User Guide

Amazon Elastic Compute Cloud

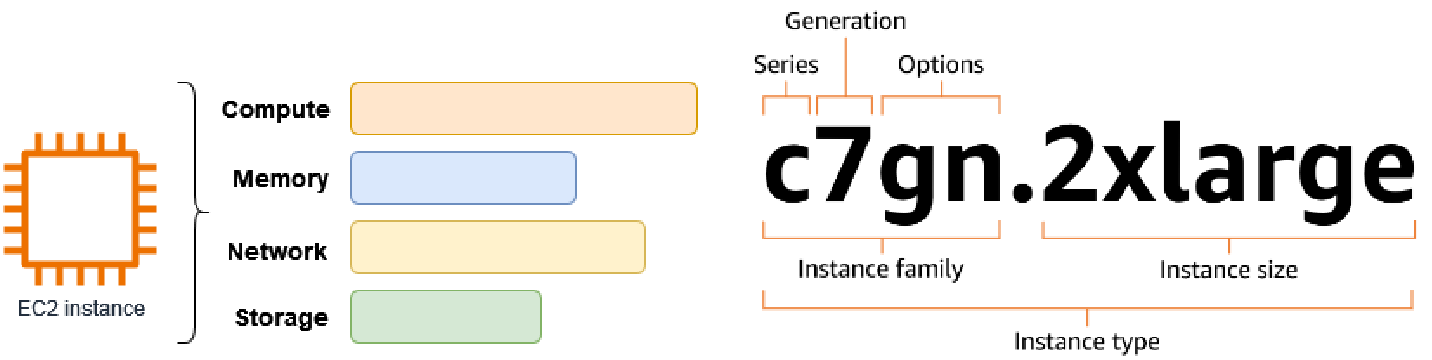
Amazon Elastic Compute Cloud: User Guide

Table of Contents

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs 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. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

An EC2 instance is a virtual server in the AWS Cloud. When you launch an EC2 instance, the instance type that you specify determines the hardware available to your instance. Each instance type offers a different balance of compute, memory, network, and storage resources. For more information, see the Amazon EC2 Instance Types Guide.



Features of Amazon EC2

Amazon EC2 provides the following high-level features:

Instances

Virtual servers.

Amazon Machine Images (AMIs)

Preconfigured templates for your instances that package the components you need for your server (including the operating system and additional software).

Instance types

Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances.

Amazon EBS volumes

Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).

Instance store volumes

Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance.

Key pairs

Secure login information for your instances. AWS stores the public key and you store the private key in a secure place.

Security groups

A virtual firewall that allows you to specify the protocols, ports, and source IP ranges that can reach your instances, and the destination IP ranges to which your instances can connect.

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.

Related services

Services to use with Amazon EC2

You can use other AWS services with the instances that you deploy using Amazon EC2.

Amazon EC2 Auto Scaling

Helps ensure you have the correct number of Amazon EC2 instances available to handle the load for your application.

AWS Backup

Automate backing up your Amazon EC2 instances and the Amazon EBS volumes attached to them.

Amazon CloudWatch

Monitor your instances and Amazon EBS volumes.

[Elastic Load Balancing]

Automatically distribute incoming application traffic across multiple instances.

[Amazon GuardDuty]

Detect potentially unauthorized or malicious use of your EC2 instances.

[EC2 Image Builder]

Automate the creation, management, and deployment of customized, secure, and up-to-date server images.

[AWS Launch Wizard]

Size, configure, and deploy AWS resources for third-party applications without having to manually identify and provision individual AWS resources.

[AWS Systems Manager]

Perform operations at scale on EC2 instances with this secure end-to-end management solution.

Additional compute services

You can launch instances using another AWS compute service instead of using Amazon EC2.

[Amazon Lightsail]

Build websites or web applications using Amazon Lightsail, a cloud platform that provides the resources that you need to deploy your project quickly, for a low, predictable monthly price. To compare Amazon EC2 and Lightsail, see [Amazon Lightsail or Amazon EC2].

[Amazon Elastic Container Service (Amazon ECS)]

Deploy, manage, and scale containerized applications on a cluster of EC2 instances. For more information, see [Choosing an AWS container service].

[Amazon Elastic Kubernetes Service (Amazon EKS)]

Run your Kubernetes applications on AWS. For more information, see [Choosing an AWS container service].

Access Amazon EC2

You can create and manage your Amazon EC2 instances using the following interfaces:

Amazon EC2 console

A simple web interface to create and manage Amazon EC2 instances and resources. 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.

AWS Command Line Interface

Enables you to interact with AWS services using commands in your command-line shell. It is supported on Windows, Mac, and Linux. For more information about the AWS CLI , see AWS Command Line Interface User Guide. You can find the Amazon EC2 commands in the AWS CLI Command Reference.

AWS CloudFormation

Amazon EC2 supports creating resources using AWS CloudFormation. You create a template, in JSON or YAML format, 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 supported resource types and properties for Amazon EC2, see EC2 resource type reference in the *AWS CloudFormation User Guide*.

AWS SDKs

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

AWS Tools for PowerShell

A set of PowerShell modules that are built on the functionality exposed by the SDK for .NET. The Tools for PowerShell enable you to script operations on your AWS resources from the PowerShell command line. To get started, see the AWS Tools for PowerShell User Guide. You can find the cmdlets for Amazon EC2, in the AWS Tools for PowerShell Cmdlet Reference.

Query API

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*.

Pricing for Amazon EC2

Amazon EC2 provides the following pricing options:

Free Tier

You can get started with Amazon EC2 for free. To explore the Free Tier options, see AWS Free Tier.

On-Demand Instances

Pay for the instances that you use by the second, with a minimum of 60 seconds, 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.

Dedicated Hosts

Reduce costs by using a physical EC2 server that is fully dedicated for your use, either On-Demand or as part of a Savings Plan. You can use your existing server-bound software licenses and get help meeting compliance requirements.

On-Demand Capacity Reservations

Reserve compute capacity for your EC2 instances in a specific Availability Zone for any duration of time.

Per-second billing

Removes the cost of unused minutes and seconds from your bill.

For a complete list of charges and prices for Amazon EC2 and more information about the purchase models, see Amazon EC2 pricing.

Estimates, billing, and cost optimization

To create estimates for your AWS use cases, use the AWS Pricing Calculator.

To estimate the cost of transforming **Microsoft workloads** to a modern architecture that uses open source and cloud-native services deployed on AWS, use the AWS Modernization Calculator for Microsoft Workloads.

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.

To calculate the cost of a sample provisioned environment, see Cloud Economics Center. When calculating the cost of a provisioned environment, remember to include incidental costs such as snapshot storage for EBS volumes.

You can optimize the cost, security, and performance of your AWS environment using AWS Trusted Advisor.

You can use AWS Cost Explorer to analyze the cost and usage of your EC2 instances. You can view data up to the last 13 months, and forecast how much you are likely to spend for the next 12 months. For more information, see Analyzing your costs and usage with AWS Cost Explorer in the *AWS Cost Management User Guide*.

Resources

- Amazon EC2 features
- AWS re:Post
- AWS Skill Builder
- AWS Support

- Hands-on Tutorials
- Web Hosting
- Windows on AWS

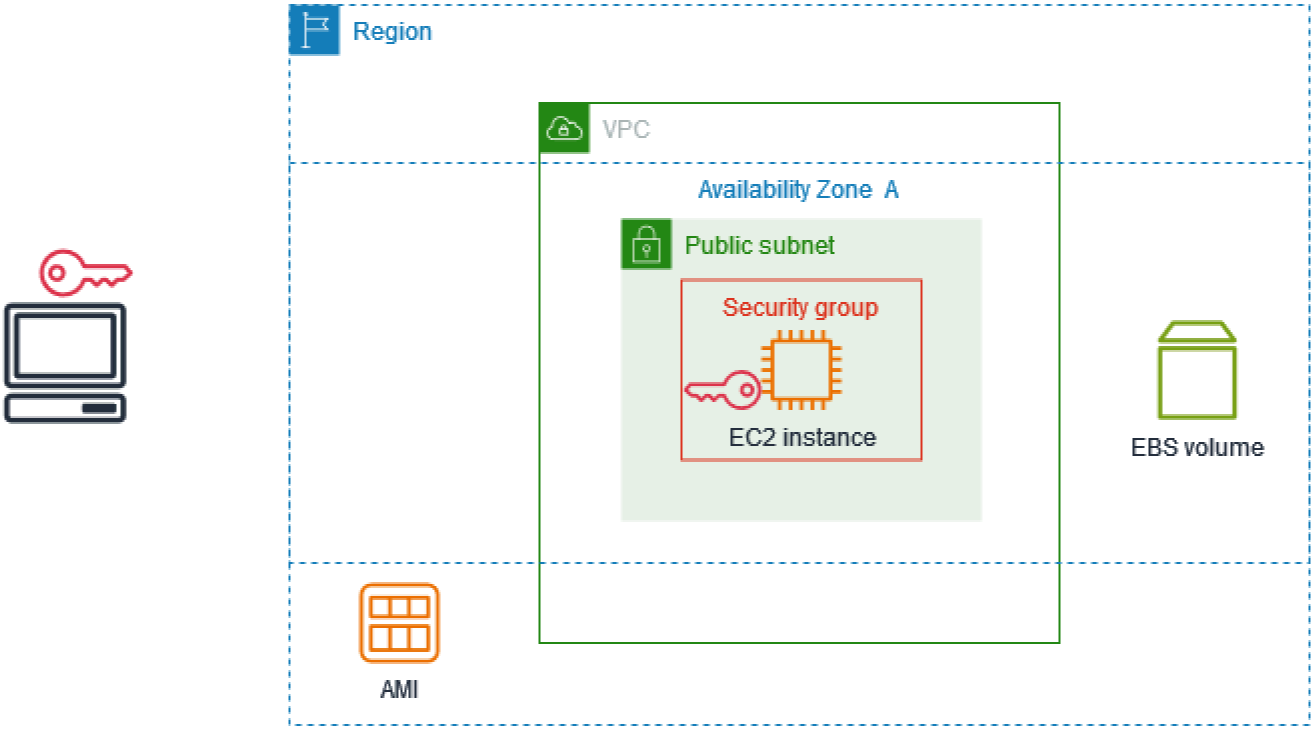
Get started with Amazon EC2

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

Overview

The following diagram shows the key components that you'll use in this tutorial:

- **An image –** A template that contains the software to run on your instance, such as the operating system.
- **A key pair –** A set of security credentials that you use to prove your identity when connecting to your instance. The public key is on your instance and the private key is on your computer.
- **A network –** A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. To help you get started quickly, your account comes with a default VPC in each AWS Region, and each default VPC has a default subnet in each Availability Zone.
- **A security group –** Acts as a virtual firewall to control inbound and outbound traffic.
- **An EBS volume –** We require a root volume for the image. You can optionally add data volumes.



Cost for this tutorial

When you create your AWS account, you can get started with Amazon EC2 for free using the AWS Free Tier.

If you created your AWS account before July 15, 2025, it's less than 12 months old, and you haven't already exceeded the Free Tier benefits for Amazon EC2, it won't cost you anything to complete this tutorial, because we help you select options that are within the Free Tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance (even if it remains idle) until you terminate it.

If you created your AWS account on or after July 15, 2025, it's less than 6 months old, and you haven't used up all your credits, it won't cost you anything to complete this tutorial, because we help you select options that are within the Free Tier benefits.

For information on how to determine whether you are eligible for the Free Tier, see the section called “Track your Free Tier usage”.

Tasks

- Step 1: Launch an instance
- Step 2: Connect to your instance
- Step 3: Clean up your instance

- [Next steps](#)

Step 1: Launch an instance

You can launch an EC2 instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, we display the current AWS Region — for example, **Ohio**. You can use the selected Region, or optionally select a Region that is closer to you.
3. From the EC2 console dashboard, in the **Launch instance** pane, choose **Launch instance**.
4. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
5. Under **Application and OS Images (Amazon Machine Image)**, do the following:

    a. Choose **Quick Start**, and then choose the operating system (OS) for your instance. For your first Linux instance, we recommend that you choose Amazon Linux.

    b. From **Amazon Machine Image (AMI)**, select an AMI that is marked **Free Tier eligible**.
6. Under **Instance type**, for **Instance type**, select an instance type that is marked **Free Tier eligible**.
7. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or choose **Create new key pair** to create your first key pair.

    > **Warning**
    >
    > If you choose **Proceed without a key pair (Not recommended)**, you won't be able to connect to your instance using the methods described in this tutorial.
8. Under **Network settings**, notice that we selected your default VPC, selected the option to use the default subnet in an Availability Zone that we choose for you, and configured a security group with a rule that allows connections to your instance from anywhere (`0.0.0.0/0`).

> **Warning**
>
> If you specify `0.0.0.0/0`, you are enabling traffic from any IP addresses in the world. For the SSH and RDP protocols, you might consider this acceptable for a short time in a test environment, but it's unsafe for production environments. In production, be sure to authorize access only from the appropriate individual IP address or range of addresses.

For your first instance, we recommend that you use the default settings. Otherwise, you can update your network settings as follows:

- (Optional) To use a specific default subnet, choose **Edit** and then choose a subnet.
- (Optional) To use a different VPC, choose **Edit** and then choose an existing VPC. If the VPC isn't configured for public internet access, you won't be able to connect to your instance.
- (Optional) To restrict inbound connection traffic to a specific network, choose **Custom** instead of **Anywhere**, and enter the CIDR block for your network.
- (Optional) To use a different security group, choose **Select existing security group** and choose an existing security group. If the security group does not have a rule that allows connection traffic from your network, you won't be able to connect to your instance. For a Linux instance, you must allow SSH traffic. For a Windows instance, you must allow RDP traffic.

9. Under **Configure storage**, notice that we configured a root volume but no data volumes. This is sufficient for test purposes.
10. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.
11. If the launch is successful, choose the ID of the instance from the **Success** notification to open the **Instances** page and monitor the status of the launch.
12. Select the checkbox for the instance. The initial instance state is `pending`. After the instance starts, its state changes to `running`. Choose the **Status and alarms** tab. After your instance passes its status checks, it is ready to receive connection requests.

Step 2: Connect to your instance

The procedure that you use depends on the operating system of the instance. If you can't connect to your instance, see Troubleshoot issues connecting to your Amazon EC2 Linux instance for assistance.

Linux instances

You can connect to your Linux instance using any SSH client. If you are running Windows on your computer, open a terminal and run the `ssh` command to verify that you have an SSH client installed. If the command is not found, install OpenSSH for Windows.

To connect to your instance using SSH

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 then choose **Connect**.
4. On the **Connect to instance** page, choose the **SSH client** tab.
5. (Optional) If you created a key pair when you launched the instance and downloaded the private key (.pem file) to a computer running Linux or macOS, run the example **chmod** command to set the permissions for your private key.
6. Copy the example SSH command. The following is an example, where *`key-pair-name`*.pem is the name of your private key file, *`ec2-user`* is the username associated with the image, and the string after the @ symbol is the public DNS name of the instance.

    ```
    ssh -i key-pair-name.pem ec2-user@ec2-198-51-100-1.us-east-2.compute.amazonaws.com
    ```

7. In a terminal window on your computer, run the **ssh** command that you saved in the previous step. If the private key file is not in the current directory, you must specify the fully-qualified path to the key file in this command.

    The following is an example response:

    ```
    The authenticity of host 'ec2-198-51-100-1.us-east-2.compute.amazonaws.com
     (198-51-100-1)' can't be established.
    ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
    Are you sure you want to continue connecting (yes/no)?
    ```

8. (Optional) Verify that the fingerprint in the security alert matches the instance fingerprint contained in the console output when you first start an instance. To get the console output, choose **Actions**, **Monitor and troubleshoot**, **Get system log**. If the fingerprints don't match, someone might be attempting a man-in-the-middle attack. If they match, continue to the next step.

9. Enter **yes**.

   The following is an example response:

```
Warning: Permanently added 'ec2-198-51-100-1.us-
east-2.compute.amazonaws.com' (ECDSA) to the list of known hosts.
```

Windows instances

To connect to a Windows instance using RDP, you must retrieve the initial administrator password and then enter this password when you connect to your instance. It takes a few minutes after instance launch before this password is available. Your account must have permission to call the GetPasswordData action. For more information, see Example policies to control access the Amazon EC2 API.

The default username for the Administrator account depends on the language of the operating system (OS) contained in the AMI. To determine the correct username, identify the language of the OS, and then choose the corresponding username. For example, for an English OS, the username is `Administrator`, for a French OS it's `Administrateur`, and for a Portuguese OS it's `Administrador`. If a language version of the OS does not have a username in the same language, choose the username `Administrator (Other)`. For more information, see Localized Names for Administrator Account in Windows in the Microsoft website.

To retrieve the initial administrator password

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 then choose **Connect**.
4. On the **Connect to instance** page, choose the **RDP client** tab.
5. For **Username**, choose the default username for the Administrator account. The username you choose must match the language of the operating system (OS) contained in the AMI that you

used to launch your instance. If there is no username in the same language as your OS, choose **Administrator (Other)**.

6. Choose **Get password**.
7. On the **Get Windows password** page, do the following:

    a. Choose **Upload private key file** and navigate to the private key (`.pem`) file that you specified when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file to this window.

    b. Choose **Decrypt password**. The **Get Windows password** page closes, and the default administrator password for the instance appears under **Password**, replacing the **Get password** link shown previously.

    c. Copy the password and save it in a safe place. This password is required to connect to the instance.

The following procedure uses the Remote Desktop Connection client for Windows (MSTSC). If you're using a different RDP client, download the RDP file and then see the documentation for the RDP client for the steps to establish the RDP connection.

To connect to a Windows instance using an RDP client

1. On the **Connect to instance** page, choose **Download remote desktop file**. When the file download is finished, choose **Cancel** to return to the **Instances** page. The RDP file is downloaded to your `Downloads` folder.
2. Run `mstsc.exe` to open the RDP client.
3. Expand **Show options**, choose **Open**, and select the .rdp file from your `Downloads` folder.
4. By default, **Computer** is the public IPv4 DNS name of the instance and **User name** is the administrator account. To connect to the instance using IPv6 instead, replace the public IPv4 DNS name of the instance with its IPv6 address. Review the default settings and change them as needed.
5. Choose **Connect**. If you receive a warning that the publisher of the remote connection is unknown, choose **Connect** to continue.
6. Enter the password that you saved previously, and then choose **OK**.
7. Due to the nature of self-signed certificates, you might get a warning that the security certificate could not be authenticated. Do one of the following:
    - If you trust the certificate, choose **Yes** to connect to your instance.

- [Windows] Before you proceed, compare the thumbprint of the certificate with the value in the system log to confirm the identity of the remote computer. Choose **View certificate** and then choose **Thumbprint** from the **Details** tab. Compare this value to the value of `RDPCERTIFICATE-THUMBPRINT` in **Actions**, **Monitor and troubleshoot**, **Get system log**.
- [Mac OS X] Before you proceed, compare the fingerprint of the certificate with the value in the system log to confirm the identity of the remote computer. Choose **Show Certificate**, expand **Details**, and choose **SHA1 Fingerprints**. Compare this value to the value of `RDPCERTIFICATE-THUMBPRINT` in **Actions**, **Monitor and troubleshoot**, **Get system log**.

8. If the RDP connection is successful, the RDP client displays the Windows login screen and then the Windows desktop. If you receive an error message instead, see the section called "Remote Desktop can't connect to the remote computer". When you are finished with the RDP connection, you can close the RDP client.

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.

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

You'll stop incurring charges for that instance or usage that counts against your Free Tier limits as soon as the instance status changes to `shutting down` or `terminated`. To keep your instance for later, but not incur charges or usage that counts against your Free Tier limits, you can stop the instance now and then start it again later. For more information, see Stop and start Amazon EC2 instances.

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Instance state**, **Terminate (delete) instance**.

3. Choose **Terminate (delete)** 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 explore the following next steps:

- Explore the Amazon EC2 core concepts with the introductory tutorials. For more information, see Tutorials for launching EC2 instances.
- Learn how to track your Amazon EC2 Free Tier usage using the console. For more information, see the section called "Track your Free Tier usage".
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier (for accounts created before July 15, 2025). For more information, see Tracking your AWS Free Tier usage in the *AWS Billing User Guide*.
- Add an EBS volume. For more information, see Create an Amazon EBS volume in the *Amazon EBS User Guide*.
- Learn how to remotely manage your EC2 instance using the Run command. For more information, see AWS Systems Manager Run Command in the *AWS Systems Manager User Guide*.
- Learn about instance purchasing options. For more information, see Amazon EC2 billing and purchasing options.
- Get advice about instance types. For more information, see Get recommendations from EC2 instance type finder.

Best practices for Amazon EC2

To ensure the maximum benefit from Amazon EC2, we recommend that you perform the following best practices.

Security

- Manage access to AWS resources and APIs using identity federation with an identity provider and IAM roles whenever possible. For more information, see Creating IAM policies in the *IAM User Guide*.
- Implement the least permissive rules for your security group.
- Regularly patch, update, and secure the operating system and applications on your instance. For more information, see Update management. For guidelines specific to Windows operating systems, see Security best practices for Windows instances.
- Use Amazon Inspector to automatically discover and scan Amazon EC2 instances for software vulnerabilities and unintended network exposure. For more information, see the Amazon Inspector User Guide.
- Use AWS Security Hub controls to monitor your Amazon EC2 resources against security best practices and security standards. For more information about using Security Hub, see Amazon Elastic Compute Cloud controls in the *AWS Security Hub User Guide*.

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see Root device type.
- 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 data when an instance is terminated.
- 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 in the *Amazon EBS User Guide*.

Resource management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see Use instance metadata to manage your EC2 instance and Tag your Amazon EC2 resources.
- 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.
- Use AWS Trusted Advisor to inspect your AWS environment, and then make recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. For more information, see AWS Trusted Advisor in the *AWS Support User Guide*.

Backup and recovery

- Regularly back up your EBS volumes using Amazon EBS snapshots, and create an Amazon Machine Image (AMI) from your instance to save the configuration as a template for launching future instances. For more information about AWS services that help achieve this use case, see AWS Backup and Amazon Data Lifecycle Manager.
- 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.
- Monitor and respond to events. For more information, see Monitor Amazon EC2 resources.
- 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. 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 to ensure data and services are restored successfully.

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.

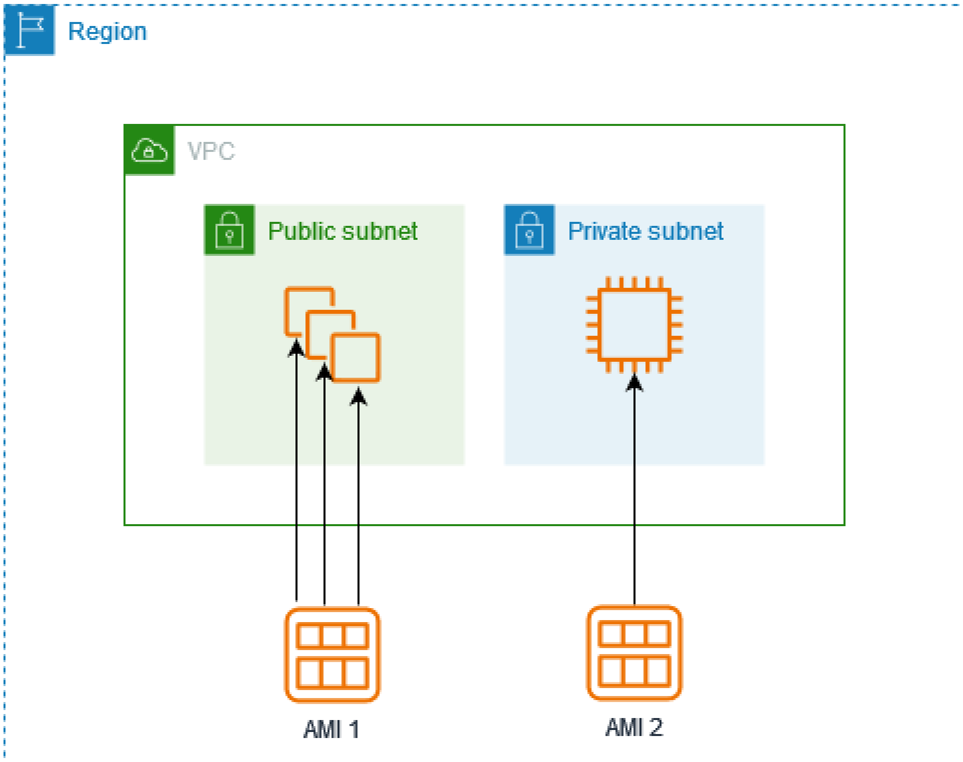
Amazon Machine Images in Amazon EC2

An Amazon Machine Image (AMI) is an image that provides the software that is required to set up and boot an Amazon EC2 instance. Each AMI also contains a block device mapping that specifies the block devices to attach to the instances that you launch. You must specify an AMI when you launch an instance. The AMI must be compatible with the instance type that you chose for your instance. You can use an AMI provided by AWS, a public AMI, an AMI that someone else shared with you, or an AMI that you purchased from the AWS Marketplace.

An AMI is specific to the following:

- Region
- Operating system
- Processor architecture
- Root device type
- Virtualization type

You can launch multiple instances from a single AMI when you require multiple instances with the same configuration. You can use different AMIs to launch instances when you require instances with different configurations, as shown in the following diagram.



You can create an AMI from your Amazon EC2 instances and then use it to launch instances with the same configuration. You can copy an AMI to another AWS Region, and then use it to launch instances in that Region. You can also share an AMI that you created with other accounts so that they can launch instances with the same configuration. You can sell your AMI using the AWS Marketplace.

Contents

AMI types and characteristics in Amazon EC2

When you launch an instance, the AMI that you choose must be compatible with the instance type that you choose. You can select an AMI to use based on the following characteristics:

- Region
- Operating system
- Processor architecture
- Launch permissions
- Root device type
- Virtualization types

Launch permissions

Launch permissions determine who can use an AMI to launch instances. You can think of launch permissions as sharing an AMI—when you grant launch permissions, you're sharing the AMI with other users. Only the owner of an AMI can determine its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts, organizat ions, or organizational units (OUs).
implicit	The owner has implicit launch permissions for an AMI.

Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, see Understand shared AMI usage in Amazon EC2. Developers can charge for their AMIs. For more information, see Paid AMIs in the AWS Marketplace for Amazon EC2 instances.

Root device type

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*.

- Amazon EBS-backed AMI – The root device for an instance launched from the AMI is an Amazon Elastic Block Store (Amazon EBS) volume created from an Amazon EBS snapshot. Supported for both Linux and Windows AMIs.
- Amazon instance store-backed AMI – The root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. Supported for Linux AMIs only. Windows AMIs do not support instance store for the root device.

For more information, see Root volumes for your Amazon EC2 instances.

Note

Instance store-backed AMIs are considered end of life and are not recommended for new usage. They are only supported on the following older instance types: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

The following table summarizes the important differences when using the two types of AMIs.

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
Root device volume	EBS volume	Instance store volume
Boot time for an instance	Usually less than 1 minute	Usually less than 5 minutes
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other EBS volumes persists after instance termination by default.	Data on any instance store volumes persists only during the life of the instance.
Stopped state	Can be in a stopped state. Even when the instance is stopped and not running, the root volume is persisted in Amazon EBS.	Cannot be in a stopped state; instances are running or terminated.

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
Modifications	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, EBS volume usage, and storing your AMI as an EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools

* By default, EBS root volumes have the `DeleteOnTermination` flag set to `true`. For information about how to change this flag so that the volume persists after termination, see Keep an Amazon EBS root volume after an Amazon EC2 instance terminates.

** Supported with `io2` EBS Block Express only. For more information, see Provisioned IOPS SSD Block Express volumes in the *Amazon EBS User Guide*.

Determine the root device type of your AMI

The AMI that you use to launch an EC2 instance determines the type of the root volume. The root volume of an EC2 instance is either an EBS volume or an instance store volume.

Nitro-based instances support only EBS root volumes. The following previous generation instance types are the only instance types that support instance store root volumes: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

Console

To determine the root device type of an AMI

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

2. In the navigation pane, choose **AMIs**, and select the AMI.
3. On the **Details** tab, check the value of **Root device type** as follows:
   - `ebs` – This is an EBS-backed AMI.
   - `instance store` – This is an instance store-backed AMI.

AWS CLI

To determine the root device type of an AMI

Use the describe-images command.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].RootDeviceType
```

The following is example output.

```
ebs
```

PowerShell

To determine the root device type of an AMI

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image `
    -ImageId ami-0abcdef1234567890).RootDeviceType.Value
```

The following is example output.

```
ebs
```

Virtualization types

Amazon Machine Images use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The main differences between PV and HVM AMIs are the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance. Windows AMIs are HVM AMIs.

The following table compares HVM and PV AMIs.

Characteristic	HVM	PV
Description	HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.	PV AMIs boot with a special boot loader called PV-GRUB, which starts the boot cycle and then chain loads the kernel specified in the `menu.lst` file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization. For more information about PV-GRUB and its use in Amazon EC2, see User provided kernels.
Supported instance types	All current generation instance types support HVM AMIs.	The following previous generation instance types support PV AMIs: C1, C3, M1, M3, M2, and T1. Current generation instance types do not support PV AMIs.
Support for hardware extensions	HVM guests can take advantage of hardware extensions that provide fast access to the underlyin g hardware on the host system. They are required to use enhanced networkin	No, they can't take advantage of special hardware extension s such as enhanced networkin g or GPU processing.

Characteristic	HVM	PV
	g and GPU processing. To pass through instructions to specialized network and GPU devices, the OS must have access to the native hardware platform, and HVM virtualiz ation provides this access. For more information, see Enhanced networking on Amazon EC2 instances.	
How to find	Verify that the virtualization type of the AMI is set to `hvm`, using the console or the describe-images command.	Verify that the virtualization type of the AMI is set to `paravirtual` , using the console or the describe-images command.

PV on HVM

Paravirtual guests traditionally performed better with storage and network operations than HVM guests because they could leverage special drivers for I/O that avoided the overhead of emulating network and disk hardware, whereas HVM guests had to translate these instructions to emulated hardware. Now PV drivers are available for HVM guests, so operating systems that cannot be ported to run in a paravirtualized environment can still see performance advantages in storage and network I/O by using them. With these PV on HVM drivers, HVM guests can get the same, or better, performance than paravirtual guests.

Find an AMI that meets the requirements for your EC2 instance

An AMI includes the components and applications, such as the operating system and type of root volume, required to launch an instance. To launch an instance, you must find an AMI that meets your needs.

When selecting an AMI, consider the following requirements you might have for the instances that you want to launch:

- The AWS Region of the AMI as AMI IDs are unique to each Region.
- The operating system (for example, Linux or Windows).
- The architecture (for example, 32-bit, 64-bit, or 64-bit ARM).
- The root device type (for example, Amazon EBS or instance store).
- The provider (for example, Amazon Web Services).
- Additional software (for example, SQL Server).

Console

You can select from the list of AMIs when you use the launch instance wizard, or you can search all available AMIs using the **Images** page.

To find a Quick Start AMI using the launch instance 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. AMI IDs are unique to each AWS Region.
3. From the console dashboard, choose **Launch instance**.
4. Under **Application and OS Images (Amazon Machine Image)**, choose **Quick Start**, choose the operating system (OS) for your instance, and then, from **Amazon Machine Image (AMI)**, select from one of the commonly used AMIs in the list. If you don't see the AMI that you want to use, choose **Browse more AMIs** to browse the full AMI catalog. For more information, see Application and OS Images (Amazon Machine Image).

To find an AMI using the AMIs 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. AMI IDs are unique to each AWS Region.
3. In the navigation pane, choose **AMIs**.
4. (Optional) Use the filter and search options to scope the list of displayed AMIs to see only the AMIs that match your criteria.

For example, to list all AMIs provided by AWS, choose **Public images**. Then use the search options to further scope the list of displayed AMIs. Choose the **Search** bar and, from the menu, choose **Owner alias**, then the **=** operator, and then the value **amazon**. To find AMIs that match a specific platform, for example Linux or Windows, choose the **Search** bar again to choose **Platform**, then the **=** operator, and then the operating system from the list provided.

5. (Optional) Choose the **Preferences** 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 on the **Details** tab.
6. Before you select an AMI, it's important that you check whether it's backed by instance store or by Amazon EBS and that you are aware of the effects of this difference. For more information, see Root device type.
7. To launch an instance from this AMI, select it and then choose **Launch instance from image**. For more information about launching an instance using the console, see Launch an EC2 instance using the launch instance wizard in the console. If you're not ready to launch the instance now, make note of the AMI ID for later.

AWS CLI

Use the describe-images command to find an AMI that meets your requirements. By default, this command returns all AMIs that are public, that you own, and that are shared with you.

To find an AMI owned by Amazon

Use the describe-images command with the `--owners` option.

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

To find a Windows AMI

Add the following filter to display only Windows AMIs.

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

To find an EBS-backed AMI

Add the following filter to display only AMIs backed by Amazon EBS.

```
--filters "Name=root-device-type,Values=ebs"
```

PowerShell

Use the Get-EC2Image cmdlet to find an AMI that meets your requirements. By default, this cmdlet returns all AMIs that are public, that you own, or that are shared with you.

To find an AMI owned by Amazon

Use the Get-EC2Image command with the `-Owner` parameter.

```
Get-EC2Image -Owner amazon
```

To find a Windows AMI

Add the following filter to display only Windows AMIs.

```
-Filter @{Name="platform"; Values="windows"}
```

For additional examples, see Find an Amazon Machine Image Using Windows PowerShell in the *AWS Tools for PowerShell User Guide*.

Related resources

For more information about AMIs for a specific operating system, see the following:

- Amazon Linux 2023 – AL2023 on Amazon EC2 in the *Amazon Linux 2023 User Guide*
- Ubuntu – Amazon EC2 AMI Locator on the *Canonical Ubuntu website*
- RHEL – Red Hat Enterprise Linux Images (AMI) Available on Amazon Web Services (AWS) on the Red Hat website
- Windows Server – AWS Windows AMI reference

For information about AMIs that you can subscribe to on the AWS Marketplace see Paid AMIs in the AWS Marketplace for Amazon EC2 instances.

For information about using Systems Manager to help your users find the latest AMI that they should use when launching an instance, see the following:

Reference AMIs using Systems Manager parameters

When you launch an instance using the EC2 launch instance wizard in the Amazon EC2 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 (described in this section). 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`. Specifying this data type ensures that when the parameter is created or modified, the parameter value is validated as an AMI ID. For more information, see Native parameter support for Amazon Machine Image IDs in the *AWS Systems Manager User Guide*.

Contents

Use cases

When you use Systems Manager parameters to point to AMI IDs, it is easier for your users to select the correct AMI when launching instances. Systems Manager parameters can also simplify the maintenance of automation code.

Easier for users

If you require instances to be launched using a specific AMI, and the AMI is regularly updated, we recommend that you require your users to select a Systems Manager parameter to find the AMI.

Requiring your users to select a Systems Manager parameter ensures 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 have to know about the periodic updates to the AMI because they continue to select the same Systems Manager parameter each time. Using a Systems Manager parameter for your AMI makes 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 can 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 have to be modified each time a new version of the AMI is created. This simplifies the maintenance of the automation and helps to drive down deployment costs.

> **Note**
>
> Running instances are not affected when you change the AMI ID pointed to by the Systems Manager parameter.

Permissions

If you use Systems Manager parameters that point to AMI IDs in the launch instance wizard, you must add the following permissions to your IAM policy:

- `ssm:DescribeParameters` – Grants permission to view and select Systems Manager parameters.
- `ssm:GetParameters` – Grants permission to retrieve the values of the Systems Manager parameters.

You can also restrict access to specific Systems Manager parameters. For more information and example IAM policies, see Example: Use the EC2 launch instance wizard.

Limitations

AMIs and Systems Manager parameters are Region specific. To use the same Systems Manager parameter name across Regions, create a Systems Manager parameter in each Region with the same name (for example, `golden-ami`). In each Region, point the Systems Manager parameter to an AMI in that Region.

Parameter names are 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.

Launch an instance using a Systems Manager parameter

When you launch an instance, instead of specifying an AMI ID, you can specify a Systems Manager parameter that points to an AMI ID.

To specify the parameter programmatically, use the following syntax, where `resolve:ssm` is the standard prefix and `parameter-name` is the unique parameter name.

```
resolve:ssm:parameter-name
```

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, where `version` is the unique version number. By default, the latest version of the parameter is used when no version is specified.

```
resolve:ssm:parameter-name:version
```

To launch an instance using a public parameter provided by AWS, see Reference the latest AMIs using Systems Manager public parameters.

Console

To find an AMI using a Systems Manager parameter

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. Under **Application and OS Images (Amazon Machine Image)**, choose **Browse more AMIs**.
5. Choose the arrow button to the right of the search bar, and then choose **Search by Systems Manager parameter**.
6. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears below **Currently resolves to**.
7. Choose **Search**. The AMIs that match the AMI ID appear in the list.
8. Select the AMI from the list, and choose **Select**.

For more information about launching an instance using the launch instance wizard, see Launch an EC2 instance using the launch instance wizard in the console.

AWS CLI

To launch an instance using a Systems Manager parameter

Use the run-instances command with the `--image-id` option. This example uses a Systems Manager parameter named **`golden-ami`**, which specifies an AMI ID.

```
--image-id resolve:ssm:/golden-ami
```

You can create versions of a Systems Manager parameter. The following example specifies version 2 of the **`golden-ami`** parameter.

```
--image-id resolve:ssm:/golden-ami:2
```

PowerShell

To launch an instance using a Systems Manager parameter

Use the New-EC2Instance cmdlet with the `-ImageId` parameter. This example uses a Systems Manager parameter named **`golden-ami`**, which specifies an AMI ID.

```
-ImageId "resolve:ssm:/golden-ami"
```

You can create versions of a Systems Manager parameter. The following example specifies version 2 of the **`golden-ami`** parameter.

```
-ImageId "resolve:ssm:/golden-ami:2"
```

Reference the latest AMIs using Systems Manager public parameters

AWS Systems Manager provides public parameters for public AMIs maintained by AWS. You can use the public parameters when launching instances to ensure that you're using the latest AMIs. For example, the public parameter `/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-arm64` is available in all Regions and always points to the latest version of the Amazon Linux 2023 AMI for arm64 architecture in a given Region.

The public parameters are available from the following paths:

- **Linux** – `/aws/service/ami-amazon-linux-latest`
- **Windows** – `/aws/service/ami-windows-latest`

For more information, see Working with public parameters in the *AWS Systems Manager User Guide*.

List the Amazon Linux AMIs

AWS CLI

To list the Linux AMIs in the current AWS Region

Use the following get-parameters-by-path command. The value for the `--path` parameter is specific to Linux AMIs.

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-amazon-linux-latest \
    --query "Parameters[].Name"
```

PowerShell

To list the Linux AMIs in the current AWS Region

Use the Get-SSMParametersByPath cmdlet.

```
Get-SSMParametersByPath `
```

```
    -Path "/aws/service/ami-amazon-linux-latest" | `
    Sort-Object Name | Format-Table Name
```

List the Windows AMIs

AWS CLI

To list the Windows AMIs in the current AWS Region

Use the following get-parameters-by-path command. The value for the `--path` parameter is specific to Windows AMIs.

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

PowerShell

To list the Windows AMIs in the current AWS Region

Use the Get-SSMParametersByPath cmdlet.

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-windows-latest" | `
    Sort-Object Name | Format-Table Name
```

Launch an instance using a public parameter

To specify the public parameter when launching an instance, use the following syntax: `resolve:ssm:`*`public-parameter`*, where `resolve:ssm` is the standard prefix and *`public-parameter`* is the path and name of the public parameter.

AWS CLI

To launch an instance using a public parameter

Use the run-instances command with the `--image-id` option. This example specifies a Systems Manager public parameter for the image ID to launch an instance using the latest Amazon Linux 2023 AMI

```
--image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64
```

PowerShell

To launch an instance using a public parameter

Use the New-EC2Instance cmdlet with the `-ImageId` parameter. This example specifies a Systems Manager public parameter for the image ID to launch an instance using the latest AMI for Windows Server 2022.

```
-ImageId "resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base"
```

For more examples that use Systems Manager parameters, see Query for the latest Amazon Linux AMI IDs Using AWS Systems Manager Parameter Store and Query for the Latest Windows AMI Using AWS Systems Manager Parameter Store.

Paid AMIs in the AWS Marketplace for Amazon EC2 instances

A *paid AMI* is an AMI that is listed for sale in the AWS Marketplace. 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 website.

You can purchase AMIs in the AWS Marketplace 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 in the AWS Marketplace to other Amazon EC2 users. 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 Understand shared AMI usage in Amazon EC2.

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 m5.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 in the AWS Marketplace

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

A paid AMI is an Amazon Machine Image (AMI) that is available for purchase. A paid AMI also has a product code. You can find AMIs that are available for purchase in the AWS Marketplace.

Console

To find a paid AMI

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. Do one of the following:
   - If you know the product code, choose **Product code**, then **=**, and then enter the product code.
   - If you do not know the product code, in the Search bar, specify the following filter: **Owner alias=aws-marketplace**. Specify additional filters as needed.
5. Save the ID of the AMI.

AWS CLI

To find a paid AMI

Use the following describe-images command.

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

The output includes a large number of images. You can specify filters to help you determine which AMI you need. After you find an AMI, specify its ID in the following command to get its product code.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].ProductCodes[].ProductCodeId
```

The following is example output.

```
[
    "cdef1234abc567def8EXAMPLE"
]
```

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

```
aws ec2 describe-images \
    --filters "Name=product-code,Values=cdef1234abc567def8EXAMPLE" \
    --query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

PowerShell

To find a paid AMI

Use the Get-EC2Image cmdlet.

```
Get-EC2Image -Owner aws-marketplace
```

The output includes a large number of images. You can specify filters to help you determine which AMI you need. After you find an AMI, specify its ID in the following command to get its product code.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ProductCodes
```

The following is example output.

```
ProductCodeId             ProductCodeType
-------------             ---------------
cdef1234abc567def8EXAMPLE 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.

```
(Get-EC2Image -Owner aws-marketplace -Filter @{"Name"="product-
code";"Value"="cdef1234abc567def8EXAMPLE"} | sort CreationDate -Descending | Select-
Object -First 1).ImageId
```

Purchase a paid AMI in the AWS Marketplace

You must sign up for (purchase) a paid AMI before you can launch an Amazon EC2 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 Amazon EC2 instance from an AWS Marketplace AMI.

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. You can use one of the following methods to subscribe to products in the AWS Marketplace:

- **AWS Marketplace website**: You can launch preconfigured software quickly with the 1-Click deployment feature. For more information, see AMI-based products in AWS Marketplace.
- **Amazon EC2 launch wizard**: You can search for an AMI and launch an instance directly from the wizard. For more information, see Launch an Amazon EC2 instance from an AWS Marketplace AMI.

Retrieve the AWS Marketplace product code from your instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. If the instance has a product code, Amazon EC2 returns it. For more information about retrieving metadata, see Access instance metadata for an EC2 instance.

IMDSv2

Linux

Run the following command from your Linux instance.

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/product-codes
```

Windows

Run the following cmdlets from your Windows instance.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/product-codes
```

IMDSv1

Linux

Run the following command from your Linux instance.

```
curl http://169.254.169.254/latest/meta-data/product-codes
```

Windows

Run the following command from your Windows instance.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/product-codes
```

Use paid support for supported AWS Marketplace offerings

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.

Limitations

- After you set the product code attribute, it can't be changed or removed.
- 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.

AWS CLI

To associate a product code with your AMI

Use the [modify-image-attribute] command.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --product-codes "cdef1234abc567def8EXAMPLE"
```

PowerShell

To associate a product code with your AMI

Use the Edit-EC2ImageAttribute cmdlet.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ProductCode "cdef1234abc567def8EXAMPLE"
```

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 username for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.
   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   b. In the navigation pane, choose **Instances**.
   c. Select the instance, and then choose **Instance state**, **Terminate (delete) instance**.
   d. Choose **Terminate (delete)** 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.

Amazon EC2 AMI lifecycle

An Amazon Machine Image (AMI) is an image that contains the software configuration required to set up and boot an instance. You must specify an AMI when you launch an instance. You can use AMIs provided by Amazon, or you can create your own AMIs. The AMI must be located in the AWS Region in which you want to launch your instance.

The lifecycle of an AMI includes creating, copying, deprecating, disabling, and deleting (deregistering) the AMI.

Create AMIs. While Amazon provides AMIs that you can use to launch your instances, you can create custom AMIs tailored to your needs. To create a custom AMI, launch an instance from an existing AMI, customize the instance (for example, install software and configure operating system settings), and then create an AMI from the instance. Any instance customizations are saved to the new AMI, so that instances launched from your new AMI include these customizations.

Copy AMIs. You can use an AMI to launch an instance only in the AWS Region in which the AMI is located. If you need to launch instances with the same configuration in multiple Regions, copy the AMI to the other Regions.

Deprecate AMIs. To mark an AMI as superseded or out of date, you can set an immediate or future deprecation date. Deprecated AMIs are hidden from AMI listings, but users and services can continue to use deprecated AMIs if they know the AMI ID.

Disable AMIs. To temporarily prevent an AMI from being used, you can disable it. When an AMI is disabled, it can't be used to launch new instances. However, if you re-enable the AMI, it can be used to launch instances again. Note that disabling an AMI doesn't affect existing instances that have already been launched from it.

Deregister (delete) AMIs. When you no longer need an AMI, you can deregister it, preventing it from being used to launch new instances. If the AMI matches a retention rule, it moves to the Recycle Bin, where it can be restored before its retention period expires, after which it is permanently deleted. If it doesn't match a retention rule, it is permanently deleted immediately. Note that deregistering an AMI does not affect existing instances that were launched from the AMI.

Automate the 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. You can also use EC2 Image Builder to automate the creation, management, an deployment of customized AMIs. For more information, see Automate backups with Amazon Data Lifecycle Manager in the *Amazon EBS User Guide* and the EC2 Image Builder User Guide.

Contents

Create an Amazon EBS-backed AMI

You can create your own Amazon EBS-backed AMI from an Amazon EC2 instance or from a snapshot of the root device of an Amazon EC2 instance.

To create an Amazon EBS-backed AMI from an instance, start by launching an instance using an existing Amazon EBS-backed AMI. This AMI can be one you obtained from the AWS Marketplace, created using VM Import/Export, or any other AMI that you can access. After customizing the instance to meet your specific requirements, create and register a new AMI. You can then use the new AMI to launch new instances with your customizations.

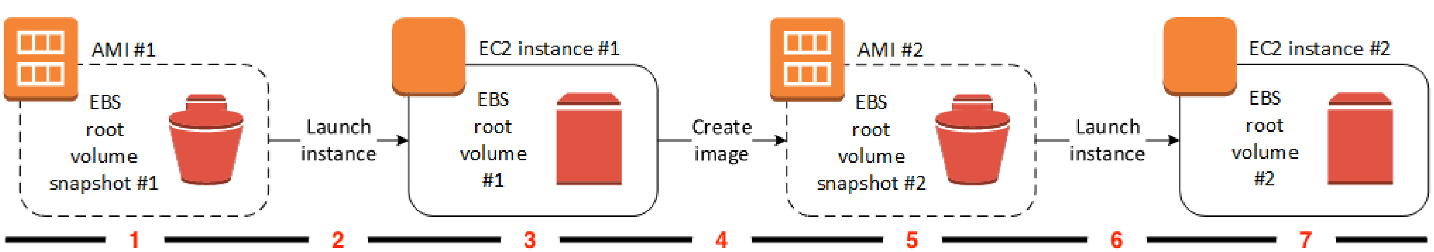
The procedures described below work for Amazon EC2 instances backed by encrypted Amazon Elastic Block Store (Amazon EBS) volumes (including the root volume) as well as for unencrypted volumes.

The AMI creation process is different for instance store-backed AMIs. For more information, see Create an instance store-backed AMI.

Contents

Overview of AMI creation from an instance

The following diagram summarizes the process for creating an Amazon EBS-backed AMI from a running EC2 instance: Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI. The numbers in the diagram match the numbers in the description that follows.



1 – AMI #1: Start with an existing AMI

Find an existing AMI that is similar to the AMI that you'd like to create. This can be an AMI you have obtained from the AWS Marketplace, an AMI that you have created using VM Import/Export, or any other AMI that you can access. You'll customize this AMI for your needs.

In the diagram, **EBS root volume snapshot #1** indicates that the AMI is an Amazon EBS-backed AMI and that information about the root volume is stored in this snapshot.

2 – Launch instance from existing AMI

The way to configure an AMI is to launch an instance from the AMI on which you'd like to base your new AMI, and then customize the instance (indicated at **3** in the diagram). Then, you'll create a new AMI that includes the customizations (indicated at **4** in the diagram).

3 – EC2 instance #1: Customize the instance

Connect to your instance and customize it for your needs. Your new AMI will include these customizations.

You can perform any of the following actions on your instance to customize it:

- Install software and applications
- Copy data
- Reduce start time by deleting temporary files and defragmenting your hard drive
- Attach additional EBS volumes

4 – Create image

When you create an AMI from an instance, Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can tell Amazon EC2 not to power down and reboot the instance. Some file systems, such as XFS, can freeze and unfreeze activity, making it safe to create the image without rebooting the instance.

During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. You're charged for the snapshots until you deregister the AMI and delete the snapshots. If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instances that support Amazon EBS encryption.

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 might find it more efficient to create snapshots of your volumes before creating your AMI. This way, only small, incremental snapshots need to be created when the AMI is created, and the process completes more quickly (the total time for snapshot creation remains the same).

5 – AMI #2: New AMI

After the process completes, you have a new AMI and snapshot (**snapshot #2**) created from the root volume of the instance. If you added instance-store volumes or EBS volumes to the instance, in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes.

Amazon EC2 automatically registers the AMI for you.

6 – Launch instance from new AMI

You can use the new AMI to launch an instance.

7 – EC2 instance #2: New instance

When you launch an instance using the new AMI, Amazon EC2 creates a new EBS volume for the instance's root volume using the snapshot. If you added instance-store volumes or EBS volumes when you customized the instance, 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 for volumes on Amazon EC2 instances.

When you create a new instance from an EBS-backed AMI, you should initialize both its root volume and any additional EBS storage before putting it into production. For more information, see Initialize Amazon EBS volumes in the *Amazon EBS User Guide*.

Create an AMI from an instance

If you have an existing instance, you can create an AMI from this instance.

Console

To create an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance from which to create the AMI, and then choose **Actions**, **Image and templates**, **Create image**.

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

4. On the **Create image** page, specify the following information:

   a. For **Image name**, enter a unique name for the image, up to 127 characters.

   b. For **Image description**, enter an optional description of the image, up to 255 characters.

   c. For **Reboot instance**, either keep the checkbox selected (the default), or clear it.

      - If **Reboot instance** is selected, when Amazon EC2 creates the new AMI, it reboots the instance so that it can take snapshots of the attached volumes while data is at rest, in order to ensure a consistent state.
      - If **Reboot instance** is cleared, when Amazon EC2 creates the new AMI, it does not shut down and reboot the instance.

      > **Warning**
      >
      > If you clear **Reboot instance**, we can't guarantee the file system integrity of the created image.

   d. **Instance volumes** – You can modify the root volume, and add additional Amazon EBS and instance store volumes, as follows:

      i. The root volume is defined in the first row.

         - To change the size of the root volume, for **Size**, enter the required value.

- If you select **Delete on termination**, when you terminate the instance created from this AMI, the EBS volume is deleted. If you clear **Delete on termination**, when you terminate the instance, the EBS volume is not deleted. For more information, see Preserve data when an instance is terminated.

ii. To add an EBS volume, choose **Add volume** (which adds a new row). For **Storage type**, choose **EBS**, and fill in the fields in the row. When you launch an instance from your new AMI, additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.

iii. To add an instance store volume, see Add instance store volumes to an Amazon EC2 AMI. When you launch an instance from your new AMI, additional volumes are automatically initialized and mounted. These volumes do not contain data from the instance store volumes of the running instance on which you based your AMI.

e. **Snapshot destination** – If your instance volumes are in a Local Zone that supports EBS local snapshots, choose where to create the AMI's snapshots:

- **AWS Region**: Create snapshots in the parent Region of the Local Zone of your volumes.
- **AWS Local Zone**: Create snapshots in the same Local Zone as your volumes.

> **Note**
>
> This option appears only in Local Zones that support EBS local snapshots, and only if your instance was created in a Local Zone. If the volume is in a Region, this option does not appear, and the snapshot is automatically created in the same Region as the volume. For more information, see Local snapshots in Local Zones in the *Amazon EBS User Guide*.

> **Important**
>
> All snapshots of the instance's volumes must be in the same location. Verify the location of existing snapshots. If any existing snapshots are in a different location than your selected destination, the AMI creation will fail.

f. **Tags** – You can tag the AMI and the snapshots with the same tags, or you can tag them with different tags.

   - To tag the AMI and the snapshots with the *same* tags, choose **Tag image and snapshots together**. The same tags are applied to the AMI and every snapshot that is created.
   - To tag the AMI and the snapshots with *different* tags, choose **Tag image and snapshots separately**. Different tags are applied to the AMI and the snapshots that are created. However, all the snapshots get the same tags; you can't tag each snapshot with a different tag.

   To add a tag, choose **Add tag**, and enter the key and value for the tag. Repeat for each tag.

g. When you're ready to create your AMI, choose **Create image**.

5. To view the status of your AMI while it is being created:

   a. In the navigation pane, choose **AMIs**.
   b. Set the filter to **Owned by me**, and find your AMI in the list.

      Initially, the status is `pending` but should change to `available` after a few minutes.

6. (Optional) To view the snapshot that was created for the new AMI:

   a. Note the ID of your AMI that you located in the previous step.
   b. In the navigation pane, choose **Snapshots**.
   c. Set the filter to **Owned by me**, and then find the snapshot with the new AMI ID in the **Description** column.

      When you launch an instance from this AMI, Amazon EC2 uses this snapshot to create its root device volume.

AWS CLI

To create an AMI

Use the [create-image](https://docs.aws.amazon.com/cli/latest/reference/ec2/create-image.html) command.

```
aws ec2 create-image \
```

```
    --instance-id i-1234567890abcdef0 \
    --name "my-web-server" \
    --description "My web server image" \
    --no-reboot
```

PowerShell

To create an AMI

Use the New-EC2Image cmdlet.

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-web-server" `
    -Description "My web server image" `
    -NoReboot $true
```

Create an AMI from a snapshot

If you have a snapshot of the root device volume of an instance, you can create an AMI from this snapshot.

> **Note**
>
> In most cases, AMIs for Windows, Red Hat, SUSE, and SQL Server require correct licensing information to be present on the AMI. For more information, see Understand AMI billing information. When creating an AMI from a snapshot, the `RegisterImage` operation derives the correct billing information from the snapshot's metadata, but this requires the appropriate metadata to be present. To verify if the correct billing information was applied, check the **Platform details** field on the new AMI. If the field is empty or doesn't match the expected operating system code (for example, Windows, Red Hat, SUSE, or SQL), the AMI creation was unsuccessful, and you should discard the AMI and follow the instructions in Create an AMI from an instance.

Console

To create an AMI from a snapshot

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

2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot from which to create the AMI, and then choose **Actions**, **Create image from snapshot**.
4. On the **Create image from snapshot** page, specify the following information:

   a. For **Image name**, enter a descriptive name for the image.

   b. For **Description**, enter a brief description for the image.

   c. For **Architecture**, choose the image architecture. Choose **i386** for 32-bit, **x86_64** for 64-bit, **arm64** for 64-bit ARM, or **x86_64** for 64-bit macOS.

   d. For **Root device name**, enter the device name to use for the root device volume. For more information, see Device names for volumes on Amazon EC2 instances.

   e. For **Virtualization type**, choose the virtualization type to be used by instances launched from this AMI. For more information, see Virtualization types.

   f. (For paravirtual virtualization only) For **Kernel ID**, select the operating system kernel for the image. If you're using a snapshot of the root device volume of an instance, select the same kernel ID as the original instance. If you're unsure, use the default kernel.

   g. (For paravirtual virtualization only) For **RAM disk ID**, select the RAM disk for the image. If you select a specific kernel, you may need to select a specific RAM disk with the drivers to support it.

   h. For **Boot mode**, choose the boot mode for the image, or choose **Use default** so that when an instance is launched with this AMI, it boots with the boot mode supported by the instance type. For more information, see Set the boot mode of an Amazon EC2 AMI.

   i. (Optional) Under **Block device mappings**, customize the root volume and add additional data volumes.

      For each volume, you can specify the size, type, performance characteristics, the behavior of delete on termination, and encryption status. For the root volume, the size can't be smaller than the size of the snapshot. For volume type, General Purpose SSD gp3 is the default selection.

   j. (Optional) Under **Tags**, you can add one or more tags to the new AMI. To add a tag, choose **Add tag**, and enter the key and value for the tag. Repeat for each tag.

   k. When you're ready to create your AMI, choose **Create image**.

5. (Windows, Red Hat, SUSE, and SQL Server only) To verify if the correct billing information was applied, check the **Platform details** field on the new AMI. If the field is empty or doesn't match the expected operating system code (for example, **Windows** or **Red Hat**), the AMI creation was unsuccessful, and you should discard the AMI and follow the instructions in Create an AMI from an instance.

AWS CLI

To create an AMI from a snapshot using the AWS CLI

Use the register-image command.

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f}
```

PowerShell

To create an AMI from a snapshot using PowerShell

Use the Register-EC2Image cmdlet.

```
$block = @{SnapshotId=snap-0db2cf683925d191f}
Register-EC2Image `
    -Name my-image `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block}
```

Create an instance store-backed AMI

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

To create an instance store-backed Linux AMI, start from an instance that you've launched from an existing instance store-backed Linux AMI. After you've customized the instance to suit your needs, bundle the volume and register a new AMI, which you can use to launch new instances with these customizations.

You can't create an instance-store backed Windows AMI because Windows AMIs don't support instance store for the root device.

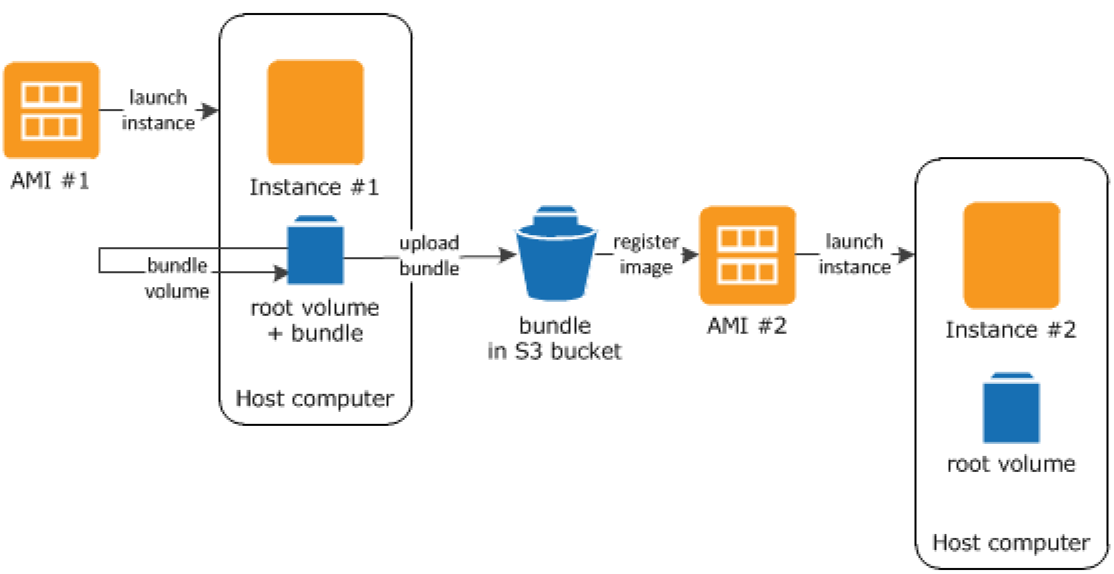
> **Important**
>
> Only the following instance types support an instance store volume as the root device and require an instance store-backed AMI: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

The AMI creation process is different for Amazon EBS-backed AMIs. For more information, see [Create an Amazon EBS-backed AMI](#).

Contents

Overview of AMI creation

The following diagram summarizes the process of creating an AMI from an instance store-backed instance.



First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, you can bundle it. It takes several minutes for the bundling process to complete. After the process completes, you have a bundle, which consists of an image manifest (`image.manifest.xml`) and files (`image.part.`*xx*) that contain a template for the root volume. Next you upload the bundle to your Amazon S3 bucket and then register your AMI.

> **Note**
>
> To upload objects to an S3 bucket for your instance store-backed Linux AMI, ACLs must be enabled for the bucket. Otherwise, Amazon EC2 will not be able to set ACLs on the objects to upload. If your destination bucket uses the bucket owner enforced setting for S3 Object Ownership, this won't work because ACLs are disabled. For more information, see Controlling ownership of objects and disabling ACLs for your bucket.

When you launch an instance using the new AMI, we create the root volume for the instance using the bundle that you uploaded to Amazon S3. The storage space used by the bundle in Amazon S3 incurs charges to your account until you delete it. For more information, see Deregister an Amazon EC2 AMI.

If you add instance store volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. For more information, see Block device mappings for volumes on Amazon EC2 instances.

Prerequisites

Before you can create an AMI, you must complete the following tasks:

- Install the AMI tools. For more information, see Set up the Amazon EC2 AMI tools.
- Install the AWS CLI. For more information, see Getting started with the AWS CLI.
- Ensure that you have an S3 bucket for the bundle, and that your bucket has ACLs enabled. For more information on configuring ACLs, see Configuring ACLs.
  - To create an S3 bucket using the AWS Management Console, open the Amazon S3 console at https://console.aws.amazon.com/s3/ and choose **Create Bucket**.

  - To create an S3 bucket with the AWS CLI, you can use the mb command. If your installed version of the AMI tools is 1.5.18 or later, you can also use the `ec2-upload-bundle` command to create the S3 bucket. For more information, see ec2-upload-bundle.
- Ensure that the files in your bundle aren't encrypted in the S3 bucket. If you require encryption for your AMI, you can use an EBS-backed AMI instead. For more information, see Use encryption with EBS-backed AMIs.
- Ensure that you have your AWS account ID. For more information, see View AWS account identifiers in the *AWS Account Management Reference Guide*.
- Ensure that you have credentials to use the AWS CLI. For more information, see Authentication and access credentials for the AWS CLI in the *AWS Command Line Interface User Guide*.
- Ensure that you have an X.509 certificate and corresponding private key.
  - If you need to create an X.509 certificate, see Manage signing certificates. The X.509 certificate and private key are used to encrypt and decrypt your AMI.
  - [China (Beijing)] Use the `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` certificate.
  - [AWS GovCloud (US-West)] Use the `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-gov.pem` certificate.
- Connect to your instance and customize it. For example, you can install software and applications, copy data, delete temporary files, and modify the Linux configuration.

Create an AMI from an Amazon Linux instance

The following procedures describe how to create an AMI from an instance store-backed instance running Amazon Linux 1. They might not work for instances running other Linux distributions.

To prepare to use the AMI tools (HVM instances only)

1. The AMI tools require GRUB Legacy to boot properly. Use the following command to install GRUB:

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

2. Install the partition management packages with the following command:

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

To create an AMI from an instance store-backed Amazon Linux instance

This procedure assumes that you have satisfied the prerequisites in Prerequisites.

In the following commands, replace each *user input placeholder* with your own information.

1. Upload your credentials to your instance. We use these credentials to ensure that only you and Amazon EC2 can access your AMI.

   a. Create a temporary directory on your instance for your credentials as follows:

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   This enables you to exclude your credentials from the created image.

   b. Copy your X.509 certificate and corresponding private key from your computer to the `/tmp/cert` directory on your instance using a secure copy tool such as scp. The `-i my-private-key.pem` option in the following **scp** command is the private key you use to connect to your instance with SSH, not the X.509 private key. For example:

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /
   path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  685     0.7KB/s   00:00
   ```

   Alternatively, because these are plain text files, you can open the certificate and key in a text editor and copy their contents into new files in `/tmp/cert`.

2. Prepare the bundle to upload to Amazon S3 by running the ec2-bundle-vol command from inside your instance. Be sure to specify the `-e` option to exclude the directory where your credentials are stored. By default, the bundle process excludes files that might contain sensitive information. These files include `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*` `*.gpg`, `*.jks`, `*/.ssh/authorized_keys`, and `*/.bash_history`. To include all of these files, use the `--no-filter` option. To include some of these files, use the `--include` option.

Important

By default, the AMI bundling process creates a compressed, encrypted collection of files in the `/tmp` directory that represents your root volume. If you do not have enough free disk space in `/tmp` to store the bundle, you need to specify a different location for the bundle to be stored with the `-d /path/to/bundle/storage` option. Some instances have ephemeral storage mounted at `/mnt` or `/media/ephemeral0` that you can use, or you can also create, attach, and mount a new Amazon EBS) volume to store the bundle. For more information, see Create an Amazon EBS volume in the *Amazon EBS User Guide*.

a. You must run the **ec2-bundle-vol** command as root. For most commands, you can use **sudo** to gain elevated permissions, but in this case, you should run **sudo -E su** to keep your environment variables.

```
[ec2-user ~]$ sudo -E su
```

Note that bash prompt now identifies you as the root user, and that the dollar sign has been replaced by a hash tag, signalling that you are in a root shell:

```
[root ec2-user]#
```

b. To create the AMI bundle, run the ec2-bundle-vol command as follows:

```
[root ec2-user]# ec2-bundle-vol -k /tmp/cert/pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 123456789012 -r x86_64 -e /tmp/cert --
partition gpt
```

Note

For the China (Beijing) and AWS GovCloud (US-West) Regions, use the `--ec2cert` parameter and specify the certificates as per the prerequisites.

It can take a few minutes to create the image. When this command completes, your `/tmp` (or non-default) directory contains the bundle (`image.manifest.xml`, plus multiple `image.part.xx` files).

c. Exit from the root shell.

```
[root ec2-user]# exit
```

3. (Optional) To add more instance store volumes, edit the block device mappings in the `image.manifest.xml` file for your AMI. For more information, see Block device mappings for volumes on Amazon EC2 instances.

   a. Create a backup of your `image.manifest.xml` file.

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

   b. Reformat the `image.manifest.xml` file so that it is easier to read and edit.

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/
image.manifest.xml
```

   c. Edit the block device mappings in `image.manifest.xml` with a text editor. The example below shows a new entry for the `ephemeral1` instance store volume.

> **Note**
>
> For a list of excluded files, see ec2-bundle-vol.

```
    <block_device_mapping>
      <mapping>
        <virtual>ami</virtual>
        <device>sda</device>
      </mapping>
      <mapping>
        <virtual>ephemeral0</virtual>
        <device>sdb</device>
      </mapping>
      <mapping>
        <virtual>ephemeral1</virtual>
```

```
      <device>sdc</device>
    </mapping>
    <mapping>
      <virtual>root</virtual>
      <device>/dev/sda1</device>
    </mapping>
  </block_device_mapping>
```

d. Save the `image.manifest.xml` file and exit your text editor.

4. To upload your bundle to Amazon S3, run the ec2-upload-bundle command as follows.

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name -
m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

> **Important**
>
> To register your AMI in a Region other than US East (N. Virginia), you must specify both the target Region with the `--region` option and a bucket path that already exists in the target Region or a unique bucket path that can be created in the target Region.

5. (Optional) After the bundle is uploaded to Amazon S3, you can remove the bundle from the `/tmp` directory on the instance using the following **rm** command:

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

> **Important**
>
> If you specified a path with the `-d` *`/path/to/bundle/storage`* option in Step 2, use that path instead of `/tmp`.

6. To register your AMI, run the register-image command as follows.

```
[ec2-user ~]$ aws ec2 register-image --image-location amzn-s3-demo-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --
virtualization-type hvm
```

> **Important**
>
> If you previously specified a Region for the ec2-upload-bundle command, specify that Region again for this command.

Set up the Amazon EC2 AMI tools

You can use the AMI tools to create and manage instance store-backed Linux AMIs. To use the tools, you must install them on your Linux instance. The AMI tools are available as both an RPM and as a .zip file for Linux distributions that don't support RPM.

To set up the AMI tools using the RPM

1. Install Ruby using the package manager for your Linux distribution, such as yum. For example:

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

2. Download the RPM file using a tool such as wget or curl. For example:

   ```
   [ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
   ```

3. Verify the RPM file's signature using the following command:

   ```
   [ec2-user ~]$ rpm -K ec2-ami-tools.noarch.rpm
   ```

   The command above should indicate that the file's SHA1 and MD5 hashes are `OK`. If the command indicates that the hashes are `NOT OK`, use the following command to view the file's Header SHA1 and MD5 hashes:

   ```
   [ec2-user ~]$ rpm -Kv ec2-ami-tools.noarch.rpm
   ```

   Then, compare your file's Header SHA1 and MD5 hashes with the following verified AMI tools hashes to confirm the file's authenticity:

   - Header SHA1: a1f662d6f25f69871104e6a62187fa4df508f880
   - MD5: 9faff05258064e2f7909b66142de6782

If your file's Header SHA1 and MD5 hashes match the verified AMI tools hashes, continue to the next step.

4. Install the RPM using the following command:

```
[ec2-user ~]$ sudo yum install ec2-ami-tools.noarch.rpm
```

5. Verify your AMI tools installation using the ec2-ami-tools-version command.

```
[ec2-user ~]$ ec2-ami-tools-version
```

> **Note**
>
> If you receive a load error such as "cannot load such file -- ec2/amitools/version (LoadError)", complete the next step to add the location of your AMI tools installation to your `RUBYLIB` path.

6. (Optional) If you received an error in the previous step, add the location of your AMI tools installation to your `RUBYLIB` path.

   a. Run the following command to determine the paths to add.

   ```
   [ec2-user ~]$ rpm -qil ec2-ami-tools | grep ec2/amitools/version
   /usr/lib/ruby/site_ruby/ec2/amitools/version.rb
   /usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
   ```

   In the above example, the missing file from the previous load error is located at `/usr/lib/ruby/site_ruby` and `/usr/lib64/ruby/site_ruby`.

   b. Add the locations from the previous step to your `RUBYLIB` path.

   ```
   [ec2-user ~]$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/site_ruby
   ```

   c. Verify your AMI tools installation using the ec2-ami-tools-version command.

   ```
   [ec2-user ~]$ ec2-ami-tools-version
   ```

To set up the AMI tools using the .zip file

1. Install Ruby and unzip using the package manager for your Linux distribution, such as **apt-get**. For example:

```
[ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. Download the .zip file using a tool such as wget or curl. For example:

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
```

3. Unzip the files into a suitable installation directory, such as `/usr/local/ec2`.

```
[ec2-user ~]$ sudo mkdir -p /usr/local/ec2
$ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
```

   Notice that the .zip file contains a folder ec2-ami-tools-*x.x.x*, where *x.x.x* is the version number of the tools (for example, `ec2-ami-tools-1.5.7`).

4. Set the `EC2_AMITOOL_HOME` environment variable to the installation directory for the tools. For example:

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. Add the tools to your `PATH` environment variable. For example:

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. You can verify your AMI tools installation using the ec2-ami-tools-version command.

```
[ec2-user ~]$ ec2-ami-tools-version
```

Manage signing certificates

Certain commands in the AMI tools require a signing certificate (also known as X.509 certificate). You must create the certificate and then upload it to AWS. For example, you can use a third-party tool such as OpenSSL to create the certificate.

To create a signing certificate

1. Install and configure OpenSSL.
2. Create a private key using the `openssl genrsa` command and save the output to a `.pem` file. We recommend that you create a 2048- or 4096-bit RSA key.

```
openssl genrsa 2048 > private-key.pem
```

3. Generate a certificate using the `openssl req` command.

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -
out certificate.pem
```

To upload the certificate to AWS, use the upload-signing-certificate command.

```
aws iam upload-signing-certificate --user-name user-name --certificate-body
 file://path/to/certificate.pem
```

To list the certificates for a user, use the list-signing-certificates command:

```
aws iam list-signing-certificates --user-name user-name
```

To disable or re-enable a signing certificate for a user, use the update-signing-certificate command. The following command disables the certificate:

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE --
status Inactive --user-name user-name
```

To delete a certificate, use the delete-signing-certificate command:

```
aws iam delete-signing-certificate --user-name user-name --certificate-
id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE
```

Amazon EC2 AMI tools reference

You can use the AMI tools commands to create and manage instance store-backed Linux AMIs. To set up the tools, see Set up the Amazon EC2 AMI tools.

For information about your access keys, see Managing access keys for IAM users in the *IAM User Guide*.

Commands

- ec2-ami-tools-version
- ec2-bundle-image
- ec2-bundle-vol
- ec2-delete-bundle
- ec2-download-bundle
- ec2-migrate-manifest
- ec2-unbundle
- ec2-upload-bundle
- Common options for AMI tools

ec2-ami-tools-version

Description

Describes the version of the AMI tools.

Syntax

`ec2-ami-tools-version`

Output

The version information.

Example

This example command displays the version information for the AMI tools that you're using.

```
[ec2-user ~]$ ec2-ami-tools-version
1.5.2 20071010
```

ec2-bundle-image

Description

Creates an instance store-backed Linux AMI from an operating system image created in a loopback file.

Syntax

ec2-bundle-image -c *path* -k *path* -u *account* -i *path* [-d *path*] [--ec2cert *path*] [-r *architecture*] [--productcodes *code1*,*code2*,...] [-B *mapping*] [-p *prefix*]

Options

`-c, --cert` *path*

The user's PEM encoded RSA public key certificate file.

Required: Yes

`-k, --privatekey` *path*

The path to a PEM-encoded RSA key file. You'll need to specify this key to unbundle this bundle, so keep it in a safe place. Note that the key doesn't have to be registered to your AWS account.

Required: Yes

`-u, --user` *account*

The user's AWS account ID, without dashes.

Required: Yes

`-i, --image` *path*

The path to the image to bundle.

Required: Yes

`-d, --destination` *path*

The directory in which to create the bundle.

Default: `/tmp`

Required: No

`--ec2cert` *path*

The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.

The `us-gov-west-1` and `cn-north-1` Regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at `/opt/aws/amitools/ec2/etc/ec2/amitools/`. If you installed the AMI tools from the RPM or ZIP file in Set up the Amazon EC2 AMI tools, the certificates are located at `$EC2_AMITOOL_HOME/etc/ec2/amitools/`.

Required: Only for the `us-gov-west-1` and `cn-north-1` Regions.

`-r, --arch` *architecture*

Image architecture. If you don't provide the architecture on the command line, you'll be prompted for it when bundling starts.

Valid values: `i386` | `x86_64`

Required: No

`--productcodes` *code1,code2,...*

Product codes to attach to the image at registration time, separated by commas.

Required: No

`-B, --block-device-mapping` *mapping*

Defines how block devices are exposed to an instance of this AMI if its instance type supports the specified device.

Specify a comma-separated list of key-value pairs, where each key is a virtual name and each value is the corresponding device name. Virtual names include the following:

- `ami`—The root file system device, as seen by the instance
- `root`—The root file system device, as seen by the kernel
- `swap`—The swap device, as seen by the instance
- `ephemeralN`—The Nth instance store volume

Required: No

`-p, --prefix` *prefix*

The filename prefix for bundled AMI files.

Default: The name of the image file. For example, if the image path is `/var/spool/my-image/version-2/debian.img`, then the default prefix is `debian.img`.

Required: No

`--kernel` *kernel_id*

Deprecated. Use register-image to set the kernel.

Required: No

`--ramdisk` *ramdisk_id*

Deprecated. Use register-image to set the RAM disk if required.

Required: No

Output

Status messages describing the stages and status of the bundling process.

Example

This example creates a bundled AMI from an operating system image that was created in a loopback file.

```
[ec2-user ~]$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -r x86_64
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
Created image.part.06
Created image.part.07
```

```
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

ec2-bundle-vol

Description

Creates an instance store-backed Linux AMI by compressing, encrypting, and signing a copy of the root device volume for the instance.

Amazon EC2 attempts to inherit product codes, kernel settings, RAM disk settings, and block device mappings from the instance.

By default, the bundle process excludes files that might contain sensitive information. These files include `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa*`, `*id_dsa*` `*.gpg`, `*.jks`, `*/.ssh/authorized_keys`, and `*/.bash_history`. To include all of these files, use the `--no-filter` option. To include some of these files, use the `--include` option.

For more information, see Create an instance store-backed AMI.

Syntax

ec2-bundle-vol -c *path* -k *path* -u *account* [-d *path*] [--ec2cert *path*] [-r *architecture*] [--productcodes *code1*,*code2*,...] [-B *mapping*] [--all] [-e *directory1*,*directory2*,...] [-i *file1*,*file2*,...] [--no-filter] [-p *prefix*] [-s *size*] [--[no-]inherit] [-v *volume*] [-P *type*] [-S *script*] [--fstab *path*] [--generate-fstab] [--grub-config *path*]

Options

`-c, --cert` *path*

The user's PEM encoded RSA public key certificate file.

Required: Yes

`-k, --privatekey` *path*

The path to the user's PEM-encoded RSA key file.

Required: Yes

`-u, --user` *account*

The user's AWS account ID, without dashes.

Required: Yes

`-d, --destination` *destination*

The directory in which to create the bundle.

Default: `/tmp`

Required: No

`--ec2cert` *path*

The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.

The `us-gov-west-1` and `cn-north-1` Regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at `/opt/aws/amitools/ec2/etc/ec2/amitools/`. If you installed the AMI tools from the RPM or ZIP file in Set up the Amazon EC2 AMI tools, the certificates are located at `$EC2_AMITOOL_HOME/etc/ec2/amitools/`.

Required: Only for the `us-gov-west-1` and `cn-north-1` Regions.

`-r, --arch` *architecture*

The image architecture. If you don't provide this on the command line, you'll be prompted to provide it when the bundling starts.

Valid values: `i386` | `x86_64`

Required: No

`--productcodes` *code1,code2,...*

Product codes to attach to the image at registration time, separated by commas.

Required: No

`-B, --block-device-mapping` *mapping*

Defines how block devices are exposed to an instance of this AMI if its instance type supports the specified device.

Specify a comma-separated list of key-value pairs, where each key is a virtual name and each value is the corresponding device name. Virtual names include the following:

- `ami`—The root file system device, as seen by the instance
- `root`—The root file system device, as seen by the kernel
- `swap`—The swap device, as seen by the instance
- `ephemeralN`—The Nth instance store volume

Required: No

`-a, --all`

Bundle all directories, including those on remotely mounted file systems.

Required: No

`-e, --exclude` *directory1,directory2,...*

A list of absolute directory paths and files to exclude from the bundle operation. This parameter overrides the `--all` option. When exclude is specified, the directories and subdirectories listed with the parameter will not be bundled with the volume.

Required: No

`-i, --include` *file1,file2,...*

A list of files to include in the bundle operation. The specified files would otherwise be excluded from the AMI because they might contain sensitive information.

Required: No

`--no-filter`

If specified, we won't exclude files from the AMI because they might contain sensitive information.

Required: No

`-p, --prefix` *prefix*

The file name prefix for bundled AMI files.

Default: `image`

Required: No

`-s, --size` *size*

The size, in MB (1024 * 1024 bytes), of the image file to create. The maximum size is 10240 MB.

Default: 10240

Required: No

`--[no-]inherit`

Indicates whether the image should inherit the instance's metadata (the default is to inherit). Bundling fails if you enable `--inherit` but the instance metadata is not accessible.

Required: No

`-v, --volume` *volume*

The absolute path to the mounted volume from which to create the bundle.

Default: The root directory (/)

Required: No

`-P, --partition` *type*

Indicates whether the disk image should use a partition table. If you don't specify a partition table type, the default is the type used on the parent block device of the volume, if applicable, otherwise the default is `gpt`.

Valid values: `mbr` | `gpt` | `none`

Required: No

`-S, --script` *script*

A customization script to be run right before bundling. The script must expect a single argument, the mount point of the volume.

Required: No

`--fstab` *path*

The path to the fstab to bundle into the image. If this is not specified, Amazon EC2 bundles /etc/fstab.

Required: No

`--generate-fstab`

Bundles the volume using an Amazon EC2-provided fstab.

Required: No

`--grub-config`

The path to an alternate grub configuration file to bundle into the image. By default, `ec2-bundle-vol` expects either `/boot/grub/menu.lst` or `/boot/grub/grub.conf` to exist on the cloned image. This option allows you to specify a path to an alternative grub configuration file, which will then be copied over the defaults (if present).

Required: No

`--kernel` *kernel_id*

Deprecated. Use register-image to set the kernel.

Required: No

`--ramdisk`*ramdisk_id*

Deprecated. Use register-image to set the RAM disk if required.

Required: No

Output

Status messages describing the stages and status of the bundling.

Example

This example creates a bundled AMI by compressing, encrypting and signing a snapshot of the local machine's root file system.

```
[ec2-user ~]$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c
 cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
  Copying / into the image file /mnt/image...
```

```
Excluding:
     sys
     dev/shm
     proc
     dev/pts
     proc/sys/fs/binfmt_misc
     dev
     media
     mnt
     proc
     sys
     tmp/image
     mnt/img-mnt
1+0 records in
1+0 records out
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
...
Created image.part.22
Created image.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
Bundle Volume complete.
```

ec2-delete-bundle

Description

Deletes the specified bundle from Amazon S3 storage. After you delete a bundle, you can't launch instances from the corresponding AMI.

Syntax

ec2-delete-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* [-t *token*] [--url *url*] [--region *region*] [--sigv *version*] [-m *path*] [-p *prefix*] [--clear] [--retry] [-y]

Options

`-b, --bucket` *bucket*

The name of the Amazon S3 bucket containing the bundled AMI, followed by an optional '/'-delimited path prefix

Required: Yes

`-a, --access-key` *access_key_id*

The AWS access key ID.

Required: Yes

`-s, --secret-key` *secret_access_key*

The AWS secret access key.

Required: Yes

`-t, --delegation-token` *token*

The delegation token to pass along to the AWS request. For more information, see [Temporary security credentials](#) in the *IAM User Guide*.

Required: Only when you are using temporary security credentials.

Default: The value of the `AWS_DELEGATION_TOKEN` environment variable (if set).

`--region`*region*

The Region to use in the request signature.

Default: `us-east-1`

Required: Required if using signature version 4

`--sigv`*version*

The signature version to use when signing the request.

Valid values: 2 | 4

Default: 4

Required: No

`-m, --manifest`*path*

The path to the manifest file.

Required: You must specify `--prefix` or `--manifest`.

`-p, --prefix` *prefix*

The bundled AMI filename prefix. Provide the entire prefix. For example, if the prefix is image.img, use `-p image.img` and not `-p image`.

Required: You must specify `--prefix` or `--manifest`.

`--clear`

Deletes the Amazon S3 bucket if it's empty after deleting the specified bundle.

Required: No

`--retry`

Automatically retries on all Amazon S3 errors, up to five times per operation.

Required: No

`-y, --yes`

Automatically assumes the answer to all prompts is yes.

Required: No

Output

Amazon EC2 displays status messages indicating the stages and status of the delete process.

Example

This example deletes a bundle from Amazon S3.

```
[ec2-user ~]$ ec2-delete-bundle -b amzn-s3-demo-bucket -a your_access_key_id -
s your_secret_access_key
Deleting files:
amzn-s3-demo-bucket/image.manifest.xml
amzn-s3-demo-bucket/image.part.00
```

```
amzn-s3-demo-bucket/image.part.01
amzn-s3-demo-bucket/image.part.02
amzn-s3-demo-bucket/image.part.03
amzn-s3-demo-bucket/image.part.04
amzn-s3-demo-bucket/image.part.05
amzn-s3-demo-bucket/image.part.06
Continue? [y/n]
y
Deleted amzn-s3-demo-bucket/image.manifest.xml
Deleted amzn-s3-demo-bucket/image.part.00
Deleted amzn-s3-demo-bucket/image.part.01
Deleted amzn-s3-demo-bucket/image.part.02
Deleted amzn-s3-demo-bucket/image.part.03
Deleted amzn-s3-demo-bucket/image.part.04
Deleted amzn-s3-demo-bucket/image.part.05
Deleted amzn-s3-demo-bucket/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

Description

Downloads the specified instance store-backed Linux AMIs from Amazon S3 storage.

Syntax

`ec2-download-bundle -b` ***`bucket`*** **`-a`** ***`access_key_id`*** **`-s`** ***`secret_access_key`*** **`-k`** ***`path`*** **`[--url`** ***`url`*** **`] [--region`** ***`region`*** **`] [--sigv`** ***`version`*** **`] [-m`** ***`file`*** **`] [-p`** ***`prefix`*** **`] [-d`** ***`directory`*** **`] [--retry]`**

Options

`-b, --bucket` *bucket*

The name of the Amazon S3 bucket where the bundle is located, followed by an optional '/'-delimited path prefix.

Required: Yes

`-a, --access-key` *access_key_id*

The AWS access key ID.

Required: Yes

`-s, --secret-key` *secret_access_key*

The AWS secret access key.

Required: Yes

`-k, --privatekey` *path*

The private key used to decrypt the manifest.

Required: Yes

`--url` *url*

The Amazon S3 service URL.

Default: `https://s3.amazonaws.com/`

Required: No

`--region` *region*

The Region to use in the request signature.

Default: `us-east-1`

Required: Required if using signature version 4

`--sigv` *version*

The signature version to use when signing the request.

Valid values: 2 | 4

Default: 4

Required: No

`-m, --manifest` *file*

The name of the manifest file (without the path). We recommend that you specify either the manifest (`-m`) or a prefix (`-p`).

Required: No

`-p, --prefix` *prefix*

The filename prefix for the bundled AMI files.

Default: `image`

Required: No

`-d, --directory` *directory*

The directory where the downloaded bundle is saved. The directory must exist.

Default: The current working directory.

Required: No

`--retry`

Automatically retries on all Amazon S3 errors, up to five times per operation.

Required: No

Output

Status messages indicating the various stages of the download process are displayed.

Example

This example creates the `bundled` directory (using the Linux **mkdir** command) and downloads the bundle from the `amzn-s3-demo-bucket` Amazon S3 bucket.

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/bundles/bundle_name
 -m image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from amzn-s3-demo-bucket to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.00 ...
Downloaded image.part.00 from amzn-s3-demo-bucket
Downloading part image.part.01 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.01 ...
Downloaded image.part.01 from amzn-s3-demo-bucket
Downloading part image.part.02 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.02 ...
Downloaded image.part.02 from amzn-s3-demo-bucket
Downloading part image.part.03 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.03 ...
```

```
Downloaded image.part.03 from amzn-s3-demo-bucket
Downloading part image.part.04 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.04 ...
Downloaded image.part.04 from amzn-s3-demo-bucket
Downloading part image.part.05 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.05 ...
Downloaded image.part.05 from amzn-s3-demo-bucket
Downloading part image.part.06 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.06 ...
Downloaded image.part.06 from amzn-s3-demo-bucket
```

ec2-migrate-manifest

Description

Modifies an instance store-backed Linux AMI (for example, its certificate, kernel, and RAM disk) so that it supports a different Region.

Syntax

ec2-migrate-manifest -c *path* -k *path* -m *path* {(-a *access_key_id* -s *secret_access_key* --region *region*) | (--no-mapping)} [--ec2cert *ec2_cert_path*] [--kernel *kernel-id*] [--ramdisk *ramdisk_id*]

Options

`-c, --cert` *path*

The user's PEM encoded RSA public key certificate file.

Required: Yes

`-k, --privatekey` *path*

The path to the user's PEM-encoded RSA key file.

Required: Yes

`--manifest` *path*

The path to the manifest file.

Required: Yes

`-a, --access-key` *access_key_id*

The AWS access key ID.

Required: Required if using automatic mapping.

`-s, --secret-key` *secret_access_key*

The AWS secret access key.

Required: Required if using automatic mapping.

`--region` *region*

The Region to look up in the mapping file.

Required: Required if using automatic mapping.

`--no-mapping`

Disables automatic mapping of kernels and RAM disks.

During migration, Amazon EC2 replaces the kernel and RAM disk in the manifest file with a kernel and RAM disk designed for the destination region. Unless the `--no-mapping` parameter is given, `ec2-migrate-bundle` might use the `DescribeRegions` and `DescribeImages` operations to perform automated mappings.

Required: Required if you're not providing the `-a`, `-s`, and `--region` options used for automatic mapping.

`--ec2cert` *path*

The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.

The `us-gov-west-1` and `cn-north-1` Regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at `/opt/aws/amitools/ec2/etc/ec2/amitools/`. If you installed the AMI tools from the ZIP file in Set up the Amazon EC2 AMI tools, the certificates are located at `$EC2_AMITOOL_HOME/etc/ec2/amitools/`.

Required: Only for the `us-gov-west-1` and `cn-north-1` Regions.

`--kernel` *kernel_id*

The ID of the kernel to select.

> **Important**
>
> We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see User provided kernels in the *Amazon Linux 2 User Guide*.

Required: No

`--ramdisk` *ramdisk_id*

The ID of the RAM disk to select.

> **Important**
>
> We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see User provided kernels in the *Amazon Linux 2 User Guide*.

Required: No

Output

Status messages describing the stages and status of the bundling process.

Example

This example copies the AMI specified in the `my-ami.manifest.xml` manifest from the US to the EU.

```
[ec2-user ~]$ ec2-migrate-manifest --manifest my-ami.manifest.xml
 --cert cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --region eu-west-1

Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-west-1.
```

ec2-unbundle

Description

Re-creates the bundle from an instance store-backed Linux AMI.

Syntax

ec2-unbundle -k *path* -m *path* [-s *source_directory*] [-d *destination_directory*]

Options

`-k, --privatekey` *path*

The path to your PEM-encoded RSA key file.

Required: Yes

`-m, --manifest` *path*

The path to the manifest file.

Required: Yes

`-s, --source` *source_directory*

The directory containing the bundle.

Default: The current directory.

Required: No

`-d, --destination` *destination_directory*

The directory in which to unbundle the AMI. The destination directory must exist.

Default: The current directory.

Required: No

Example

This Linux and UNIX example unbundles the AMI specified in the `image.manifest.xml` file.

```
[ec2-user ~]$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -s mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

Output

Status messages indicating the various stages of the unbundling process are displayed.

ec2-upload-bundle

Description

Uploads the bundle for an instance store-backed Linux AMI to Amazon S3 and sets the appropriate access control lists (ACLs) on the uploaded objects. For more information, see [Create an instance store-backed AMI](#).

> **Note**
>
> To upload objects to an S3 bucket for your instance store-backed Linux AMI, ACLs must be enabled for the bucket. Otherwise, Amazon EC2 will not be able to set ACLs on the objects to upload. If your destination bucket uses the bucket owner enforced setting for S3 Object Ownership, this won't work because ACLs are disabled. For more information, see [Controlling ownership of objects and disabling ACLs for your bucket](#).

Syntax

ec2-upload-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* [-t *token*] -m *path* [--url *url*] [--region *region*] [--sigv *version*] [--acl *acl*] [-d *directory*] [--part *part*] [--retry] [--skipmanifest]

Options

`-b, --bucket` *bucket*

The name of the Amazon S3 bucket in which to store the bundle, followed by an optional '/'-delimited path prefix. If the bucket doesn't exist, it's created if the bucket name is available. Additionally, if the bucket doesn't exist and the AMI tools version is 1.5.18 or later, this command sets the ACLs for the bucket.

Required: Yes

`-a, --access-key` *access_key_id*

Your AWS access key ID.

Required: Yes

`-s, --secret-key` *secret_access_key*

Your AWS secret access key.

Required: Yes

`-t, --delegation-token` *token*

The delegation token to pass along to the AWS request. For more information, see [Temporary security credentials](#) in the *IAM User Guide*.

Required: Only when you are using temporary security credentials.

Default: The value of the `AWS_DELEGATION_TOKEN` environment variable (if set).

`-m, --manifest` *path*

The path to the manifest file. The manifest file is created during the bundling process and can be found in the directory containing the bundle.

Required: Yes

`--url` *url*

Deprecated. Use the `--region` option instead unless your bucket is constrained to the `EU` location (and not `eu-west-1`). The `--location` flag is the only way to target that specific location restraint.

The Amazon S3 endpoint service URL.

Default: `https://s3.amazonaws.com/`

Required: No

`--region` *region*

The Region to use in the request signature for the destination S3 bucket.

- If the bucket doesn't exist and you don't specify a Region, the tool creates the bucket without a location constraint (in `us-east-1`).
- If the bucket doesn't exist and you specify a Region, the tool creates the bucket in the specified Region.
- If the bucket exists and you don't specify a Region, the tool uses the bucket's location.
- If the bucket exists and you specify `us-east-1` as the Region, the tool uses the bucket's actual location without any error message, any existing matching files are over-written.

- If the bucket exists and you specify a Region (other than `us-east-1`) that doesn't match the bucket's actual location, the tool exits with an error.

If your bucket is constrained to the EU location (and not `eu-west-1`), use the `--location` flag instead. The `--location` flag is the only way to target that specific location restraint.

Default: `us-east-1`

Required: Required if using signature version 4

`--sigv` *version*

The signature version to use when signing the request.

Valid values: `2` **|** `4`

Default: `4`

Required: No

`--acl` *acl*

The access control list policy of the bundled image.

Valid values: `public-read` **|** `aws-exec-read`

Default: `aws-exec-read`

Required: No

`-d, --directory` *directory*

The directory containing the bundled AMI parts.

Default: The directory containing the manifest file (see the `-m` option).

Required: No

`--part` *part*

Starts uploading the specified part and all subsequent parts. For example, `--part 04`.

Required: No

`--retry`

Automatically retries on all Amazon S3 errors, up to five times per operation.

Required: No

`--skipmanifest`

Does not upload the manifest.

Required: No

`--location` *location*

Deprecated. Use the `--region` option instead, unless your bucket is constrained to the EU location (and not `eu-west-1`). The `--location` flag is the only way to target that specific location restraint.

The location constraint of the destination Amazon S3 bucket. If the bucket exists and you specify a location that doesn't match the bucket's actual location, the tool exits with an error. If the bucket exists and you don't specify a location, the tool uses the bucket's location. If the bucket doesn't exist and you specify a location, the tool creates the bucket in the specified location. If the bucket doesn't exist and you don't specify a location, the tool creates the bucket without a location constraint (in `us-east-1`).

Default: If `--region` is specified, the location is set to that specified Region. If `--region` is not specified, the location defaults to `us-east-1`.

Required: No

Output

Amazon EC2 displays status messages that indicate the stages and status of the upload process.

Example

This example uploads the bundle specified by the `image.manifest.xml` manifest.

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundles/bundle_name -m
 image.manifest.xml -a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket amzn-s3-demo-bucket ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
```

```
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

Common options for AMI tools

Most of the AMI tools accept the following optional parameters.

`--help, -h`

Displays the help message.

`--version`

Displays the version and copyright notice.

`--manual`

Displays the manual entry.

`--batch`

Runs in batch mode, suppressing interactive prompts.

`--debug`

Displays information that can be useful when troubleshooting problems.

Convert your instance store-backed AMI to an EBS-backed AMI

You can convert an instance store-backed Linux AMI that you own to an Amazon EBS-backed Linux AMI.

> **Important**
>
> You can't convert an AMI that you don't own.

To convert an instance store-backed AMI to an Amazon EBS-backed AMI

1. Launch an Amazon Linux instance from an Amazon EBS-backed AMI. For more information, see Launch an EC2 instance using the launch instance wizard in the console. Amazon Linux instances have the AWS CLI and AMI tools pre-installed.
2. Upload the X.509 private key that you used to bundle your instance store-backed AMI to your instance. We use this key to ensure that only you and Amazon EC2 can access your AMI.

   a. Create a temporary directory on your instance for your X.509 private key as follows:

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   b. Copy your X.509 private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as scp. The *`my-private-key`* parameter in the following command is the private key you use to connect to your instance with SSH. For example:

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   ```

3. Configure your environment variables to use the AWS CLI. For more information, see Environment variables.

   a. (Recommended) Set environment variables for your AWS access key, secret key, and session token.

   ```
   [ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
   [ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
   [ec2-user ~]$ export AWS_SESSION_TOKEN=your_session_token
   ```

   b. Set environment variables for your AWS access key, and secret key.

   ```
   [ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
   [ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
   ```

4. Prepare an Amazon Elastic Block Store (Amazon EBS) volume for your new AMI.

   a. Create an empty EBS volume in the same Availability Zone as your instance using the create-volume command. Note the volume ID in the command output.

> **Important**
>
> This EBS volume must be the same size or larger than the original instance store root volume.

```
aws ec2 create-volume \
    --size 10 \
    --region us-west-2 \
    --availability-zone us-west-2b
```

b. Attach the volume to your Amazon EBS-backed instance using the attach-volume command.

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --region us-west-2
```

5. Create a folder for your bundle.

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

6. Download the bundle for your instance store-based AMI to `/tmp/bundle` using the ec2-download-bundle command.

```
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name
 -m image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --
privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. Reconstitute the image file from the bundle using the ec2-unbundle command.

   a. Change directories to the bundle folder.

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

   b. Run the ec2-unbundle command.

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. Copy the files from the unbundled image to the new EBS volume.

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

9. Probe the volume for any new partitions that were unbundled.

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

10. List the block devices to find the device name to mount.

```
[ec2-user bundle]$ lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/sda     202:0    0   8G  0 disk
##/dev/sda1 202:1    0   8G  0 part /
/dev/sdb     202:80   0  10G  0 disk
##/dev/sdb1 202:81   0  10G  0 part
```

In this example, the partition to mount is `/dev/sdb1`, but your device name will likely be different. If your volume is not partitioned, then the device to mount will be similar to `/dev/sdb` (without a device partition trailing digit).

11. Create a mount point for the new EBS volume and mount the volume.

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. Open the `/etc/fstab` file on the EBS volume with your favorite text editor (such as **vim** or **nano**) and remove any entries for instance store (ephemeral) volumes. Because the EBS volume is mounted on `/mnt/ebs`, the `fstab` file is located at `/mnt/ebs/etc/fstab`.

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/     /           ext4    defaults,noatime  1   1
tmpfs       /dev/shm    tmpfs   defaults        0   0
devpts      /dev/pts    devpts  gid=5,mode=620  0   0
sysfs       /sys        sysfs   defaults        0   0
proc        /proc       proc    defaults        0   0
```

```
/dev/sdb         /media/ephemeral0       auto     defaults,comment=cloudconfig    0
     2
```

In this example, the last line should be removed.

13. Unmount the volume and detach it from the instance.

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id vol-01234567890abcdef --
region us-west-2
```

14. Create an AMI from the new EBS volume as follows.

    a. Create a snapshot of the new EBS volume.

    ```
    [ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description
     "your_snapshot_description" --volume-id vol-01234567890abcdef
    ```

    b. Check to see that your snapshot is complete.

    ```
    [ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
    id snap-0abcdef1234567890
    ```

    c. Identify the processor architecture, virtualization type, and the kernel image (`aki`) used on the original AMI with the **describe-images** command. You need the AMI ID of the original instance store-backed AMI for this step.

    ```
    [ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-
    id ami-0abcdef1234567890 --output text
    IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon
     available public machine aki-fc8f11cc instance-store paravirtual xen
    ```

    In this example, the architecture is `x86_64` and the kernel image ID is `aki-fc8f11cc`. Use these values in the following step. If the output of the above command also lists an `ari` ID, take note of that as well.

    d. Register your new AMI with the snapshot ID of your new EBS volume and the values from the previous step. If the previous command output listed an `ari` ID, include that in the following command with `--ramdisk-id` *`ari_id`*.

    ```
    [ec2-user bundle]$ aws ec2 register-image --region us-west-2 --
    name your_new_ami_name --block-device-mappings DeviceName=device-
    ```

```
name,Ebs={SnapshotId=snap-0abcdef1234567890} --virtualization-type paravirtual
 --architecture x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
```

15. (Optional) After you have tested that you can launch an instance from your new AMI, you can delete the EBS volume that you created for this procedure.

```
aws ec2 delete-volume --volume-id vol-01234567890abcdef
```

Create an Amazon EC2 AMI using Windows Sysprep

The Microsoft System Preparation (Windows Sysprep) tool creates a generalized version of the operating system, with instance-specific system configuration removed before it captures a new 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.

You can also use Windows Sysprep to create a standardized AMI using the Windows launch agents. For more information, see the section called “Use Windows Sysprep with a launch agent”.

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

To troubleshoot Windows Sysprep, see Troubleshoot Sysprep issues with Amazon EC2 Windows instances.

Contents

Windows Sysprep phases

Windows Sysprep runs through the following phases:

- **Generalize**: The Sysprep tool removes image-specific information and configurations. For example, Windows 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 Windows Sysprep with the Windows launch agents, the system prevents drivers from being removed because `PersistAllDeviceInstalls` is set to true by default.

- **Specialize**: Plug and Play scans the computer and installs drivers for any detected devices. The Sysprep 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 Windows Sysprep with Windows launch agents, the answer file automates this phase.

Before you begin

- Before performing Windows Sysprep, we recommend that you remove all local user accounts and all account profiles other than a single administrator account under which Windows Sysprep will be run. If you perform Windows Sysprep with additional accounts and profiles, unexpected behavior could result, including loss of profile data or failure to complete Windows Sysprep.
- Learn more about Sysprep Overview.
- Learn which Sysprep Support for Server Roles.

Use Windows Sysprep with a launch agent

You can use Windows Sysprep to create a standardized Amazon Machine Image (AMI) when you start with an AMI that has one of the Windows launch agents installed.

Use Windows Sysprep with EC2Launch v2

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

Windows Sysprep with EC2Launch v2 topics

- Windows Sysprep actions
- Post Sysprep
- Run Windows Sysprep with EC2Launch v2

Windows Sysprep actions

Windows 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:

   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 Windows 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 Windows Sysprep by running the following command:

   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.
- Windows 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**: Windows 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 Windows 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`. Windows 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 Windows Sysprep completes if you did not configure the `setAdminAccount` task.

  The system runs the following command from your local launch agent directory (`C:\Program Files\Amazon\EC2Launch\`).

  ```
  EC2Launch.exe internal randomize-password --username Administrator
  ```

- To enable remote desktop connections, the system sets the Terminal Server `fDenyTSConnections` registry key to false.

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>`
   - `<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 Windows 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 Use the EC2Launch v2 agent to perform tasks during EC2 Windows instance launch.

Run Windows Sysprep with EC2Launch v2

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

1. In the Amazon EC2 console, locate an AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize settings
   a. 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 Configure EC2Launch v2 settings for Windows instances.
   b. If you've made changes, choose **Save** before you shut down.
4. Choose **Shutdown with Sysprep** or **Shutdown without Sysprep**.

When you are asked to confirm that you want to run Windows Sysprep and shut down the instance, click **Yes**. EC2Launch v2 runs Windows 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 Windows Sysprep tool from the command line using the following command:

```
"%programfiles%\amazon\ec2launch\ec2launch.exe" sysprep --shutdown=true
```

Use Windows Sysprep with EC2Launch

EC2Launch offers a default answer file and batch files for Windows 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 Windows Sysprep to create an instance backup. Windows Sysprep removes system-specific information. If you remove this information there might be unintended consequences for an instance backup.

Windows Sysprep with EC2Launch topics

- EC2Launch answer and batch files for Windows Sysprep
- Run Windows Sysprep with EC2Launch
- Update metadata/KMS routes for Server 2016 and later when launching a custom AMI

EC2Launch answer and batch files for Windows Sysprep

The EC2Launch answer file and batch files for Windows 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 Windows Sysprep.

`SysprepSpecialize.cmd`

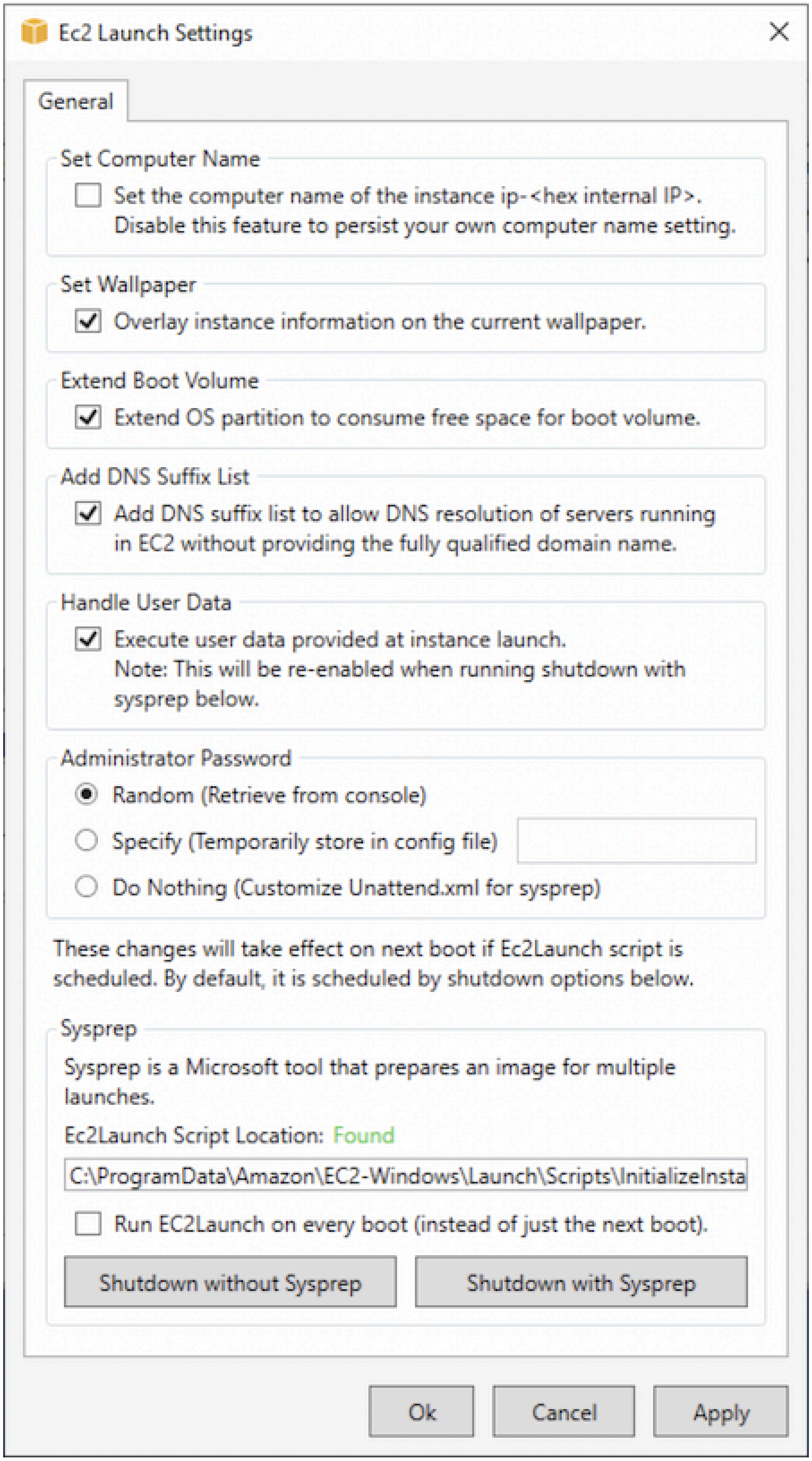
Customize this batch file to run commands during the Windows Sysprep specialize phase.

Run Windows Sysprep with EC2Launch

On the full installation of Windows Server 2016 and later (with a desktop experience), you can run Windows Sysprep with EC2Launch manually or by using the **EC2 Launch Settings** application.

To run Windows 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`.



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 Windows 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 Windows 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, EC2Lauch generates a random password instead. The password is stored in `LaunchConfig.json` as clear text and is deleted after Windows 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 Windows Sysprep.
- Run EC2LaunchSettings and shut down without Windows 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.

> **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".

Use Windows Sysprep with EC2Config

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

Windows Sysprep with EC2Config topics

Windows Sysprep actions

Windows 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 Windows Sysprep automatically. You do not need to change this value if you are running Windows 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 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:

```
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 Windows 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 Windows Sysprep by running the following command:

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

Generalize phase

- The tool removes image-specific information and configurations such as the computer name and the SID. If the instance is a member of a domain, it is removed from the domain. The `sysprep2008.xml` answer file includes the following settings 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.
- Windows 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**: Windows 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 Windows 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. Windows 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 /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 Windows 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 Windows 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 Use the EC2Config service to perform tasks during EC2 legacy Windows operating system instance launch.

Run Windows Sysprep with the EC2Config service

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

1. In the Amazon EC2 console, locate or create 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.
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 Windows Sysprep answer file in unencrypted form (clear text). When Windows 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 Windows 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 Windows Sysprep and shut down the instance, click **Yes**. You'll notice that EC2Config runs Windows 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 Windows 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. For an example of configuring and then running Windows Sysprep from the command line, see `Ec2ConfigService\Scripts\InstallUpdates.ps1`.

Copy an Amazon EC2 AMI

When you need a consistent Amazon EC2 instance configuration across multiple Regions, you can use a single Amazon Machine Image (AMI) as your template to launch all the instances. However, AMIs are Region-specific resources—to launch an instance in a specific AWS Region, the AMI must be located in that Region. Therefore, to use the same AMI in multiple Regions, you must copy it from the source Region to each target Region.

The method you use to copy an AMI depends on whether you're copying across Regions *within the same partition* or *across different partitions*:

- **Cross-Region copying** – Copy AMIs across Regions *within the same partition*, for example, across the Regions within the commercial partition. This copy method is described in this topic.
- **Cross-partition copying** – Copy AMIs *from one partition to another partition*, for example, from the commercial partition to the AWS GovCloud (US) partition. For information about this copy method, see Store and restore an AMI.
- **Cross-account copying** – Create a copy of an AMI that another AWS account has shared with your AWS account. This copy method is described in this topic.

The time taken to complete the copy operation for cross-Region and cross-account AMI copying is on a best-effort basis. If you need control over the completion time, you can specify a completion window ranging from 15 minutes to 48 hours, ensuring your AMI is copied within your required timeframe. Additional charges apply for time-based AMI copy operations. For more information, see Time-based copies in the *Amazon EBS User Guide*.

Contents

Considerations

- **Permission to copy AMIs –** You can use IAM policies to grant or deny users permission to copy AMIs. Starting October 28, 2024, you can specify resource-level permissions for the `CopyImage` action on the source AMI. Resource-level permissions for the new AMI are available as before.
- **Launch permissions and Amazon S3 bucket permissions –** AWS does not copy launch permissions or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions and Amazon S3 bucket permissions to the new AMI.
- **Tags –** You can only copy user-defined AMI tags that you attached to the source AMI. System tags (prefixed with `aws:`) and user-defined tags that are attached by other AWS accounts will not be copied. When copying an AMI, you can attach new tags to the new AMI and its backing snapshots.
- **Quotas for time-based AMI copies –** After you reach your *cumulative snapshot copy throughput quota*, subsequent time-based AMI copy requests fail. For more information, see Quotas for time-based copies in the *Amazon EBS User Guide*.
- **Supported source-destination copies –** The location of the source AMI determines whether you can copy it and the allowed destinations for the new AMI:
  - If the source AMI is in a Region, you can copy it within that Region, to another Region, to an Outpost associated with that Region, or to a Local Zone in that Region.
  - If the source AMI is in a Local Zone, you can copy it within that Local Zone, to the parent Region of that Local Zone, or to certain other Local Zones with the same parent Region.
  - If the source AMI is on an Outpost, you can't copy it.
- **CLI parameters for source and destination –** When using the CLI, the following parameters are supported for specifying the source location of the AMI to copy and the destination of the new AMI. Note that the copy operation must be initiated in the destination Region; if you omit the

`--region` parameter, the destination assumes the default Region configured in your AWS CLI settings.

Source to destination	Source parameter	Destination parameter
Region to Region	`--source-region`	`--region`
Region to Outpost	`--source-region`	`--destination-outpost-arn` (the ARN of the Outpost)
Region to Local Zone	`--source-region` Must be the parent Region of the Local Zone.	`--destination-availability-zone` (the name of the Local Zone) or `--destination-availability-zone-id` (the ID of the Local Zone)
Local Zone to Region	`--source-region` Must be the parent Region of the Local Zone. The source Local Zone is assumed from the location of the specified source AMI ID.	`--region` Must be the parent Region of the Local Zone.
Local Zone to Local Zone	`--source-region` Must be the parent Region of the Local Zone. The source Local Zone is assumed from the location of the specified source AMI ID.	`--destination-availability-zone` (the name of the Local Zone) or `--destination-availability-zone-id` (the ID of the Local Zone)

Costs

There is no charge for copying an AMI when no completion time is specified. However, additional charges apply for time-based AMI copy operations. For more information, see Time-based copies in the *Amazon EBS User Guide*.

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.

Grant permissions to copy Amazon EC2 AMIs

To copy an EBS-backed or instance store-backed AMI, you need the following IAM permissions:

- `ec2:CopyImage` – To copy the AMI. For EBS-backed AMIs, it also grants permission to copy the AMI's backing snapshots.
- `ec2:CreateTags` – To tag the target AMI. For EBS-backed AMIs, it also grants permission to tag the target AMI’s backing snapshots.

If you're copying an instance stored-backed AMI, you need the following *additional* IAM permissions:

- `s3:CreateBucket` – To create the S3 bucket in the target Region for the new AMI
- `s3:GetBucketAcl` – To read the ACL permissions for the source bucket
- `s3:ListAllMyBuckets` – To find an existing S3 bucket for AMIs in the target Region
- `s3:GetObject` – To read the objects in the source bucket
- `s3:PutObject` – To write the objects in the target bucket
- `s3:PutObjectAcl` – To write the permissions for the new objects in the target bucket

> **Note**
>
> Starting October 28, 2024, you can specify resource-level permissions for the `CopyImage` action on the source AMI. Resource-level permissions for the target AMI are available as before. For more information, see **CopyImage** in the table under Actions defined by Amazon EC2 in the *Service Authorization Reference*.

Example IAM policy for copying an EBS-backed AMI and tagging the target AMI and snapshots

The following example policy grants you permission to copy any EBS-backed AMI and tag the target AMI and its backing snapshots.

> **Note**
>
> Starting October 28, 2024, you can specify snapshots in the `Resource` element. For more information, see **CopyImage** in the table under [Actions defined by Amazon EC2](#) in the *Service Authorization Reference*.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "PermissionToCopyAllImages",
        "Effect": "Allow",
        "Action": [
            "ec2:CopyImage",
            "ec2:CreateTags"
        ],
        "Resource": [
            "arn:aws:ec2:*::image/*",
            "arn:aws:ec2:*::snapshot/*"
        ]
    }]
}
```

Example IAM policy for copying an EBS-backed AMI but denying tagging the new snapshots

The `ec2:CopySnapshot` permission is automatically granted when you get the `ec2:CopyImage` permission. Permission to tag the new backing snapshots can be explicitly denied, overriding the `Allow` effect for the `ec2:CreateTags` action.

The following example policy grants you permission to copy any EBS-backed AMI, but denies you from tagging the new backing snapshots of the target AMI.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:::snapshot/*"
        }
    ]
}
```

Example IAM policy for copying an instance store-backed AMI and tagging the target AMI

The following example policy grants you permission to copy any instance store-backed AMI in the specified source bucket to the specified Region, and tag the target AMI.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "PermissionToCopyAllImages",
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amis-for-111122223333-in-us-east-2-hash"
            ]
        }
    ]
}
```

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 you copy 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.

Copy an AMI

You can copy an AMI that you own or an AMI that was shared with you from another account. For the supported source and destination combinations, see Considerations.

Console

To copy an AMI

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.
3. In the navigation pane, choose **AMIs** to display the list of AMIs available to you in the Region.
4. If you don't see the AMI you want to copy, choose a different filter. You can filter by AMIs **Owned by me**, **Private images**, **Public images**, and **Disabled images**.
5. Select the AMI to copy, and then choose **Actions**, **Copy AMI**.
6. On the **Copy Amazon Machine Image (AMI)** page, specify the following information:

    a. **AMI copy name**: A name for the new AMI. You can include the operating system information in the name because Amazon EC2 does not provide this information when displaying details about the AMI.

    b. **AMI copy 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.

    c. **Destination Region**: The Region in which to copy the AMI. For more information, see Cross-Region copying and Cross-account copying.

    d. **Copy tags**: Select this checkbox to include your user-defined AMI tags when copying the AMI. System tags (prefixed with `aws:`) and user-defined tags that are attached by other AWS accounts will not be copied.

    e. **Time-based copy**: You can specify whether the copy operation completes within a specific timeframe or on a best-effort basis, as follows:

        - To complete the copy within a specific timeframe:
            - Select **Enable time-based copy**.

    - For **Completion duration**, enter the number of minutes (in 15-minute increments) allowed for the copy operation. The completion duration applies to all snapshots associated with the AMI.

      For more information, see [Time-based copies](#) in the *Amazon EBS User Guide*.

  - To complete the copy on a best-effort basis:
    - Leave **Enable time-based copy** unselected.

f. (EBS-backed AMIs only) **Encrypt EBS snapshots of AMI copy**: Select this checkbox to encrypt the target snapshots, or to re-encrypt them using a different key. If encryption by default is enabled, the **Encrypt EBS snapshots of AMI copy** checkbox is selected and cannot be cleared. For more information, see [Encryption and copying](#).

g. (EBS-backed AMIs only) **KMS key**: The KMS key to used to encrypt the target snapshots.

h. **Tags**: You can tag the new AMI and the new snapshots with the same tags, or you can tag them with different tags.

  - To tag the new AMI and the new snapshots with the *same* tags, choose **Tag image and snapshots together**. The same tags are applied to the new AMI and every snapshot that is created.
  - To tag the new AMI and the new snapshots with *different* tags, choose **Tag image and snapshots separately**. Different tags are applied to the new AMI and the snapshots that are created. Note, however, that all the new snapshots that are created get the same tags; you can't tag each new snapshot with a different tag.

  To add a tag, choose **Add tag**, and enter the key and value for the tag. Repeat for each tag.

i. When you're ready to copy the AMI, choose **Copy AMI**.

  The initial status of the new AMI is `Pending`. The AMI copy operation is complete when the status is `Available`.

AWS CLI

To copy an AMI from one Region to another Region

Use 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 the `--region` parameter (or omit this parameter to assume the default Region configured in your AWS CLI settings).

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region us-west-2 \
    --name my-ami \
    --region us-east-1
```

When you encrypt a target snapshot during AMI copy, you must specify these additional parameters: `--encrypted` and `--kms-key-id`.

To copy an AMI from a Region to a Local Zone

Use the copy-image command. You must specify both the source and destination. You specify the source Region using the `--source-region` parameter. You specify the destination Local Zone using the `--destination-availability-zone` parameter (you can use `--destination-availability-zone-id` instead). Note that you can only copy an AMI from a Region to a Local Zone within that same Region.

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --destination-availability-zone cn-north-1-pkx-1a \
    --name my-ami \
    --region cn-north-1
```

To copy an AMI from a Local Zone to a Region

Use the copy-image command. You must specify both the source and destination. You specify the source Region using the `--source-region` parameter. You specify the destination Region using the `--region` parameter (or omit this parameter to assume the default Region configured in your AWS CLI settings). The source Local Zone is assumed from the location of the specified source AMI ID. Note that you can only copy an AMI from a Local Zone to its parent Region.

```
aws ec2 copy-image \
```

```
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --name my-ami \
    --region cn-north-1
```

To copy an AMI from one Local Zone to another Local Zone

Use the copy-image command. You must specify both the source and destination. You specify the source Region of the Local Zone using the `--source-region` parameter. You specify the destination Local Zone using the `--destination-availability-zone` parameter (you can use `--destination-availability-zone-id` instead). The source Local Zone is assumed from the location of the specified source AMI ID. You specify the parent Region of the destination Local Zone using the `--region` parameter (or omit this parameter to assume the default Region configured in your AWS CLI settings).

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --destination-availability-zone cn-north-1-pkx-1a \
    --name my-ami \
    --region cn-north-1
```

PowerShell

To copy an AMI from one Region to another Region

Use the Copy-EC2Image cmdlet. 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 the `-Region` parameter or the Set-AWSDefaultRegion cmdlet.

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion us-west-2 `
    -Name my-ami `
    -Region us-east-1
```

When you encrypt a target snapshot during AMI copy, you must specify these additional parameters: `-Encrypted` and `-KmsKeyId`.

To copy an AMI from a Region to a Local Zone

Use the Copy-EC2Image cmdlet. You must specify both the source and destination. You specify the source Region using the -SourceRegion parameter. You specify the destination Local Zone using the -DestinationAvailabilityZone parameter (you can use -DestinationAvailabilityZoneId instead). Note that you can only copy an AMI from a Region to a Local Zone within that same Region.

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -DestinationAvailabilityZone cn-north-1-pkx-1a `
    -Name my-ami `
    -Region cn-north-1
```

To copy an AMI from a Local Zone to a Region

Use the Copy-EC2Image cmdlet. You must specify both the source and destination. You specify the source Region using the -SourceRegion parameter. You specify the destination Region using the -Region parameter or the Set-AWSDefaultRegion cmdlet. The source Local Zone is assumed from the location of the specified source AMI ID. Note that you can only copy an AMI from a Local Zone to its parent Region.

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -Name my-ami `
    -Region cn-north-1
```

To copy an AMI from one Local Zone to another Local Zone

Use the Copy-EC2Image cmdlet. You must specify both the source and destination. You specify the source Region of the Local Zone using the -SourceRegion parameter. You specify the destination Local Zone using the -DestinationAvailabilityZone parameter (you can use -DestinationAvailabilityZoneId instead). The source Local Zone is assumed from the location of the specified source AMI ID. You specify the parent Region of the destination Local Zone using the -Region parameter or the Set-AWSDefaultRegion cmdlet.

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -DestinationAvailabilityZone cn-north-1-pkx-1a `
```

```
    -Name my-ami `
    -Region cn-north-1
```

Stop a pending AMI copy operation

You can stop a pending AMI copy using the following procedures.

Console

To stop an AMI copy operation

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 then choose **Actions**, **Deregister AMI**.
5. When asked for confirmation, choose **Deregister AMI**.

AWS CLI

To stop an AMI copy operation

Use the deregister-image command.

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

PowerShell

To stop an AMI copy operation using

Use the Unregister-EC2Image cmdlet.

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

How Amazon EC2 AMI copy works

Copying a source AMI results in an identical but distinct new AMI that we also refer to as the *target* AMI. The target AMI has its own unique AMI ID. You can change or deregister the source AMI with no effect on the target AMI. The reverse is also true.

With an 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.

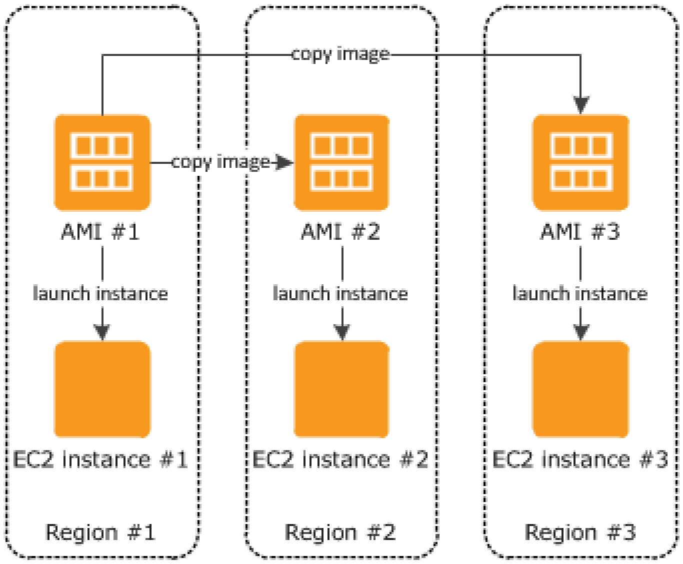
Contents

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 relationship between 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 might still use the resources from the source Region, which can impact performance and cost.

Limitations

- Destination Regions are limited to 300 concurrent AMI copy operations. This also applies to time-based AMI copy operations.
- You can't copy a paravirtual (PV) AMI to a Region that does not support PV AMIs. For more information, see Virtualization types.

Cross-account copying

If an AMI from another AWS account is shared with your AWS account, you can copy the shared AMI. This is known as cross-account copying. The AMI that is shared with you is the source AMI.

When you copy the source AMI, you create a new AMI. The new AMI is often referred to as the target AMI.

AMI costs

- For a shared AMI, the account of the shared AMI is charged for the storage in the Region.
- If you copy an AMI that is 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.
  - 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, not just for the AMI itself. The storage is 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. For more information about granting resource permissions, for EBS snapshots, see Share an Amazon EBS snapshot with other AWS accounts in the *Amazon EBS User Guide*, and for S3 buckets, see Identity and access management for Amazon S3 in the *Amazon S3 User Guide*.

> **Note**
>
> Tags that are attached to the source AMI are not copied across accounts to the target AMI.

Time-based AMI copy operations

When you initiate a time-based AMI copy operation for an EBS-backed AMI with a single associated snapshot, it behaves in the same way as an **individual time-based snapshot copy operation**, and the same throughput limitations apply.

When you initiate a time-based AMI copy operation for an EBS-backed AMI with a multiple associated snapshots, it behaves in the same way as **concurrent time-based snapshot copy operations**, and the same throughput limitations apply. Each associated snapshot results in a separate snapshot copy request, each of which contributes to your cumulative snapshot copy throughput quota. The completion duration that you specify applies to each associated snapshot.

For more information, see [Time-based copies] in the *Amazon EBS User Guide*.

Encryption and copying

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

Scenario	Description	Supported
1	Unencrypted to unencrypted	Yes
2	Encrypted to encrypted	Yes
3	Unencrypted to encrypted	Yes
4	Encrypted to unencrypted	No

> **Note**
>
> Encrypting during the `CopyImage` action applies only to Amazon EBS-backed AMIs. Because an instance store-backed AMI does not use snapshots, you can't use copying to change its encryption status.

When you copy an AMI without specifying encryption parameters, the backing snapshot is copied with its original encryption status by default. Therefore, if the source AMI is backed by an unencrypted snapshot, the resulting target snapshot will also be unencrypted. Similarly, if the source AMI's snapshot is encrypted, the resulting target snapshot will also be encrypted by the same AWS KMS key. For AMIs backed by multiple snapshots, each target snapshot preserves the encryption state of its corresponding source snapshot.

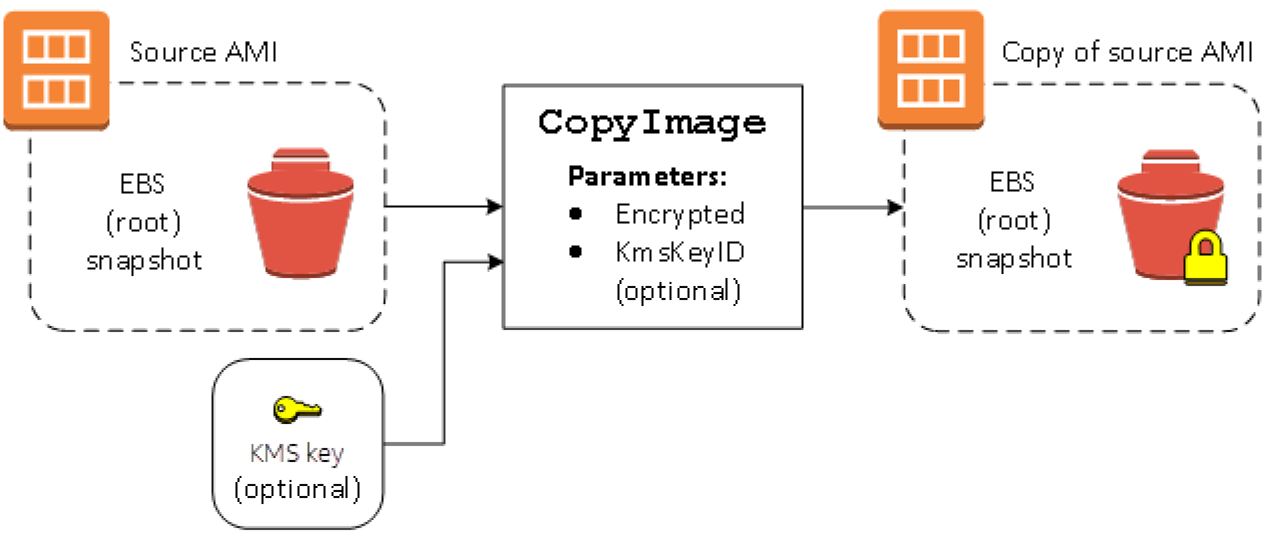
To change the encryption state of the target backing snapshots during an AMI copy, you can specify encryption parameters. The following example shows a non-default case, where encryption parameters are specified with the `CopyImage` action to change the target AMI's encryption state.

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 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.

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

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.

For information about the AWS partitions, see `partition` on the Amazon Resource Names (ARNs) page in the *IAM User Guide*.

> **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.

Contents

Use cases

Use the store and restore APIs to do the following:

- Copy an AMI between AWS partitions
- Make archival copies of AMIs

Copy an AMI between AWS partitions

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`.
- Monitor the progress of the store task by using `DescribeStoreImageTasks`. The object becomes visible in the S3 bucket when the task is completed.
- 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 target 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`.
- 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. 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

Limitations

- To store an AMI, your AWS account must either own the AMI and its snapshots, or the AMI and its snapshots must be shared directly with your account. You can't store an AMI if it is only publicly shared.
- 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 5,000 GB.
- Quota on store image requests: 1,200 GB of storage work (snapshot data) in progress.
- Quota on restore image requests: 600 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.

How AMI store and restore works

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
- DescribeStoreImageTasks
- CreateRestoreImageTask
- File paths

CreateStoreImageTask

The `CreateStoreImageTask` 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` API.

The sum of the sizes of all the AMIs in progress is limited to 1,200 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 200 GB of snapshot data and another AMI with 400 GB of snapshot data are currently being stored, another request will be accepted, because the total in progress is 600 GB, which is less than the limit. But if a single AMI with 1,200 GB of snapshot data is currently being stored, further tasks are rejected until the task is completed.

DescribeStoreImageTasks

The `DescribeStoreImageTasks` 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` API starts a task that restores an AMI from an S3 object that was previously created by using a `CreateStoreImageTask` 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.

File paths

You can use file paths when storing and restoring AMIs, in the following way:

- When storing an AMI in S3, the file path can be added to the bucket name. Internally, the system separates the path from the bucket name, and then adds the path to the object key that is generated to store the AMI. The full object path is shown in the response from the API call.
- When restoring the AMI, because an object key parameter is available, the path can be added to the beginning of the object key value.

Example: Bucket name with appended file path

When you store the AMI, specify the file path after the bucket name.

```
amzn-s3-demo-bucket/path1/path2
```

The following is the resulting object key.

```
path1/path2/ami-0abcdef1234567890.bin
```

When you restore the AMI, you specify both the bucket name and the object key. For examples, see the section called "Create a store image task".

Create a store image task

When you store an AMI in an S3 bucket, a store image task is created. You can use the store image task to monitor the progress and outcome of the process.

Contents

Securing your AMIs

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 can 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 Amazon 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 create IAM policies that grant principals access to specific resources only. For more example policies, see Access management for AWS resources in the *IAM User Guide*.

> **Note**
>
> If the snapshots that make up the AMI are encrypted, or if your account is enabled for encryption by default, your IAM principal must have permission to use the KMS key.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
```

```
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:PutObjectTagging",
                "s3:AbortMultipartUpload",
                "ebs:CompleteSnapshot",
                "ebs:GetSnapshotBlock",
                "ebs:ListChangedBlocks",
                "ebs:ListSnapshotBlocks",
                "ebs:PutSnapshotBlock",
                "ebs:StartSnapshot",
                "ec2:CreateStoreImageTask",
                "ec2:DescribeStoreImageTasks",
                "ec2:CreateRestoreImageTask",
                "ec2:GetEbsEncryptionByDefault",
                "ec2:DescribeTags",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

Create a store image task

To store an AMI in an S3 bucket, start by creating a store image task. The time it takes to complete the task depends on the size of the AMI. You can track the progress of the task until it either succeeds or fails.

AWS CLI

To create the store image task

Use the create-store-image-task command.

```
aws ec2 create-store-image-task \
    --image-id ami-0abcdef1234567890 \
    --bucket amzn-s3-demo-bucket
```

The following is example output.

```
{
```

```
  "ObjectKey": "ami-0abcdef1234567890.bin"
}
```

To describe the progress of the store image task

Use the describe-store-image-tasks command.

```
aws ec2 describe-store-image-tasks \
    --image-ids ami-0abcdef1234567890 \
    --query StoreImageTaskResults[].StoreTaskState \
    --output text
```

The following is example output.

```
InProgress
```

PowerShell

To create the store image task

Use the New-EC2StoreImageTask cmdlet.

```
New-EC2StoreImageTask `
    -ImageId ami-0abcdef1234567890 `
    -Bucket amzn-s3-demo-bucket
```

The following is example output.

```
ObjectKey         : ami-0abcdef1234567890.bin
```

To describe the progress of the store image task

Use the Get-EC2StoreImageTask cmdlet.

```
(Get-EC2StoreImageTask -ImageId ami-0abcdef1234567890).StoreTaskState
```

The following is example output.

```
InProgress
```

Create a restore image task

You must specify a name for the restored AMI. The name must be unique for AMIs in the Region for this account. The restored AMI gets a new AMI ID.

AWS CLI

To create a restore image task

Use the create-restore-image-task command.

```
aws ec2 create-restore-image-task \
    --object-key ami-0abcdef1234567890.bin \
    --bucket amzn-s3-demo-bucket \
    --name "my-restored-ami"
```

The following is example output.

```
{
   "ImageId": "ami-1234567890abcdef0"
}
```

PowerShell

To create a restore image task

Use the New-EC2RestoreImageTask cmdlet.

```
New-EC2RestoreImageTask `
    -ObjectKey ami-0abcdef1234567890.bin `
    -Bucket amzn-s3-demo-bucket `
    -Name "my-restored-ami"
```

The following is example output.

```
ImageId         : ami-1234567890abcdef0
```

Identify the source AMI used to create a new Amazon EC2 AMI

You can identify the source AMI used to create a new AMI by checking the **Source AMI ID** (console) or `sourceImageId` (AWS CLI) field on the new AMI. This field contains the ID of the original AMI that was copied to create the new AMI.

You can also find the Region where the source AMI was located by checking the **Source AMI Region** (console) or `sourceImageRegion` (AWS CLI) field.

Considerations

- The ID and Region of the source AMI only appear if the AMI was created by using the following API commands:
  - CreateImage – Creates an AMI from an instance.
  - CopyImage – Copies an AMI within the same Region or across Regions in the same partition.
  - CreateRestoreImageTask – Copies an AMI to another partition.

  If the AMI was created with any other API command, the ID and Region of the source AMI don't appear.
- For some older AMIs, the ID and Region of the source AMI might not be available.
- If the source AMI has been deleted, the ID and Region fields of the source AMI still appear on the new AMI.
- For AMIs created by using CreateImage (creates an AMI from an instance), the source AMI ID is the ID of the AMI used to launch the instance.

Console

To identify the source AMI used to create an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select the AMI to view its details.

   The source AMI information appears in the following fields: **Source AMI ID** and **Source AMI Region**

AWS CLI

To identify the source AMI used to create an AMI

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

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].{ID:SourceImageId,Region:SourceImageRegion}"
```

The following is example output.

```
[
    {
        "ID": "ami-0abcdef1234567890",
        "Region": "us-west-2"
    }
}
```

PowerShell

To identify the source AMI used to create an AMI

Use the Get-EC2Image cmdlet.

```
Get-EC2Image -ImageId ami-0abcdef1234567890 | Select SourceImageId,
 SourceImageRegion
```

The following is example output.

```
SourceImageId           SourceImageRegion
-------------           -----------------
ami-0abcdef1234567890 us-west-2
```

Manage and monitor AMI usage

AWS provides several features to help you manage and monitor your AMI usage effectively. You can track which accounts are using your shared AMIs, identify when your AMIs were last used, and discover which resources in your AWS account are referencing specific AMIs.

The following table provides an overview of the features for managing and monitoring AMI usage:

Feature	Use case	Key benefits
AMI usage reports	Gain visibility into which AWS accounts are using your AMIs and how much each AMI is being used.	• Identify the AWS accounts and resource types referencing your AMIs so that you can safely deregister or disable AMIs. • Identify unused AMIs for deregistration to reduce storage costs. • Identify your most used AMIs.
Last used tracking	Check when your AMI was last used.	• Identify unused AMIs so that you can safely deregister AMIs. • Identify unused AMIs for deregistration to reduce storage costs.
AMI reference check	Ensure your AWS resources are using the latest compliant AMIs.	• Audit the use of AMIs in your account. • Check where specific AMIs are being referenced. • Maintain compliance by updating your resources to reference the latest AMIs.

Topics

View your AMI usage

If you share your Amazon Machine Images (AMIs) with other AWS accounts—whether with specific AWS accounts, organizations, organizational units (OUs), or publicly—you can see how these AMIs are used by creating AMI usage reports. The reports provide visibility into:

- Which AWS accounts are using your AMIs in EC2 instances or launch templates
- How many EC2 instances or launch templates are referencing each AMI

AMI usage reports help you manage your AMIs more effectively by helping you:

- Identify the AWS accounts and resource types referencing your AMIs so that you can safely deregister or disable AMIs.
- Identify unused AMIs for deregistration to reduce storage costs.
- Identify your most used AMIs.

Contents

How AMI usage reports work

When you create an AMI usage report, you specify:

- The AMI to report on.
- The AWS accounts to check (specific accounts or all accounts).
- The resource types to check (EC2 instances, launch templates, or both).

- For launch templates, the number of versions to check (defaults to the 20 most recent versions).

Amazon EC2 creates a separate report for each AMI. Each report provides:

- A list of the AWS accounts using the AMI.
- A count of the resources referencing the AMI by resource type per account. Note that for launch templates, if an AMI is referenced in multiple versions of a launch template, the count is only 1.

> **Important**
>
> When you generate an AMI usage report, it might not contain the most recent activity. Instance activity from the past 24 hours and launch template activity from the past few days might not appear in the report.

Amazon EC2 automatically deletes a report 30 days after creation. You can download reports from the EC2 console to retain locally.

Create an AMI usage report

To view how your AMI is being used, you must first create an AMI usage report, specifying the accounts and resource types to report on. Once the report is created, you can view the contents of the report. You can also download the report from the EC2 console.

Console

To create an AMI usage report

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select an AMI and choose **Actions**, **AMI usage**, **View my AMI usage**.
4. On the **Create my AMI usage report** page, do the following:

   a. For **Resource types**, select one or more resource types to report on.

   b. For **Account IDs**, do one of the following:

      - Choose **Specify accounts IDs** and then choose **Add account ID** for each account to report on.

- Choose **Include all accounts** to report on all accounts.

c. Choose **Create my AMI usage report**.

5. On the AMI page, choose the **My AMI usage** tab.
6. Choose a report ID to view its details.

AWS CLI

To create an AMI usage report for a list of accounts

Use the create-image-usage-report command with the following required parameters:

- `--image-id` – The ID of the AMI to report on.
- `--resource-types` – The types of resources to check. In the following example, the resource types to check are EC2 instances and launch templates. In addition, the number of launch template versions to check is also specified (`version-depth=100`).

To report on specific accounts, use the `--account-ids` parameter to specify the ID of each account to report on.

```
aws ec2 create-image-usage-report \
    --image-id ami-0abcdef1234567890 \
    --account-ids 111122223333 444455556666 123456789012 \
    --resource-types ResourceType=ec2:Instance \
      'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=100}]'
```

To create an AMI usage report of all accounts

To report on all accounts using the specified AMI, use the same command but omit the `--account-ids` parameter.

```
aws ec2 create-image-usage-report \
    --image-id ami-0abcdef1234567890 \
    --resource-types ResourceType=ec2:Instance \
      'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=100}]'
```

The following is example output.

```
{
    "ReportId": "amiur-00b877d192f6b02d0"
}
```

To monitor the report creation status

Use the describe-image-usage-reports command and specify the report ID.

```
aws ec2 describe-image-usage-reports --report-ids amiur-00b877d192f6b02d0
```

The following is example output. The initial value of the `State` field is `pending`. To be able to view the report entries, the state must be `available`.

```
{
    "ImageUsageReports": [
        {
            "ImageId": "ami-0e9ae3dc21c2b3a64",
            "ReportId": "amiur-abcae3dc21c2b3999",
            "ResourceTypes": [
                {"ResourceType": "ec2:Instance"}
            ],
            "State": "pending",
            "CreationTime": "2025-09-29T13:27:12.322000+00:00",
            "ExpirationTime": "2025-10-28T13:27:12.322000+00:00"
        }
    ]
}
```

PowerShell

To create an AMI usage report for a list of accounts

Use the New-EC2ImageUsageReport cmdlet with the following required parameters:

- `-ImageId` – The ID of the AMI to report on.
- `-ResourceType` – The types of resources to check. In the following example, the resource types to check are EC2 instances and launch templates. In addition, the number of launch template versions to check is also specified (`'version-depth' = 100`).

To report on specific accounts, use the -AccountId parameter to specify the ID of each account to report on.

```
New-EC2ImageUsageReport `
    -ImageId ami-0abcdef1234567890 `
    -AccountId 111122223333 444455556666 123456789012 `
    -ResourceType @(
        @{ResourceType = 'ec2:Instance'},
        @{ResourceType = 'ec2:LaunchTemplate'ResourceTypeOptions = @{'version-depth'
 = 100}
        })
```

To create an AMI usage report of all accounts

To report on all accounts using the specified AMI, use the same command but omit the -AccountId parameter.

```
New-EC2ImageUsageReport `
    -ImageId ami-0abcdef1234567890 `
    -ResourceType @(
        @{ResourceType = 'ec2:Instance'},
        @{ResourceType = 'ec2:LaunchTemplate'ResourceTypeOptions = @{'version-depth'
 = 100}
        })
```

The following is example output.

```
ReportId
--------
amiur-00b877d192f6b02d0
```

To monitor the report creation status

Use the Get-EC2ImageUsageReport command and specify the report ID.

```
Get-EC2ImageUsageReport -ReportId amiur-00b877d192f6b02d0
```

The following is example output. The initial value of the State field is pending. To be able to view the report entries, the state must be available.

```
ImageUsageReports
-----------------
{@{ImageId=ami-0e9ae3dc21c2b3a64; ReportId=amiur-abcae3dc21c2b3999;
 ResourceTypes=System.Object[]; State=pending; CreationTime=2025-09-29;
 ExpirationTime=2025-10-28}}
```

View AMI usage reports

You can view all the usage reports you've created for an AMI in the past 30 days. Amazon EC2 automatically deletes a report 30 days after creation.

For each report, you can see the AWS accounts that are using the AMI, and for each account, a count of the resources referencing the AMI by resource type. You can also see when the report creation was initiated. This information is only available when the report is in the **Complete** (console) or `available` (AWS CLI) state.

> **Important**
>
> When you generate an AMI usage report, it might not contain the most recent activity. Instance activity from the past 24 hours and launch template activity from the past few days might not appear in the report.

Console

To view an AMI usage report

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select an AMI.
4. Choose the **My usage reports** tab.

   The report list shows:

   - All reports generated in the past 30 days for the selected AMI.
   - For each report, the **Report initiated time** column shows the date the report was created.
5. Choose the ID of a report to view its contents.

6. To go back to the **My usage reports** tab on the AMI details page, choose **View all reports for this AMI**.

AWS CLI

To list all the AMI usage reports for the specified AMI

Use the describe-image-usage-reports command and specify the ID of the AMI to get a list of its reports.

```
aws ec2 describe-image-usage-reports --image-ids ami-0abcdef1234567890
```

The following is example output. Each report ID is listed along with the resource types that were scanned and the report creation and expiration dates. You can use this information to identify the reports whose entries you want to view.

```
{
  "ImageUsageReports": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ReportId": "amiur-1111111111111111",
      "ResourceTypes": [
        {
          "ResourceType": "ec2:Instance"
        }
      ],
      "State": "available",
      "CreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ExpirationTime": "2025-10-28T13:27:12.322000+00:00",
      "Tags": []
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ReportId": "amiur-22222222222222222",
      "ResourceTypes": [
        {
          "ResourceType": "ec2:Instance"
        },
        {
          "ResourceType": "ec2:LaunchTemplate"
        }
      ],
```

```
      "State": "available",
      "CreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ExpirationTime": "2025-10-30T13:27:12.322000+00:00",
      "Tags": []
    }
  ],
  "NextToken": "opaque"
}
```

To view the contents of an AMI usage report for the specified AMI

Use the [describe-image-usage-report-entries](#) command and specify the ID of the AMI. The response returns all the reports for the specified AMI, showing the accounts that have used the AMI and their resource counts.

```
aws ec2 describe-image-usage-report-entries --image-ids ami-0abcdef1234567890
```

The following is example output.

```
{
  "ImageUsageReportEntries": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 15,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-1111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 2,
      "ReportCreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ReportId": "amiur-22222222222222222"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "001100110011",
      "UsageCount": 39,
```

```
      "ReportCreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ReportId": "amiur-22222222222222222"
    }
  ],
  "NextToken": "opaque"
}
```

To view the contents of an AMI usage report for the specified report

Use the describe-image-usage-report-entries command and specify the ID of the report. The response returns all the entries for the specified report, showing the accounts that have used the AMI and their resource counts.

```
aws ec2 describe-image-usage-report-entries --report-ids amiur-11111111111111111
```

The following is example output.

```
{
  "ImageUsageReportEntries": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 15,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-11111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:LaunchTemplate",
      "AccountId": "123412341234",
      "UsageCount": 4,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-11111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:LaunchTemplate",
      "AccountId": "001100110011",
      "UsageCount": 2,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-11111111111111111"
```

```
    }
  ],
  "NextToken": "opaque"
}
```

PowerShell

To list all the AMI usage reports for the specified AMI

Use the Get-EC2ImageUsageReport cmdlet and specify the ID of the AMI to get a list of its reports.

```
Get-EC2ImageUsageReport -ImageId ami-0abcdef1234567890
```

The following is example output. Each report ID is listed along with the resource types that were scanned and the report creation and expiration dates. You can use this information to identify the reports whose entries you want to view.

```
@{
    ImageUsageReports = @(
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-1111111111111111"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-09-29T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-28T13:27:12.322000+00:00"
        },
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-2222222222222222"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-09-30T13:27:12.322000+00:00"
```

```
            ExpirationTime = "2025-10-29T13:27:12.322000+00:00"
        },
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-33333333333333333"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-10-01T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-30T13:27:12.322000+00:00"
        }
    )
    NextToken = "opaque"
}
```

To view the contents of an AMI usage report for the specified AMI

Use the Get-EC2ImageUsageReportEntry cmdlet and specify the ID of the AMI. The response returns all the reports for the specified AMI, showing the accounts that have used the AMI and their resource counts.

```
Get-EC2ImageUsageReportEntry -ImageId ami-0abcdef1234567890
```

The following is example output.

```
ImageUsageReportEntries : {@{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 15
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 7
    ReportCreationTime = "2025-09-30T13:27:12.322000+00:00"
    ReportId = "amiur-22222222222222222"
```

```
    }...}
NextToken : opaque
```

To view the contents of an AMI usage report for the specified report

Use the Get-EC2ImageUsageReportEntry cmdlet and specify the ID of the report. The response returns all the entries for the specified report, showing the accounts that have used the AMI and their resource counts.

```
Get-EC2ImageUsageReportEntry -ReportId amiur-1111111111111111
```

The following is example output.

```
ImageUsageReportEntries : {@{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 15
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:LaunchTemplate"
    AccountId = "123412341234"
    UsageCount = 4
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:LaunchTemplate"
    AccountId = "************"
    UsageCount = 2
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }}
NextToken : opaque
```

Delete an AMI usage report

Amazon EC2 automatically deletes a report 30 days after it was created. You can delete it manually before that time.

Console

To delete an AMI usage report

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select an AMI.
4. Choose the **My AMI usage** tab.
5. Choose the option button next to the report to delete, and then choose **Delete**.

AWS CLI

To delete an AMI usage report

Use the delete-image-usage-report command and specify the ID of the report.

```
aws ec2 delete-image-usage-report --report-id amiur-0123456789abcdefg
```

PowerShell

To delete an AMI usage report

Use the Remove-EC2ImageUsageReport cmdlet and specify the ID of the report.

```
Remove-EC2ImageUsageReport -ReportId amiur-0123456789abcdefg
```

Check when an Amazon EC2 AMI was last used

Amazon EC2 automatically tracks the date and time when an AMI was last used to launch an instance. If you have an AMI that has not been used to launch an instance in a long time, consider whether the AMI is a good candidate for deregistration or deprecation.

Considerations

- When an AMI is used to launch an instance, there is a 24-hour delay before that usage is reported.
- You must be the owner of the AMI to get the last launched time.
- AMI usage data is available starting April 2017.

Console

To view the last launched time of an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **AMIs**.
3. From the filter bar, choose **Owned by me**.
4. Select the checkbox for the AMI.
5. On the **Details** tab, find **Last launched time**.

AWS CLI

To view the last launched time by describing the AMI

Use the describe-images command. If `LastLaunchedTime` is not present in the output, verify that you own the AMI.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].LastLaunchedTime \
    --output text
```

The following is example output.

```
2025-02-17T20:22:19Z
```

To view the last launched time attribute of an AMI

Use the describe-image-attribute command. You must be the owner of the specified AMI.

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute lastLaunchedTime \
    --query LastLaunchedTime.Value \
    --output text
```

The following is example output.

```
2025-02-17T20:22:19Z
```

PowerShell

To view the last launched time by describing the AMI

Use the Get-EC2Image cmdlet. If `LastLaunchedTime` is not present in the output, verify that you own the AMI.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).LastLaunchedTime
```

The following is example output.

```
2025-02-17T20:22:19Z
```

To view the last launched time attribute of an AMI

Use the Get-EC2ImageAttribute cmdlet. You must be the owner of the specified AMI.

```
(Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute LastLaunchedTime).LastLaunchedTime
```

The following is example output.

```
2025-02-17T20:22:19Z
```

Identify your resources referencing specified AMIs

You can identify your AWS resources that reference specified Amazon Machine Images (AMIs), regardless of whether the AMIs are public or private, or who owns them. This visibility helps you ensure your resources use the latest compliant AMIs.

Key benefits

Checking AMI references helps you:

- Audit the use of AMIs in your account.
- Check where specific AMIs are being referenced.
- Maintain compliance by updating your resources to reference the latest AMIs.

Topics

Supported resources

AMI references can be checked in:

- EC2 instances
- Launch templates
- SSM parameters
- Image Builder image recipes
- Image Builder container recipes

How AMI reference checks work

Basic operation

When you run an AMI reference check, you:

- Specify which AMIs to check.
- Choose which resource types to scan.
- Receive a list of your resources that reference the specified AMIs.

Resource type selection

In the console, you select the resource types to scan.

In the CLI, you specify resource types to scan using one or both of the following CLI parameters:

- `IncludeAllResourceTypes`: Scans all supported resource types.
- `ResourceTypes`: Scans your specified resource types.

Response scoping

You can scope the response for EC2 instances and launch templates by customizing the `ResourceTypeOptions` values using the `ResourceTypes` parameter. The console and `IncludeAllResourceTypes` parameter both use default option values. When `ResourceTypes` and `IncludeAllResourceTypes` are used together, the `ResourceTypes` option values take precedence over the defaults.

The following are the default values:

Resource type	Scoping option (`OptionName`)	Purpose	Default values for `OptionValue` and console
EC2 instances	`state-name`	Filter by instance state	`pending`, `running`, `shutting-down`, `terminated`, `stopping`, `stopped` (all states)
Launch templates	`version-depth`	Specify the number of launch template versions to check (starting from the most recent version)	`10` (most recent versions)

Required IAM permissions

To use the DescribeImageReferences API to identify your resources that are referencing specified AMIs, you need the following IAM permissions to describe the resources:

- `ec2:DescribeInstances`
- `ec2:DescribeLaunchTemplates`
- `ec2:DescribeLaunchTemplateVersions`
- `ssm:DescribeParameters`
- `ssm:GetParameters`

- `imagebuilder:ListImageRecipes`
- `imagebuilder:ListContainerRecipes`
- `imagebuilder:GetContainerRecipe`

Example IAM policy for using the DescribeImageReferences API

The following example policy grants you the permissions to use the DescribeImageReferences API, which includes the permissions to describe EC2 instances, launch templates, Systems Manager parameters, Image Builder image recipes, and Image Builder container recipes.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "ec2:DescribeImageReferences",
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeLaunchTemplateVersions",
    "ssm:DescribeParameters",
    "ssm:GetParameters",
    "imagebuilder:ListImageRecipes",
    "imagebuilder:ListContainerRecipes",
    "imagebuilder:GetContainerRecipe"
   ],
   "Resource": "*",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "ec2-images.amazonaws.com"
     ]
    }
   }
  }
 ]
}
```

> **Important**
>
> We strongly recommend using the AWS managed policy `AmazonEC2ImageReferencesAccessPolicy` instead of creating the policy yourself. Creating a custom IAM policy that provides only the required permissions requires time and expertise, and will require updates as new resource types become available.
> The `AmazonEC2ImageReferencesAccessPolicy` managed policy:
>
> - Grants all the permissions needed to use the DescribeImageReferences API (these include the permissions to describe EC2 instances, launch templates, Systems Manager parameters, and Image Builder container and image recipes).
> - Automatically supports new resource types as they become available (especially important when using the `IncludeAllResourceTypes` parameter).
>
> You can attach the `AmazonEC2ImageReferencesAccessPolicy` policy to your IAM identities (users, groups, and roles).
> To view the permissions included in this policy, see AmazonEC2ImageReferencesAccessPolicy in the *AWS Managed Policy Reference*.

Steps for checking AMI references

Use the following procedures to identify which of your AWS resources are referencing specified AMIs.

Console

To identify resources referencing specified AMIs

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select one or more AMIs to check for references.
4. Choose **Actions**, **AMI usage**, **View referenced resources**.
5. On the **View resources referencing selected AMIs** page:
    a. For **Resource types**, select one or more resource types.
    b. Choose **View resources**.

6. The **Resources referencing selected AMIs** section appears. The list displays the resources referencing the specified AMIs. Each row provides the following information:

   - **AMI ID** – The ID of the referenced AMI.
   - **Resource type** – The resource type of the resource referencing the AMI.
   - **Resource ID** – The ID of the resource referencing the AMI.

AWS CLI

To check AMI references for specific resource types

Use the describe-image-references command with the `--resource-types` parameter. The following example checks EC2 instances (scoping by instance state), launch templates (scoping to the 20 most recent launch template versions), and other specific resource types.

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --resource-types \
        'ResourceType=ec2:Instance,ResourceTypeOptions=[{OptionName=state-
name,OptionValues=[running,pending]}]' \
        'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=[20]}]' \
        'ResourceType=ssm:Parameter' \
        'ResourceType=imagebuilder:ImageRecipe' \
        'ResourceType=imagebuilder:ContainerRecipe'
```

The following is example output.

```
{
    "ImageReferences": [
        {
            "ImageId": "ami-0abcdef1234567890",
            "ResourceType": "ec2:Instance",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
        },
        {
            "ImageId": "ami-1234567890abcdef0",
            "ResourceType": "ec2:LaunchTemplate",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:launch-template/
lt-1234567890abcdef0"
        }
```

```
    ]
}
```

To check AMI references for all the supported resource types

Use the describe-image-references command with the `--include-all-resource-types` parameter.

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --include-all-resource-types
```

To check AMI references for all supported resource types and specific options

Use the describe-image-references command with both the `--include-all-resource-types` and `--resource-types` parameters. This example checks all resource types while scoping the response for EC2 instances to running or pending instances.

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --include-all-resource-types \
    --resource-types
 'ResourceType=ec2:Instance,ResourceTypeOptions=[{OptionName=state-
name,OptionValues=[running,pending]}]'
```

PowerShell

To check AMI references for specific resource types

Use the Get-EC2ImageReference cmdlet with the `-ResourceType` parameter. The following example checks EC2 instances (scoping by instance state), launch templates (scoping to the 20 most recent launch template versions), and other specific resource types.

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -ResourceType @(
        @{
            ResourceType = 'ec2:Instance'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'state-name'
                    OptionValues = @('running', 'pending')
```

```
                }
            )
        },
        @{
            ResourceType = 'ec2:LaunchTemplate'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'version-depth'
                    OptionValues = @('20')
                }
            )
        },
        @{
            ResourceType = 'ssm:Parameter'
        },
        @{
            ResourceType = 'imagebuilder:ImageRecipe'
        },
        @{
            ResourceType = 'imagebuilder:ContainerRecipe'
        }
    )
```

To check AMI references for all the supported resource types

Use the [Get-EC2ImageReference](#) cmdlet with the `-IncludeAllResourceTypes` parameter.

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -IncludeAllResourceTypes
```

To check AMI references for all supported resource types and specific options

Use the [Get-EC2ImageReference](#) cmdlet with both the `-IncludeAllResourceTypes` and `-ResourceType` parameters. This example checks all resource types while scoping the response for EC2 instances to running or pending instances.

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -IncludeAllResourceTypes `
    -ResourceType @(
        @{
```

```
            ResourceType = 'ec2:Instance'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'state-name'
                    OptionValues = @('running', 'pending')
                }
            )
        }
    )
```

Deprecate an Amazon EC2 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 it.

After an AMI is deprecated:

- For AMI users, the deprecated AMI does not appear in DescribeImages API calls unless you specify its ID or specify that deprecated AMIs must appear. AMI owners continue to see deprecated AMIs in DescribeImages 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.

Contents

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 it.

Considerations

- To deprecate an AMI, you must be the owner of the AMI.
- AMIs that have not been used recently to launch an instance might be good candidates for deprecation or deregistering. For more information, see the section called “Check when an AMI was last used”.
- You can create Amazon Data Lifecycle Manager EBS-backed AMI policies to automate the deprecation of EBS-backed AMIs. For more information, see Create AMI lifecycle policies.
- By default, the deprecation date of all public AMIs is set to two years from the AMI creation date. You can set the deprecation date to earlier than two years. To cancel the deprecation date, or to move the deprecation to a later date, you must make the AMI private by only sharing it with specific AWS accounts.

Deprecate an AMI

You can deprecate an AMI on a specific date and time. You must be the owner of the AMI.

The upper limit for the deprecation date is 10 years from now, except for public AMIs, where the upper limit is 2 years from the creation date. You can't specify a date in the past.

Console

To deprecate an AMI on a specific date

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigator, choose **AMIs**.
3. From the filter bar, choose **Owned by me**.
4. Select the AMI, and then choose **Actions**, **Manage AMI Deprecation**. You can select multiple AMIs to set the same deprecation date of several AMIs at once.
5. Select the **Enable** checkbox, and then enter the deprecation date and time.
6. Choose **Save**.

AWS CLI

To deprecate an AMI on a specific date

Use the enable-image-deprecation command. If you specify a value for seconds, Amazon EC2 rounds the seconds to the nearest minute.

```
aws ec2 enable-image-deprecation \
    --image-id ami-0abcdef1234567890 \
    --deprecate-at "2025-04-15T13:17:12.000Z"
```

PowerShell

To deprecate an AMI on a specific date

Use the Enable-EC2ImageDeprecation cmdlet. If you specify a value for seconds, Amazon EC2 rounds the seconds to the nearest minute.

```
Enable-EC2ImageDeprecation `
    -ImageId ami-0abcdef1234567890 `
    -DeprecateAt 2025-04-15T13:17:12.000Z
```

Describe deprecated AMIs

You can view the deprecation date and time of an AMI, and filter AMIs by deprecation date.

Console

To view the deprecation date of an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigator, choose **AMIs**, and then select the AMI.
3. Check the **Deprecation time** field (if you selected the checkbox next to the AMI, it's located on the **Details** tab). The field shows the deprecation date and time of the AMI. If the field is empty, the AMI is not deprecated.

To filter AMIs by deprecation date

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigator, choose **AMIs**.
3. From the filter bar, choose **Owned by me** or **Private images** (private images include AMIs that are shared with you as well as owned by you).
4. In the Search bar, enter **`Deprecation time`** (as you enter the letters, the **Deprecation time** filter appears), and then choose an operator and a date and time.

AWS CLI

When you describe all AMIs, the results depend on whether you are an AMI user or the AMI owner.

- **AMI user** – By default, when you describe all AMIs, deprecated AMIs that are shared with you but not owned by you are excluded. To include deprecated AMIs in the results, specify the `--include-deprecated` option.
- **AMI owner** – When you describe all AMIs, all AMIs that you own, including deprecated AMIs, are included. You can't exclude deprecated AMIs that you own by using the `--no-include-deprecated` option.

To include deprecated AMIs when describing all AMIs for an account

Use the following describe-images command.

```
aws ec2 describe-images
    --owners 123456789012 \
```

```
    --include-deprecated
```

To describe the deprecated AMIs for your account

Use the following describe-images command.

```
aws ec2 describe-images \
    --owners self \
    --query "Images[?DeprecationTime!=null].ImageId" \
    --output text
```

The following is example output.

```
ami-0abcdef1234567890
```

To describe the deprecation date of an AMI

Use the following describe-images command. If `DeprecationTime` is not present in the output, the AMI is not deprecated or set to deprecate at a future date.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].DeprecationTime \
    --output text
```

The following is example output.

```
2025-05-01T00:00:00.000Z
```

PowerShell

To list the deprecated AMIs for your account

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image -Owner self | Where-Object {$_.DeprecationTime -ne $null}).ImageId
```

The following is example output.

```
ami-0abcdef1234567890
```

To describe the deprecation date of an AMI

Use the [Get-EC2Image](#) cmdlet. If `DeprecationTime` is not present in the output, the AMI is not deprecated or set to deprecate at a future date.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).DeprecationTime
```

The following is example output.

```
2025-05-01T00:00:00.000Z
```

Cancel AMI deprecation

You can cancel the deprecation of an AMI, which removes the deprecation date and time. You must be the AMI owner to perform this procedure.

Console

To cancel the deprecation of an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigator, choose **AMIs**.
3. From the filter bar, choose **Owned by me**.
4. Select the AMI, and then choose **Actions**, **Manage AMI Deprecation**. You can select multiple AMIs to cancel the deprecation of several AMIs at once.
5. Clear the **Enable** checkbox, and then choose **Save**.

AWS CLI

To cancel the deprecation of an AMI

Use the following disable-image-deprecation command.

```
aws ec2 disable-image-deprecation --image-id ami-0abcdef1234567890
```

PowerShell

To cancel the deprecation of an AMI

Use the Disable-EC2ImageDeprecation cmdlet.

```
Disable-EC2ImageDeprecation -ImageId ami-0abcdef1234567890
```

Disable an Amazon EC2 AMI

You can disable an AMI to prevent it from being used for instance launches. You can't launch new instances from a disabled AMI. You can re-enable a disabled AMI so that it can be used again for instance launches.

You can disable both private and public AMIs.

To reduce storage costs for disabled EBS-backed AMIs that are rarely used, but which need to be retained long term, you can archive their associated snapshots. For more information, see Archive Amazon EBS snapshots in the *Amazon EBS User Guide*.

Contents

How AMI disable works

> **Warning**
>
> Disabling an AMI removes all its launch permissions.

When an AMI is disabled:

- The AMI's state changes to `disabled`.

- A disabled AMI can't be shared. If an AMI was public or previously shared, it is made private. If an AMI was shared with an AWS account, organization, or Organizational Unit, they lose access to the disabled AMI.
- A disabled AMI does not appear in [DescribeImages] API calls by default.
- A disabled AMI does not appear under the **Owned by me** console filter. To find disabled AMIs, use the **Disabled images** console filter.
- A disabled AMI is not available to select for instance launches in the EC2 console. For example, a disabled AMI does not appear in the AMI catalog in the launch instance wizard or when creating a launch template.
- Launch services, such as launch templates and Auto Scaling groups, can continue to reference disabled AMIs. Subsequent instance launches from a disabled AMI will fail, so we recommend updating launch templates and Auto Scaling groups to reference available AMIs only.
- EC2 instances that were previously launched using an AMI that is subsequently disabled are not affected, and can be stopped, started, and rebooted.
- You can't delete snapshots associated with disabled AMIs. Attempting to delete an associated snapshot results in the `snapshot is currently in use` error.

When an AMI is re-enabled:

- The AMI's state changes to `available`, and it can be used to launch instances.
- The AMI can be shared.
- AWS accounts, organizations, and Organizational Units that lost access to the AMI when it was disabled do not regain access automatically, but the AMI can be shared with them again.

Costs

When you disable an AMI, the AMI is not deleted. If the AMI is an EBS-backed AMI, you continue to pay for the AMI's EBS snapshots. If you want to keep the AMI, you might be able to reduce your storage costs by archiving the snapshots. For more information, see Archive Amazon EBS snapshots in the *Amazon EBS User Guide*. If you don't want to keep the AMI and its snapshots, you must deregister the AMI and delete the snapshots. For more information, see Deregister an AMI.

Prerequisites

To disable or re-enable an AMI, you must be the owner of the AMI.

Required IAM permissions

To disable and re-enable an AMI, you must have the following IAM permissions:

- `ec2:DisableImage`
- `ec2:EnableImage`

Disable an AMI

You can disable an AMI by using the EC2 console or the AWS Command Line Interface (AWS CLI). You must be the AMI owner to perform this procedure.

Console

To disable an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **AMIs**.
3. From the filter bar, choose **Owned by me**.
4. Select the AMI, and then choose **Actions**, **Disable AMI**. You can select multiple AMIs to disable at once.
5. In the **Disable AMI** window, choose **Disable AMI**.

AWS CLI

To disable an AMI

Use the following disable-image command.

```
aws ec2 disable-image --image-id ami-0abcdef1234567890
```

PowerShell

To disable an AMI

Use the Disable-EC2Image cmdlet.

```
Disable-EC2Image -ImageId ami-0abcdef1234567890
```

Describe disabled AMIs

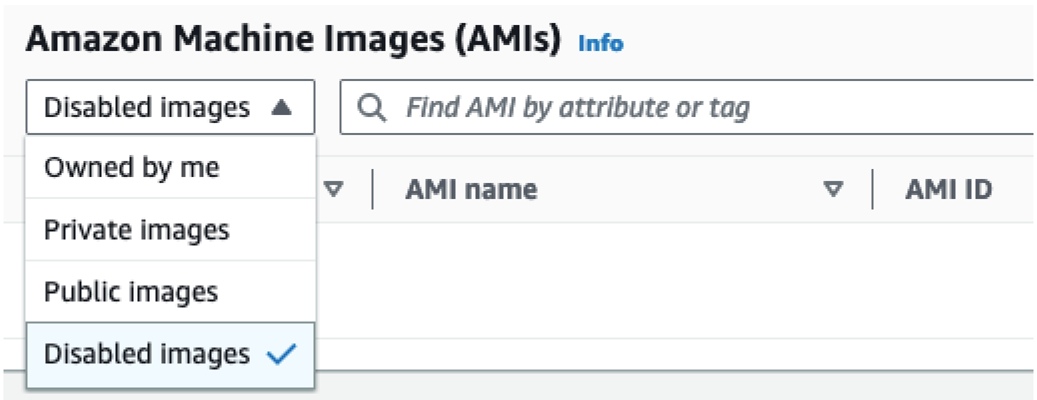
You can view disabled AMIs in the EC2 console and by using the AWS CLI.

You must be the AMI owner to view disabled AMIs. Because disabled AMIs are made private, you can't view disabled AMIs if you're not the owner.

Console

To view disabled AMIs

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **AMIs**.
3. From the filter bar, choose **Disabled images**.



AWS CLI

By default, when you describe all AMIs, the disabled AMIs are not included in the results. To include disabled AMIs in the results, specify the `--include-disabled` option. The `State` field for an AMI is `disabled` if the AMI is disabled.

To include disabled AMIs when describing all AMIs for an account

Use the following describe-images command.

```
aws ec2 describe-images \
    --owners 123456789012 \
    --include-disabled
```

To list the disabled AMIs for your account

Use the following describe-images command.

```
aws ec2 describe-images \
    --owners self \
    --include-disabled \
    --filters Name=state,Values=disabled \
    --query Images[].ImageId \
    --output text
```

The following is example output.

```
ami-0abcdef1234567890
```

To describe the status of an AMI

Use the following describe-images command. If `DeprecationTime` is not present in the output, the AMI is not deprecated or set to deprecate at a future date.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].State \
    --output text
```

The following is example output.

```
disabled
```

PowerShell

By default, when you describe all AMIs, the disabled AMIs are not included in the results. To include disabled AMIs in the results, specify the `-IncludeDisabled` parameter. The `State` field for an AMI is `disabled` if the AMI is disabled.

To list the disabled AMIs for your account

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image `
    -Owner self `
    -IncludeDisabled $true | Where-Object {$_.State -eq "disabled"}).ImageId
```

The following is example output.

```
ami-0abcdef1234567890
```

To describe the status of an AMI

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).State.Value
```

The following is example output.

```
disabled
```

Re-enable a disabled AMI

You can re-enable a disabled AMI. You must be the AMI owner to perform this procedure.

Console

To re-enable a disabled AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **AMIs**.
3. From the filter bar, choose **Disabled images**.
4. Select the AMI, and then choose **Actions**, **Enable AMI**. You can select multiple AMIs to re-enable several AMIs at once.
5. In the **Enable AMI** window, choose **Enable**.

AWS CLI

To re-enable a disabled AMI

Use the following enable-image command.

```
aws ec2 enable-image --image-id ami-0abcdef1234567890
```

PowerShell

To re-enable a disabled AMI

Use the Enable-EC2Image cmdlet.

```
Enable-EC2Image -ImageId ami-0abcdef1234567890
```

Deregister an Amazon EC2 AMI

When you deregister an AMI, Amazon EC2 permanently deletes it. After you deregister an AMI, you can't use it to launch new instances. You might consider deregistering an AMI when you have finished using it.

To protect against accidental or malicious deregistering of an AMI, you can turn on deregistration protection. If you accidentally deregister an EBS-backed AMI, you can use the Recycle Bin to restore it only if you restore it within the allowed time period before it is permanently deleted.

When deregistering an AMI, you can optionally delete its associated snapshots at the same time. However, if a snapshot is associated with multiple AMIs, it won't be deleted even if specified for deletion, although the AMI will still be deregistered. Any snapshots not deleted will continue to incur storage costs.

Deregistering an AMI has no effect on any instances that were launched from the AMI. You can continue to use these instances. By default, deregistering an AMI also has no effect on any snapshots that were created during the AMI creation process. You'll continue to incur usage costs for these instances and storage costs for the snapshots. Therefore, to avoid incurring unnecessary costs, we recommend that you terminate any instances and delete any snapshots that you do not need. You can delete the snapshots either automatically during deregistration or manually after deregistration. For more information, see Avoid costs from unused resources.

For instances launched from an AMI that is subsequently deregistered, you can still view some high-level information about the AMI by using the `describe-instance-image-metadata` AWS CLI command. For more information, see describe-instance-image-metadata.

Contents

- Protect an Amazon EC2 AMI from deregistration

Considerations

- You can't deregister an AMI that is not owned by your account.
- You can't use Amazon EC2 to deregister an AMI that is managed by the AWS Backup service. Instead, use AWS Backup to delete the corresponding recovery points in the backup vault. For more information, see Deleting backups in the *AWS Backup Developer Guide*.

Deregister an AMI

You can deregister EBS-backed AMIs and instance store-backed AMIs. For EBS-backed AMIs, you can optionally delete the associated snapshots at the same time. However, if a snapshot is associated with other AMIs, it will not be deleted even if specified for deletion.

Console

To deregister an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. From the filter bar, choose **Owned by me** to list your available AMIs, or choose **Disabled images** to list your disabled AMIs.
4. Select the AMI to deregister.
5. Choose **Actions**, **Deregister AMI**.
6. (Optional) To delete the associated snapshots during deregistration, select the **Delete associated snapshots** checkbox.

   > **Note**
   >
   > If a snapshot is associated with other AMIs, it is not deleted, even if the checkbox is selected.

7. Choose **Deregister AMI**.

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

AWS CLI

To deregister an AMI

Use the following deregister-image command.

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

To deregister an AMI and delete its associated snapshots

Use the following deregister-image command and specify the `--delete-associated-snapshots` parameter. Note that if a snapshot is associated with other AMIs, it is not deleted, even if you specify this parameter.

```
aws ec2 deregister-image \
    --image-id ami-0abcdef1234567890 \
    --delete-associated-snapshots
```

PowerShell

To deregister an AMI

Use the Unregister-EC2Image cmdlet.

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

To deregister an AMI and delete its associated snapshots

Use the Unregister-EC2Image cmdlet and specify the `-DeleteAssociatedSnapshots` parameter. Note that if a snapshot is associated with other AMIs, it is not deleted, even if you specify this parameter.

```
Unregister-EC2Image `
    -ImageId ami-0abcdef1234567890 `
    -DeleteAssociatedSnapshots
```

Avoid costs from unused resources

Deregistering an AMI doesn't, by default, delete all of the resources that are associated with the AMI. These resources include the snapshots for EBS-backed AMIs and the files in Amazon S3 for instance store-backed AMIs. When you deregister an AMI, you also don't terminate or stop any instances launched from the AMI.

You will continue to incur costs for storing the snapshots and files, and you will incur costs for any running instances.

To avoid incurring these types of unnecessary costs, we recommend deleting any resources that you don't need.

EBS-backed AMIs

- Delete the associated snapshots while deregistering the AMI. For more information, see Deregister an AMI.
- If you deregister an AMI without deleting its associated snaphots, you can manually delete the snapshots. The snapshot of the instance root volume created during AMI creation has the following description format:

```
Created by CreateImage(i-1234567890abcdef0) for ami-0abcdef1234567890
```

- If you no longer need the instances that were launched from the AMI, you can stop or terminate them. To list the instances, filter by the ID of the AMI.

Instance store-backed AMIs

- Delete the bundle in Amazon S3 by using the ec2-delete-bundle (AMI tools) command.
- If the Amazon S3 bucket is empty after you delete the bundle, and you have no further use for that bucket, you can delete the bucket.
- If you no longer need the instances that were launched from the AMI, you can terminate them. To list the instances, filter by the ID of the AMI.

Protect an Amazon EC2 AMI from deregistration

You can turn on *deregistration protection* on an AMI to prevent accidental or malicious deletion. When you turn on deregistration protection, the AMI can’t be deregistered by any user, regardless of their IAM permissions. If you want to deregister the AMI, you must first turn off the deregistration protection on it.

When you turn on deregistration protection on an AMI, you have the option to include a 24-hour cooldown period. This cooldown period is the time during which deregistration protection remains in effect after you turn it off. During this cooldown period, the AMI can't be deregistered. When the cooldown period ends, the AMI can be deregistered.

Deregistration protection is turned off by default on all existing and new AMIs.

Turn on deregistration protection

Use the following procedures to turn on deregistration protection.

Console

To turn on deregistration protection

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. From the filter bar, choose **Owned by me** to list your available AMIs, or choose **Disabled images** to list your disabled AMIs.
4. Select the AMI on which you want to turn on deregistration protection, and then choose **Actions**, **Manage AMI deregistration protection**.
5. In the **Manage AMI deregistration protection** dialog box, you can turn on deregistration protection with or without a cooldown period. Choose one of the following options:
   - **Enable with a 24-hour cooldown period** – With a cooldown period, the AMI can't be deregistered for 24 hours when deregistration protection is turned off.
   - **Enable without cooldown** – Without a cooldown period, the AMI can be deregistered immediately when deregistration protection is turned off.
6. Choose **Save**.

AWS CLI

To turn on deregistration protection

Use the enable-image-deregistration-protection command. To enable the optional cooldown period, include the `--with-cooldown` option.

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0abcdef1234567890 \
    --with-cooldown
```

PowerShell

To turn on deregistration protection

Use the Enable-EC2ImageDeregistrationProtection cmdlet. To enable the optional cooldown period, set the `-WithCooldown` parameter to `true`.

```
Enable-EC2ImageDeregistrationProtection `
    -ImageId ami-0abcdef1234567890 `
    -WithCooldown $true
```

Turn off deregistration protection

Use the following procedures to turn off deregistration protection.

If you chose to include a 24-hour cooldown period when you turned on deregistration protection for the AMI, then, when you turn off deregistration protection, you won't immediately be able to deregister the AMI. The cooldown period is the 24-hour time period during which deregistration protection remains in effect even after you turn it off. During this cooldown period, the AMI can't be deregistered. After the cooldown period ends, the AMI can be deregistered.

Console

To turn off deregistration protection

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. From the filter bar, choose **Owned by me** to list your available AMIs, or choose **Disabled images** to list your disabled AMIs.
4. Select the AMI to turn off deregistration protection, and then choose **Actions**, **Manage AMI deregistration protection**.
5. In the **Manage AMI deregistration protection** dialog box, choose **Disable**.
6. Choose **Save**.

AWS CLI

To turn off deregistration protection

Use the disable-image-deregistration-protection command.

```
aws ec2 disable-image-deregistration-protection --image-id ami-0abcdef1234567890
```

PowerShell

To turn off deregistration protection

Use the Disable-EC2ImageDeregistrationProtection cmdlet.

```
Disable-EC2ImageDeregistrationProtection -ImageId ami-0abcdef1234567890
```

Instance launch behavior with Amazon EC2 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.

In Amazon EC2, two variants of the boot mode software are supported: Unified Extensible Firmware Interface (UEFI) and Legacy BIOS.

Possible boot mode parameters on an AMI

An AMI can have one of the following boot mode parameter values: `uefi`, `legacy-bios`, or `uefi-preferred`. The AMI boot mode parameter is optional. For AMIs with no boot mode parameter, the instances launched from these AMIs use the default boot mode value of the instance type.

Purpose of the AMI boot mode parameter

The AMI boot mode parameter signals to Amazon 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 will be unsuccessful.

UEFI Preferred boot mode parameter

You can create AMIs that support both UEFI and Legacy BIOS by using the `uefi-preferred` boot mode parameter. When the boot mode parameter is set to `uefi-preferred`, and if the instance type supports UEFI, the instance is launched on UEFI. If the instance type does not support UEFI, the instance is launched on Legacy BIOS.

> **Warning**
>
> Some features, like UEFI Secure Boot, are only available on instances that boot on UEFI. When you use the `uefi-preferred` AMI boot mode parameter with an instance type that does not support UEFI, the instance will launch as Legacy BIOS and the UEFI-dependent feature will be disabled. If you rely on the availability of a UEFI-dependent feature, set your AMI boot mode parameter to `uefi`.

Default boot modes for instance types

- Graviton instance types: UEFI
- Intel and AMD instance types: Legacy BIOS

Zone support

UEFI boot is not supported in Wavelength Zones or AWS Outposts.

Boot mode topics

- Requirements to launch an EC2 instance in UEFI boot mode
- Determine the boot mode parameter of an Amazon EC2 AMI
- Determine the supported boot modes of an EC2 instance type
- Determine the boot mode of an EC2 instance
- Determine the boot mode of the operating system for your EC2 instance
- Set the boot mode of an Amazon EC2 AMI
- UEFI variables for Amazon EC2 instances
- UEFI Secure Boot for Amazon EC2 instances

Requirements to launch an EC2 instance in UEFI boot mode

The boot mode of an instance is determined by the configuration of the AMI, the operating system contained in it, and the instance type. To launch an instance in UEFI boot mode, you must meet the following requirements.

AMI

The AMI must be configured for UEFI as follows:

- **Operating system** – 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 operating system for your EC2 instance.
- **AMI boot mode parameter** – The boot mode parameter of the AMI must be set to `uefi` or `uefi-preferred`. For more information, see Determine the boot mode parameter of an Amazon EC2 AMI.

Linux – The following Linux AMIs support UEFI:

- Amazon Linux 2023
- Amazon Linux 2 (Graviton instance types only)

For other Linux AMIs, you must configure the AMI, import the AMI through VM Import/Export, or import the AMI through CloudEndure.

Windows – The following Windows AMIs support UEFI:

- Windows_Server-2025-* (except for AMIs with the `BIOS-` name prefix)
- TPM-Windows_Server-2022-English-Full-Base
- TPM-Windows_Server-2022-English-Core-Base
- TPM-Windows_Server-2019-English-Full-Base
- TPM-Windows_Server-2019-English-Core-Base
- TPM-Windows_Server-2016-English-Full-Base
- TPM-Windows_Server-2016-English-Core-Base

Instance type

All instances built on the AWS Nitro System support both UEFI and Legacy BIOS, except the following: bare metal instances, DL1, G4ad, P4, u-3tb1, u-6tb1, u-9tb1, u-12tb1, u-18tb1, u-24tb1, and VT1. For more information, see the section called “Instance type boot mode”.

The following table shows that the boot mode of an instance (indicated by the **Resulting instance boot mode** column) is determined by a combination of the boot mode parameter of the AMI (column 1), the boot mode configuration of the operating system contained in the AMI (column 2), and the boot mode support of the instance type (column 3).

AMI boot mode parameter	Operating system boot mode configuration	Instance type boot mode support	Resulting instance boot mode
UEFI	UEFI	UEFI	UEFI
Legacy BIOS	Legacy BIOS	Legacy BIOS	Legacy BIOS
UEFI Preferred	UEFI	UEFI	UEFI
UEFI Preferred	UEFI	UEFI and Legacy BIOS	UEFI
UEFI Preferred	Legacy BIOS	Legacy BIOS	Legacy BIOS
UEFI Preferred	Legacy BIOS	UEFI and Legacy BIOS	Legacy BIOS
No boot mode specified - ARM	UEFI	UEFI	UEFI
No boot mode specified - x86	Legacy BIOS	UEFI and Legacy BIOS	Legacy BIOS

Determine the boot mode parameter of an Amazon EC2 AMI

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

Some AMIs don't 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.

Console

To determine the boot mode parameter of an AMI

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. Inspect the **Boot mode** field.

- A value of **uefi** indicates that the AMI supports UEFI.
- A value of **uefi-preferred** indicates that the AMI supports both UEFI and Legacy BIOS.
- If there is no value, the instances launched from the AMI use the default value of the instance type.

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

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 Application and OS Images (Amazon Machine Image).

AWS CLI

To determine the boot mode parameter of an AMI

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

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-id ami-0abcdef1234567890 \
    --query Images[].BootMode \
    --output text
```

The following is example output.

```
uefi
```

In the output, a value of `uefi` indicates that the AMI supports UEFI. A value of `uefi-preferred` indicates that the AMI supports both UEFI and Legacy BIOS. If there is no value, the instances launched from the AMI use the default value of the instance type.

PowerShell

To determine the boot mode parameter of an AMI

Use the Get-EC2Image cmdlet to determine the boot mode of an AMI.

```
Get-EC2Image -Region us-east-1 `
    -ImageId ami-0abcdef1234567890 | Format-List Name, BootMode, TpmSupport
```

The following is example output.

```
Name        : TPM-Windows_Server-2016-English-Full-Base-2023.05.10
BootMode    : uefi
TpmSupport : v2.0
```

In the output, the value of `BootMode` indicates the boot mode of the AMI. A value of `uefi` indicates that the AMI supports UEFI. A value of `uefi-preferred` indicates that the AMI supports both UEFI and Legacy BIOS. If there is no value, the instances launched from the AMI use the default value of the instance type.

Determine the supported boot modes of an EC2 instance type

You can determine the supported boot modes of an instance type.

The Amazon EC2 console does not display the supported boot modes of an instance type.

AWS CLI

Use the describe-instance-types command to determine the supported boot modes of an instance type. The `--query` parameter filters the output to return only the supported boot modes.

The following example shows that the specified instance type supports both UEFI and Legacy BIOS boot modes.

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

The following is example output.

```
[
    [
        "legacy-bios",
        "uefi"
    ]
]
```

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

```
aws ec2 describe-instance-types \
    --instance-types t2.xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

The following is example output.

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

PowerShell

Use the [Get-EC2InstanceType](#) cmdlet to determine the supported boot modes of an instance type.

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

```
Get-EC2InstanceType -InstanceType m5.2xlarge | Format-List InstanceType,
 SupportedBootModes
```

The following is example output.

```
InstanceType       : m5.2xlarge
SupportedBootModes : {legacy-bios, uefi}
```

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

```
Get-EC2InstanceType -InstanceType t2.xlarge | Format-List InstanceType,
 SupportedBootModes
```

The following is example output.

```
InstanceType       : t2.xlarge
SupportedBootModes : {legacy-bios}
```

To determine the instance types that support UEFI

You can determine the instance types that support UEFI. The Amazon EC2 console does not display the UEFI support of an instance type.

AWS CLI

The available instance types vary by AWS Region. To see the available instance types that support UEFI in a Region, use the [describe-instance-types](#) command. Include the `--filters` parameter to scope the results to the instance types that support UEFI and the `--query` parameter to scope the output to the value of `InstanceType`.

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi \
    --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

PowerShell

The available instance types vary by AWS Region. To see the available instance types that support UEFI in a Region, use the [Get-EC2InstanceType](#) cmdlet.

```
Get-EC2InstanceType | `
 Where-Object {$_.SupportedBootModes -Contains "uefi"} | `
 Sort-Object InstanceType | `
 Format-Table InstanceType -GroupBy CurrentGeneration
```

To determine the instance types that support UEFI Secure Boot and persist non-volatile variables

Bare metal instances do not support UEFI Secure Boot and non-volatile variables, so these examples exclude them from the output. For information about UEFI Secure Boot, see [UEFI Secure Boot for Amazon EC2 instances](#).

AWS CLI

Use the [describe-instance-types](#) command, and exclude the bare metal instances from the output.

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi Name=bare-metal,Values=false \
```

```
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

Use the Get-EC2InstanceType cmdlet, and exclude the bare metal instances from the output.

```
Get-EC2InstanceType | `
    Where-Object { `
        $_.SupportedBootModes -Contains "uefi" -and `
        $_.BareMetal -eq $False
        } | `
    Sort-Object InstanceType  | `
    Format-Table InstanceType, SupportedBootModes, BareMetal, `
        @{Name="SupportedArchitectures";
 Expression={$_.ProcessorInfo.SupportedArchitectures}}
```

Determine the boot mode of an EC2 instance

The boot mode of an instance is displayed in the **Boot mode** field in the Amazon EC2 console, and by the `currentInstanceBootMode` parameter in the AWS CLI.

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 `currentInstanceBootMode` parameter of `uefi`.
- An AMI with a boot mode parameter of `legacy-bios` creates an instance with a `currentInstanceBootMode` parameter of `legacy-bios`.
- An AMI with a boot mode parameter of `uefi-preferred` creates an instance with a `currentInstanceBootMode` parameter of `uefi` if the instance type supports UEFI; otherwise, it creates an instance with a `currentInstanceBootMode` parameter of `legacy-bios`.
- An AMI with no boot mode parameter value creates an instance with a `currentInstanceBootMode` parameter value that is dependent on whether the AMI architecture is ARM or x86 and the supported boot mode of the instance type. The default boot mode is `uefi` on Graviton instance types, and `legacy-bios` on Intel and AMD instance types.

Console

To determine the boot mode of an instance

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.

AWS CLI

To determine the boot mode of an instance

Use the describe-instances command to determine the boot mode of an instance. You can also determine the boot mode of the AMI that was used to the create the instance.

```
aws ec2 describe-instances \
    --region us-east-1 \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BootMode \
    --output text
```

The following is example output.

```
uefi
```

PowerShell

To determine the boot mode of an instance

Use the Get-EC2Image cmdlet to determine the boot mode of an instance. You can also determine the boot mode of the AMI that was used to the create the instance.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | Format-List BootMode,
 CurrentInstanceBootMode, InstanceType, ImageId
```

The following is example output.

```
BootMode                : uefi
CurrentInstanceBootMode : uefi
```

```
InstanceType                 : c5a.large
ImageId                      : ami-0abcdef1234567890
```

Determine the boot mode of the operating system for your EC2 instance

The boot mode of the AMI guides Amazon 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 using SSH (Linux instances) or RDP (Windows instances).

Use the instructions for your instance's operating system.

Linux

To determine the boot mode of the instance's operating system

1. Connect to your Linux instance using SSH.
2. To view the boot mode of the operating system, try one of the following:

   - Run the following command.

     ```
     [ec2-user ~]$ sudo /usr/sbin/efibootmgr
     ```

     Expected output from an instance booted in UEFI boot mode

     ```
     BootCurrent: 0001
     Timeout: 0 seconds
     BootOrder: 0000,0001
     Boot0000* UiApp
     Boot0001* UEFI Amazon Elastic Block Store vol-xyz
     ```

   - Run the following command to verify the existence of the `/sys/firmware/efi` directory. This directory exists only if the instance boots using UEFI. If this directory doesn't exist, the command returns `Legacy BIOS Boot Detected`.

     ```
     [ec2-user ~]$ [ -d /sys/firmware/efi ] && echo "UEFI Boot Detected" || echo
      "Legacy BIOS Boot Detected"
     ```

     Expected output from an instance booted in UEFI boot mode

```
UEFI Boot Detected
```

Expected output from an instance booted in Legacy BIOS boot mode

```
Legacy BIOS Boot Detected
```

- Run the following command to verify that EFI appears in the `dmesg` output.

```
[ec2-user ~]$ dmesg | grep -i "EFI"
```

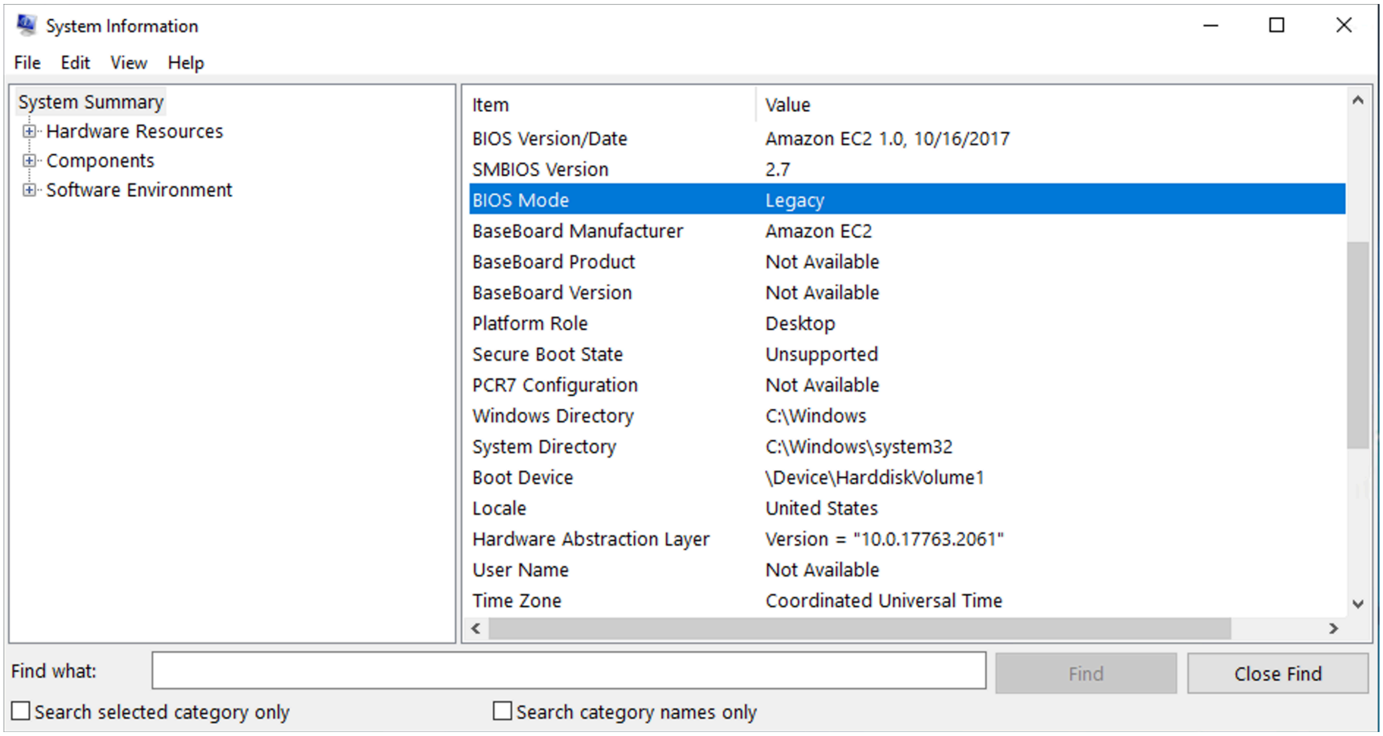
Expected output from an instance booted in UEFI boot mode

```
[    0.000000] efi: Getting EFI parameters from FDT:
[    0.000000] efi: EFI v2.70 by EDK II
```

Windows

To determine the boot mode of the instance's operating system

1. Connect to your Windows instance using RDP.
2. Go to **System Information** and check the **BIOS Mode** row.



Set the boot mode of an Amazon EC2 AMI

By default, an AMI inherits the boot mode of the EC2 instance used to create the AMI. For example, if you create an AMI from an EC2 instance running on Legacy BIOS, the boot mode of the new AMI is `legacy-bios`. If you create an AMI from an EC2 instance with a boot mode of `uefi-preferred`, the boot mode of the new AMI is `uefi-preferred`.

When you register an AMI, you can set the boot mode of the AMI to `uefi`, `legacy-bios`, or `uefi-preferred`.

When the AMI boot mode is set to `uefi-preferred`, the instance boots as follows:

- For instance types that support both UEFI and Legacy BIOS (for example, `m5.large`), the instance boots using UEFI.
- For instance types that support only Legacy BIOS (for example, `m4.large`), the instance boots using Legacy BIOS.

If you set the AMI boot mode to `uefi-preferred`, the operating system must support the ability to boot both UEFI and Legacy BIOS.

To convert an existing Legacy BIOS-based instance to UEFI, or an existing UEFI-based instance to Legacy BIOS, you must first modify the instance's volume and operating system to support the selected boot mode. Then, create a snapshot of the volume. Finally, create an AMI from the snapshot.

Considerations

- Setting the AMI boot mode parameter does not automatically configure the operating system for the specified boot mode. You must first make suitable modifications to the instance's volume and operating system to support booting using the selected boot mode. Otherwise, the resulting AMI is not usable. For example, if you are converting a Legacy BIOS-based Windows 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.
- You can't use the register-image command or the Register-EC2Image cmdlet to create an AMI that supports both NitroTPM and UEFI Preferred.
- Some features, like UEFI Secure Boot, are only available on instances that boot on UEFI. When you use the `uefi-preferred` AMI boot mode parameter with an instance type that does not support UEFI, the instance launches as Legacy BIOS and the UEFI-dependent feature is disabled. If you rely on the availability of a UEFI-dependent feature, set your AMI boot mode parameter to `uefi`.

AWS CLI

To set the boot mode of an AMI

1. Make suitable modifications to the instance's volume and operating system 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.

   > **Warning**
   >
   > 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.

   ```
   aws ec2 describe-instances \
   ```

```
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BlockDeviceMappings
```

The following is example output.

```
[
    [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "AttachTime": "2024-07-11T01:05:51+00:00",
                "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.

```
aws ec2 create-snapshot \
    --volume-id vol-01234567890abcdef \
    --description "my snapshot"
```

The following is example output.

```
{
    "Description": "my snapshot",
    "Encrypted": false,
    "OwnerId": "123456789012",
    "Progress": "",
    "SnapshotId": "snap-0abcdef1234567890",
    "StartTime": "",
    "State": "pending",
    "VolumeId": "vol-01234567890abcdef",
    "VolumeSize": 30,
    "Tags": []
}
```

4. Wait until the state of the snapshot is `completed` before you go to the next step. To get the state of the snapshot, use the [describe-snapshots](#) command with the snapshot ID from the previous step.

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-0abcdef1234567890 \
    --query Snapshots[].State \
    --output text
```

   The following is example output.

```
completed
```

5. To create a new AMI, use the [register-image](#) command. Use the value of `SnapshotId` from the output of **CreateSnapshot**.

   - To set the boot mode to UEFI, add the `--boot-mode` parameter with a value of `uefi`.

```
aws ec2 register-image \
   --description "my image" \
   --name "my-image" \
   --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
   --root-device-name /dev/sda1 \
   --virtualization-type hvm \
   --ena-support \
   --boot-mode uefi
```

   - To set the boot mode to `uefi-preferred`, set the value of `--boot-mode` to `uefi-preferred`

```
aws ec2 register-image \
   --description "my description" \
   --name "my-image" \
   --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
   --root-device-name /dev/sda1 \
   --virtualization-type hvm \
   --ena-support \
   --boot-mode uefi-preferred
```

6. (Optional) To verify that the newly-created AMI has the boot mode that you specified, use the describe-images command.

```
aws ec2 describe-images \
    --image-id ami-1234567890abcdef0 \
    --query Images[].BootMode \
    --output text
```

The following is example output.

```
uefi
```

PowerShell

To set the boot mode of an AMI

1. Make suitable modifications to the instance's volume and operating system 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.

> **Warning**
> If you don't perform this step, the AMI will not be usable.

2. To find the volume ID of the instance, use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

The following is example output.

```
AssociatedResource  :
AttachTime          : 7/11/2024 1:05:51 AM
DeleteOnTermination : True
Operator            :
Status              : attached
VolumeId            : vol-01234567890abcdef
```

3. To create a snapshot of the volume, use the [New-EC2Snapshot](#) cmdlet. Use the volume ID from the previous step.

```
New-EC2Snapshot `
    -VolumeId vol-01234567890abcdef `
    -Description "my snapshot"
```

   The following is example output.

```
AvailabilityZone          :
Description               : my snapshot
Encrypted                 : False
FullSnapshotSizeInBytes   : 0
KmsKeyId                  :
OwnerId                   : 123456789012
RestoreExpiryTime         :
SnapshotId                : snap-0abcdef1234567890
SseType                   :
StartTime                 : 4/25/2025 6:08:59 PM
State                     : pending
StateMessage              :
VolumeId                  : vol-01234567890abcdef
VolumeSize                : 30
```

4. Wait until the state of the snapshot is `completed` before you go to the next step. To get the state of the snapshot, use the [Get-EC2Snapshot](#) cmdlet with the snapshot ID from the previous step.

```
(Get-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890).State.Value
```

   The following is example output.

```
completed
```

5. To create a new AMI, use the [Register-EC2Image](#) cmdlet. Use the value of `SnapshotId` from the output of **New-EC2Snapshot**.

   - To set the boot mode to UEFI, add the `-BootMode` parameter with a value of `uefi`.

```
$block = @{SnapshotId=snap-0abcdef1234567890}
```

```
Register-EC2Image `
    -Description "my image" `
    -Name "my-image" `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
    -RootDeviceName /dev/xvda `
    -EnaSupport $true `
    -BootMode uefi
```

- To set the boot mode to `uefi-preferred`, set the value of `-BootMode` to `uefi-preferred`

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
    -Description "my image" `
    -Name "my-image" `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
    -RootDeviceName /dev/xvda `
    -EnaSupport $true `
    -BootMode uefi-preferred
```

6. (Optional) To verify that the newly-created AMI has the boot mode that you specified, use the Get-EC2Image cmdlet.

```
(Get-EC2Image `
    -ImageId ami-1234567890abcdef0).BootMode.Value
```

The following is example output.

```
uefi
```

UEFI variables for Amazon EC2 instances

When you launch an instance where the boot mode is set to UEFI, a key-value store for variables is created. The store can be used by UEFI and the instance operating system for storing UEFI variables.

UEFI variables are used by the boot loader and the operating system to configure early system startup. They allow the operating system to manage certain settings of the boot process, like the boot order, or managing the keys for UEFI Secure Boot.

> **Warning**
>
> Anyone who can connect to the instance (and potentially any software running on the instance), or anyone with permissions to use the GetInstanceUefiData API on the instance can read the variables. You should never store sensitive data, such as passwords or personally identifiable information, in the UEFI variable store.

UEFI variable persistence

- For instances that were launched on or before May 10, 2022, UEFI variables are wiped on reboot or stop.
- For instances that are launched on or after May 11, 2022, UEFI variables that are marked as non-volatile are persisted on reboot and stop/start.
- Bare metal instances don't preserve UEFI non-volatile variables across instance stop/start operations.

UEFI Secure Boot for Amazon EC2 instances

UEFI Secure Boot builds on the long-standing secure boot process of Amazon EC2, and provides additional defense-in-depth that helps customers secure software from threats that persist across reboots. It ensures that the instance only boots software that is signed with cryptographic keys. The keys are stored in the key database of the UEFI non-volatile variable store. UEFI Secure Boot prevents unauthorized modification of the instance boot flow.

Contents

How UEFI Secure Boot works with Amazon EC2 instances

UEFI Secure Boot is a feature specified in UEFI, which provides verification about the state of the boot chain. It is designed to ensure that only cryptographically verified UEFI binaries are executed after the self-initialization of the firmware. These binaries include UEFI drivers and the main bootloader, as well as chain-loaded components.

UEFI Secure Boot specifies four key databases, which are used in a chain of trust. The databases are stored in the UEFI variable store.

The chain of trust is as follows:

Platform key (PK) database

The PK database is the root of trust. It contains a single public PK key that is used in the chain of trust for updating the key exchange key (KEK) database.

To change the PK database, you must have the private PK key to sign an update request. This includes deleting the PK database by writing an empty PK key.

Key exchange key (KEK) database

The KEK database is a list of public KEK keys that are used in the chain of trust for updating the signature (db) and denylist (dbx) databases.

To change the public KEK database, you must have the private PK key to sign an update request.

Signature (db) database

The db database is a list of public keys and hashes that are used in the chain of trust to validate all UEFI boot binaries.

To change the db database, you must have the private PK key or any of the private KEK keys to sign an update request.

Signature denylist (dbx) database

The dbx database is a list of public keys and binary hashes that are not trusted, and are used in the chain of trust as a revocation file.

The dbx database always takes precedence over all other key databases.

To change the dbx database, you must have the private PK key or any of the private KEK keys to sign an update request.

The UEFI Forum maintains a publicly available dbx for many known-bad binaries and certs at https://uefi.org/revocationlistfile.

Important

UEFI Secure Boot enforces signature validation on any UEFI binaries. To permit execution of a UEFI binary in UEFI Secure Boot, you sign it with any of the private db keys described above.

By default, UEFI Secure Boot is disabled and the system is in `SetupMode`. When the system is in `SetupMode`, all key variables can be updated without a cryptographic signature. When the PK is set, UEFI Secure Boot is enabled and the SetupMode is exited.

Requirements for UEFI Secure Boot on Amazon EC2

When you launch an Amazon EC2 instance with a supported AMI and a supported instance type, that instance will automatically validate UEFI boot binaries against its UEFI Secure Boot database. No additional configuration is required. You can also configure UEFI Secure Boot on an instance after launch.

Note

UEFI Secure Boot protects your instance and its operating system against boot flow modifications. If you create a new AMI from a source AMI that has UEFI Secure Boot enabled and modify certain parameters during the copy process, such as changing the `UefiData` within the AMI, you can disable UEFI Secure Boot.

Contents

Supported AMIs

Linux AMIs

To launch a Linux instance, the Linux AMI must have UEFI Secure Boot enabled.

Amazon Linux supports UEFI Secure Boot starting with AL2023 release 2023.1. However, UEFI Secure Boot isn't enabled in the default AMIs. For more information, see UEFI Secure Boot in the

AL2023 User Guide. Older versions of Amazon Linux AMIs aren't enabled for UEFI Secure Boot. To use a supported AMI, you must perform a number of configuration steps on your own Linux AMI. For more information, see Create a Linux AMI with custom UEFI Secure Boot keys.

Windows AMIs

To launch a Windows instance, the Windows AMI must have UEFI Secure Boot enabled. To find an AWS Windows AMI that's preconfigured for UEFI Secure Boot with Microsoft keys, see Find Windows Server AMIs configured with NitroTPM and UEFI Secure Boot in the *AWS Windows AMIs Reference*.

Currently, we do not support importing Windows with UEFI Secure Boot by using the import-image command.

Supported instance types

All virtualized instance types that support UEFI also support UEFI Secure Boot. For the instance types that support UEFI Secure Boot, see Requirements for UEFI boot mode.

> **Note**
>
> Bare metal instance types do not support UEFI Secure Boot.

Verify whether an Amazon EC2 instance is enabled for UEFI Secure Boot

You can use the following procedures to determine whether an Amazon EC2 is enabled for UEFI Secure Boot.

Linux instances

You can use the `mokutil` utility to verify whether a Linux instance is enabled for UEFI Secure Boot. If `mokutil` is not installed on your instance, you must install it. For the installation instructions for Amazon Linux 2, see Find and install software packages on an Amazon Linux 2 instance. For other Linux distributions, see their specific documentation.

To verify whether a Linux instance is enabled for UEFI Secure Boot

Connect to your instance and run the following command as `root` in a terminal window.

```
mokutil --sb-state
```

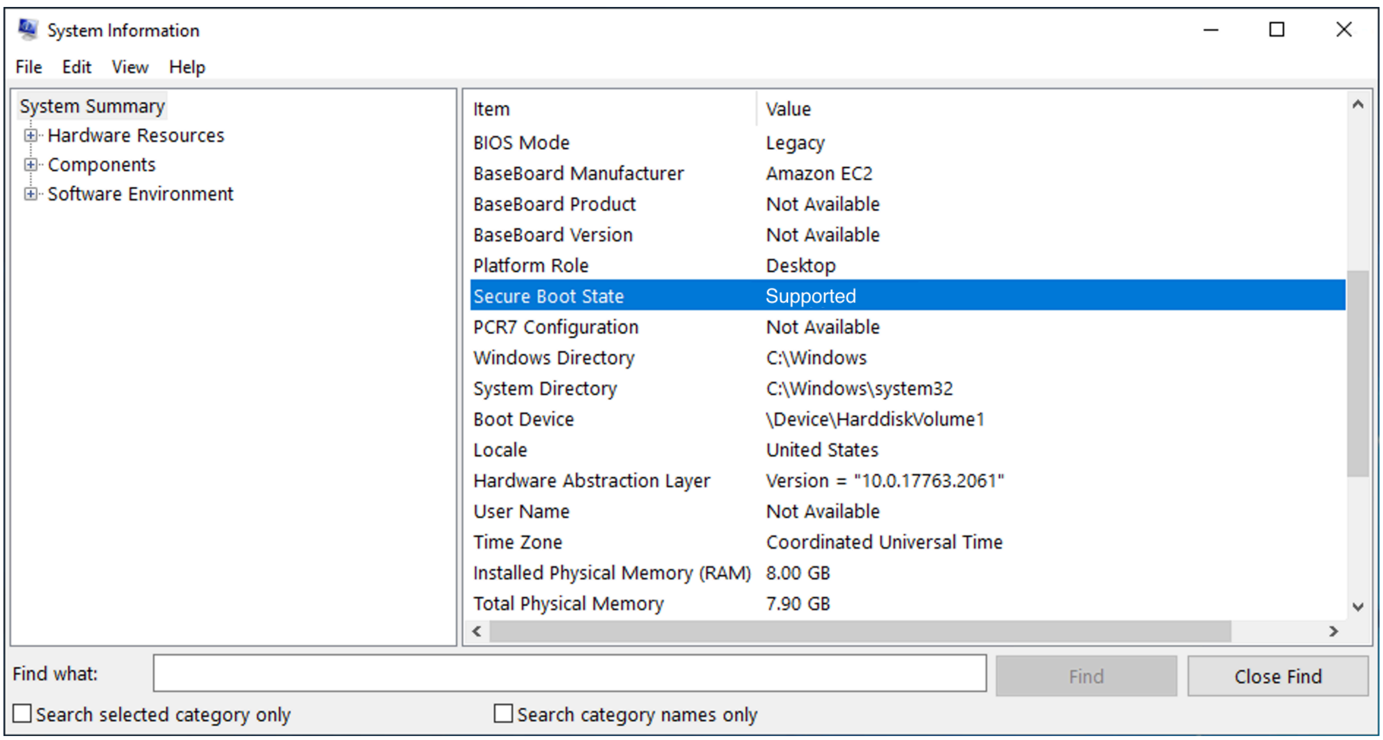
The following is example output.

- If UEFI Secure Boot is enabled, the output contains `SecureBoot enabled`.
- If UEFI Secure Boot is not enabled, the output contains `SecureBoot disabled` or `Failed to read SecureBoot`.

Windows instances

To verify whether a Windows instance is enabled for UEFI Secure Boot

1. Connect to your instance.
2. Open the msinfo32 tool.
3. Check the **Secure Boot State** field. If UEFI Secure Boot is enabled, the value is **Supported**, as shown in the following image.



You can also use the Windows PowerShell Cmdlet `Confirm-SecureBootUEFI` to check the the Secure Boot status. For more information about the cmdlet, see Confirm-SecureBootUEFI in the Microsoft Documentation.

Create a Linux AMI with custom UEFI Secure Boot keys

These instructions show you how to create a Linux AMI with UEFI Secure Boot and custom-made private keys. Amazon Linux supports UEFI Secure Boot starting with AL2023 release 2023.1. For more information, see [UEFI Secure Boot on AL2023](#) in the *Amazon Linux 2023 User Guide*.

> **Important**
>
> The following procedure is intended for **advanced users only**. You must have sufficient knowledge of SSL and Linux distribution boot flow to use these procedures.

Prerequisites

- The following tools will be used:
  - OpenSSL – https://www.openssl.org/
  - efivar – https://github.com/rhboot/efivar
  - efitools – https://git.kernel.org/pub/scm/linux/kernel/git/jejb/efitools.git/
  - get-instance-uefi-data command
- Your Linux instance must have been launched with a Linux AMI that supports UEFI boot mode, and have non-volatile data present.

Newly created instances without UEFI Secure Boot keys are created in `SetupMode`, which allows you to enroll your own keys. Some AMIs come preconfigured with UEFI Secure Boot and you cannot change the existing keys. If you want to change the keys, you must create a new AMI based on the original AMI.

You have two ways to propagate the keys in the variable store, which are described in Option A and Option B that follow. Option A describes how to do this from within the instance, mimicking the flow of real hardware. Option B describes how to create a binary blob, which is then passed as a base64-encoded file when you create the AMI. For both options, you must first create the three key pairs, which are used for the chain of trust.

To create a Linux AMI that supports UEFI Secure Boot, first create the three key pairs, and then complete either Option A or Option B, but not both:

- Task 1: Create key pairs
- Task 2 - Option A: Add keys to the variable store from within the instance

- Task 2 - Option B: Create a binary blob containing a pre-filled variable store

Task 1: Create key pairs

UEFI Secure Boot is based on the following three key databases, which are used in a chain of trust: the platform key (PK), the key exchange key (KEK), and the signature database (db).[1]

You create each key on the instance. To prepare the public keys in a format that is valid for the UEFI Secure Boot standard, you create a certificate for each key. DER defines the SSL format (binary encoding of a format). You then convert each certificate into a UEFI signature list, which is the binary format that is understood by UEFI Secure Boot. And finally, you sign each certificate with the relevant key.

Tasks

- Prepare to create the key pairs
- Key pair 1: Create the platform key (PK)
- Key pair 2: Create the key exchange key (KEK)
- Key pair 3: Create the signature database (db)
- Sign the boot image (kernel) with the private key

Prepare to create the key pairs

Before creating the key pairs, create a globally unique identifier (GUID) to be used in key generation.

1. Connect to the instance.
2. Run the following command in a shell prompt.

```
uuidgen --random > GUID.txt
```

Key pair 1: Create the platform key (PK)

The PK is the root of trust for UEFI Secure Boot instances. The private PK is used to update the KEK, which in turn can be used to add authorized keys to the signature database (db).

The X.509 standard is used for creating the key pair. For information about the standard, see X.509 on *Wikipedia*.

To create the PK

1. Create the key. You must name the variable PK.

```
openssl req -newkey rsa:4096 -nodes -keyout PK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Platform key/" -out PK.crt
```

   The following parameters are specified:

   - `-keyout PK.key` – The private key file.
   - `-days 3650` – The number of days that the certificate is valid.
   - `-out PK.crt` – The certificate that is used to create the UEFI variable.
   - `CN=Platform key` – The common name (CN) for the key. You can enter the name of your own organization instead of *Platform key*.

2. Create the certificate.

```
openssl x509 -outform DER -in PK.crt -out PK.cer
```

3. Convert the certificate into a UEFI signature list.

```
cert-to-efi-sig-list -g "$(< GUID.txt)" PK.crt PK.esl
```

4. Sign the UEFI signature list with the private PK (self-signed).

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt PK PK.esl PK.auth
```

Key pair 2: Create the key exchange key (KEK)

The private KEK is used to add keys to the db, which is the list of authorized signatures to boot on the system.

To create the KEK

1. Create the key.

```
openssl req -newkey rsa:4096 -nodes -keyout KEK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Key Exchange Key/" -out KEK.crt
```

2. Create the certificate.

```
openssl x509 -outform DER -in KEK.crt -out KEK.cer
```

3. Convert the certificate into a UEFI signature list.

```
cert-to-efi-sig-list -g "$(< GUID.txt)" KEK.crt KEK.esl
```

4. Sign the signature list with the private PK.

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt KEK KEK.esl KEK.auth
```

Key pair 3: Create the signature database (db)

The db list contains authorized keys that are authorized to be booted on the system. To modify the list, the private KEK is necessary. Boot images will be signed with the private key that is created in this step.

To create the db

1. Create the key.

```
openssl req -newkey rsa:4096 -nodes -keyout db.key -new -x509 -sha256 -days 3650 -
subj "/CN=Signature Database key/" -out db.crt
```

2. Create the certificate.

```
openssl x509 -outform DER -in db.crt -out db.cer
```

3. Convert the certificate into a UEFI signature list.

```
cert-to-efi-sig-list -g "$(< GUID.txt)" db.crt db.esl
```

4. Sign the signature list with the private KEK.

```
sign-efi-sig-list -g "$(< GUID.txt)" -k KEK.key -c KEK.crt db db.esl db.auth
```

Sign the boot image (kernel) with the private key

For Ubuntu 22.04, the following images require signatures.

```
/boot/efi/EFI/ubuntu/shimx64.efi
/boot/efi/EFI/ubuntu/mmx64.efi
/boot/efi/EFI/ubuntu/grubx64.efi
/boot/vmlinuz
```

To sign an image

Use the following syntax to sign an image.

```
sbsign --key db.key --cert db.crt --output /boot/vmlinuz /boot/vmlinuz
```

> **Note**
>
> You must sign all new kernels. */boot/vmlinuz* will usually symlink to the last installed kernel.

Refer to the documentation for your distribution to find out about your boot chain and required images.

[1] Thanks to the ArchWiki community for all of the work they have done. The commands for creating the PK, creating the KEK, creating the DB, and signing the image are from Creating keys, authored by the ArchWiki Maintenance Team and/or the ArchWiki contributors.

Task 2 - Option A: Add keys to the variable store from within the instance

After you have created the three key pairs, you can connect to your instance and add the keys to the variable store from within the instance by completing the following steps. Alternatively, complete the steps in the section called "Task 2 - Option B: Create a binary blob containing a pre-filled variable store".

Option A steps:

- Step 1: Launch an instance that will support UEFI Secure Boot
- Step 2: Configure an instance to support UEFI Secure Boot
- Step 3: Create an AMI from the instance

Step 1: Launch an instance that will support UEFI Secure Boot

When you launch an instance with the following prerequisites, the instance will then be ready to be configured to support UEFI Secure Boot. You can only enable support for UEFI Secure Boot on an instance at launch; you can't enable it later.

Prerequisites

- **AMI –** The Linux AMI must support UEFI boot mode. To verify that the AMI supports UEFI boot mode, the AMI boot mode parameter must be **uefi**. For more information, see Determine the boot mode parameter of an Amazon EC2 AMI.

  Note that AWS only provides Linux AMIs configured to support UEFI for Graviton-based instance types. AWS currently does not provide x86_64 Linux AMIs that support UEFI boot mode. You can configure your own AMI to support UEFI boot mode for all architectures. To configure your own AMI to support UEFI boot mode, you must perform a number of configuration steps on your own AMI. For more information, see Set the boot mode of an Amazon EC2 AMI.
- **Instance type –** All virtualized instance types that support UEFI also support UEFI Secure Boot. Bare metal instance types do not support UEFI Secure Boot. For the instance types that support UEFI Secure Boot, see Requirements for UEFI boot mode.
- Launch your instance after the release of UEFI Secure Boot. Only instances launched after May 10, 2022 (when UEFI Secure Boot was released) can support UEFI Secure Boot.

After you've launched your instance, you can verify that it is ready to be configured to support UEFI Secure Boot (in other words, you can proceed to Step 2) by checking whether UEFI data is present. The presence of UEFI data indicates that non-volatile data is persisted.

To verify whether your instance is ready for Step 2

Use the get-instance-uefi-data command and specify the instance ID.

```
aws ec2 get-instance-uefi-data --instance-id i-1234567890abcdef0
```

The instance is ready for Step 2 if UEFI data is present in the output. If the output is empty, the instance cannot be configured to support UEFI Secure Boot. This can happen if your instance was launched before UEFI Secure Boot support became available. Launch a new instance and try again.

Step 2: Configure an instance to support UEFI Secure Boot

Enroll the key pairs in your UEFI variable store on the instance

> **Warning**
>
> You must sign your boot images *after* you enroll the keys, otherwise you won't be able to boot your instance.

After you create the signed UEFI signature lists (PK, KEK, and db), they must be enrolled into the UEFI firmware.

Writing to the PK variable is possible only if:

- No PK is enrolled yet, which is indicated if the `SetupMode` variable is `1`. Check this by using the following command. The output is either `1` or `0`.

  ```
  efivar -d -n 8be4df61-93ca-11d2-aa0d-00e098032b8c-SetupMode
  ```
- The new PK is signed by the private key of the existing PK.

To enroll the keys in your UEFI variable store

The following commands must be run on the instance.

If SetupMode is enabled (the value is 1), the keys can be enrolled by running the following commands on the instance:

```
[ec2-user ~]$ efi-updatevar -f db.auth db
```

```
[ec2-user ~]$ efi-updatevar -f KEK.auth KEK
```

```
[ec2-user ~]$ efi-updatevar -f PK.auth PK
```

To verify that UEFI Secure Boot is enabled

To verify that UEFI Secure Boot is enabled, follow the steps in Verify whether an Amazon EC2 instance is enabled for UEFI Secure Boot.

You can now export your UEFI variable store with the get-instance-uefi-data CLI command, or you continue to the next step and sign your boot images to reboot into a UEFI Secure Boot-enabled instance.

Step 3: Create an AMI from the instance

To create an AMI from the instance, you can use the console or the `CreateImage` API, CLI, or SDKs. For the console instructions, see Create an Amazon EBS-backed AMI. For the API instructions, see CreateImage.

> **Note**
>
> The `CreateImage` API automatically copies the UEFI variable store of the instance to the AMI. The console uses the `CreateImage` API. After you launch instances using this AMI, the instances will have the same UEFI variable store.

Task 2 - Option B: Create a binary blob containing a pre-filled variable store

After you have created the three key pairs, you can create a binary blob containing a pre-filled variable store containing the UEFI Secure Boot keys. Alternatively, complete the steps in the section called "Task 2 - Option A: Add keys to the variable store from within the instance".

> **Warning**
>
> You must sign your boot images *before* you enroll the keys, otherwise you won't be able to boot your instance.

Option B steps:

- Step 1: Create a new variable store or update an existing one
- Step 2: Upload the binary blob on AMI creation

Step 1: Create a new variable store or update an existing one

You can create the variable store *offline* without a running instance by using the python-uefivars tool. The tool can create a new variable store from your keys. The script currently supports the EDK2 format, the AWS format, and a JSON representation that is easier to edit with higher-level tooling.

To create the variable store offline without a running instance

1. Download the tool at the following link.

```
https://github.com/awslabs/python-uefivars
```

2. Create a new variable store from your keys by running the following command. This will create a base64-encoded binary blob in *your_binary_blob*.bin. The tool also supports updating a binary blob via the `-I` parameter.

```
./uefivars.py -i none -o aws -O your_binary_blob.bin -P PK.esl -K KEK.esl --db
 db.esl --dbx dbx.esl
```

Step 2: Upload the binary blob on AMI creation

Use [register-image](https://docs.aws.amazon.com/cli/latest/reference/ec2/register-image.html) to pass your UEFI variable store data. For the `--uefi-data` parameter, specify your binary blob, and for the `--boot-mode` parameter, specify `uefi`.

```
aws ec2 register-image \
    --name uefi_sb_tpm_register_image_test \
    --uefi-data $(cat your_binary_blob.bin) \
    --block-device-mappings "DeviceName=/dev/sda1,Ebs=
 {SnapshotId=snap-0123456789example,DeleteOnTermination=true}" \
    --architecture x86_64 \
    --root-device-name /dev/sda1 \
    --virtualization-type hvm \
    --ena-support \
    --boot-mode uefi
```

Create the AWS binary blob for UEFI Secure Boot

You can use the following steps to customize the UEFI Secure Boot variables during AMI creation. The KEK that is used in these steps is current as of September 2021. If Microsoft updates the KEK, you must use the latest KEK.

To create the AWS binary blob

1. Create an empty PK signature list.

```
touch empty_key.crt
```

```
cert-to-efi-sig-list empty_key.crt PK.esl
```

2. Download the KEK certificates.

```
https://go.microsoft.com/fwlink/?LinkId=321185
```

3. Wrap the KEK certificates in a UEFI signature list (`siglist`).

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_KEK.esl MicCorKEKCA2011_2011-06-24.crt
```

4. Download Microsoft's db certificates.

```
https://www.microsoft.com/pkiops/certs/MicWinProPCA2011_2011-10-19.crt
https://www.microsoft.com/pkiops/certs/MicCorUEFCA2011_2011-06-27.crt
```

5. Generate the db signature list.

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_db.esl MicWinProPCA2011_2011-10-19.crt
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_UEFI_db.esl MicCorUEFCA2011_2011-06-27.crt
cat MS_Win_db.esl MS_UEFI_db.esl > MS_db.esl
```

6. The Unified Extensible Firmware Interface Forum no longer provides the DBX files. They are now provided by Microsoft on GitHub. Download the latest DBX update from the Microsoft Secure Boot updates repository at https://github.com/microsoft/secureboot_objects.

7. Unpack the signed update-binary.

   Create `SplitDbxContent.ps1` with the script content below. Alternatively, you can install the script from [PowerShell Gallery](#) using `Install-Script -Name SplitDbxContent`.

```
<#PSScriptInfo

.VERSION 1.0

.GUID ec45a3fc-5e87-4d90-b55e-bdea083f732d

.AUTHOR Microsoft Secure Boot Team

.COMPANYNAME Microsoft

```

```
.COPYRIGHT Microsoft

.TAGS Windows Security

.LICENSEURI

.PROJECTURI

.ICONURI

.EXTERNALMODULEDEPENDENCIES

.REQUIREDSCRIPTS

.EXTERNALSCRIPTDEPENDENCIES

.RELEASENOTES
Version 1.0: Original published version.

#>

<#
.DESCRIPTION
 Splits a DBX update package into the new DBX variable contents and the signature
 authorizing the change.
 To apply an update using the output files of this script, try:
 Set-SecureBootUefi -Name dbx -ContentFilePath .\content.bin -SignedFilePath .
\signature.p7 -Time 2010-03-06T19:17:21Z -AppendWrite'
.EXAMPLE
.\SplitDbxAuthInfo.ps1 DbxUpdate_x64.bin
#>


# Get file from script input
$file  = Get-Content -Encoding Byte $args[0]

# Identify file signature
$chop = $file[40..($file.Length - 1)]
if (($chop[0] -ne 0x30) -or ($chop[1] -ne 0x82 )) {
    Write-Error "Cannot find signature"
    exit 1
}

# Signature is known to be ASN size plus header of 4 bytes
```

```
$sig_length = ($chop[2] * 256) + $chop[3] + 4
$sig = $chop[0..($sig_length - 1)]

if ($sig_length -gt ($file.Length + 40)) {
    Write-Error "Signature longer than file size!"
    exit 1
}

# Content is everything else
$content = $file[0..39] + $chop[$sig_length..($chop.Length - 1)]

# Write signature and content to files
Set-Content -Encoding Byte signature.p7 $sig
Set-Content -Encoding Byte content.bin $content
```

Use the script to unpack the signed DBX files.

```
PS C:\Windows\system32> SplitDbxContent.ps1 .\dbx.bin
```

This produces two files — `signature.p7` and `content.bin`. Use `content.bin` in the next step.

8. Build a UEFI variable store using the `uefivars.py` script.

```
./uefivars.py -i none -o aws -O uefiblob-microsoft-keys-empty-pk.bin -P ~/PK.esl -K
 ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx ~/content.bin
```

9. Check the binary blob and the UEFI variable store.

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o json | less
```

10. You can update the blob by passing it to the same tool again.

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o aws -O uefiblob-
microsoft-keys-empty-pk.bin -P ~/PK.esl -K ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx
 ~/content.bin
```

Expected output

```
Replacing PK
Replacing KEK
```

```
Replacing db
Replacing dbx
```

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 [Understand shared AMI usage in Amazon EC2](#).

Encryption with EBS-backed AMIs topics

- [Instance-launching scenarios](#)
- [Image-copying scenarios](#)

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, see [Block device mappings for volumes on Amazon EC2 instances](#). 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 is enabled, all volumes created from the AMI (whether from encrypted or unencrypted snapshots) are encrypted. If encryption by default is not enabled, 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.

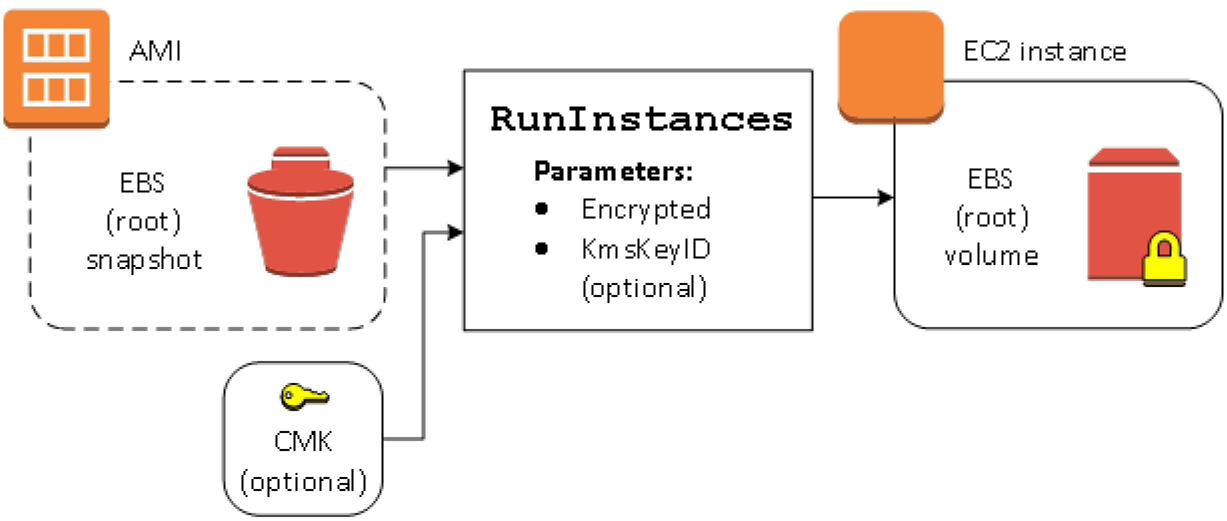
Submitting a `KmsKeyId` without also setting the `Encrypted` parameter results in an error.

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 an Amazon EC2 instance.

Encrypt a volume during launch

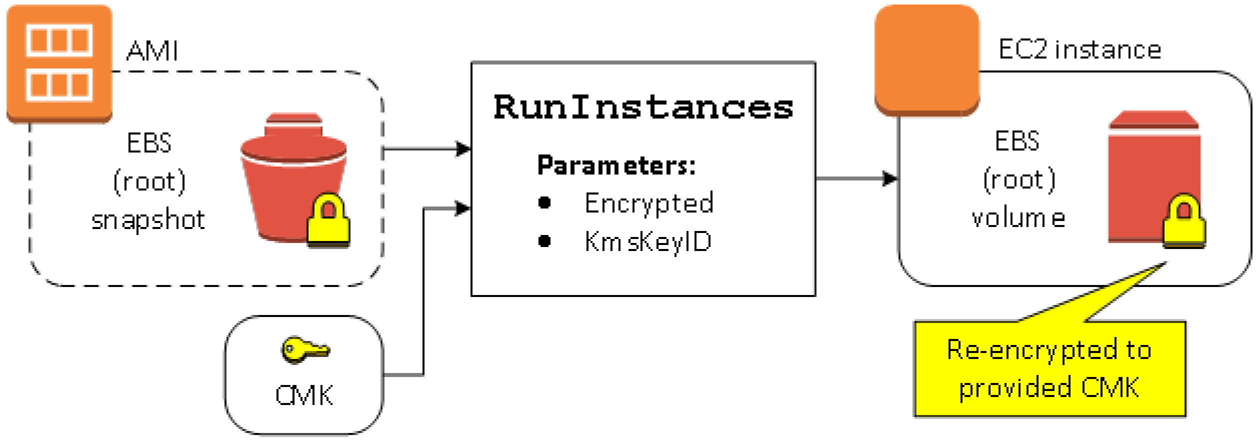
In this example, an AMI backed by an unencrypted snapshot is used to launch an EC2 instance with an encrypted EBS volume.



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

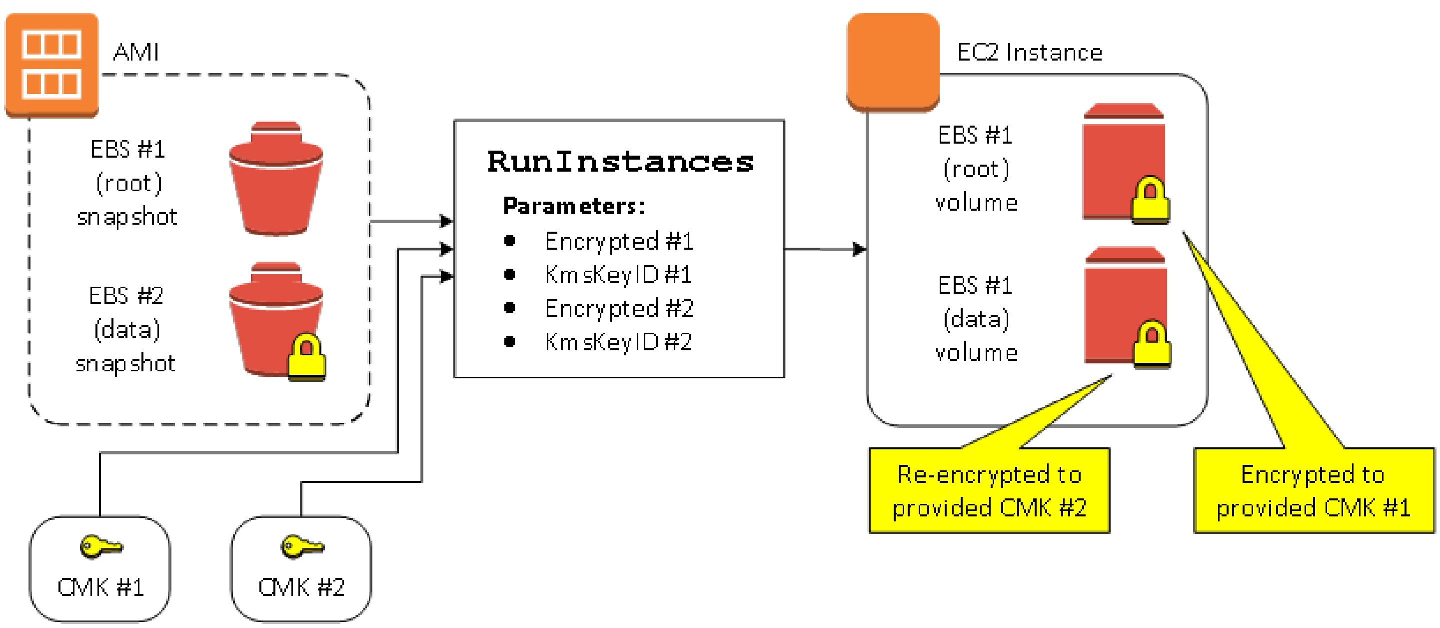
In this example, an AMI backed by an encrypted snapshot is used to launch an EC2 instance with an EBS volume encrypted by a new KMS key.



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.

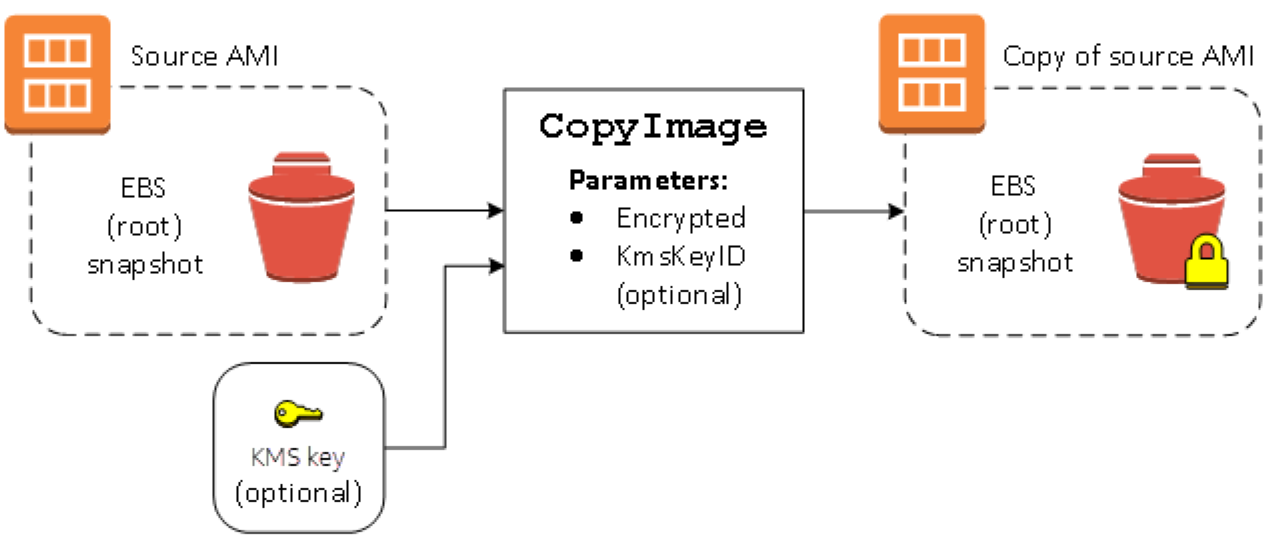
The following section provides an example of copying an AMI using non-default encryption parameters, resulting in a change of encryption state.

For detailed instructions using the console, see Copy an Amazon EC2 AMI.

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 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 shared AMI usage in Amazon EC2

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, such as a verified provider.

Verified provider

In the Amazon EC2 console, public AMIs that are owned by Amazon or a verified Amazon partner are marked **Verified provider**.

You can also use the describe-images AWS CLI command to identify the public AMIs that come from a verified provider. Public images that are owned by Amazon or a verified partner have an aliased owner, which is either `amazon`, `aws-backup-vault`, or `aws-marketplace`. In the CLI output, these values appear for `ImageOwnerAlias`. Other users can't alias their AMIs. This enables you to easily find AMIs from Amazon or verified partners.

To become a verified provider, you must register as a seller on the AWS Marketplace. Once registered, you can list your AMI on the AWS Marketplace. For more information, see Getting started as a seller and AMI-based products in the *AWS Marketplace Seller Guide*.

Shared AMI topics

- Find shared AMIs to use for Amazon EC2 instances

- Prepare to use shared AMIs for Linux
- Control the discovery and use of AMIs in Amazon EC2 with Allowed AMIs
- Make your AMI publicly available for use in Amazon EC2
- Understand block public access for AMIs
- Share an AMI with organizations and organizational units
- Share an AMI with specific AWS accounts
- Cancel having an AMI shared with your AWS account
- Recommendations for creating shared Linux AMIs

If you're looking for information about other topics

- For information about creating an AMI, see the section called “Create an instance store-backed AMI” or the section called “Create an AMI”.
- For information about building, delivering, and maintaining your applications on the AWS Marketplace, see the AWS Marketplace Documentation.

Find shared AMIs to use for Amazon EC2 instances

You can use the Amazon EC2 console or the command line to find public or private shared AMIs to use with your Amazon EC2 instances.

AMIs are a Regional resource. When you search for a shared AMI (public or private), you must search for it from the same Region from which it is 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 Amazon EC2 AMI.

Console

The console provides an AMI filter field. You can also scope your searches using the filters provided in the **Search** field.

To find a shared or AMI

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 one of the following options:

- **Private images** – Lists all AMIs that are shared with you.
- **Public images** – Lists all public AMIs.

4. (Optional) To display only the public images from Amazon, choose the **Search** field and then, from the menu options, choose **Owner alias**, then **=**, and then **amazon**.
5. (Optional) Add filters to scope your search to AMIs that meet your requirements.

To find a shared public AMI from a verified provider

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMI Catalog**.
3. Choose **Community AMIs**.
4. In the **Refine results** pane, select **Verified provider**. The **Verified provider** label indicates that the AMIs are from Amazon or a verified partner.

AWS CLI

Use the describe-images command to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

To 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
```

To 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
```

To list AMIs owned by verified providers

The following command lists the AMIs owned by verified providers. Public AMIs owned by verified providers (either Amazon or verified partners) have an aliased owner, which appears as amazon, aws-backup-vault, or aws-marketplace in the account field. This helps you to easily find AMIs from verified providers. Other users can't alias their AMIs.

```
aws ec2 describe-images \
    --owners amazon aws-marketplace \
    --query 'Images[*].[ImageId]' \
    --output text
```

To list AMIs owned by an account

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

```
aws ec2 describe-images --owners 123456789012
```

To 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"
```

PowerShell

Use the Get-EC2Image cmdlet (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.

To list all public AMIs

The following command lists all public AMIs, including any public AMIs that you own.

```
Get-EC2Image -ExecutableUser all
```

To 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.

```
Get-EC2Image -ExecutableUser self
```

To list AMIs owned by verified providers

The following command lists the AMIs owned by verified providers. Public AMIs owned by verified providers (either Amazon or verified partners) have an aliased owner, which appears as `amazon`, `aws-backup-vault`, or `aws-marketplace` in the account field. This helps you to easily find AMIs from verified providers. Other users can't alias their AMIs.

```
Get-EC2Image -Owner amazon aws-marketplace
```

To list AMIs owned by an account

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

```
Get-EC2Image -Owner 123456789012
```

To 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"}
```

Prepare to use shared AMIs for Linux

Before you use a shared AMI for Linux, take the following steps to confirm that there are no pre-installed credentials that would allow unwanted access to your instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party. Check the documentation for the Linux distribution used by the AMI for information about improving the security of the system.

To ensure that you don't accidentally lose access to your instance, we recommend that you initiate two SSH sessions and keep the second session open until you've removed credentials that you don't recognize and confirmed that you can still log into your instance using SSH.

1. Identify and disable any unauthorized public SSH keys. The only key in the file should be the key you used to launch the AMI. The following command locates `authorized_keys` files:

   ```
   [ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
   ```

2. Disable password-based authentication for the root user. Open the `sshd_config` file and edit the `PermitRootLogin` line as follows:

```
PermitRootLogin without-password
```

Alternatively, you can disable the ability to log into the instance as the root user:

```
PermitRootLogin No
```

Restart the sshd service.

3. Check whether there are any other users that are able to log in to your instance. Users with superuser privileges are particularly dangerous. Remove or lock the password of any unknown accounts.
4. Check for open ports that you aren't using and running network services listening for incoming connections.
5. To prevent preconfigured remote logging, you should delete the existing configuration file and restart the `rsyslog` service. For example:

```
[ec2-user ~]$ sudo rm /etc/rsyslog.conf
[ec2-user ~]$ sudo service rsyslog restart
```

6. Verify that all cron jobs are legitimate.

If you discover a public AMI that you feel presents a security risk, contact the AWS security team. For more information, see the AWS Security Center.

Control the discovery and use of AMIs in Amazon EC2 with Allowed AMIs

To control the discovery and use of Amazon Machine Images (AMIs) by users in your AWS account, you can use the *Allowed AMIs* feature. This feature allows you to specify criteria that AMIs must meet to be visible and available within your account. When the criteria are enabled, users launching instances will only see and have access to AMIs that comply with the specified criteria. For example, you can specify a list of trusted AMI providers as the criteria, and only AMIs from these providers will be visible and available for use.

Before enabling the Allowed AMIs settings, you can enable *audit mode* to preview which AMIs will or won't be visible and available for use. This lets you refine the criteria as needed to ensure that only the intended AMIs are visible and available to users in your account. Additionally, you can run the describe-instance-image-metadata command and filter the response to identify any instances that were launched with AMIs that don't meet the specified criteria. This information can guide your decision to either update your launch configurations to use compliant AMIs (for example, specifying a different AMI in a launch template) or adjust your criteria to allow these AMIs.

You specify the Allowed AMIs settings at the account level, either directly in the account or by using a declarative policy. These settings must be configured in each AWS Region where you want to control the discovery and use of AMIs. Using a declarative policy allows you to apply the settings across multiple Regions simultaneously, as well as across multiple accounts simultaneously. When a declarative policy is in use, you can't modify the settings directly within an account. This topic describes how to configure the settings directly within an account. For information about using declarative policies, see Declarative policies in the *AWS Organizations User Guide*.

> **Note**
>
> The Allowed AMIs feature only controls the discovery and use of public AMIs or AMIs shared with your account. It does not restrict the AMIs owned by your account. Regardless of the criteria you set, the AMIs created by your account are always discoverable and usable by users in your account.

Key benefits of Allowed AMIs

- **Compliance and security**: Users can only discover and use AMIs that meet the specified criteria, reducing the risk of non-compliant AMI usage.
- **Efficient management**: By reducing the number of allowed AMIs, managing the remaining ones becomes easier and more efficient.
- **Centralized account-level implementation**: Configure the Allowed AMIs settings at the account level, either directly within the account or through a declarative policy. This provides a centralized and efficient way to control AMI usage across the entire account.

Contents

How Allowed AMIs works

You control AMI access by specifying *criteria* in a JSON configuration. These criteria determine which AMIs can be discovered and used in your account. After defining your criteria, you enable them using the API or console. Once enabled, all AMIs are evaluated against the criteria. If an AMI matches at least one of the criteria, it is allowed in your account.

Topics

Allowed AMIs criteria

Currently, the only supported criteria are AMI providers. Valid values are aliases that are defined by AWS, and AWS account IDs, as follows:

- `amazon` – An alias that identifies AMIs created by AWS
- `aws-marketplace` – An alias that identifies AMIs created by verified providers in the AWS Marketplace
- `aws-backup-vault` – An alias that identifies backup AMIs that reside in logically air-gapped AWS Backup vault accounts. If you use the AWS Backup logically air-gapped vault feature, ensure this alias is included as an AMI provider.
- AWS account IDs – One or more 12-digit AWS account IDs
- `none` – Indicates that only AMIs created by your account can be discovered and used. Public or shared AMIs can't be discovered and used. If you specify `none`, you can't specify an alias or account ID.

JSON configuration

The core configuration for Allowed AMIs is the JSON configuration that defines the criteria for allowed AMIs. Here's an example that specifies two aliases and three AWS account IDs:

```
{
```

```
    "ImageCriteria": [
        {
            "ImageProviders": [
                "amazon",
                "aws-marketplace",
                "123456789012",
                "112233445566",
                "009988776655"
            ]
        }
    ]
}
```

How criteria are evaluated

The criteria in the JSON configuration are evaluated using an `or` condition. This means that an AMI only needs to match one of the specified providers in any `ImageCriteria` object to be considered allowed. For example, if you specify both `amazon` and an AWS account ID as providers, AMIs from either provider will be allowed.

Limits for the JSON configuration

- `ImageCriteria` objects: Maximum of 10 `ImageCriteria` objects can be specified in a single configuration.
- `ImageProviders` values: Maximum of 200 values across all `ImageCriteria` objects.

Example of limits

Consider the following example to illustrate these limits, where different `ImageProviders` lists are used to group the AMI provider accounts:

```
{
    "ImageCriteria": [
        {
            "ImageProviders": ["amazon", "aws-marketplace"]
        },
        {
            "ImageProviders": ["123456789012", "112233445566", "121232343454"]
        },
        {
            "ImageProviders": ["998877665555", "987654321098"]
        }
```

```
        // Up to 7 more ImageCriteria objects can be added
        // Up to 193 more ImageProviders values can be added
    ]
}
```

In this example:

- There are 3 `imageCriteria` objects (up to 7 more can be added to reach the limit of 10).
- There are 7 total `imageProviders` values across all objects (up to 193 more can be added to reach the limit of 200).

In this example, AMIs are allowed from any of the specified AMI providers in all of the `ImageCriteria` objects, since they are evaluated using an `or` condition.

Allowed AMIs operations

The Allowed AMIs feature has three operational modes for managing the image criteria: **enabled**, **disabled**, and **audit mode**. These allow you to enable or disable the image criteria, or review them as needed.

Enabled

When Allowed AMIs is enabled:

- The `ImageCriteria` are applied.
- Only allowed AMIs are discoverable in the EC2 console and by APIs that use images (for example, that describe, copy, store, or perform other actions that use images).
- Instances can only be launched using allowed AMIs.

Disabled

When Allowed AMIs is disabled:

- The `ImageCriteria` are not applied.
- No restrictions are placed on AMI discoverability or usage.

Audit mode

In audit mode:

- The `ImageCriteria` are applied, but no restrictions are placed on AMI discoverability or usage.
- In the EC2 console, for each AMI, the **Allowed image** field displays either **Yes** or **No** to indicate whether the AMI will be discoverable and available to users in the account when Allowed AMIs is enabled.
- In the command line, the response for the `describe-image` operation includes `"ImageAllowed": true` or `"ImageAllowed": false` to indicate whether the AMI will be discoverable and available to users in the account when Allowed AMIs is enabled.
- In the EC2 console, the AMI Catalog displays **Not allowed** next to AMIs that won't be discoverable or available to users in the account when Allowed AMIs is enabled.

Best practices for implementing Allowed AMIs

When implementing Allowed AMIs, consider these best practices to ensure a smooth transition and minimize potential disruptions to your AWS environment.

1. **Enable audit mode**

   Begin by enabling Allowed AMIs in audit mode. This mode allows you to see which AMIs would be affected by your criteria without actually restricting access, providing a risk-free evaluation period.

2. **Set Allowed AMIs criteria**

   Carefully establish which AMI providers align with your organization's security policies, compliance requirements, and operational needs.

   > **Note**
   >
   > We recommend specifying the `amazon` alias to allow AMIs created by AWS, ensuring that AWS managed services you use can continue to launch EC2 instances in your account.

3. **Check for impact on expected business processes**

   You can use the console or the CLI to identify any instances that were launched with AMIs that don't meet the specified criteria. This information can guide your decision to either update your launch configurations to use compliant AMIs (for example, specifying a different AMI in a launch template) or adjust your criteria to allow these AMIs.

Console: Use the ec2-instance-launched-with-allowed-ami AWS Config rule to check if running or stopped instances were launched with AMIs that meet your Allowed AMIs criteria. The rule is **NON_COMPLIANT** if an AMI doesn't meet the Allowed AMIs criteria, and **COMPLIANT** if it does. The rule only operates when the Allowed AMIs setting is set to **enabled** or **audit mode**.

CLI: Run the describe-instance-image-metadata command and filter the response to identify any instances that were launched with AMIs that don't meet the specified criteria.

For the console and CLI instructions, see Find instances launched from AMIs that aren't allowed.

4. **Enable Allowed AMIs**

   Once you've confirmed that the criteria will not adversely affect expected business processes, enable Allowed AMIs.

5. **Monitor instance launches**

   Continue to monitor instance launches from AMIs across your applications and the AWS managed services you use, such as Amazon EMR, Amazon ECR, Amazon EKS, and AWS Elastic Beanstalk. Check for any unexpected issues and make necessary adjustments to the Allowed AMIs criteria.

6. **Pilot new AMIs**

   To test third-party AMIs that do not comply with your current Allowed AMIs settings, AWS recommends the following approaches:

   - Use a separate AWS account: Create an account with no access to your business-critical resources. Ensure that the Allowed AMIs setting is not enabled in this account, or that the AMIs you want to test are explicitly allowed, so that you can test them.
   - Test in another AWS Region: Use a Region where the third-party AMIs are available, but where you have not yet enabled the Allowed AMIs settings.

   These approaches help ensure your business-critical resources remain secure while you test new AMIs.

Required IAM permissions

To use the Allowed AMIs feature, you need the following IAM permissions:

- `GetAllowedImagesSettings`

- `EnableAllowedImagesSettings`
- `DisableAllowedImagesSettings`
- `ReplaceImageCriteriaInAllowedImagesSettings`

Manage the settings for Allowed AMIs

You can manage the settings for Allowed AMIs. These settings are per Region per account.

Tasks

Enable Allowed AMIs

You can enable Allowed AMIs and specify Allowed AMIs criteria. We recommend that you begin in audit mode, which shows you which AMIs would be affected by the criteria without actually restricting access.

Console

To enable Allowed AMIs

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dashboard**.
3. Under **Account attributes** (top right), choose **Allowed AMIs**.
4. On the **Allowed AMIs** tab, choose **Manage**.
5. For **Allowed AMIs settings**, choose **Audit mode** or **Enabled**. We recommend that you begin in audit mode, test the criteria, and then return to this step to enable Allowed AMIs.
6. (Optional) For **AMI criteria**, enter the criteria in JSON format.
7. Choose **Update**.

AWS CLI

To enable Allowed AMIs

Use the enable-allowed-images-settings command.

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state enabled
```

To enable audit mode instead, specify `audit-mode` instead of `enabled`.

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state audit-mode
```

PowerShell

To enable Allowed AMIs

Use the Enable-EC2AllowedImagesSetting cmdlet.

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState enabled
```

To enable audit mode instead, specify `audit-mode` instead of `enabled`.

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState audit-mode
```

Set the Allowed AMIs criteria

After you enable Allowed AMIs, you can set or replace the Allowed AMIs criteria.

For the correct configuration and valid values, see JSON configuration and Allowed AMIs criteria.

Console

To set the Allowed AMIs criteria

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dashboard**.
3. Under **Account attributes** (top right), choose **Allowed AMIs**.
4. On the **Allowed AMIs** tab, choose **Manage**.

5. For **AMI criteria**, enter the criteria in JSON format.
6. Choose **Update**.

AWS CLI

To set the Allowed AMIs criteria

Use the replace-image-criteria-in-allowed-images-settings command as follows to allow AMIs from Amazon and the specified account.

```
aws ec2 replace-image-criteria-in-allowed-images-settings \
    --image-criteria ImageProviders=amazon,123456789012
```

PowerShell

To set the Allowed AMIs criteria

Use the Set-EC2ImageCriteriaInAllowedImagesSetting cmdlet as follows to allow AMIs from Amazon and the specified account.

```
$imageCriteria = New-Object Amazon.EC2.Model.ImageCriterionRequest
$imageCriteria.ImageProviders = @("amazon", "123456789012")
Set-EC2ImageCriteriaInAllowedImagesSetting -ImageCriterion $imageCriteria
```

Disable Allowed AMIs

You can disable Allowed AMIs as follows.

Console

To disable Allowed AMIs

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dashboard**.
3. Under **Account attributes** (top right), choose **Allowed AMIs**.
4. On the **Allowed AMIs** tab, choose **Manage**.
5. For **Allowed AMIs settings**, choose **Disabled**.
6. Choose **Update**.

AWS CLI

To disable Allowed AMIs

Use the disable-allowed-images-settings command.

```
aws ec2 disable-allowed-images-settings
```

PowerShell

To disable Allowed AMIs

Use the Disable-EC2AllowedImagesSetting cmdlet.

```
Disable-EC2AllowedImagesSetting
```

Get the Allowed AMIs criteria

You can get the current state of the Allowed AMIs setting and the Allowed AMIs criteria.

Console

To get the Allowed AMIs state and criteria

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dashboard**.
3. Under **Account attributes** (top right), choose **Allowed AMIs**.
4. On the **Allowed AMIs** tab, **Allowed AMIs settings** is set to **Enabled**, **Disabled**, or **Audit mode**.
5. If the state of Allowed AMIs is either **Enabled** or **Audit mode**, **AMI criteria**, displays the AMI criteria in JSON format.

AWS CLI

To get the Allowed AMIs state and criteria

Use the get-allowed-images-settings command.

```
aws ec2 get-allowed-images-settings
```

In the following example output, the state is `audit-mode` and the image provider list includes two providers (`amazon`, plus the specified account).

```
{
    "State": "audit-mode",
    "ImageCriteria": [
        {
            "ImageProviders": [
                "amazon",
                "123456789012"
            ]
        }
    ],
    "ManagedBy": "account"
}
```

PowerShell

To get the Allowed AMIs state and criteria

Use the Get-EC2AllowedImagesSetting cmdlet.

```
Get-EC2AllowedImagesSetting | `
    Select State, ManagedBy, @{Name='ImageProviders';
 Expression={($_.ImageCriteria.ImageProviders)}}
```

In the following example output, the state is `audit-mode` and the image provider list includes two providers (`amazon`, plus the specified account).

```
State      ManagedBy ImageProviders
-----      --------- --------------
audit-mode account   {amazon, 123456789012}
```

Find AMIs that are allowed

You can find the AMIs that are allowed or not allowed by the current Allowed AMIs criteria.

> **Note**
>
> Allowed AMIs must be in audit mode.

Console

To check whether an AMI meets the Allowed AMIs criteria

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select the AMI.
4. On the **Details** tab (if you selected the checkbox) or in the summary area (if you selected the AMI ID), find the **Allowed image** field.
   - **Yes –** The AMI meets the Allowed AMIs criteria. This AMI will be available to users in your account after you enable Allowed AMIs.
   - **No –** The AMI does not meet the Allowed AMIs criteria.
5. In the navigation pane, choose **AMI Catalog**.

   An AMI marked **Not allowed** indicates an AMI that does not meet the Allowed AMIs criteria. This AMI won't be visible or available to users in your account when Allowed AMIs is enabled.

AWS CLI

To check whether an AMI meets the Allowed AMIs criteria

Use the describe-images command.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].ImageAllowed \
    --output text
```

The following is example output.

```
True
```

To find AMIs that meet the Allowed AMIs criteria

Use the describe-images command.

```
aws ec2 describe-images \
```

```
    --filters "Name=image-allowed,Values=true" \
    --max-items 10 \
    --query Images[].ImageId
```

The following is example output.

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

PowerShell

To check whether an AMI meets the Allowed AMIs criteria

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ImageAllowed
```

The following is example output.

```
True
```

To find AMIs that meet the Allowed AMIs criteria

Use the Get-EC2Image cmdlet.

```
Get-EC2Image `
    -Filter @{Name="image-allows";Values="true"} `
    -MaxResult 10 | `
    Select ImageId
```

The following is example output.

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
```

```
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

Find instances launched from AMIs that aren't allowed

You can identify the instances that were launched using an AMI that does not meet the Allowed AMIs criteria.

Console

To check whether an instance was launched using an AMI that isn't allowed

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 **Allowed image**.

   - **Yes** – The AMI meets the Allowed AMIs criteria.
   - **No** – The AMI does not meet the Allowed AMIs criteria.

AWS CLI

To find instances launched using AMIs that aren't allowed

Use the describe-instance-image-metadata command with the `image-allowed` filter.

```
aws ec2 describe-instance-image-metadata \
    --filters "Name=image-allowed,Values=false" \
    --query "InstanceImageMetadata[*].[InstanceId,ImageMetadata.ImageId]" \
    --output table
```

The following is example output.

```
--------------------------------------------------
```

```
|          DescribeInstanceImageMetadata         |
+----------------------+------------------------+
|  i-08fd74f3f1595fdbd |  ami-09245d5773578a1d6  |
|  i-0b1bf24fd4f297ab9 |  ami-07cccf2bd80ed467f  |
|  i-026a2eb590b4f7234 |  ami-0c0ec0a3a3a4c34c0  |
|  i-006a6a4e8870c828f |  ami-0a70b9d193ae8a799  |
|  i-0781e91cfeca3179d |  ami-00c257e12d6828491  |
|  i-02b631e2a6ae7c2d9 |  ami-0bfddf4206f1fa7b9  |
+----------------------+------------------------+
```

PowerShell

To find instances launched using AMIs that aren't allowed

Use the Get-EC2InstanceImageMetadata cmdlet.

```
Get-EC2InstanceImageMetadata `
    -Filter @{Name="image-allowed";Values="false"} | `
    Select InstanceId, @{Name='ImageId'; Expression={($_.ImageMetadata.ImageId)}}
```

The following is example output.

```
InstanceId          ImageId
----------          -------
i-08fd74f3f1595fdbd ami-09245d5773578a1d6
i-0b1bf24fd4f297ab9 ami-07cccf2bd80ed467f
i-026a2eb590b4f7234 ami-0c0ec0a3a3a4c34c0
i-006a6a4e8870c828f ami-0a70b9d193ae8a799
i-0781e91cfeca3179d ami-00c257e12d6828491
i-02b631e2a6ae7c2d9 ami-0bfddf4206f1fa7b9
```

AWS Config

You can add the **ec2-instance-launched-with-allowed-ami** AWS Config rule, configure it for your requirements, and then use it to evaluate your instances.

For more information, see Adding AWS Config rules and ec2-instance-launched-with-allowed-ami in the *AWS Config Developer Guide*.

Make your AMI publicly available for use in Amazon EC2

You can make your AMI publicly available by sharing it with all AWS accounts.

If you want to prevent the public sharing of your AMIs, you can enable *block public access for AMIs*. This blocks any attempts to make an AMI public, helping to prevent unauthorized access and potential misuse of AMI data. Note that enabling block public access does not affect your AMIs that are already publicly available; they remain publicly available. For more information, see Understand block public access for AMIs.

To allow only specific accounts to use your AMI to launch instances, see Share an AMI with specific AWS accounts.

Contents

Considerations

Consider the following before making an AMI public.

- **Ownership** – To make an AMI public, your AWS account must own the AMI.
- **Region** – AMIs are a Regional resource. When you share an AMI, it is available only in the Region from which you shared it. 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 Amazon EC2 AMI.
- **Block public access** – To publicly share an AMI, block public access for AMIs must be disabled in each Region in which the AMI will be publicly shared. After you've publicly shared the AMI, you can re-enable block public access for AMIs to prevent further public sharing of your AMIs.
- **Some AMIs can't be made public** – If your AMI includes one of the following components, you can't make it public (but you can share the AMI with specific AWS accounts):
  - Encrypted volumes
  - Snapshots of encrypted volumes
  - Product codes
- **Avoid exposing sensitive data** – To avoid exposing sensitive data when you share an AMI, read the security considerations in Recommendations for creating shared Linux AMIs and follow the recommended actions.
- **Usage** – When you share an AMI, users can only launch instances from the AMI. They can’t delete, share, or modify it. However, after they have launched an instance using your AMI, they can then create an AMI from the instance they launched.

- **Automatic deprecation –** By default, the deprecation date of all public AMIs is set to two years from the AMI creation date. You can set the deprecation date to earlier than two years. To cancel the deprecation date, or to move the deprecation to a later date, you must make the AMI private by only sharing it with specific AWS accounts.
- **Remove obsolete AMIs –** After a public AMI reaches its deprecation date, if no new instances were launched from the AMI for six or more months, AWS eventually removes the public sharing property so that obsolete AMIs don't appear in the public AMI lists.
- **Billing –** You are not billed when your AMI is used by other AWS accounts to launch instances. The accounts that launch instances using the AMI are billed for the instances that they launch.

Share an AMI with all AWS accounts (share publicly)

After you make an AMI public, it is available in **Community AMIs** in the console, which you can access from the **AMI Catalog** in the left navigator in the EC2 console or when launching an instance using the console. Note that it can take a short while for an AMI to appear in **Community AMIs** after you make it public.

Console

To make an AMI public

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**, **Edit AMI permissions**.
4. Under **AMI availability**, choose **Public**.
5. Choose **Save changes**.

AWS CLI

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts), or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

1. Use the modify-image-attribute command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

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

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

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

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

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

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.

```
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.

```
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.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -UserGroup all
```

Understand block public access for AMIs

To prevent the public sharing of your AMIs, you can enable *block public access for AMIs* at the account level.

When block public access is enabled, any attempt to make an AMI public is automatically blocked. However, if you already have public AMIs, they remain publicly available.

To publicly share AMIs, you must disable block public access. When you’re done sharing, it's best practice to re-enable block public access to prevent any unintended public sharing of your AMIs.

Note

This setting is configured at the account level, either directly in the account or by using a declarative policy. It must be configured in each AWS Region where you want to prevent the public sharing of your AMIs. Using a declarative policy allows you to apply the setting across multiple Regions simultaneously, as well as across multiple accounts simultaneously. When a declarative policy is in use, you can't modify the setting directly within an account. This topic describes how to configure the setting directly within an account. For information about using declarative policies, see Declarative policies in the *AWS Organizations User Guide.*

You can restrict IAM permissions to an administrator user so that only they can enable or disable block public access for AMIs.

Topics

- Default settings
- Manage the block public access setting for AMIs

Default settings

The **Block public access for AMIs** setting is either enabled or disabled by default depending on whether your account is new or existing, and whether you have public AMIs. The following table lists the default settings:

AWS account	Block public access for AMIs default setting
New accounts	Enabled
Existing accounts with no public AMIs [1]	Enabled
Existing accounts with one or more public AMIs	Disabled

[1] If your account had one or more public AMIs on or after July 15, 2023, **Block public access for AMIs** is disabled by default for your account, even if you subsequently made all the AMIs private.

Manage the block public access setting for AMIs

You can manage the block public access setting for your AMIs to control whether they can be publicly shared. You can enable, disable, or view the current block public access state for your AMIs using the Amazon EC2 console or the AWS CLI.

View the block public access state for AMIs

To see whether the public sharing of your AMIs is blocked in your account, you can view the state for block public access for AMIs. You must view the state in each AWS Region in which you want to see whether the public sharing of your AMIs is blocked.

Required permissions

To get the current block public access setting for AMIs, you must have the `GetImageBlockPublicAccessState` IAM permission.

Console

To view the block public access state for AMIs in the specified Region

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar (at the top of the screen), select the Region in which to view the block public access state for AMIs.
3. If the dashboard is not displayed, in the navigation pane, choose **EC2 Dashboard**.
4. Under **Account attributes**, choose **Data protection and security**.
5. Under **Block public access for AMIs**, check the **Public access** field. The value is either **New public sharing blocked** or **New public sharing allowed**.

AWS CLI

To get the block public access state for AMIs

Use the get-image-block-public-access-state command. The value is either `block-new-sharing` or `unblocked`.

Example: For a specific Region

```
aws ec2 get-image-block-public-access-state --region us-east-1
```

The `ManagedBy` field indicates the entity that configured the setting. In this example, `account` indicates that the setting was configured directly in the account. A value of `declarative-policy` would mean the setting was configured by a declarative policy. For more information, see Declarative policies in the *AWS Organizations User Guide*.

```
{
    "ImageBlockPublicAccessState": "block-new-sharing",
    "ManagedBy": "account"
}
```

Example: For all Regions in your account

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-image-block-public-access-state \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

The following is example output.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       unblocked
eu-west-3        block-new-sharing
...
```

PowerShell

To get the block public access state for AMIs

Use the Get-EC2ImageBlockPublicAccessState cmdlet. The value is either `block-new-sharing` or `unblocked`.

Example: For a specific Region

```
Get-EC2ImageBlockPublicAccessState -Region us-east-1
```

The following is example output.

```
block-new-sharing
```

Example: For all Regions in your account

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
        [PSCustomObject]@{
            Region   = $_
            PublicAccessState = (Get-EC2ImageBlockPublicAccessState -Region $_)
        }
} | `
Format-Table -AutoSize
```

The following is example output.

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

Enable block public access for AMIs

To prevent the public sharing of your AMIs, enable block public access for AMIs at the account level. You must enable block public access for AMIs in each AWS Region in which you want to prevent the public sharing of your AMIs. If you already have public AMIs, they will remain publicly available.

Required permissions

To enable the block public access setting for AMIs, you must have the `EnableImageBlockPublicAccess` IAM permission.

Considerations

- It can take up to 10 minutes to configure this setting. During this time, if you describe the public access state, the response is `unblocked`. When the configuration is completed, the response is `block-new-sharing`.

Console

To enable block public access for AMIs in the specified Region

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar (at the top of the screen), select the Region in which to enable block public access for AMIs.
3. If the dashboard is not displayed, in the navigation pane, choose **EC2 Dashboard**.
4. Under **Account attributes**, choose **Data protection and security**.
5. Under **Block public access for AMIs**, choose **Manage**.
6. Select the **Block new public sharing** checkbox, and then choose **Update**.

AWS CLI

To enable block public access for AMIs

Use the enable-image-block-public-access command.

Example: For a specific Region

```
aws ec2 enable-image-block-public-access \
--region us-east-1 \
--image-block-public-access-state block-new-sharing
```

The following is example output.

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

Example: For all Regions in your account

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
```

```
    );
    do (output=$(
        aws ec2 enable-image-block-public-access \
            --region $region \
            --image-block-public-access-state block-new-sharing \
            --output text)
        echo -e "$region \t $output"
    );
done
```

The following is example output.

```
Region            Public Access State
--------------    ----------------------
ap-south-1        block-new-sharing
eu-north-1        block-new-sharing
eu-west-3         block-new-sharing
...
```

PowerShell

To enable block public access for AMIs

Use the Enable-EC2ImageBlockPublicAccess command.

Example: For a specific Region

```
Enable-EC2ImageBlockPublicAccess `
    -Region us-east-1 `
    -ImageBlockPublicAccessState block-new-sharing
```

The following is example output.

```
Value
-----
block-new-sharing
```

Example: For all Regions in your account

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
```

```
        Region            = $_
        PublicAccessState = (
        Enable-EC2ImageBlockPublicAccess `
         -Region $_ `
         -ImageBlockPublicAccessState block-new-sharing)
    }
} | `
Format-Table -AutoSize
```

The following is example output.

```
Region          PublicAccessState
------          -----------------
ap-south-1      block-new-sharing
eu-north-1      block-new-sharing
eu-west-3       block-new-sharing
...
```

Disable block public access for AMIs

To allow the users in your account to publicly share your AMIs, disable block public access at the account level. You must disable block public access for AMIs in each AWS Region in which you want to allow the public sharing of your AMIs.

Required permissions

To disable the block public access setting for AMIs, you must have the `DisableImageBlockPublicAccess` IAM permission.

Considerations

- It can take up to 10 minutes to configure this setting. During this time, if you describe the public access state, the response is `block-new-sharing`. When the configuration is completed, the response is `unblocked`.

Console

To disable block public access for AMIs in the specified Region

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

2. From the navigation bar (at the top of the screen), select the Region in which to disable block public access for AMIs.
3. If the dashboard is not displayed, in the navigation pane, choose **EC2 Dashboard**.
4. Under **Account attributes**, choose **Data protection and security**.
5. Under **Block public access for AMIs**, choose **Manage**.
6. Clear the **Block new public sharing** checkbox, and then choose **Update**.
7. Enter **`confirm`** when prompted for confirmation, and then choose **Allow public sharing**.

AWS CLI

To disable block public access for AMIs

Use the disable-image-block-public-access command.

Example: For a specific Region

```
aws ec2 disable-image-block-public-access --region us-east-1
```

The following is example output.

```
{
   "ImageBlockPublicAccessState": "unblocked"
}
```

Example: For all Regions in your account

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 disable-image-block-public-access \
            --region $region \
            --output text)
```

```
        echo -e "$region \t $output"
    );
done
```

The following is example output.

```
Region            Public Access State
--------------    ----------------------
ap-south-1        unblocked
eu-north-1        unblocked
eu-west-3         unblocked
...
```

PowerShell

To disable block public access for AMIs

Use the Disable-EC2ImageBlockPublicAccess cmdlet.

Example: For a specific Region

```
Disable-EC2ImageBlockPublicAccess -Region us-east-1
```

The following is example output.

```
Value
-----
unblocked
```

Example: For all Regions in your account

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Disable-EC2ImageBlockPublicAccess -Region $_)
    }
} | `
Format-Table -AutoSize
```

The following is example output.

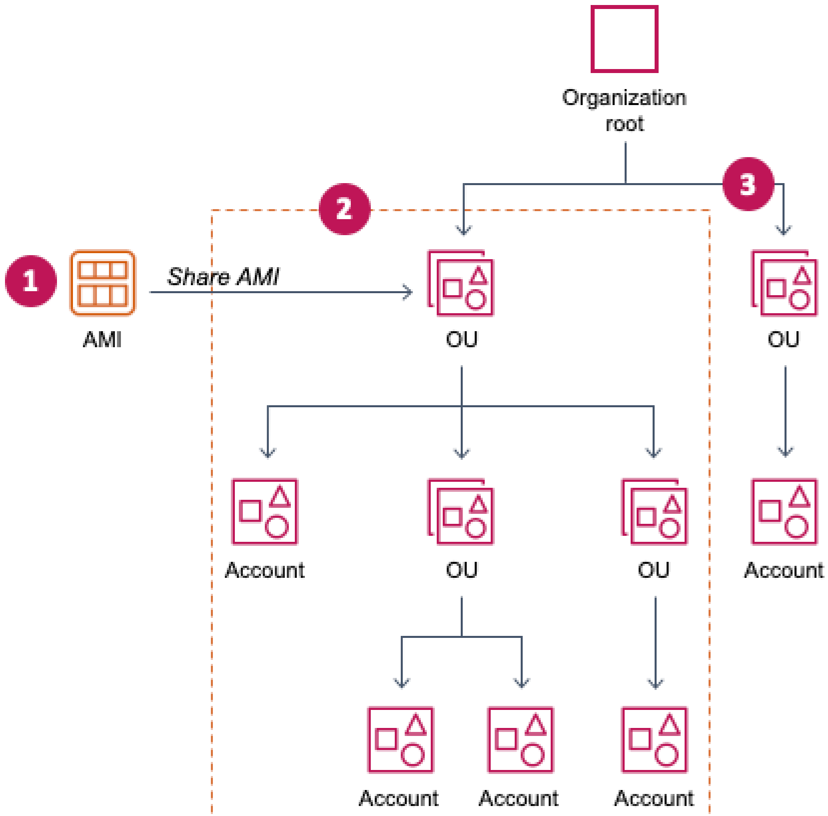
```
Region          PublicAccessState
------          -----------------
ap-south-1      unblocked
eu-north-1      unblocked
eu-west-3       unblocked
...
```

Share an AMI with organizations and organizational units

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage. You can share an AMI with an organization or an organizational unit (OU) that you have created, in addition to sharing it with specific accounts.

An organization is an entity that you create to consolidate and centrally manage your AWS accounts. You can organize the accounts in a hierarchical, tree-like structure, with a root at the top and organizational units nested under the organization root. Each account can be added directly to the root, or placed in one of the OUs in the hierarchy. For more information, see AWS Organizations terminology and concepts in the *AWS Organizations User Guide*.

When you share an AMI with an organization or an OU, all of the children accounts gain access to the AMI. For example, in the following diagram, the AMI is shared with a top-level OU (indicated by the arrow at the number **1**). All of the OUs and accounts that are nested underneath that top-level OU (indicated by the dotted line at number **2**) also have access to the AMI. The accounts in the organization and OU outside the dotted line (indicated by the number **3**) do not have access to the AMI because they are not children of the OU that the AMI is shared with.



Topics

- Considerations
- Get the ARN of an organization or organizational unit
- Allow organizations and OUs to use a KMS key
- Manage AMI sharing with an organization or OU

Considerations

Consider the following when sharing AMIs with specific organizations or organizational units.

- **Ownership** – To share an AMI, your AWS account must own the AMI.
- **Sharing limits** – The AMI owner can share an AMI with any organization or OU, including organizations and OUs that they're not a member of.

  For the maximum number of entities to which an AMI can be shared within a Region, see the Amazon EC2 service quotas.

- **Tags** – You can't share user-defined tags (tags that you attach to an AMI). When you share an AMI, your user-defined tags are not available to any AWS account in an organization or OU with which the AMI is shared.
- **ARN format** – When you specify an organization or OU in a command, make sure to use the correct ARN format. You'll get an error if you specify only the ID, for example, if you specify only `o-123example` or `ou-1234-5example`.

  Correct ARN formats:

  - Organization ARN: `arn:aws:organizations::111122223333:organization/organization-id`
  - OU ARN: `arn:aws:organizations::111122223333:ou/organization-id/ou-id`

  Where:

  - *`111122223333`* is an example of the 12-digit account ID for the management account. If you don't know the management account number, you can describe the organization or the organizational unit to get the ARN, which includes the management account number. For more information, see Get the ARN of an organization or organizational unit.
  - *`organization-id`* is the organization ID, for example, `o-123example`.
  - *`ou-id`* is the organizational unit ID, for example, `ou-1234-5example`.

  For more information about the format of ARNs, see Amazon Resource Names (ARNs) in the *IAM User Guide*.
- **Encryption and keys** – You can share AMIs that are backed by unencrypted and encrypted snapshots.
  - The encrypted snapshots must be encrypted with a customer managed key. You can't share AMIs that are backed by snapshots that are encrypted with the default AWS managed key.
  - If you share an AMI that is backed by encrypted snapshots, you must allow the organizations or OUs to use the customer managed keys that were used to encrypt the snapshots. For more information, see Allow organizations and OUs to use a KMS key.
- **Region** – AMIs are a Regional resource. When you share an AMI, it is available only in the Region from which you shared it. 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 Amazon EC2 AMI.
- **Usage** – When you share an AMI, users can only launch instances from the AMI. They can't delete, share, or modify it. However, after they have launched an instance using your AMI, they can then create an AMI from the instance they launched.

- **Billing** – You are not billed when your AMI is used by other AWS accounts to launch instances. The accounts that launch instances using the AMI are billed for the instances that they launch.

Get the ARN of an organization or organizational unit

The organization and the organizational unit ARNs contain the 12-digit management account number. If you don't know the management account number, you can describe the organization and the organizational unit to get the ARN for each. In the following examples, `123456789012` is the account ID of the management account.

Required permissions

Before you can get the ARNs, you must have the permission to describe organizations and organizational units. The following policy provides the necessary permission.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "organizations:Describe*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS CLI

To get the ARN of an organization

Use the describe-organization command. Add the `--query` option to return only the organization ARN.

```
aws organizations describe-organization --query 'Organization.Arn'
```

The following is example output.

```
"arn:aws:organizations::123456789012:organization/o-1234567abc"
```

To get the ARN of an organizational unit

Use the describe-organizational-unit command. Use the `--query` parameter to return only the organizational unit ARN.

```
aws organizations describe-organizational-unit \
    --organizational-unit-id ou-a123-b4567890 \
    --query 'OrganizationalUnit.Arn'
```

The following is example output.

```
"arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890"
```

PowerShell

To get the ARN of an organization

Use the Get-ORGOrganization cmdlet.

```
(Get-ORGOrganization).Arn
```

The following is example output.

```
arn:aws:organizations::123456789012:organization/o-1234567abc
```

To get the ARN of an organizational unit

Use the Get-ORGOrganizationalUnit cmdlet.

```
(Get-ORGOrganizationalUnit -OrganizationalUnitId "ou-a123-b4567890").Arn
```

The following is example output.

```
arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890
```

Allow organizations and OUs to use a KMS key

If you share an AMI that is backed by encrypted snapshots, you must also allow the organizations or organizational units (OUs) to use the KMS keys that were used to encrypt the snapshots.

> **Note**
>
> The encrypted snapshots must be encrypted with a *customer managed* key. You can't share AMIs that are backed by snapshots that are encrypted with the default AWS managed key.

To control access to the KMS key, in the key policy you can use the `aws:PrincipalOrgID` and `aws:PrincipalOrgPaths` condition keys to allow only specific principals permission to the specified actions. A principal can be a user, IAM role, federated user, or AWS account root user.

The condition keys are used as follows:

- `aws:PrincipalOrgID` – Allows any principal belonging to the organization represented by the specified ID.
- `aws:PrincipalOrgPaths` – Allows any principal belonging to the OUs represented by the specified paths.

To give an organization (including the OUs and accounts that belong to it) permission to use a KMS key, add the following statement to the key policy.

```
{
    "Sid": "Allow access for organization root",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
```

```
    "Condition": {
        "StringEquals": {
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

To give specific OUs (and the accounts that belong to it) permission to use a KMS key, you can use a policy similar to the following example.

```
{
        "Sid": "Allow access for specific OUs and their descendants",
        "Effect": "Allow",
        "Principal": "*",
        "Action": [
            "kms:Describe*",
            "kms:List*",
            "kms:Get*",
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:CreateGrant"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": "o-123example"
            },
            "ForAnyValue:StringLike": {
                "aws:PrincipalOrgPaths": [
                    "o-123example/r-ab12/ou-ab12-33333333/*",
                    "o-123example/r-ab12/ou-ab12-22222222/*"
                ]
            }
        }
}
```

For more example condition statements, see aws:PrincipalOrgID and aws:PrincipalOrgPaths in the *IAM User Guide*.

For information about cross-account access, see Allowing users in other accounts to use a KMS key in the *AWS Key Management Service Developer Guide*.

Manage AMI sharing with an organization or OU

You can manage AMI sharing with organizations and organization units (OU) to control whether they can launch Amazon EC2 instances.

View the organizations and OUs with which an AMI is shared

You can find the organizations and OUs with which you've shared your AMI.

Console

To check with which organizations and OUs you've shared your AMI

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, choose the **Permissions** tab, and scroll down to **Shared organizations/OUs**.

   To find AMIs that are shared with you, see Find shared AMIs to use for Amazon EC2 instances.

AWS CLI

To check with which organizations and OUs you've shared your AMI

Use the describe-image-attribute command with the `launchPermission` attribute.

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

The following is an example response.

```
{
    "ImageId": "ami-0abcdef1234567890",
    "LaunchPermissions": [
        {
            "OrganizationalUnitArn": "arn:aws:organizations::111122223333:ou/
o-123example/ou-1234-5example"
        }
```

```
    ]
}
```

PowerShell

To check with which organizations and OUs you've shared your AMI

Use the Get-EC2ImageAttribute cmdlet.

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

Share an AMI with an organization or OU

You can share an AMI with an organization or OU.

> **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, and the system automatically provides the instance with access to the referenced EBS snapshots for the launch. However, you do need to share the KMS keys used to encrypt snapshots that the AMI references. For more information, see Allow organizations and OUs to use a KMS key.

Console

To share an AMI with an organization or an OU

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**, **Edit AMI permissions**.
4. Under **AMI availability**, choose **Private**.
5. Next to **Shared organizations/OUs**, choose **Add organization/OU ARN**.
6. For **Organization/OU ARN**, enter the organization ARN or OU ARN with which you want to share the AMI, and then choose **Share AMI**. Note that you must specify the full ARN, not just the ID.

To share this AMI with multiple organizations or OUs, repeat this step until you have added all of the required organizations or OUs.

7. Choose **Save changes** when you're done.
8. (Optional) To view the organizations or OUs with which you have shared the AMI, select the AMI in the list, choose the **Permissions** tab, and scroll down to **Shared organizations/OUs**. To find AMIs that are shared with you, see Find shared AMIs to use for Amazon EC2 instances.

AWS CLI

To share an AMI with an organization

Use the modify-image-attribute command to grant launch permissions for the specified AMI to the specified organization.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

To share an AMI with an OU

The modify-image-attribute command grants launch permissions for the specified AMI to the specified OU. Note that you must specify the full ARN, not just the ID.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationalUnitArn=arn:aws:organizations::123456789012:ou/o-123example/
ou-1234-5example}]"
```

PowerShell

Use the Edit-EC2ImageAttribute command (Tools for Windows PowerShell) to share an AMI as shown in the following examples.

To share an AMI with an organization or an OU

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

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

To stop sharing an AMI with an organization or OU

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

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

To stop sharing an AMI with all organizations, OUs, and AWS accounts

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.

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

Stop sharing an AMI with an organization or OU

You can stop sharing an AMI with an organization or OU.

> **Note**
>
> You can't stop sharing an AMI with a specific account if it's in an organization or OU with which an AMI is shared. If you try to stop sharing the AMI by removing launch permissions

for the account, Amazon EC2 returns a success message. However, the AMI continues to be shared with the account.

Console

To stop sharing an AMI with an organization or OU

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**, **Edit AMI permissions**.
4. Under **Shared organizations/OUs**, select the organizations or OUs with which you want to stop sharing the AMI, and then choose **Remove selected**.
5. Choose **Save changes** when you're done.
6. (Optional) To confirm that you have stopped sharing the AMI with the organizations or OUs, select the AMI in the list, choose the **Permissions** tab, and scroll down to **Shared organizations/OUs**.

AWS CLI

To stop sharing an AMI with an organization or OU

Use the modify-image-attribute command. This example removes launch permissions for the specified AMI from the specified organization.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Remove=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

To stop sharing an AMI with all organizations, OUs, and AWS accounts

Use the reset-image-attribute command. This example removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
aws ec2 reset-image-attribute \
```

```
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

To stop sharing an AMI with an organization or OU

Use the Edit-EC2ImageAttribute cmdlet. This example removes launch permissions for the specified AMI from the specified organization.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

To stop sharing an AMI with all organizations, OUs, and AWS accounts

Use the Reset-EC2ImageAttribute cmdlet. This example 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.

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute LaunchPermission
```

Share an AMI with specific AWS accounts

You can share an AMI with specific AWS accounts without making the AMI public. All you need are the AWS account IDs.

An AWS account ID is a 12-digit number, such as `012345678901`, that uniquely identifies an AWS account. For more information, see View AWS account identifiers in the *AWS Account Management Reference Guide*.

Considerations

Consider the following when sharing AMIs with specific AWS accounts.

- **Ownership** – To share an AMI, your AWS account must own the AMI.

- **Sharing limits** – For the maximum number of entities to which an AMI can be shared within a Region, see the Amazon EC2 service quotas.
- **Tags** – You can't share user-defined tags (tags that you attach to an AMI). When you share an AMI, your user-defined tags are not available to any AWS account that the AMI is shared with.
- **Snapshots** – You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. You can share only the AMI itself; the system provides the instance access to the referenced EBS snapshots for the launch. However, you must share any KMS keys used to encrypt snapshots that an AMI references. For more information, see Share an Amazon EBS snapshot in the *Amazon EBS User Guide*.
- **Encryption and keys** – You can share AMIs that are backed by unencrypted and encrypted snapshots.
  - The encrypted snapshots must be encrypted with a KMS key. You can’t share AMIs that are backed by snapshots that are encrypted with the default AWS managed key.
  - If you share an AMI that is backed by encrypted snapshots, you must allow the AWS accounts to use the KMS keys that were used to encrypt the snapshots. For more information, see Allow organizations and OUs to use a KMS key. To set up the key policy that you need to launch Auto Scaling instances when you use a customer managed key for encryption, see Required AWS KMS key policy for use with encrypted volumes in the *Amazon EC2 Auto Scaling User Guide*.
- **Region** – AMIs are a Regional resource. When you share an AMI, it is only 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 Amazon EC2 AMI.
- **Usage** – When you share an AMI, users can only launch instances from the AMI. They can’t delete, share, or modify it. However, after they have launched an instance using your AMI, they can then create an AMI from their instance.
- **Copying shared AMIs** – If users in another account want to copy a shared AMI, you must grant them read permissions for the storage that backs the AMI. For more information, see Cross-account copying.
- **Billing** – You are not billed when your AMI is used by other AWS accounts to launch instances. The accounts that launch instances using the AMI are billed for the instances that they launch.

Console

To grant explicit launch permissions

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**, **Edit AMI permissions**.
4. Choose **Private**.
5. Under **Shared accounts**, choose **Add account ID**.
6. For **AWS account ID**, enter the AWS account ID with which you want to share the AMI, and then choose **Share AMI**.

   To share this AMI with multiple accounts, repeat Steps 5 and 6 until you have added all the required account IDs.
7. Choose **Save changes** when you are done.
8. (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 to use for Amazon EC2 instances.

AWS CLI

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

To grant explicit launch permissions

The following example 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}]"
```

To remove launch permissions for an account

The following example removes launch permissions for the specified AMI from the specified AWS account.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

To remove all launch permissions

The following example removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

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

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 example grants launch permissions for the specified AMI to the specified AWS account.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserId "123456789012"
```

To remove launch permissions for an account

The following example removes launch permissions for the specified AMI from the specified AWS account.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission -OperationType remove `
    -UserId "123456789012"
```

To remove all launch permissions

The following exaple 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.

```
Reset-EC2ImageAttribute `
```

```
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

Cancel having an AMI shared with your AWS account

An Amazon Machine Image (AMI) can be shared with specific AWS accounts by adding the accounts to the AMI's launch permissions. If an AMI has been shared with your AWS account and you no longer want it shared with your account, you can remove your account from the AMI's launch permissions. You do this by running the `cancel-image-launch-permission` AWS CLI command. When running this command, your AWS account is removed from the launch permissions for the specified AMI. To find the AMIs that are shared with your AWS account, see Find shared AMIs to use for Amazon EC2 instances.

You might cancel having an AMI shared with your account, for example, to reduce the likelihood of launching an instance with an unused or deprecated AMI that was shared with you. When you cancel having an AMI shared with your account, it no longer appears in any AMI lists in the EC2 console or in the output for describe-images.

Topics

Limitations

- You can remove your account from the launch permissions of an AMI that is shared with your AWS account only. You can't use `cancel-image-launch-permission` to remove your account from the launch permissions of an AMI shared with an organization or organizational unit (OU) or to remove access to public AMIs.
- You can't permanently remove your account from the launch permissions of an AMI. An AMI owner can share an AMI with your account again.
- AMIs are a Regional resource. When running `cancel-image-launch-permission`, you must specify the Region in which the AMI is located. Either specify the Region in the command, or use the AWS_DEFAULT_REGION environment variable.
- Only the AWS CLI and SDKs support removing your account from the launch permissions of an AMI. The EC2 console does not currently support this action.

Cancel having an AMI shared with your account

> **Note**
>
> After you cancel having an AMI shared with your account, you can't undo it. To regain access to the AMI, the AMI owner must share it with your account.

AWS CLI

To cancel having an AMI shared with your account

Use the cancel-image-launch-permission command.

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0abcdef1234567890 \
    --region us-east-1
```

PowerShell

To cancel having an AMI shared with your account

Use the Stop-EC2ImageLaunchPermission cmdlet.

```
Stop-EC2ImageLaunchPermission `
    -ImageId ami-0abcdef1234567890 `
    -Region us-east-1
```

Recommendations for creating shared Linux AMIs

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

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

Contents

If you are building AMIs for AWS Marketplace, see Best practices for building AMIs in the *AWS Marketplace Seller Guide* for guidelines, policies, and best practices.

For additional information about sharing AMIs safely, see the following articles:

- How To Share and Use Public AMIs in A Secure Manner
- Public AMI Publishing: Hardening and Clean-up Requirements

Disable password-based remote logins for the root user

Using a fixed root password for a public AMI is a security risk that can quickly become known. Even relying on users to change the password after the first login opens a small window of opportunity for potential abuse.

To solve this problem, disable password-based remote logins for the root user.

To disable password-based remote logins for the root user

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

   ```
   #PermitRootLogin yes
   ```

2. Change the line to:

   ```
   PermitRootLogin without-password
   ```

   The location of this configuration file might differ for your distribution, or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Disable local root access

When you work with shared AMIs, a best practice is to disable direct root logins. To do this, log into your running instance and issue the following command:

```
[ec2-user ~]$ sudo passwd -l root
```

> **Note**
>
> This command does not impact the use of `sudo`.

Remove SSH host key pairs

If you plan to share an AMI derived from a public AMI, remove the existing SSH host key pairs located in `/etc/ssh`. This forces SSH to generate new unique SSH key pairs when someone launches an instance using your AMI, improving security and reducing the likelihood of "man-in-the-middle" attacks.

Remove all of the following key files that are present on your system.

- ssh_host_dsa_key
- ssh_host_dsa_key.pub
- ssh_host_key
- ssh_host_key.pub
- ssh_host_rsa_key
- ssh_host_rsa_key.pub
- ssh_host_ecdsa_key
- ssh_host_ecdsa_key.pub
- ssh_host_ed25519_key
- ssh_host_ed25519_key.pub

You can securely remove all of these files with the following command.

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

> **Warning**
>
> Secure deletion utilities such as **shred** might not remove all copies of a file from your storage media. Hidden copies of files may be created by journalling file systems (including Amazon Linux default ext4), snapshots, backups, RAID, and temporary caching. For more information, see the [shred documentation](#).

> **Important**
>
> If you forget to remove the existing SSH host key pairs from your public AMI, our routine auditing process notifies you and all customers running instances of your AMI of the potential security risk. After a short grace period, we mark the AMI private.

Install public key credentials

After configuring the AMI to prevent logging in using a password, you must make sure users can log in using another mechanism.

Amazon EC2 allows users to specify a public-private key pair name when launching an instance. When a valid key pair name is provided to the `RunInstances` API call (or through the command line API tools), the public key (the portion of the key pair that Amazon EC2 retains on the server after a call to `CreateKeyPair` or `ImportKeyPair`) is made available to the instance through an HTTP query against the instance metadata.

To log in through SSH, your AMI must retrieve the key value at boot and append it to `/root/.ssh/authorized_keys` (or the equivalent for any other user account on the AMI). Users can launch instances of your AMI with a key pair and log in without requiring a root password.

Many distributions, including Amazon Linux and Ubuntu, use the `cloud-init` package to inject public key credentials for a configured user. If your distribution does not support `cloud-init`, you can add the following code to a system start-up script (such as `/etc/rc.local`) to pull in the public key you specified at launch for the root user.

Note

In the following example, the IP address http://169.254.169.254/ is a link-local address and is valid only from the instance.

IMDSv2

```
if [ ! -d /root/.ssh ] ; then
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
        cat /tmp/my-key >> /root/.ssh/authorized_keys
        chmod 700 /root/.ssh/authorized_keys
        rm /tmp/my-key
fi
```

IMDSv1

```
if [ ! -d /root/.ssh ] ; then
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
        cat /tmp/my-key >> /root/.ssh/authorized_keys
        chmod 700 /root/.ssh/authorized_keys
        rm /tmp/my-key
fi
```

This can be applied to any user; you do not need to restrict it to the `root` user.

> **Note**
>
> Rebundling an instance based on this AMI includes the key with which it was launched. To prevent the key's inclusion, you must clear out (or delete) the `authorized_keys` file or exclude this file from rebundling.

Disable sshd DNS checks (optional)

Disabling sshd DNS checks slightly weakens your sshd security. However, if DNS resolution fails, SSH logins still work. If you do not disable sshd checks, DNS resolution failures prevent all logins.

To disable sshd DNS checks

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

    ```
    #UseDNS yes
    ```

2. Change the line to:

    ```
    UseDNS no
    ```

> **Note**
>
> The location of this configuration file can differ for your distribution or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Remove sensitive data

We recommend against storing sensitive data or software on any AMI that you share. Users who launch a shared AMI might be able to rebundle it and register it as their own. Follow these guidelines to help you to avoid some easily overlooked security risks:

- We recommend using the `--exclude` *`directory`* option on `ec2-bundle-vol` to skip any directories and subdirectories that contain secret information that you would not like to include in your bundle. In particular, exclude all user-owned SSH public/private key pairs and SSH `authorized_keys` files when bundling the image. The Amazon public AMIs store these in

/root/.ssh for the root user, and /home/*user_name*/.ssh/ for regular users. For more information, see ec2-bundle-vol.

- Always delete the shell history before bundling. If you attempt more than one bundle upload in the same AMI, the shell history contains your access key. The following example should be the last command you run before bundling from within the instance.

```
[ec2-user ~]$ shred -u ~/.*history
```

> **Warning**
>
> The limitations of **shred** described in the warning above apply here as well.
> Be aware that bash writes the history of the current session to the disk on exit. If you log out of your instance after deleting ~/.bash_history, and then log back in, you will find that ~/.bash_history has been re-created and contains all of the commands you ran during your previous session.
> Other programs besides bash also write histories to disk, Use caution and remove or exclude unnecessary dot-files and dot-directories.

- Bundling a running instance requires your private key and X.509 certificate. Put these and other credentials in a location that is not bundled (such as the instance store).

Monitor AMI events using Amazon EventBridge

When the state of an Amazon Machine Image (AMI) changes, Amazon EC2 generates an event that is sent to Amazon EventBridge (formerly known as Amazon CloudWatch Events). The events are sent to the default EventBridge event bus in JSON format. You can use Amazon EventBridge to detect and react to these events. You do this by creating rules in EventBridge that trigger an action in response to an event. For example, you can create an EventBridge rule that detects when the AMI creation process has completed and then invokes an Amazon SNS topic to send an email notification to you.

Amazon EC2 generates an `EC2 AMI State Change` event when an AMI enters any of the following states:

- `available`
- `failed`

- deregistered
- disabled

Events are generated on a best effort basis.

The following table lists the AMI operations and the states that an AMI can enter. In the table, **Yes** indicates the states that the AMI can enter when the corresponding operation runs.

AMI operations	available	failed	deregistered	disabled
CopyImage	Yes	Yes		
CreateImage	Yes	Yes		
CreateRes toreImageTask	Yes	Yes		
DeregisterImage			Yes	
DisableImage				Yes
EnableImage	Yes			
RegisterImage	Yes	Yes		

EC2 AMI State Change events

- Event details
- available events
- failed events
- deregistered events
- disabled events

Event details

You can use the following fields in the event to create rules that trigger an action:

`"source": "aws.ec2"`

Identifies that the event is from Amazon EC2.

`"detail-type": "EC2 AMI State Change"`

Identifies the event name.

`"detail": { "ImageId": "ami-0abcdef1234567890", "State": "available", }`

Provides the AMI ID and the state of the AMI (`available`, `failed`, `deregistered`, or `disabled`).

For more information, see the following in the *Amazon EventBridge User Guide*:

- Amazon EventBridge events
- Amazon EventBridge event patterns
- Amazon EventBridge rules

For a tutorial about 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*.

available events

The following is an example of an event that Amazon EC2 generates when the AMI enters the `available` state following a successful `CreateImage`, `CopyImage`, `RegisterImage`, `CreateRestoreImageTask`, or `EnableImage` operation.

`"State": "available"` indicates that the operation was successful.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
```

```
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "available",
        "ErrorMessage": ""
    }
}
```

failed events

The following is an example of an event that Amazon EC2 generates when the AMI enters the `failed` state following a failed `CreateImage`, `CopyImage`, `RegisterImage`, or `CreateRestoreImageTask` operation.

The following fields provide pertinent information:

- `"State": "failed"` – Indicates that the operation failed.
- `"ErrorMessage": ""` – Provides the reason for the failed operation.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "failed",
        "ErrorMessage": "Description of failure"
    }
}
```

deregistered events

The following is an example of an event that Amazon EC2 generates when the AMI enters the `deregistered` state following a successful `DeregisterImage` operation. If the operation fails, no event is generated. Any failure is known immediately because `DeregisterImage` is a synchronous operation.

`"State": "deregistered"` indicates that the `DeregisterImage` operation was successful.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "deregistered",
        "ErrorMessage": ""
    }
}
```

disabled events

The following is an example of an event that Amazon EC2 generates when the AMI enters the `disabled` state following a successful `DisableImage` operation. If the operation fails, no event is generated. Any failure is known immediately because `DisableImage` is a synchronous operation.

`"State": "disabled"` indicates that the `DisableImage` operation was successful.

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
```

```
        "State": "disabled",
        "ErrorMessage": ""
    }
}
```

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.

Contents

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 or Get-EC2Image 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 or Get-EC2Image command.

Platform details	Usage operation [2]
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0 [3]
Red Hat Enterprise Linux	RunInstances:0010
Red Hat Enterprise Linux with HA	RunInstances:1010
Red Hat Enterprise Linux with SQL Server Standard and HA	RunInstances:1014
Red Hat Enterprise Linux with SQL Server Enterprise and HA	RunInstances:1110
Red Hat Enterprise Linux with SQL Server Standard	RunInstances:0014
Red Hat Enterprise Linux with SQL Server Web	RunInstances:0210
Red Hat Enterprise Linux with SQL Server Enterprise	RunInstances:0110

Platform details	Usage operation [2]
SQL Server Enterprise	RunInstances:0100
SQL Server Standard	RunInstances:0004
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Ubuntu Pro	RunInstances:0g00
Windows	RunInstances:0002
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise [1]	RunInstances:0102
Windows with SQL Server Standard [1]	RunInstances:0006
Windows with SQL Server Web [1]	RunInstances:0202

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

[2] 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 (N. Virginia) in Zone 6.

[3] This appears as RunInstances (Linux/UNIX) in your usage reports.

Finding AMI billing and usage details

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

- **Platform details**
- **Usage operation**
- **AMI ID**

Console

To find the AMI billing information for an AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select the AMI.
4. On the **Details** tab, find **Platform details** and **Usage operation**.

To find the AMI billing information 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.
4. On the **Details** tab, expand **Instance details** and find **Platform details** and **Usage operation**.

AWS CLI

To find the AMI billing information for an AMI

Use the describe-images command.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

The following is example output for a Linux AMI.

```
[
    {
        "PlatformDetails": "Linux/UNIX",
        "UsageOperation": "RunInstances"
    }
]
```

To find the AMI billing information for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

The following is example output for a Windows instance.

```
[
    {
        "PlatformDetails": "Windows",
        "UsageOperation": "RunInstances:0002"
    }
]
```

PowerShell

To find the AMI billing information for an AMI

Use the Get-EC2Image cmdlet.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Format-List PlatformDetails, UsageOperation
```

The following is example output for a Linux AMI.

```
PlatformDetails : Linux/UNIX
```

```
UsageOperation  : RunInstances
```

To find the AMI billing information for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | `
    Format-List PlatformDetails, UsageOperation
```

The following is example output for a Windows instance.

```
PlatformDetails : Windows
UsageOperation  : RunInstances:0002
```

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`.

AMI quotas in Amazon EC2

The following quotas apply to creating and sharing AMIs. The quotas apply per AWS Region.

Quota name	Description	Default quota per Region
AMIs	The maximum number of public and private AMIs allowed per Region. These include available, pending, and disabled AMIs, and AMIs in the Recycle Bin.	50,000
Public AMIs	The maximum number of public AMIs, including public AMIs in the Recycle Bin, allowed per Region.	5
AMI sharing	The maximum number of entities (organizations, organizational units (OUs), and accounts) that an AMI can be shared with in a Region. Note that if you share an AMI with an organization or OU, the number of accounts in the organization or OU does not count towards the quota.	1,000

If you exceed your quotas and you want to create or share more AMIs, you can do the following:

- If you exceed your total AMIs or public AMIs quota, consider deregistering unused images.
- If you exceed your public AMIs quota, consider making one or more public AMIs private.
- If you exceed your AMI sharing quota, consider sharing your AMIs with an organization or OU instead of separate accounts.
- Request a quota increase for AMIs.

Request a quota increase for AMIs

If you need more than the default quota for AMIs, you can request a quota increase.

To request a quota increase for AMIs

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/.
2. In the navigation pane, choose **AWS services**.
3. Choose **Amazon Elastic Compute Cloud (Amazon EC2)** from the list, or type the name of the service in the search box.
4. Choose the AMI quota to request an increase. The AMI quotas you can select are:
   - AMIs
   - Public AMIs
   - AMI sharing
5. Choose **Request quota increase**.
6. For **Change quota value**, enter the new quota value, and then choose **Request**.

To view any pending or recently resolved requests, choose **Dashboard** from the navigation pane. For pending requests, choose the status of the request to open the request receipt. The initial status of a request is **Pending**. After the status changes to **Quota requested**, you'll see the case number under **Support Center case number**. Choose the case number to open the ticket for your request.

After the request is resolved, the **Applied quota value** for the quota is set to the new value.

For more information, see the Service Quotas User Guide.

Amazon EC2 instances

An Amazon EC2 instance is a virtual server in the AWS cloud environment. You have full control over your instance, from the time that you first start it (referred to as launching an instance) until you delete it (referred to as terminating an instance). You can choose from a variety of operating systems when you launch your instance. You can connect to your instance and customize it to meet your needs. For example, you can configure the operating system, install operating system updates, and install applications on your instance.

Amazon EC2 provides a wide range of instance types. You can choose an instance type that provides the compute resources, memory, storage, and network performance that you need to run your applications.

With Amazon EC2, you pay only for what you use. Billing for your instance starts when you launch your instance and it transitions to the running state. Billing stops when you stop your instance and resumes when you start your instance. When you terminate your instance, billing stops when it transitions to the shutting down state.

Amazon EC2 provides features that you can use to optimize the performance and the cost of your instances. For example, you can use Amazon EC2 Fleet or Amazon EC2 Auto Scaling to scale your capacity up or down as your instance utilization changes. You can reduce the costs for your instances using Spot Instances or Savings Plans.

A *managed instance* is managed by a service provider, such as Amazon EKS Auto Mode. You can't directly modify the settings of a managed instance. Managed instances are identified by a **true** value in the **Managed** field. For more information, see [Amazon EC2 managed instances].

Features and tasks

- [Amazon EC2 instance types]
- [Amazon EC2 managed instances]
- [Amazon EC2 billing and purchasing options]
- [Store instance launch parameters in Amazon EC2 launch templates]
- [Launch an Amazon EC2 instance]
- [Connect to your EC2 instance]
- [Amazon EC2 instance state changes]

Amazon EC2 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. For more information about features and use cases, see Amazon EC2 Instance Types.

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

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

Instance type naming conventions

Names are based on instance family, generation, processor family, capabilities, and size. For more information, see Naming conventions in the *Amazon EC2 Instance Types Guide*.

Find an instance type

To determine which instance types meet your requirements, such as supported Regions, compute resources, or storage resources, see Find an Amazon EC2 instance type and Amazon EC2 instance type specifications in the *Amazon EC2 Instance Types Guide*.

Hardware specifications

For detailed instance type specifications, see Specifications in the *Amazon EC2 Instance Types Guide*. For pricing information, see Amazon EC2 On-Demand Pricing.

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 change the instance type later. For more information, see Amazon EC2 instance type changes.

Hypervisor type

Amazon EC2 supports the following hypervisors: Xen and Nitro.

Nitro-based instances

- **General purpose:** M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8g | M8gd | M8i | M8i-flex | T3 | T3a | T4g
- **Compute optimized:** C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8g | C8gd | C8gn
- **Memory optimized:** R5 | R5a | R5ad | R5b | R5d | R5dn | R5n | R6a | R6g | R6gd | R6i | R6idn | R6in | R6id | R7a | R7g | R7gd | R7i | R7iz | R8g | R8gd | R8i | R8i-flex | U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | U-24tb1 | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X2gd | X2idn | X2iedn | X2iezn | X8g | z1d
- **Storage optimized:** D3 | D3en | I3en | I4g | I4i | I7i | I7ie | I8g | I8ge | Im4gn | Is4gen
- **Accelerated computing:** DL1 | DL2q | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | G6f | Gr6 | Gr6f | Inf1 | Inf2 | P3dn | P4d | P4de | P5 | P5e | P5en | P6-B200 | P6e-GB200 | Trn1 | Trn1n | Trn2 | Trn2u | VT1
- **High-performance computing:** Hpc6a | Hpc6id | Hpc7a | Hpc7g
- **Previous generation:** A1

For more information about the supported versions of Nitro hypervisor, see Network feature support in the *Amazon EC2 Instance Types Guide*.

Xen-based instances

- **General purpose**: M1 | M2 | M3 | M4 | T1 | T2

- **Compute optimized**: C1 | C3 | C4
- **Memory optimized**: R3 | R4 | X1 | X1e
- **Storage optimized**: D2 | H1 | I2 | I3
- **Accelerated computing**: F1 | G3 | P2 | P3

AMI virtualization types

The virtualization type of your instance is determined by the AMI that you use to launch it. Current generation instance types support hardware virtual machine (HVM) only. Some previous generation instance types support paravirtual (PV) and some AWS Regions support PV instances. For more information, see Virtualization types.

For best performance, we recommend that you use an HVM AMI. In addition, HVM AMIs are required to take advantage of enhanced networking. HVM virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses PV network and storage drivers for improved performance.

Processors

EC2 instances support a variety of processors.

Processors

- Intel processors
- AMD processors
- AWS Graviton processors
- AWS Trainium
- AWS Inferentia

Intel processors

Amazon EC2 instances that run on Intel processors might include the following processor features. Not all instances that run on Intel processors support all of these processor features. For 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`, `D3en`, and `C6i`. `C5` and `C5d` instances support VNNI for only `12xlarge`, `24xlarge`, and `metal` instances.

Confusion can 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 instance running Ubuntu or Windows displays the CPU architecture as AMD64 even though the instances are running on Intel hardware.

AMD processors

Amazon EC2 instances that run on AMD EPYC processors can help you optimize both cost and performance for your workloads. These instances might support the following processor features. Not all instances that run on AMD processors support all of these processor features. For information about which features are available for each instance type, see Amazon EC2 Instance types.

- AMD Secure Memory Encryption (SME)
- AMD Transparent Single Key Memory Encryption (TSME)
- AMD Advanced Vector Extensions (AVX)
- AMD Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP)
- Vector Neural Network Instructions (VNNI)
- BFloat16

AWS Graviton processors

AWS Graviton is a family of processors designed to deliver the best price performance for your workloads running on Amazon EC2 instances.

For more information, see Getting started with Graviton.

AWS Trainium

Instances powered by AWS Trainium are purpose built for high-performance, cost-effective deep learning training. You can use these instances to train natural language processing, computer vision, and recommender models used across a broad set of applications, such as speech recognition, recommendation, fraud detection, and image and video classification. Use your existing workflows in popular ML frameworks, such as PyTorch and TensorFlow.

AWS Inferentia

Instances powered by AWS Inferentia are designed to accelerate machine learning. They provide high performance and low latency machine learning inference. These instances are optimized for deploying deep learning (DL) models for applications, such as natural language processing, object detection and classification, content personalization and filtering, and speech recognition.

There are a variety of ways that you can get started:

- Use SageMaker AI, a fully-managed service that is the easiest way to get started with machine learning models. For more information, see Get Started with SageMaker AI in the *Amazon SageMaker AI Developer Guide*.
- Launch an Inf1 or Inf2 instance using the Deep Learning AMI. For more information, see AWS Inferentia with DLAMI in the *AWS Deep Learning AMIs Developer Guide*.
- Launch an Inf1 or Inf2 instance using your own AMI and install the AWS Neuron SDK, which enables you to compile, run, and profile deep learning models for AWS Inferentia.

- Launch a container instance using an Inf1 or Inf2 instance and an Amazon ECS-optimized AMI. For more information, see Amazon Linux 2 (Inferentia) AMIs in the *Amazon Elastic Container Service Developer Guide*.
- Create an Amazon EKS cluster with nodes running Inf1 instances. For more information, see Inferentia support in the Amazon EKS User Guide.

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 the resources that your workload requires, such as compute, memory, or storage resources. It can be beneficial to identify several instance types that might suit your workload and evaluate their performance in a test environment. There is no substitute for measuring the performance of your application under load.

You can get suggestions and guidance for EC2 instance types using the EC2 instance type finder. For more information, see the section called "EC2 instance type finder".

If you already have running EC2 instances, you can use AWS Compute Optimizer to get recommendations about the instance types that you should use to improve performance, save money, or both. For more information, see the section called "Compute Optimizer recommendations".

Tasks

- Find an instance type using the console
- Describe an instance type using the AWS CLI
- Find an instance type using the AWS CLI
- Find an instance type using the Tools for PowerShell

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 the name of an instance type to open its details page and view all attributes available through the console. The console does not display all the attributes available through the API or the command line.
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 filter on the following attributes:

   - **Availability zones** – The name of the Availability Zone, Local Zone, or Wavelength Zone. For more information, see the section called "Regions and Zones".
   - **vCPUs** or **Cores** – The number of vCPUs or cores.
   - **Memory (GiB)** – The memory size, in GiB.
   - **Network performance** – The network performance, in Gigabits.
   - **Local instance storage** – Indicates whether the instance type has local instance storage (`true` | `false`).
6. (Optional) To see a side-by-side comparison, select the checkbox for multiple instance types. The comparison is displayed at the bottom of the screen.
7. (Optional) To save the list of instance types to a comma-separated values (.csv) file for further review, choose **Actions**, **Download list CSV**. The file includes all instance types that match the filters you set.
8. (Optional) To launch instances using an instance type that meet your needs, select the checkbox for the instance type and choose **Actions**, **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

Describe an instance type using the AWS CLI

You can use the describe-instance-types command to describe a specific instance type.

To fully describe an instance type

The following command displays all available details for the specified instance type. The output is lengthy, so it is omitted here.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
```

```
    --region us-east-2
```

The describe an instance type and filter the output

The following command displays the networking details for the specified instance type.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].NetworkInfo"
```

The following is example output.

```
[
    {
        "NetworkPerformance": "Low to Moderate",
        "MaximumNetworkInterfaces": 2,
        "MaximumNetworkCards": 1,
        "DefaultNetworkCardIndex": 0,
        "NetworkCards": [
            {
                "NetworkCardIndex": 0,
                "NetworkPerformance": "Low to Moderate",
                "MaximumNetworkInterfaces": 2,
                "BaselineBandwidthInGbps": 0.064,
                "PeakBandwidthInGbps": 1.024
            }
        ],
        "Ipv4AddressesPerInterface": 2,
        "Ipv6AddressesPerInterface": 2,
        "Ipv6Supported": true,
        "EnaSupport": "unsupported",
        "EfaSupported": false,
        "EncryptionInTransitSupported": false,
        "EnaSrdSupported": false
    }
]
```

The following command displays the available memory for the specified instance type.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
```

```
    --region us-east-2 \
    --query "InstanceTypes[].MemoryInfo"
```

The following is example output.

```
[
    {
        "SizeInMiB": 1024
    }
]
```

Find an instance type using the AWS CLI

You can use the describe-instance-types and describe-instance-type-offerings commands to find the instance types that meet your needs.

Examples

- Example 1: Find an instance type by Availability Zone
- Example 2: Find an instance type by available memory size
- Example 3: Find an instance type by available instance storage
- Example 4: Find an instance type that supports hibernation

Example 1: Find an instance type by Availability Zone

The following example displays only the instance types offered in the specified Availability Zone.

```
aws ec2 describe-instance-type-offerings \
    --location-type "availability-zone" \
    --filters "Name=location,Values=us-east-2a" \
    --region us-east-2 \
    --query "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

The output is a list of instance types, sorted alphabetically. The following is the start of the output only.

```
a1.2xlarge
a1.4xlarge
a1.large
```

```
a1.medium
a1.metal
a1.xlarge
c4.2xlarge
    ...
```

Example 2: Find an instance type by available memory size

The following example displays only current generation instance types with 64 GiB (65536 MiB) of memory.

```
aws ec2 describe-instance-types \
    --filters "Name=current-generation,Values=true" "Name=memory-info.size-in-mib,Values=65536" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

The output is a list of instance types, sorted alphabetically. The following is the start of the output only.

```
c5a.8xlarge
c5ad.8xlarge
c6a.8xlarge
c6g.8xlarge
c6gd.8xlarge
c6gn.8xlarge
c6i.8xlarge
c6id.8xlarge
c6in.8xlarge
    ...
```

Example 3: Find an instance type by available instance storage

The following example displays the total size of instance storage for all R7 instances with instance store volumes.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r7*" "Name=instance-storage-supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[].[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
```

```
    --output table
```

The following is example output.

```
---------------------------
|  DescribeInstanceTypes  |
+----------------+--------+
|  r7gd.xlarge   |  237   |
|  r7gd.8xlarge  |  1900  |
|  r7gd.16xlarge |  3800  |
|  r7gd.medium   |  59    |
|  r7gd.4xlarge  |  950   |
|  r7gd.2xlarge  |  474   |
|  r7gd.metal    |  3800  |
|  r7gd.large    |  118   |
|  r7gd.12xlarge |  2850  |
+----------------+--------+
```

Example 4: Find an instance type that supports hibernation

The following example displays the instance types that support hibernation.

```
aws ec2 describe-instance-types \
    --filters "Name=hibernation-supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

The output is a list of instance types, sorted alphabetically. The following is the start of the output only.

```
c4.2xlarge
c4.4xlarge
c4.8xlarge
c4.large
c4.xlarge
c5.12xlarge
c5.18xlarge
c5.2xlarge
c5.4xlarge
c5.9xlarge
```

```
...
```

Find an instance type using the Tools for PowerShell

You can use the Get-EC2InstanceType and Get-EC2InstanceTypeOffering cmdlets to find the instance types that meet your needs.

Examples

- Find an instance type by Availability Zone
- Find an instance type by available memory size
- Find an instance type by available instance storage
- Find an instance type that supports hibernation

Find an instance type by Availability Zone

The following example displays only the instance types offered in the specified Availability Zone.

```
(Get-EC2InstanceTypeOffering `
    -LocationType "availability-zone" `
    -Region us-east-2 `
    -Filter @{Name="location"; Values="us-east-2a"}).InstanceType | Sort-Object `
```

Find an instance type by available memory size

The following example displays only current generation instance types with 64 GiB (65536 MiB) of memory.

```
(Get-EC2InstanceType `
    -Filter @{Name="current-generation"; Values="true"},
            @{Name="memory-info.size-in-mib"; Values="65536"}).InstanceType | Sort-
Object
```

Find an instance type by available instance storage

The following example displays the total size of instance storage for all R7 instances with instance store volumes.

```
Get-EC2InstanceType `
```

```
    -Filter @{Name="instance-type"; Values="r7*"},
            @{Name="instance-storage-supported"; Values="true"} | `
     Select InstanceType, @{Name="TotalSizeInGB";
 Expression={($_.InstanceStorageInfo.TotalSizeInGB)}}
```

The following is example output.

```
InstanceType  TotalSizeInGB
------------  -------------
r7gd.8xlarge           1900
r7gd.16xlarge          3800
r7gd.xlarge             237
r7gd.4xlarge            950
r7gd.medium              59
r7gd.2xlarge            474
r7gd.large              118
r7gd.metal             3800
r7gd.12xlarge          2850
```

Find an instance type that supports hibernation

The following example displays the instance types that support hibernation.

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-Object
```

Get recommendations from EC2 instance type finder

EC2 instance type finder considers your use case, workload type, CPU manufacturer preference, and how you prioritize price and performance, as well as additional parameters that you can specify. It then uses this data to provide suggestions and guidance for Amazon EC2 instance types that are best suited to your new workloads.

With so many instance types available, finding the right instance types for your workload can be time-consuming and complex. By using the EC2 instance type finder, you can remain up to date with the latest instance types and achieve the best price-performance for your workloads.

You can get suggestions and guidance for EC2 instance types using the Amazon EC2 console. You can also go directly to Amazon Q to ask for instance type advice. For more information, see the Amazon Q Developer User Guide.

If you're looking for instance type recommendations for an *existing* workload, use AWS Compute Optimizer. For more information, see Get EC2 instance recommendations from Compute Optimizer.

Use the EC2 instance type finder

In the Amazon EC2 console, you can get instance type suggestions from the EC2 instance type finder in the launch instance wizard, when creating a launch template, or on the **Instance types** page.

Use the following instructions to get suggestions and guidance for EC2 instance types using the EC2 instance type finder in the Amazon EC2 console. To view an animation of the steps, see View an animation: Get instance type suggestions using the EC2 instance type finder.

To get instance type suggestions using the EC2 instance type finder

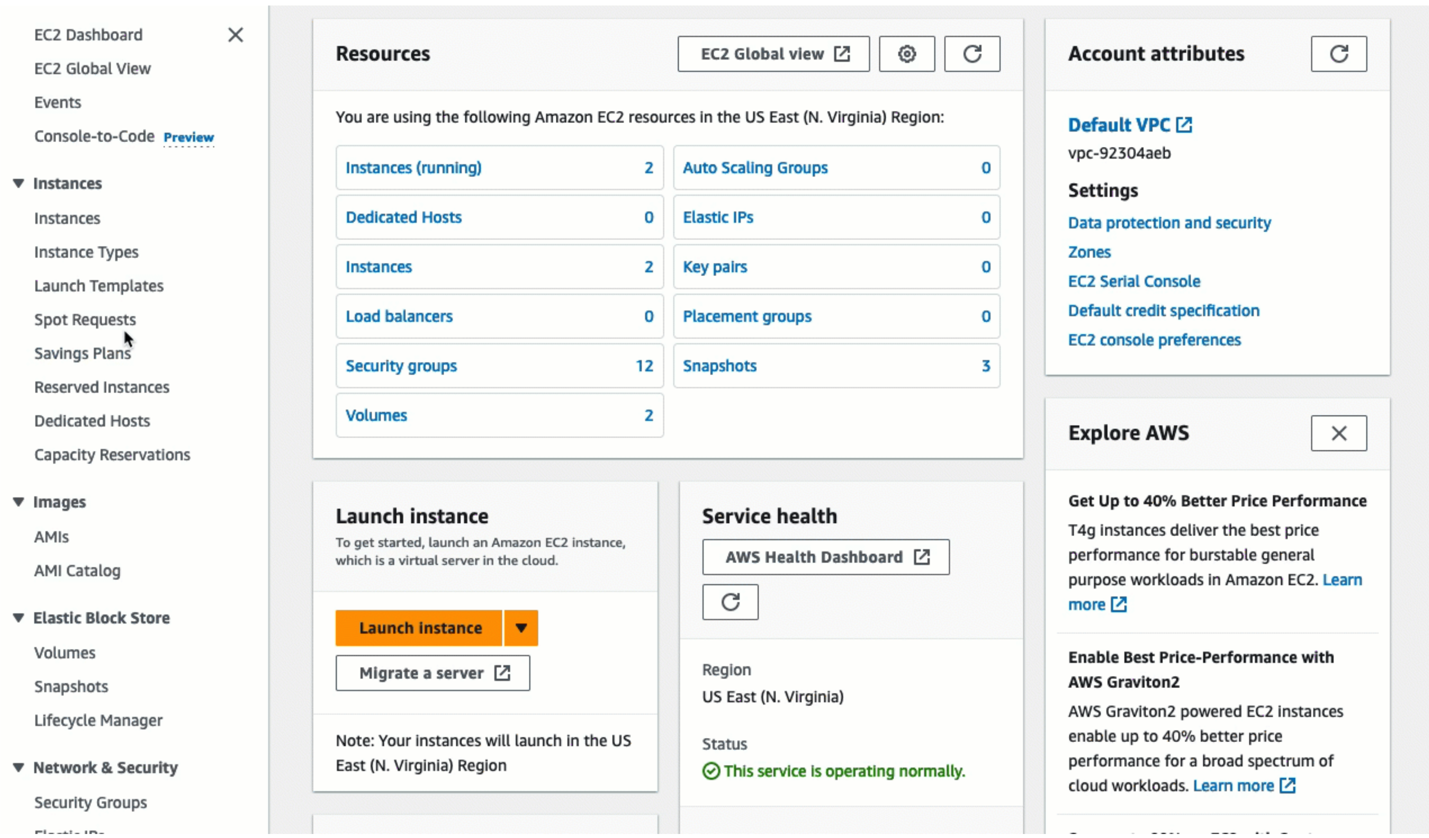
1. Start your process using any of the following:
   - Follow the procedure to launch an instance. Next to **Instance type**, choose the **Get advice** link.
   - Follow the procedure to create a launch template. Next to **Instance type**, choose the **Get advice** link.
   - In the navigation pane, choose **Instance Types**, and then choose the **Instance type finder** button.
2. In the **Get advice on instance type selection** screen, do the following:
   a. Specify your instance type requirements by selecting options for **Workload type**, **Use case**, **Priority**, and **CPU manufacturers**.
   b. (Optional) To specify more detailed requirements for your workload, do the following:
      i. Expand **Advanced parameters**.
      ii. To add a parameter, select a parameter, choose **Add**, and specify a value for the parameter. Repeat for each additional parameter that you want to add. To indicate no minimum or maximum value, leave the field empty.
      iii. To remove a parameter after adding it, choose the **X** next to the parameter.
   c. Choose **Get instance type advice**.

      Amazon EC2 provides you with suggestions for instance families matching your specified requirements.

3. To view the details of each instance type within the suggested instance families, choose **View recommended instance family details**.
4. Select an instance type that meets your requirements, and then choose **Actions**, **Launch instance** or **Actions**, **Create launch template**.

   Alternatively, if you started the process in the launch instance wizard or the launch template page, and you prefer to go back to your original flow, make note of the instance type you'd like to use. Then, in the launch instance wizard or launch template, for **Instance type**, choose the instance type, and complete the procedure to launch an instance or create a launch template.

View an animation: Get instance type suggestions using the EC2 instance type finder



Get EC2 instance recommendations from Compute Optimizer

AWS Compute Optimizer provides Amazon EC2 recommendations to help you improve performance, save money, or both. You can use these recommendations to decide whether to change 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. For more information, see Amazon EC2 instance metrics in the *AWS Compute Optimizer User Guide*.

Contents

Requirements

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*.

Compute Optimizer generates recommendations for some instance types, but not all instance types. If you're using an unsupported instance type, Compute Optimizer will not generate recommendations. For the list of supported instance types, see Amazon EC2 instance requirements in the *AWS Compute Optimizer User Guide*.

Finding classifications

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.

View recommendations

After you opt in to Compute Optimizer, you can view the findings that Compute Optimizer generates for your EC2 instances in the Amazon 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.

To view recommendations for an instance using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose the instance ID to open the instance detail page.
4. On the instance detail page, in the top summary section, locate **AWS Compute Optimizer finding**. If there is a finding, we display the finding classification and a link to view the details. Otherwise, we display **No recommendations available for this instance.**
5. If there is a finding, choose **View detail**. This opens the **Recommendations for EC2 instances** page in the Compute Optimizer console. The current instance type is labeled **Current**. There are also up to three instance type recommendations, labeled **Option 1**, **Option 2**, and **Option 3**. This page also shows recent CloudWatch metric data for the instance.

To view recommendations for all instances in all Regions

You can view recommendations for all of your Amazon EC2 instances in all Regions using the Compute Optimizer console. For more information, see Viewing EC2 instances recommendations and Viewing EC2 instance details in the *AWS Compute Optimizer User Guide*.

Considerations for evaluating recommendations

When you receive a recommendation, you must decide whether to act on it. 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 for burstable performance instances.
- 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 changing the instance type.
- 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.

Amazon EC2 instance type changes

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 increase its size by changing it to a bigger T2 instance type, such as `t2.large`. Or you can change it to another instance type, such as `m5.large`. You might also want to change from a previous generation to a current generation instance type to take advantage of some features, such as 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 EC2 instance recommendations from Compute Optimizer.

When you change the instance type, you'll start paying the rate of the new instance type. For the on-demand rates of all instance types, see Amazon EC2 On-Demand Pricing.

To add additional storage to your instance without changing the instance type, add an EBS volume to the instance. For more information, see Attach an Amazon EBS volume to an instance in the *Amazon EBS User Guide*.

Which instructions to follow?

There are different instructions for changing the instance type. The instructions to use depend on the instance's root volume, and whether the instance type is compatible with the instance's current configuration. For information about how compatibility is determined, see Compatibility for changing the instance type.

Use the following table to determine which instructions to follow.

Root volume	Compatibility	Use these instructions
EBS	Compatible	Change the instance type
EBS	Not compatible	Migrate to a new instance type
Instance store	Not applicable	Migrate to a new instance type

Compatibility for changing the instance type

You can change the instance type only if the instance's current configuration is compatible with the instance type that you want. If the instance type that you want is not compatible with the instance's current configuration, you must launch a new instance with a configuration that is compatible with the instance type, and then migrate your application to the new instance.

Compatibility is determined through the following:

Virtualization type

Linux AMIs use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). If an instance was launched from a PV AMI, you can't change to an instance type that is HVM only. For more information, see Virtualization types. To check the virtualization type of your instance, check the **Virtualization** value on the details pane of the **Instances** screen in the Amazon EC2 console.

Architecture

AMIs are specific to the architecture of the processor, so you must select an instance type with the same processor architecture as the current instance type. For example:

- If the current instance type has 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 changing the instance type of a 32-bit instance, you are limited to these instance types.

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 require the necessary drivers installed. For example, Nitro-based instances require EBS-backed AMIs with the Elastic Network Adapter (ENA) drivers installed. To change from an instance type that does not support enhanced networking to an instance type that supports enhanced networking, you must install the ENA drivers or ixgbevf drivers on the instance, as appropriate.

> **Note**
>
> When you resize an instance with ENA Express enabled, the new instance type must also support ENA Express. For a list of instance types that support ENA Express, see Supported instance types for ENA Express.
>
> To change from an instance type that supports ENA Express to an instance type that does not support it, ensure that ENA Express is not currently enabled before you resize the instance.

NVMe

EBS volumes are exposed as NVMe block devices on Nitro-based instances. If you change from an instance type that does not support NVMe to an instance type that supports NVMe, you must first install the NVMe drivers 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`).

[Linux instances] Therefore, to mount file systems at boot time using `/etc/fstab`, you must use UUID/Label instead of device names.

Volume limit

The maximum number of Amazon EBS volumes that you can attach to an instance depends on the instance type and instance size. For more information, see Amazon EBS volume limits for Amazon EC2 instances.

You can only change to an instance type or instance size that supports the same number or a larger number of volumes than is currently attached to the instance. If you change to an instance type or instance size that does not support the number of currently attached volumes, the request fails. For example, if you change from an `m7i.4xlarge` instance with 32 attached volumes to an `m6i.4xlarge`, which supports a maximum of 27 volumes, the request fails.

NitroTPM

If you launched the instance using an AMI with NitroTPM enabled and an instance type that supports NitroTPM, the instance launches with NitroTPM enabled. You can only change to an instance type that also supports NitroTPM.

Change the instance type for your Amazon EC2 instance

Use the following instructions to change the instance type of an Amazon EBS-backed instance if the instance type that you need is compatible with the current configuration of your instance. For more information, see the section called "Compatibility".

Considerations

- You must stop your instance before you can change its instance type. Ensure that you plan for downtime while your instance is stopped. Stopping the instance and changing its instance type might take a few minutes, and restarting your instance might take a variable amount of time depending on your application's startup scripts. For more information, see Stop and start Amazon EC2 instances.
- When you stop and start an instance, we move the instance to new hardware. If your instance has a public IPv4 address, that is not an Elastic IP, we release the address and give your instance a new public IPv4 address. For more information on IP address behavior throughout the lifecycle of an instance, see Differences between instance states.
- You can't change the instance type of a Spot Instance.
- [Windows instances] We recommend that you update the AWS PV driver package before changing the instance type. For more information, see the section called "Upgrade PV drivers".
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and might terminate it and launch a replacement instance. To prevent this, you can suspend the scaling processes for the group while you're changing the instance type. For more information, see Suspending and resuming a process for an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.
- When you change the instance type of an instance with NVMe instance store volumes, the updated instance might have additional instance store volumes, because all NVMe instance store volumes are available even if they are not specified in the AMI or instance block device mapping. Otherwise, the updated instance has the same number of instance store volumes that you specified when you launched the original instance.
- The maximum number of Amazon EBS volumes that you can attach to an instance depends on the instance type and instance size. You can't change to an instance type or instance size that does not support the number of volumes that are already attached to your instance. For more information, see Amazon EBS volume limits for Amazon EC2 instances.
- [Linux instances] You can use the `AWSSupport-MigrateXenToNitroLinux` runbook to migrate compatible Linux instances from a Xen instance type to a Nitro instance type. For more

information, see [AWSSupport-MigrateXenToNitroLinux runbook]() in the *AWS Systems Manager Automation runbook reference*.

- [Windows instances] For additional guidance on migrating compatible Windows instances from a Xen instance type to a Nitro instance type, see [Migrate to latest generation instance types]().

To change the instance type of 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 changing the instance type]().
2. [Windows instances] If you configured your Windows instance to use [static IP addressing]() and you change from an instance 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**, open **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 at https://console.aws.amazon.com/ec2/.
4. In the navigation pane, choose **Instances**.
5. Select the instance and choose **Instance state**, **Stop instance**. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
6. With the instance still selected, choose **Actions**, **Instance settings**, **Change instance type**. This option is grayed out if the instance state is not `stopped`.
7. On the **Change instance type** page, do the following:

   a. For **Instance type**, select the instance type that you want.

      If the instance type is not in the list, then it's not compatible with the configuration of your instance. Instead, use the following instructions: [Migrate to a new instance type by launching a new EC2 instance]().

   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. (Optional) Configure vCPU options on the new instance type.

When you change the instance type of an existing instance, Amazon EC2 applies the CPU option settings from the existing instance to the new instance, if possible. If the new instance type doesn't support those settings, the CPU options are reset to **None**. This option uses the default number of vCPUs for the new instance type.

If the instance type that you selected supports vCPU configuration, select **Specify CPU options** in the **Advanced details** panel to configure vCPUs for your new instance type.

d. Choose **Change** to accept the new settings.

8. To start the instance, select the instance and choose **Instance state**, **Start instance**. It can take a few minutes for the instance to enter the `running` state. If your instance won't start, see Troubleshoot changing the instance type.

9. [Windows instances] If your instance runs Windows Server 2016 or Windows Server 2019 with EC2Launch v1, connect to your Windows instance and run the following EC2Launch PowerShell script to configure the instance after the instance type is changed.

> **Important**
>
> The administrator password will reset when you enable the initialize instance EC2 Launch 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 (EC2Launch) or Change settings (EC2Launch v2).

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

Migrate to a new instance type by launching a new EC2 instance

You can change the instance type of an EC2 instance only if it is an EBS-backed instance with a configuration that is compatible with the new instance type that you want. Otherwise, if the configuration or your instance is not compatible with the new instance type, or it is an instance store-based instance, you must launch a replacement instance that is compatible with the instance type that you want. For more information about how compatibility is determined, see Compatibility for changing the instance type.

Overview of the migration process

- Back up the data on the original instance.
- Launch a new instance with a configuration that is compatible with the new instance type that you want, attaching any EBS volumes that were attached to your original instance.
- Install your application on your new instance.
- Restore any data.
- If the original instance has an Elastic IP address, you must associate it with your new instance to ensure that your users can continue to use your application without interruption.

To migrate an instance to a new instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Back up any data that you still need as follows:
    - Connect to your instance and copy the data on your instance store volumes to persistent storage.
    - Create snapshots of your EBS volumes so that you can create new volumes with the same data, or detach the volumes from the original instance so that you can attach them to the new instance.
3. In the navigation pane, choose **Instances**.
4. Choose **Launch instances**. When you configure the instance, do the following:
    a. Select an AMI that supports the instance type that you want. For example, you must select an AMI that supports the processor type of the new instance type. Also, current generation instance types require an HVM AMI.
    b. Select the new instance type that you want. If the instance type that you want isn't available, then it's not compatible with the configuration of the AMI that you selected.
    c. If you want to allow the same traffic to reach the new instance, select the same VPC and security group that are used with the original instance.

d. When you're done configuring your new instance, complete the steps to select a key pair and launch your instance. It can take a few minutes for the instance to enter the `running` state.

5. If you backed up data to an EBS snapshot, create a volume from the snapshot and then attach the volume to the new instance.

   To move an EBS volume from the original instance to the new instance, detach the volume from the original instance and then attach the volume to the new instance.

6. Install your application and any required software on the new instance.
7. Restore any data that you backed up from the instance store volumes of the original instance.
8. If the original instance has an Elastic IP address, assign it to the new 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 new instance.
   f. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
   g. Choose **Associate**.

9. (Optional) You can terminate the original instance if it's no longer needed. Select the instance, verify that you are about to terminate the original instance and not the new instance (for example, check the name or launch time), and then choose **Instance state**, **Terminate instance**.

Troubleshoot changing the instance type

Use the following information to help diagnose and fix issues that you might encounter when changing the instance type.

Instance won't start after changing instance type

Possible cause: Requirements for new instance type not met

If your instance won't boot, it is possible that one of the requirements for the new instance type was not met. For more information, see Why is my Linux instance not booting after I changed its type?

Possible cause: AMI does not support instance type

If you use the EC2 console to change the instance type, only the instance types that are supported by the selected AMI are available. However, if you use the AWS CLI to launch an instance, you can specify an incompatible AMI and instance type. If the AMI and instance type are incompatible, the instance can't start. For more information, see Compatibility for changing the instance type.

Possible cause: Instance is in cluster placement group

If your instance is in a cluster placement group and, after changing the instance type, the instance fails to start, try the following:

1. Stop all the instances in the cluster placement group.
2. Change the instance type of the affected instance.
3. Start all the instances in the cluster placement group.

Application or website not reachable from the internet after changing instance type

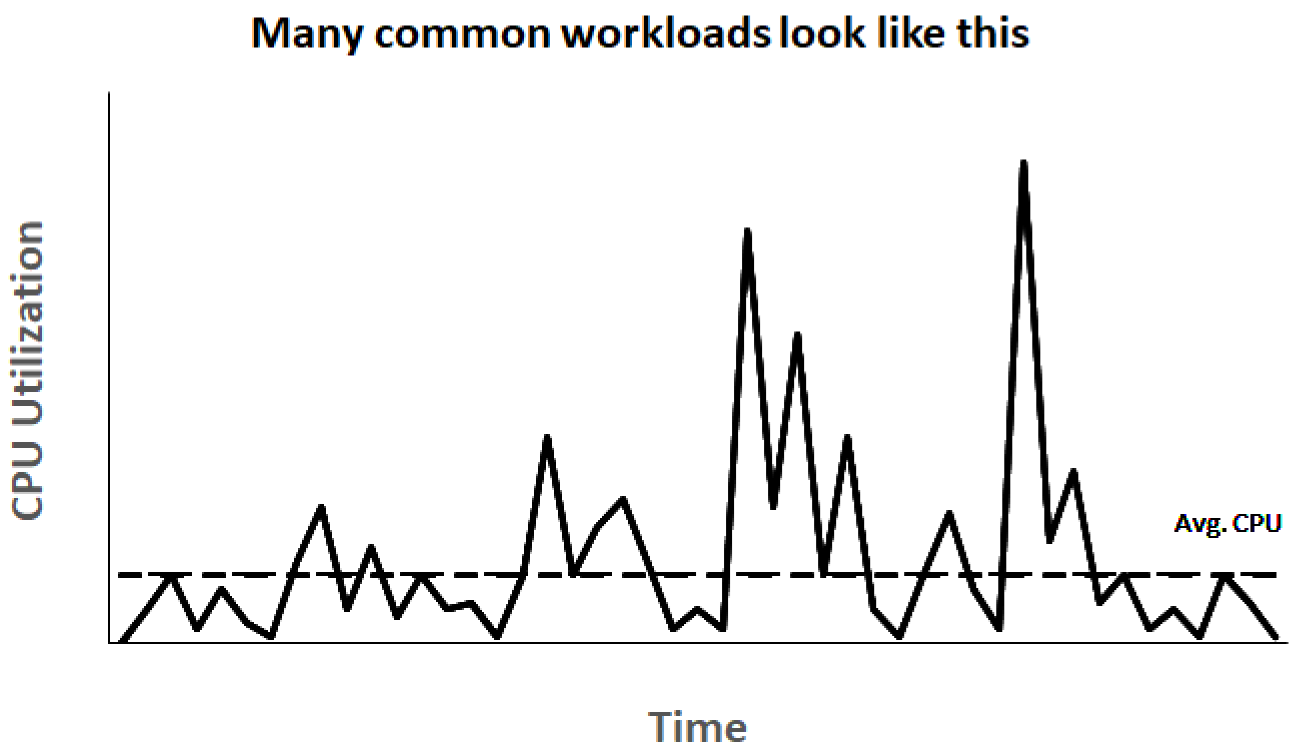
Possible cause: Public IPv4 address is released

When you change the instance type, you must first stop the instance. When you stop an instance, we release the public IPv4 address and give your instance a new public IPv4 address.

To retain the public IPv4 address between instance stops and starts, we recommend that you use an Elastic IP address, at no extra cost provided your instance is running. For more information, see Elastic IP addresses.

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.



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.

Note

This topic describes burstable CPU. For information about burstable network performance, see Amazon EC2 instance network bandwidth.

EC2 burstable instance types

The EC2 burstable instances consist of T4g, 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.

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

Type	Description	Processor family
T2	Previous generation burstable instances	Intel Xeon processors

For information about instance pricing and additional specifications, see Amazon EC2 Pricing and Amazon EC2 Instance Types. For information about burstable network performance, see Amazon EC2 instance network bandwidth.

If your created your AWS account before July 15, 2025 and it's 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. If you created your AWS account on or after July 15, 2025, you can use `t3.micro`, `t3.small`, `t4g.micro`, `t4g.small` instance types for 6 months or until your credits are used up. For more information, see AWS Free Tier.

Supported purchasing options for T instances

- On-Demand Instances
- Reserved Instances
- Dedicated Instances (T3 only)
- Dedicated Hosts (T3 only, in `standard` mode only)
- Spot Instances

For more information, see Amazon EC2 billing and purchasing options.

Contents

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. Compute Optimizer may also recommend a different instance type based on your scenario. For more information, see Viewing EC2 instance recommendations in the *AWS Compute Optimizer User Guide*.

Key concepts 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 credits 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.

Contents

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.

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 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: Baseline utilization % = (number of credits earned/number of vCPUs)/60 minutes.

For the baseline utilization of each burstable performance instance type, see the credit table.

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.

Type	Type of CPU credits supported	Credit configuration modes	Accrued CPU credits lifespan between instance starts and stops
Latest generation			
T4g	Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)	Standard, Unlimited (default)	7 days (credits persist for 7 days after an instance stops)
T3a	Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)	Standard, Unlimited (default)	7 days (credits persist for 7 days after an instance stops)
T3	Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)	Standard, Unlimited (default)	7 days (credits persist for 7 days after an instance stops)
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)

> **Note**
>
> Unlimited mode is not supported for T3 instances that are launched on a Dedicated Host.

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).

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

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs***	Baseline utilization per vCPU
`t2.2xlarge`	81.6	1958.4	8	17%**
T3				
`t3.nano`	6	144	2	5%**
`t3.micro`	12	288	2	10%**
`t3.small`	24	576	2	20%**
`t3.medium`	24	576	2	20%**
`t3.large`	36	864	2	30%**
`t3.xlarge`	96	2304	4	40%**
`t3.2xlarge`	192	4608	8	40%**
T3a				
`t3a.nano`	6	144	2	5%**
`t3a.micro`	12	288	2	10%**
`t3a.small`	24	576	2	20%**
`t3a.medium`	24	576	2	20%**
`t3a.large`	36	864	2	30%**
`t3a.xlarge`	96	2304	4	40%**
`t3a.2xlarge`	192	4608	8	40%**
T4g				

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs***	Baseline utilization per vCPU
t4g.nano	6	144	2	5%**
t4g.micro	12	288	2	10%**
t4g.small	24	576	2	20%**
t4g.medium	24	576	2	20%**
t4g.large	36	864	2	30%**
t4g.xlarge	96	2304	4	40%**
t4g.2xlarge	192	4608	8	40%**

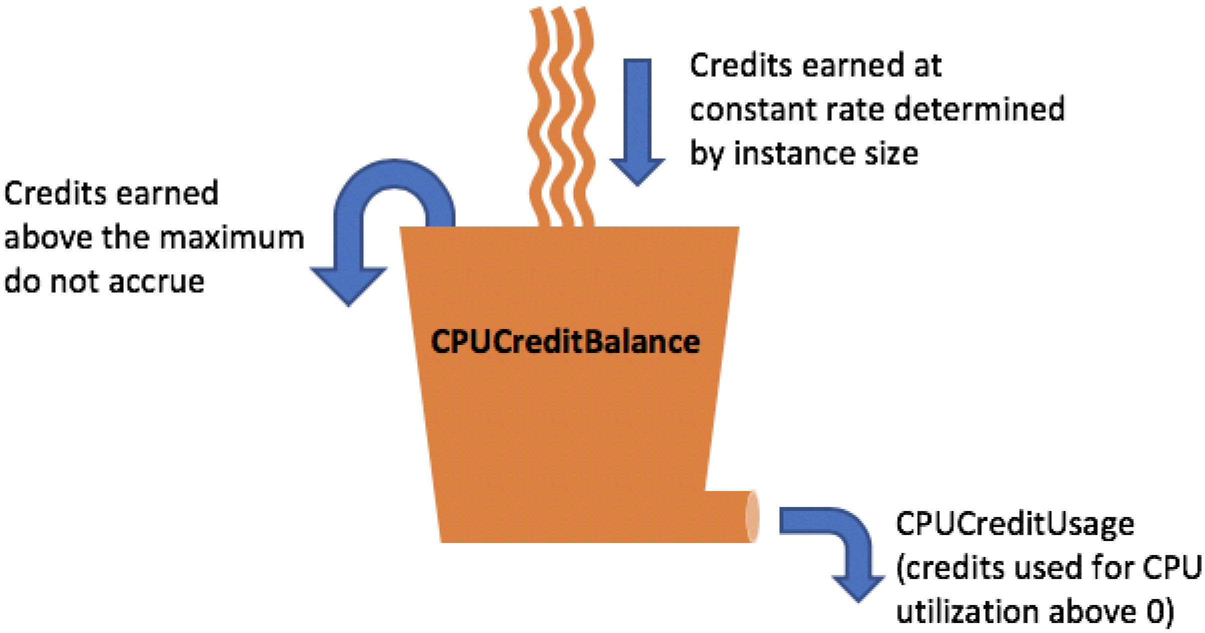
* The number of credits that can be accrued is equivalent to the number of credits that can be earned in a 24-hour period.
** The percentage baseline utilization in the table is per vCPU. In CloudWatch, CPU utilization is shown per vCPU. For example, the CPU utilization for a `t3.large` instance operating at the baseline level is shown as 30% in CloudWatch CPU metrics. For information about how to calculate the baseline utilization, see Baseline utilization.
*** Each vCPU is a thread of either an Intel Xeon core or an AMD EPYC core, except for T2 and T4g 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 bucket indicates the CPU credit balance limit, and the spillover indicates the newly earned credits that exceed the limit.



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

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

T4g, 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. T3 instances launched on a Dedicated Host launch as `standard` by default; `unlimited` mode is not supported for T3 instances on a Dedicated Host.

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 T4g, 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.

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:

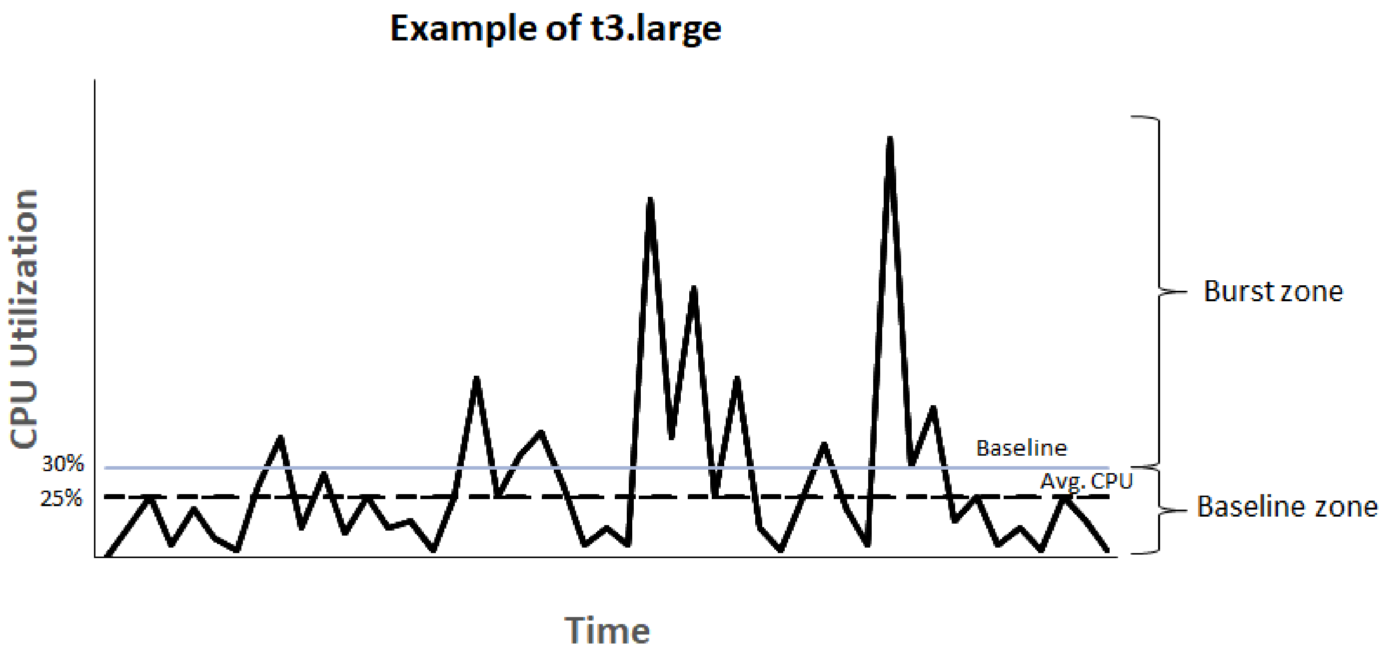
```
(number of credits earned/number of vCPUs)/60 minutes = % baseline
utilization
```

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

```
(6 credits earned/2 vCPUs)/60 minutes = 5% baseline utilization
```

A `t3.large` instance, with 2 vCPUs, earns 36 credits per hour, resulting in a baseline utilization of 30% (`(36/2)/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 your created your AWS account before July 15, 2025 and 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] of the instance.

T4g, T3a, and T3 instances launch as `unlimited` by default (unless you [change the 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] mode to avoid paying higher costs.

For more information, see Surplus credits can incur charges and Launch burstable performance instances.

> **Note**
>
> T3 instances launched on a Dedicated Host launch as `standard` by default; `unlimited` mode is not supported for T3 instances on a Dedicated Host.

Contents

Unlimited mode concepts for burstable instances

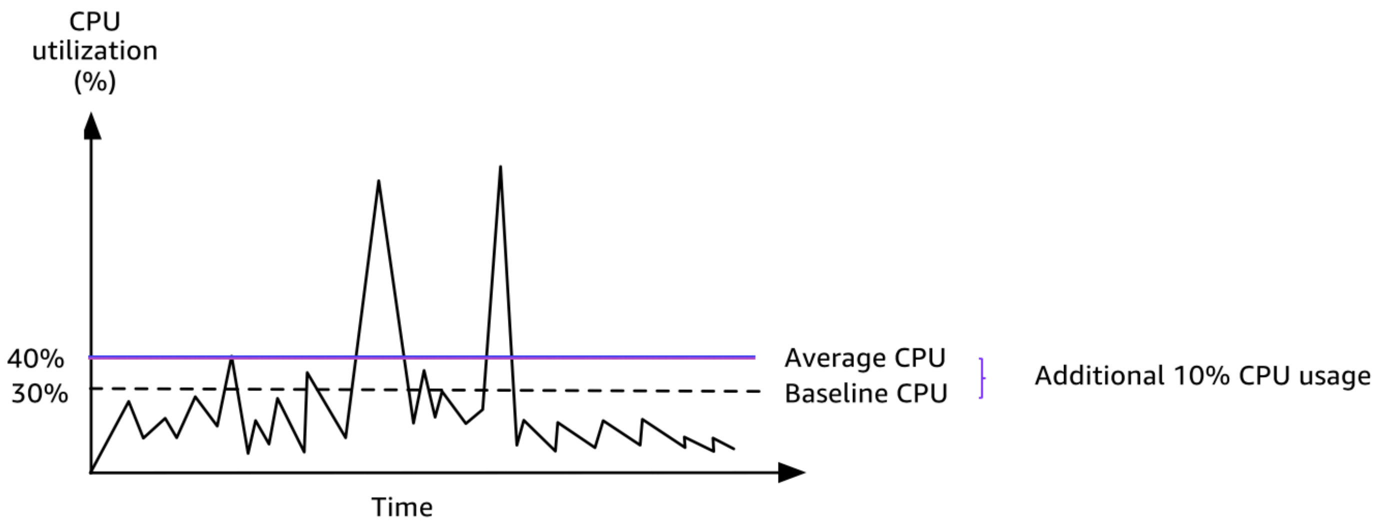
The `unlimited` mode is a credit configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance. You can set `unlimited` as the default credit option at the account level per AWS Region, per burstable performance instance family,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

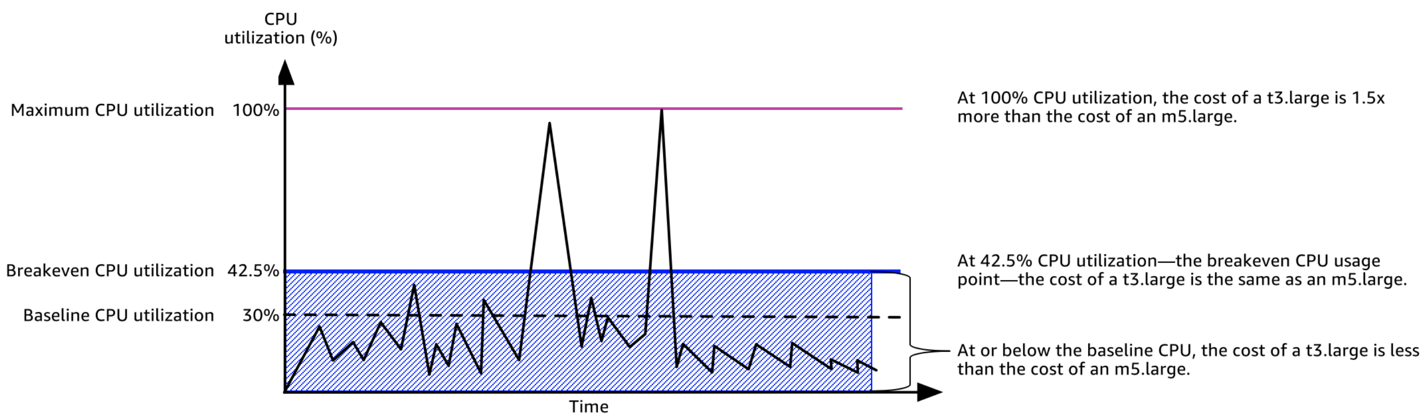
When to use unlimited mode versus fixed CPU

When determining whether you should use a burstable performance instance in `unlimited` mode, such as T3, or a fixed performance instance, such as M5, you need to determine the breakeven CPU usage. The breakeven CPU usage for a burstable performance instance is the point at which a burstable performance instance costs the same as a fixed performance instance. The breakeven CPU usage helps you determine the following:

- If the average CPU usage over a 24-hour period is at or below the breakeven CPU usage, use a burstable performance instance in `unlimited` mode so that you can benefit from the lower price of a burstable performance instance while getting the same performance as a fixed performance instance.

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

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



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

Instance type	vCPUs	T3 price*/ hour	M5 price*/ hour	Price difference	T3 baseline utilization per vCPU (%)	Charge per vCPU hour for surplus credits	Charge per vCPU minute	Additional burst minutes available per vCPU	Additional CPU % available	Breakeven CPU %
A	B	C	D	E = D - C	F	G	H = G / 60	I = E / H	J = (I /	K = F + J

Instance type	vCPUs	T3 price*/ hour	M5 price*/ hour	Price difference	T3 baseline utilization per vCPU (%)	Charge per vCPU hour for surplus credits	Charge per vCPU minute	Additional burst minutes available per vCPU	Additional CPU % available	Breakeven CPU %
									60) / B	
t3.large	2	$0.0835	$0.096	$0.0125	30%	$0.05	$0.00083	15	12.5%	42.5%

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

The table provides the following information:

- Column A shows the instance type, `t3.large`.
- Column B shows the number of vCPUs for the `t3.large`.
- Column C shows the price of a `t3.large` per hour.
- Column D shows the price of an `m5.large` per hour.
- Column E shows the price difference between the `t3.large` and the `m5.large`.
- Column F shows the baseline utilization per vCPU of the `t3.large`, which is 30%. At the baseline, the hourly cost of the instance covers the cost of the CPU usage.
- Column G shows the flat additional rate per vCPU-hour that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column H shows the flat additional rate per vCPU-minute that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column I shows the number of additional minutes that the `t3.large` can burst per hour at 100% CPU while paying the same price per hour as an `m5.large`.
- Column J shows the additional CPU usage (in %) over baseline that the instance can burst while paying the same price per hour as an `m5.large`.

- Column K shows the breakeven CPU usage (in %) that the `t3.large` can burst without paying more than the `m5.large`. Anything above this, and the `t3.large` costs more than the `m5.large`.

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

T3 instance type	Breakeven CPU usage (in %) for T3 compared to M5
`t3.large`	42.5%
`t3.xlarge`	52.5%
`t3.2xlarge`	52.5%

Surplus credits can incur charges

If the average CPU utilization of an instance is at or below the baseline, the instance incurs no additional charges. Because an instance earns a maximum number of credits in a 24-hour period (for example, a `t3.micro` instance can earn a maximum of 288 credits in a 24-hour period), it can spend surplus credits up to that maximum without being charged.

However, if CPU utilization stays above the baseline, the instance cannot earn enough credits to pay down the surplus credits that it has spent. The surplus credits that are not paid down are charged at a flat additional rate per vCPU-hour. For information about the rate, see 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 the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour.
- The instance is stopped or terminated.
- The instance is switched from `unlimited` to `standard`.

Spent surplus credits are tracked by the CloudWatch metric `CPUSurplusCreditBalance`. Surplus credits that are charged are tracked by the CloudWatch metric `CPUSurplusCreditsCharged`. For more information, see Additional CloudWatch metrics for burstable performance instances.

No launch credits for T2 Unlimited instances

T2 Standard instances receive launch credits, 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.

T4g, T3a, and T3 instances never receive launch credits because they launch in Unlimited mode by default, and therefore can burst immediately upon start. The Unlimited mode credit configuration enables T4g, T3a, and T3 instances to use as much CPU as needed to burst beyond the 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 Configure the credit specification at launch and Manage the credit specification of a burstable performance 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. For more information, see Manage the default credit specification for an account.

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 Configure burstable performance instances.

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 CPU credits for burstable instances](#).

Unlimited mode examples for burstable instances

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

Examples

- [Example 1: Explain credit use with T3 Unlimited](#)
- [Example 2: Explain credit use with T2 Unlimited](#)

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.

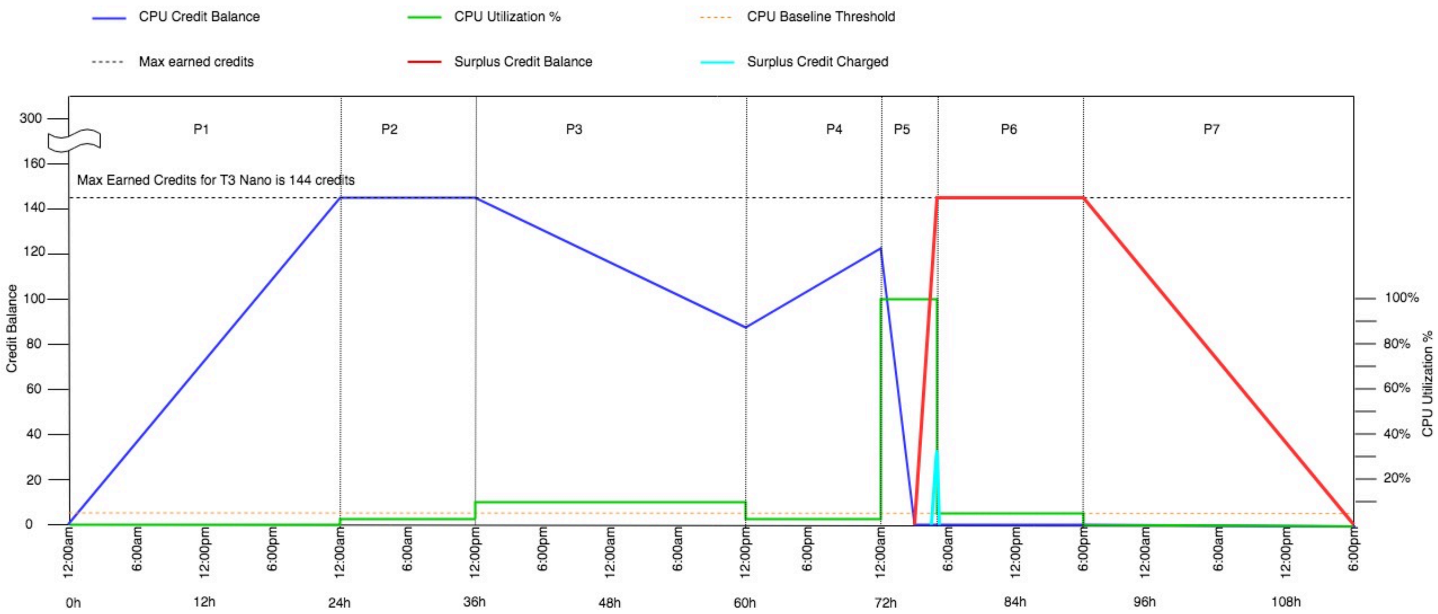
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

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 `t2.nano` instance launched as `unlimited`, and how it spends *earned* and *surplus* credits to sustain CPU utilization.

A `t2.nano` instance earns 72 CPU credits over a rolling 24-hour period, which it can redeem for 72 minutes of vCPU use. When it depletes its CPU credit balance (represented by the CloudWatch metric `CPUCreditBalance`), it can spend *surplus* CPU credits—that it has *not yet earned*—to burst for as long as it needs. Because a `t2.nano` instance earns a maximum of 72 credits in a 24-hour period, it can spend surplus credits up to that maximum without being charged immediately. If it spends more than 72 CPU credits, it is charged for the difference at the end of the hour.

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

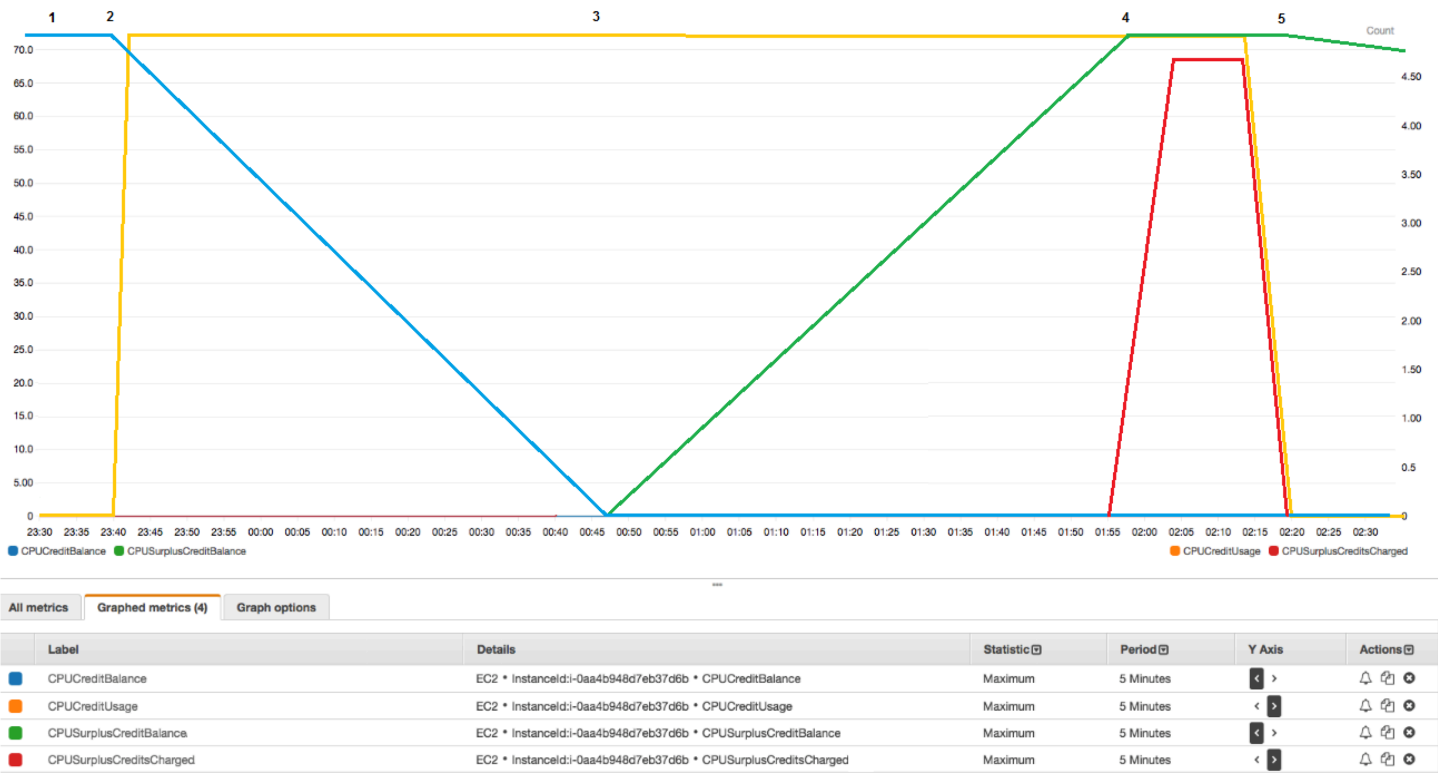
1 – In the first 10 minutes, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value remains at its maximum of 72.

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

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

4 – Surplus credits are spent until 01:55, when the `CPUSurplusCreditBalance` value reaches 72 CPU credits. This is equal to the maximum a `t2.nano` instance can earn in a 24-hour period. Any surplus credits spent thereafter cannot be offset by earned credits within the 24-hour period, which results in a small additional charge at the end of the hour.

5 – The instance continues to spend surplus credits until around 02:20. At this time, CPU utilization falls below the baseline, and the instance starts to earn credits at 3 credits per hour (or 0.25 credits every 5 minutes), which it uses to pay down the `CPUSurplusCreditBalance`. After the `CPUSurplusCreditBalance` value reduces to 0, the instance starts to accrue earned credits in its `CPUCreditBalance` at 0.25 credits every 5 minutes.



Calculating the bill (Linux instance)

Surplus credits cost $0.05 per vCPU-hour. The instance spent approximately 25 surplus credits between 01:55 and 02:20, which is equivalent to 0.42 vCPU-hours. Additional charges for this instance are 0.42 vCPU-hours x $0.05/vCPU-hour = $0.021, rounded to $0.02. Here is the month-end bill for this T2 Unlimited instance:

Amazon Elastic Compute Cloud running Linux/UNIX		
$0.0058 per On Demand Linux t2.nano Instance Hour	720.000 Hrs	$4.18

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.05 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.02

Calculating the bill (Windows instance)

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:

Amazon Elastic Compute Cloud running Windows		
$0.0081 per On Demand Windows t2.nano Instance Hour	720.000 Hrs	$5.83

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.096 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.04

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

Standard mode 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 for burstable performance instances.

Contents

Standard mode concepts for burstable instances

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 T4g, 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.

T4g, 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, and T1 Standard instances get 15 launch credits. 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.

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

Instance type	Launch credits	vCPUs
`t1.micro`	15	1
`t2.nano`	30	1
`t2.micro`	30	1
`t2.small`	30	1
`t2.medium`	60	2
`t2.large`	60	2
`t2.xlarge`	120	4
`t2.2xlarge`	240	8

Launch credit limits

There is a limit to the number of times T2 Standard instances can receive launch credits. The default limit is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. For example, the limit is reached when one instance is stopped and started 100 times within a 24-hour period, or when 100 instances are launched within a 24-hour period, or other combinations that equate to 100 starts. New accounts may have a lower limit, which increases over time based on your usage.

> **Tip**
>
> To ensure that your workloads always get the performance they need, switch to [Unlimited mode for burstable performance instances]() or consider using a larger instance size.

Differences between launch credits and earned credits

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

	Launch credits	Earned credits
Credit earn rate	T2 Standard instances get 30 launch credits per vCPU at launch or start. If a T2 instance is switched from `unlimited` to `standard`, it does not get launch credits at the time of switching.	Each T2 instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. For more information about the number of CPU credits earned per instance size, see the [credit table]().
Credit earn limit	The limit for receiving launch credits is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. New accounts may have a lower limit, which increases over time based on your usage.	A T2 instance cannot accrue more credits than the CPU credit balance limit. If the CPU credit balance has reached its limit, any credits that are earned after the limit is reached are discarded. Launch credits do not count towards the limit. For more information about the CPU credit

	Launch credits	Earned credits
		balance limit for each T2 instance size, see the credit table.
Credit use	Launch credits are spent first, before earned credits.	Earned credits are spent only after all launch credits are spent.
Credit expiration	When a T2 Standard instance is running, launch credits do not expire. When a T2 Standard instance stops or is switched to T2 Unlimited, all launch credits are lost.	When a T2 instance is running, earned credits that have accrued do not expire. When the T2 instance stops, all accrued earned credits are lost.

The number of accrued launch credits and accrued earned credits is tracked by the CloudWatch metric `CPUCreditBalance`. For more information, see `CPUCreditBalance` in the CloudWatch metrics table.

Standard mode examples for burstable instances

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

Examples

- Example 1: Explain credit use with T3 Standard
- Example 2: Explain credit use with T2 Standard

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.

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 it 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.

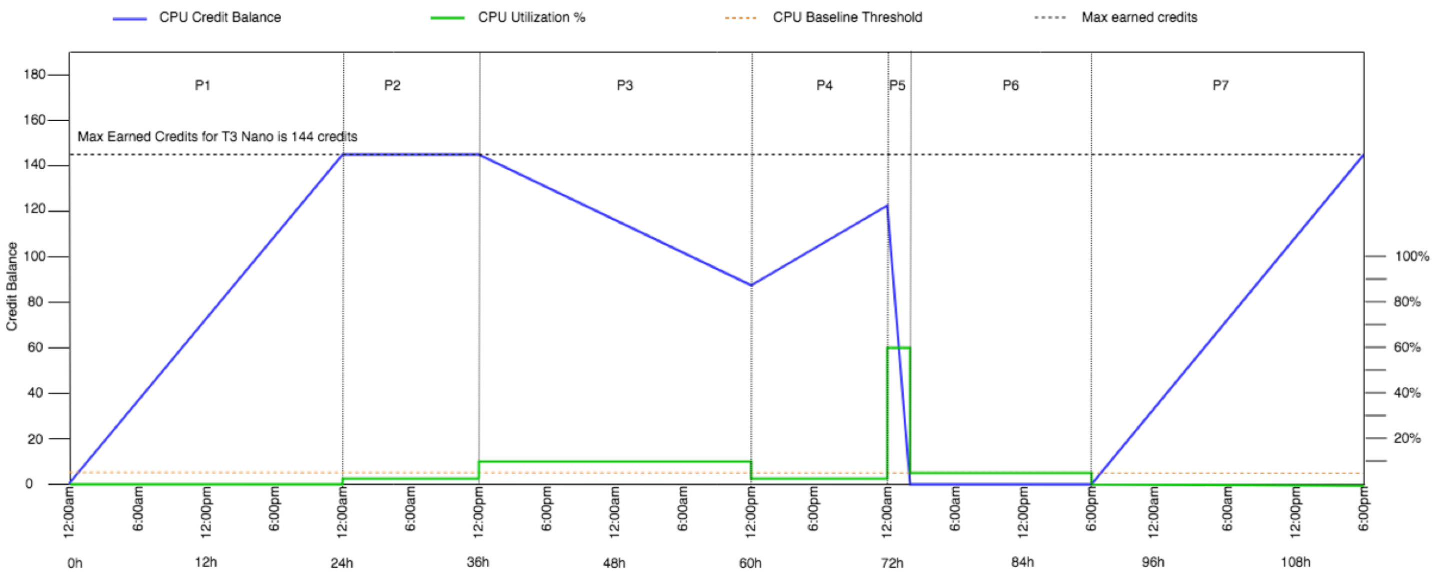
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

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 60% 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.

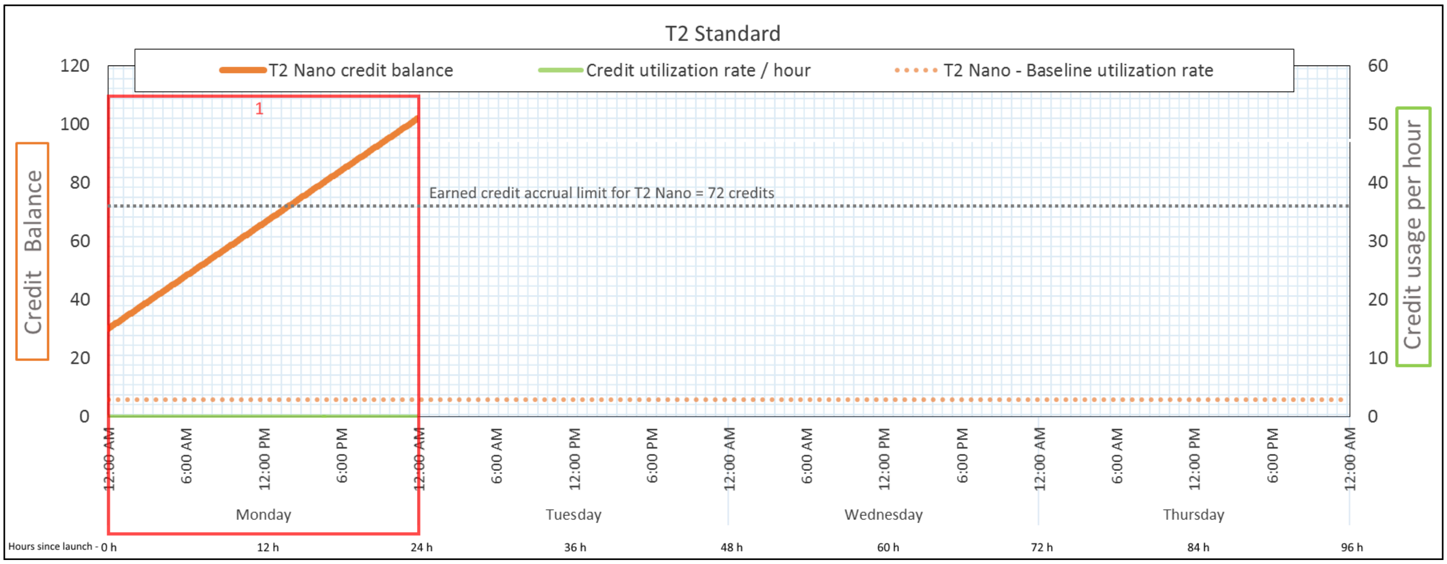
A `t2.nano` instance gets 30 launch credits when it is launched, and earns 72 credits every 24 hours. Its credit balance limit is 72 earned credits; launch credits do not count towards the limit. After the limit is reached, new credits that are earned are discarded. For more information about the number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

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

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

Period 1: 1 – 24 hours

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



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

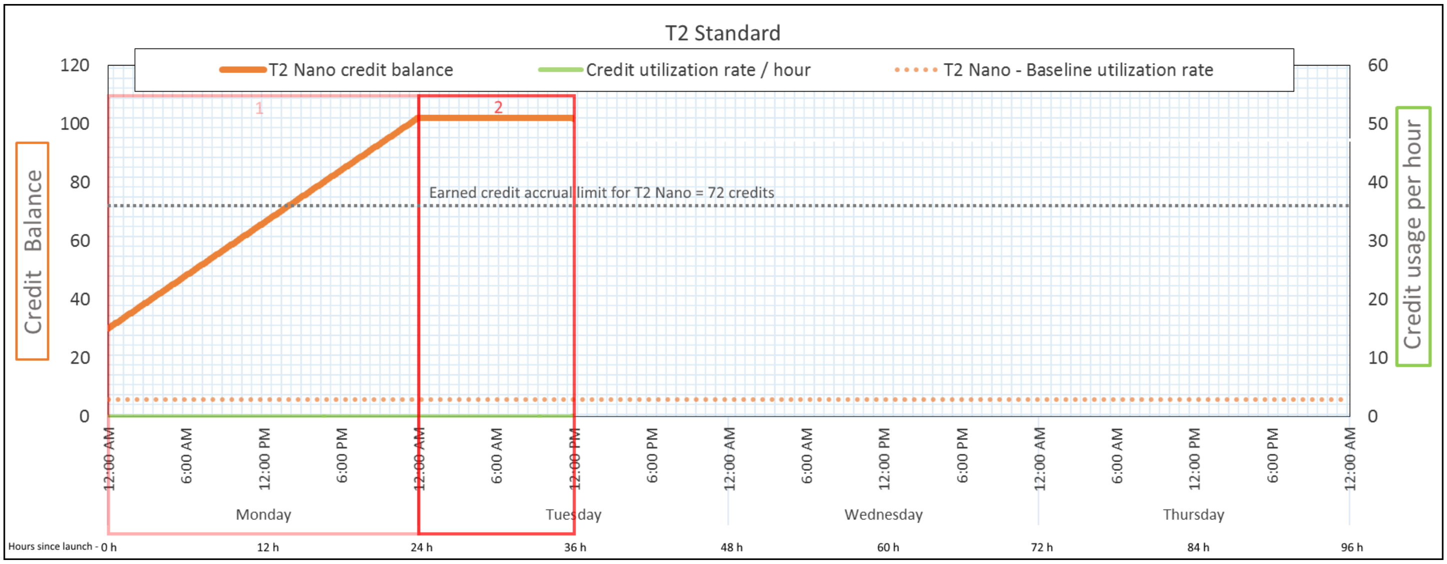
Conclusion

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

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

Period 2: 25 – 36 hours

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



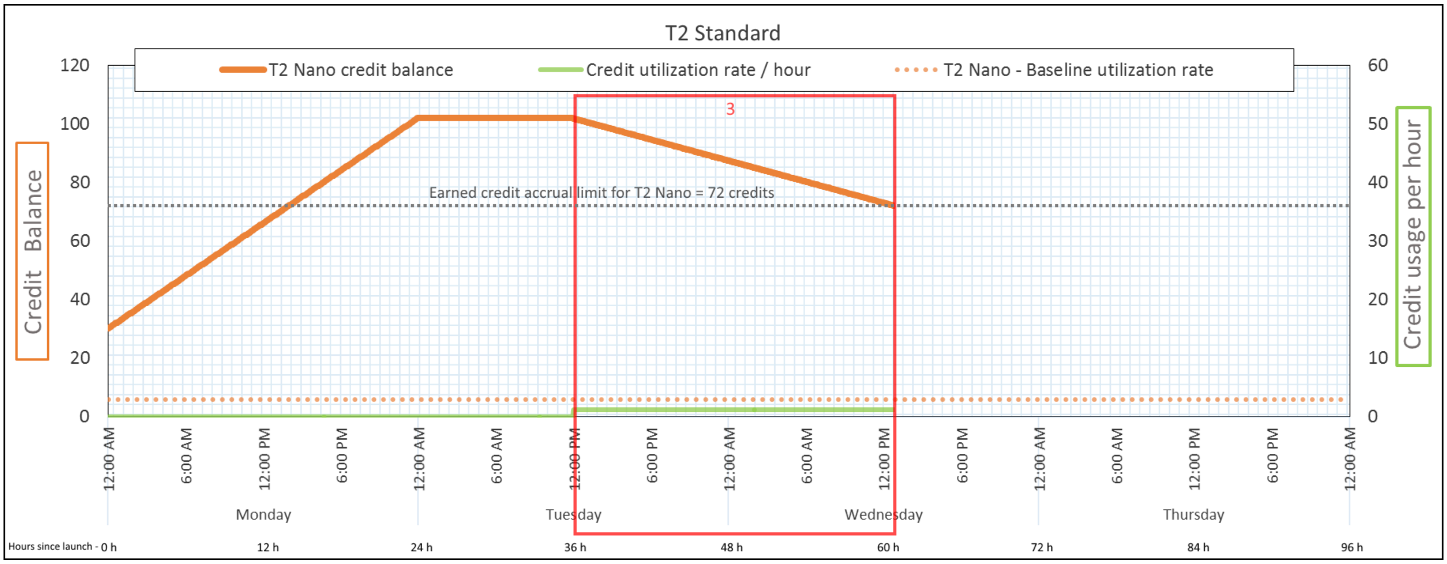
Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours (3 credits per hour)
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	102 credits (30 launch credits + 72 earned credits)—balance is unchanged

Conclusion

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

Period 3: 37 – 61 hours

For the next 25 hours, the instance uses 2% CPU, which requires 30 credits. In the same period, it earns 75 credits, but the credit balance decreases. The balance decreases because the accrued *launch* credits are spent first, while newly earned credits are discarded because the credit balance is already at its limit of 72 earned credits.



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

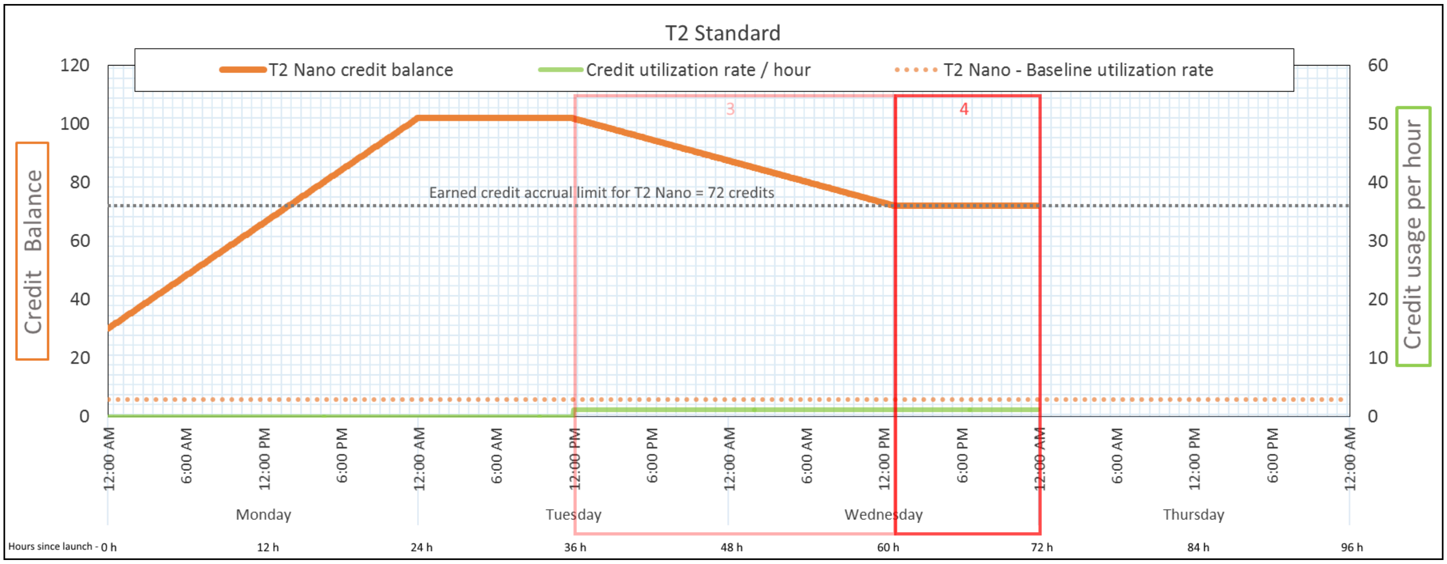
Conclusion

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

Period 4: 62 – 72 hours

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

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



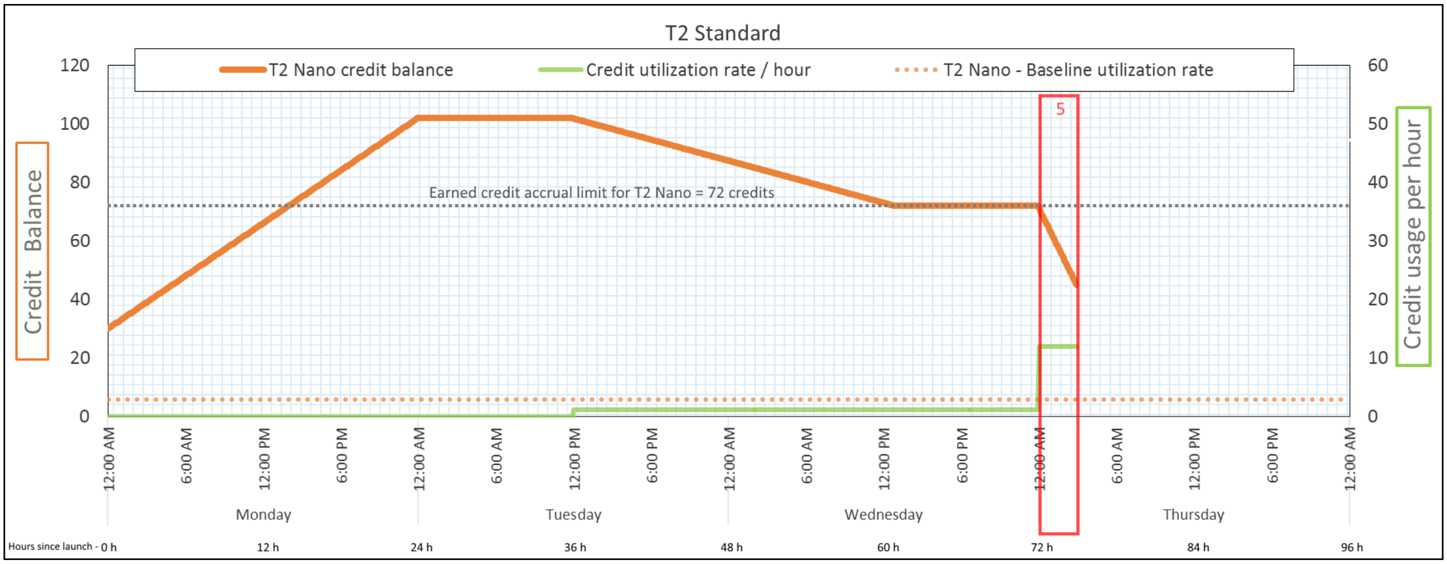
Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—13.2 credits over 11 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	43.2 credits per 24 hours (60% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned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

Conclusion

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

Period 5: 73 – 75 hours

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



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

Conclusion

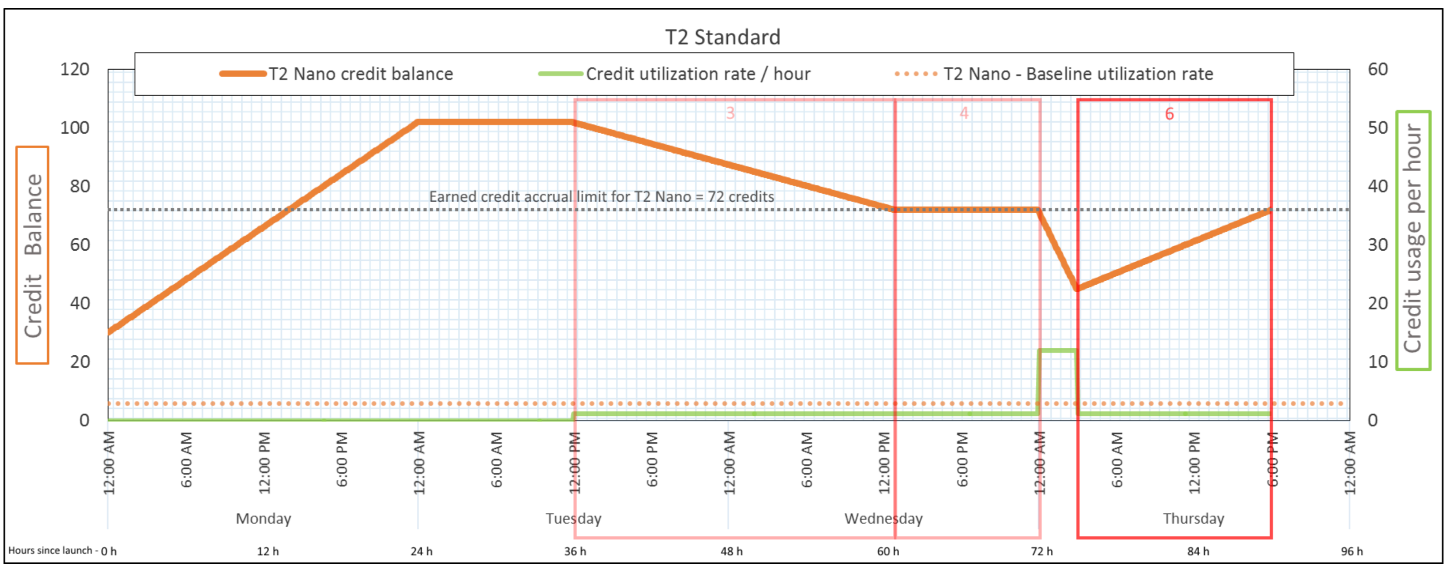
If an instance spends more credits than it earns, its credit balance decreases.

Period 6: 76 – 90 hours

For the next 15 hours, the instance uses 2% CPU, which requires 18 credits. This is the same CPU utilization as in Periods 3 and 4. However, the balance increases in this period, whereas it decreased in Period 3 and plateaued in Period 4.

In Period 3, the accrued launch credits were spent, and any earned credits that exceeded the credit limit were discarded, resulting in a decrease in the credit balance. In Period 4, the instance spent fewer credits than it earned. Any earned credits that exceeded the limit were discarded, so the balance plateaued at its maximum of 72 credits.

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



Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—18 credits over 15 hours
Credit Earn Rate	72 credits per 24 hours (45 credits over 15 hours)
Credit Discard Rate	0 credits per 24 hours

Credit Balance	72 credits (balance increases at a rate of 43.2 credits per 24 hours—change rate = spend rate 28.8/24 + earn rate 72/24)

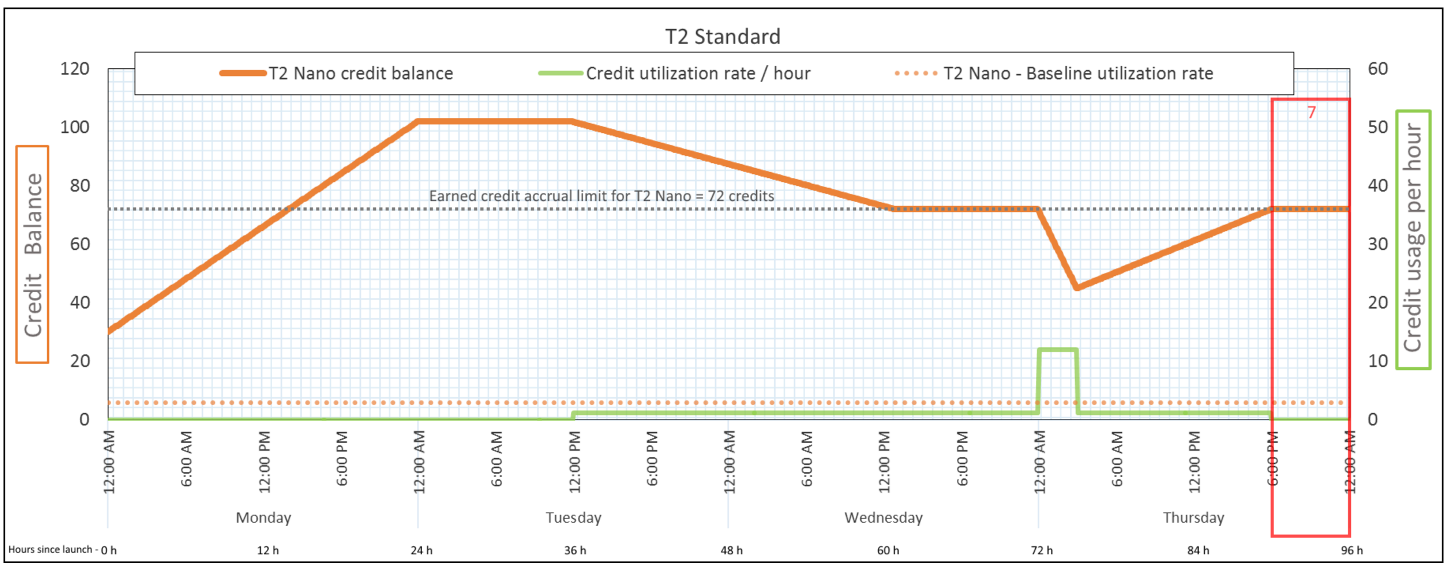
Conclusion

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

Period 7: 91 – 96 hours

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

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



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

Conclusion

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

Configure burstable performance instances

The steps for launching, monitoring, and modifying burstable performance instances (T instances) are similar. The key difference is the default credit specification when they launch.

Each T instance family comes with the following *default credit specification*:

- T4g, T3a, and T3 instances launch as `unlimited`
- T3 instances on a Dedicated Host can only launch as `standard`
- T2 instances launch as `standard`

You can change the default credit specification for the account.

Tasks

- Configure the credit specification at launch
- Configure an Auto Scaling group to set the credit specification as unlimited
- Manage the credit specification of a burstable performance instance
- Manage the default credit specification for an account

Configure the credit specification at launch

You can launch your T instances with a credit specification of `unlimited` or `standard`.

The following procedures describe how to use the EC2 console or the AWS CLI. For information about using an Auto Scaling group, see Configure an Auto Scaling group to set the credit specification as unlimited.

Console

To configure the credit specification of an instance at launch

1. Follow the procedure to launch an instance.

2. Under **Instance type**, select a T instance type.
3. Expand **Advanced details**. For **Credit specification**, select a credit specification.
4. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**.

AWS CLI

To set the credit specification of an instance at launch

Use the run-instances command with the `--credit-specification` option.

```
--credit-specification CpuCredits=unlimited
```

PowerShell

To set the credit specification of an instance at launch

Use the New-EC2Instance cmdlet with the `-CreditSpecification_CpuCredit` parameter.

```
-CreditSpecification_CpuCredit unlimited
```

Configure an Auto Scaling group to set the credit specification as unlimited

When T 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`.

Console

To create a launch template that sets the credit specification

1. Follow the Create a launch template using advanced settings procedure in the *Amazon EC2 Auto Scaling User Guide*.

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**.

AWS CLI

To create a launch template that sets the credit specification

Use the create-launch-template command.

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --version-description FirstVersion \
    --launch-template-data CreditSpecification={CpuCredits=unlimited}
```

PowerShell

To create a launch template that sets the credit specification

Use the New-EC2LaunchTemplate cmdlet. Define the credit specification for the launch template data as follows.

```
$creditSpec = New-Object Amazon.EC2.Model.CreditSpecificationRequest
$creditSpec.CpuCredits = "unlimited"
$launchTemplateData = New-Object Amazon.EC2.Model.RequestLaunchTemplateData
$launchTemplateData.CreditSpecification = $creditSpec
```

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.

Console

To create an Auto Scaling group using a launch template

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 add a launch template to an existing Auto Scaling group

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**.

AWS CLI

To create an Auto Scaling group using a launch template

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

To add a launch template to an existing Auto Scaling group

Use the update-auto-scaling-group command and specify the `--launch-template` parameter.

PowerShell

To create an Auto Scaling group using a launch template

Use the New-ASAutoScalingGroup cmdlet and specify the `-LaunchTemplate_LaunchTemplateId` or `-LaunchTemplate_LaunchTemplateName` parameter.

To add a launch template to an existing Auto Scaling group

Use the Update-ASAutoScalingGroup cmdlet and specify the -LaunchTemplate_LaunchTemplateId or -LaunchTemplate_LaunchTemplateName parameter.

Manage the credit specification of a burstable performance instance

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

Note that in `unlimited` mode, an instance can spend surplus credits, which might incur an additional charge. For more information, see Surplus credits can incur charges.

Console

To manage the credit specification of an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**.
3. (Optional) Select an instance. On its **Details** tab, find **Credit specification**. The value is either `unlimited` or `standard`.
4. (Optional) To modify the credit specification for multiple instances at the same time, select them all.
5. Choose **Actions**, **Instance settings**, **Change credit specification**. This option is enabled only if you selected a T instance.
6. For **Unlimited mode**, select or clear the checkbox next to each instance ID.

AWS CLI

To get the credit specification of an instance

Use the describe-instance-credit-specifications command. If you do not specify an instance ID, all instances with the credit specification of `unlimited` are returned. The output would also include 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.

```
aws ec2 describe-instance-credit-specifications \
```

```
    --instance-id i-1234567890abcdef0 \
    --query InstanceCreditSpecifications[].CpuCredits \
    --output text
```

The following is example output.

```
unlimited
```

To set the credit specification of an instance

Use the modify-instance-credit-specification command.

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

PowerShell

To get the credit specification of an instance

Use the Get-EC2CreditSpecification cmdlet.

```
(Get-EC2CreditSpecification `
    -InstanceId i-1234567890abcdef0).CpuCredits
```

The following is example output.

```
unlimited
```

To set the credit specification of an instance

Use the Edit-EC2InstanceCreditSpecification cmdlet.

```
Edit-EC2InstanceCreditSpecification `
    -Region us-east-1 `
    -InstanceCreditSpecification @({InstanceId="i-1234567890abcdef0"
 CpuCredits="unlimited"})
```

Manage the default credit specification for an account

Each T instance family comes with a default credit specification. You can change the default credit specification for each T instance family at the account level per AWS Region. The valid values for the default credit specification are `unlimited` and `standard`.

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 T 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.

Console

To manage the default credit specification

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the left navigation pane, choose **EC2 Dashboard**.
4. From **Account attributes**, choose **Default credit specification**.
5. Choose **Manage**.
6. For each instance family, choose **Unlimited** or **Standard**, and then choose **Update**.

AWS CLI

To get the default credit specification

Use the get-default-credit-specification command.

```
aws ec2 get-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --query InstanceFamilyCreditSpecifications[].CpuCredits \
    --output text
```

The following is example output.

```
standard
```

To set the default credit specification

Use the modify-default-credit-specification command. The following example sets the value to `unlimited`.

```
aws ec2 modify-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --cpu-credits unlimited
```

PowerShell

To get the default credit specification

Use the Get-EC2DefaultCreditSpecification cmdlet.

```
(Get-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2).CpuCredits
```

The following is example output.

```
standard
```

To set the default credit specification

Use the Edit-EC2DefaultCreditSpecification cmdlet. The following example sets the value to `unlimited`.

```
Edit-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2 `
    -CpuCredit unlimited
```

Monitor CPU credits for burstable instances

EC2 sends metrics to Amazon CloudWatch. You can see the CPU credit metrics in the Amazon EC2 per-instance metrics of the CloudWatch console or by using the AWS CLI to list the metrics for each instance. For more information, see CloudWatch metrics that are available for your instances.

Contents

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 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 CloudWatch metrics that are available for your instances.

Metric	Description
`CPUCreditUsage`	The number of CPU credits spent by the instance for CPU utilization. One CPU credit equals one vCPU running at 100% utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes).

Metric	Description
	CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the `Sum` statistic instead of the `Average` statistic. Units: Credits (vCPU-minutes)
`CPUCreditBalance`	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. 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. The credits in the `CPUCreditBalance` are available for the instance to spend to burst beyond its baseline CPU utilization. When an instance is running, credits in the `CPUCreditBalance` do not expire. When a T4g, T3a or T3 instance stops, the `CPUCreditBalance` value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the `CPUCreditBalance` value does not persist, and all accrued credits are lost. CPU credit metrics are available at a five-minute frequency only. Units: Credits (vCPU-minutes)

Metric	Description
CPUSurplusCreditBalance	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. Units: Credits (vCPU-minutes)
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`. Units: Credits (vCPU-minutes)

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

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

The size of the instance determines the number of credits that the instance can earn per hour and the number of earned credits that it can accrue in the credit balance. For information about the number of credits earned per hour, and the credit balance limit for each instance size, see the credit table.

Example

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

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

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

Example

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

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

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

A value of `0` for `Adjusted balance` indicates that the instance spent all its earned credits for bursting, and no surplus credits were spent. As a result, both `CPUCreditBalance` and `CPUSurplusCreditBalance` are set to `0`.

A positive `Adjusted balance` value indicates that the instance accrued earned credits, and previous surplus credits, if any, were paid down. As a result, the `Adjusted balance` value is assigned to `CPUCreditBalance`, and the `CPUSurplusCreditBalance` is set to `0`. The instance size determines the [maximum number of credits](#) that it can accrue.

Example

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

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

Example

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

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

Example

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

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

Performance acceleration with 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.

Before you can activate or optimize a GPU-based instance, you must install the appropriate drivers, as follows:

- To install NVIDIA drivers on an instance with an attached NVIDIA GPU, such as a P3 or G4dn instance, see NVIDIA drivers.
- To install AMD drivers on an instance with an attached AMD GPU, such as a G4ad instance, see AMD drivers.

Contents

Activate NVIDIA GRID Virtual Applications on your Amazon EC2 GPU-based instances

To activate the GRID Virtual Applications on GPU-based instances that have NVIDIA GPUs (NVIDIA GRID Virtual Workstation is enabled by default), you must define the product type for the driver. The process that you use depends on the operating system of your instance.

Linux instances

To activate GRID Virtual Applications on your Linux instances

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

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. Open the `/etc/nvidia/gridd.conf` file in your favorite text editor.
3. Find the `FeatureType` line, and set it equal to `0`. Then add a line with `IgnoreSP=TRUE`.

```
FeatureType=0 IgnoreSP=TRUE
```

4. Save the file and exit.
5. Reboot the instance to pick up the new configuration.

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

Windows instances

To activate GRID Virtual Applications on your 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 on Amazon EC2 instances

There are several GPU setting optimizations that you can perform to achieve the best performance on NVIDIA GPU instances. With some of these instance types, the NVIDIA driver uses an autoboost feature, which varies the GPU clock speeds. By disabling autoboost and setting the GPU clock speeds to their maximum frequency, you can consistently achieve the maximum performance with your GPU instances.

Optimize GPU settings on Linux

1. Configure the GPU settings to be persistent. This command can take several minutes to run.

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. [G3, and P2 instances only] Disable the autoboost feature for all GPUs on the instance.

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

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
```

   - G4dn instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
```

   - G5 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 6250,1710
```

- G6 and Gr6 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 6251,2040
```

- G6e instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 9001,2520
```

- P2 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,875
```

- P3 and P3dn instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 877,1530
```

- P4d instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 1215,1410
```

- P4de instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 1593,1410
```

- P5 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2619,1980
```

Optimize GPU settings on Windows

1. Open a PowerShell window and navigate to the NVIDIA installation folder.

```
PS C:\> cd "C:\Windows\System32\DriverStore\FileRepository\nvgridsw_aws.inf_*\"
```

2. [G3, and P2 instances only] Disable the autoboost feature for all GPUs on the instance.

```
PS C:\> .\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:

     ```
     PS C:\> .\nvidia-smi -ac "2505,1177"
     ```

   - G4dn instances:

     ```
     PS C:\> .\nvidia-smi -ac "5001,1590"
     ```

   - G5 instances:

     ```
     PS C:\> .\nvidia-smi -ac "6250,1710"
     ```

   - G6 and Gr6 instances:

     ```
     PS C:\> .\nvidia-smi -ac "6251,2040"
     ```

   - G6e instances:

     ```
     PS C:\> .\nvidia-smi -ac "9001,2520"
     ```

   - P2 instances:

     ```
     PS C:\> .\nvidia-smi -ac "2505,875"
     ```

   - P3 and P3dn instances:

     ```
     PS C:\> .\nvidia-smi -ac "877,1530"
     ```

Set up Dual 4K displays on G4ad Linux instances

After you launch a G4ad instance, you can set up dual 4K displays.

To install the AMD drivers and configure dual screens

1. Connect to your Linux instance to get the PCI Bus address of the GPU you want to target for dual 4K (2x4k):

```
lspci -vv | grep -i amd
```

   You will get output similar to the following:

```
00:1e.0 Display controller: Advanced Micro Devices, Inc. [*AMD*/ATI] Device 7362 (rev
 c3)
Subsystem: Advanced Micro Devices, Inc. [AMD/ATI] Device 0a34
```

2. Note the PCI bus address is 00:1e.0 in the above output. Create a file named `/etc/modprobe.d/amdgpu.conf` and add:

```
options amdgpu virtual_display=0000:00:1e.0,2
```

3. To install the AMD drivers on Linux, see AMD drivers for your EC2 instance. If you already have the AMD GPU driver installed, you will need to rebuild the amdgpu kernel modules through dkms.

4. Use the below xorg.conf file to define the dual (2x4K) screen topology and save the file in `/etc/X11/xorg.conf`:

```
~$ cat /etc/X11/xorg.conf
Section "ServerLayout"
    Identifier     "Layout0"
    Screen          0 "Screen0"
    Screen        1 "Screen1"
    InputDevice     "Keyboard0" "CoreKeyboard"
    InputDevice     "Mouse0" "CorePointer"
    Option          "Xinerama" "1"
EndSection
Section "Files"
    ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
    ModulePath "/opt/amdgpu/lib/xorg/modules"
    ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
    ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
    ModulePath "/usr/lib64/xorg/modules"
    ModulePath "/usr/lib/xorg/modules"
EndSection
```

```
Section "InputDevice"
    # generated from default
    Identifier     "Mouse0"
    Driver         "mouse"
    Option         "Protocol" "auto"
    Option         "Device" "/dev/psaux"
    Option         "Emulate3Buttons" "no"
    Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
    # generated from default
    Identifier     "Keyboard0"
    Driver         "kbd"
EndSection

Section "Monitor"
    Identifier     "Virtual"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "Primary" "true"
EndSection

Section "Monitor"
    Identifier     "Virtual-1"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "RightOf" "Virtual"
EndSection

Section "Device"
    Identifier     "Device0"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection

Section "Device"
    Identifier     "Device1"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection
```

```
Section "Extensions"
    Option          "DPMS" "Disable"
EndSection

Section "Screen"
    Identifier      "Screen0"
    Device          "Device0"
    Monitor         "Virtual"
    DefaultDepth    24
    Option          "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual     3840 2160
        Depth       32
    EndSubSection
EndSection

Section "Screen"
    Identifier      "Screen1"
    Device          "Device1"
    Monitor         "Virtual"
    DefaultDepth    24
    Option          "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual     3840 2160
        Depth       32
    EndSubSection
EndSection
```

5. Set up DCV by following the instructions in setting up an interactive desktop.
6. After the DCV set up is complete, reboot.
7. Confirm that the driver is functional:

   ```
   dmesg | grep amdgpu
   ```

   The response should look like the following:

   ```
   Initialized amdgpu
   ```

8. You should see in the output for `DISPLAY=:0 xrandr -q` that you have 2 virtual displays connected:

```
~$ DISPLAY=:0 xrandr -q
Screen 0: minimum 320 x 200, current 3840 x 1080, maximum 16384 x 16384
Virtual connected primary 1920x1080+0+0 (normal left inverted right x axis y axis)
 0mm x 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
 1280x800 59.96
 1280x720 59.97
 1152x768 59.95
 1024x768 60.00 59.95
 800x600  60.32 59.96 56.25
 848x480  60.00 59.94
 720x480  59.94
 640x480  59.94 59.94
Virtual-1 connected 1920x1080+1920+0 (normal left inverted right x axis y axis) 0mm x
 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
 1280x800 59.96
 1280x720 59.97
 1152x768 59.95
```

```
 1024x768 60.00 59.95
 800x600  60.32 59.96 56.25
 848x480  60.00 59.94
 720x480  59.94
640x480  59.94 59.94
```

9. When you connect into DCV, change the resolution to 2x4K, confirming the dual monitor support is registered by DCV.

Set up an interactive desktop for Linux

After you confirm that your Linux instance has the AMD GPU driver installed and amdgpu is in use, you can install an interactive desktop manager. We recommend the MATE desktop environment for the best compatibility and performance.

Prerequisite

Open a text editor and save the following as a file named `xorg.conf`. You'll need this file on your instance.

```
Section "ServerLayout"
Identifier     "Layout0"
Screen          0 "Screen0"
InputDevice     "Keyboard0" "CoreKeyboard"
InputDevice     "Mouse0" "CorePointer"
EndSection
Section "Files"
ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
ModulePath "/opt/amdgpu/lib/xorg/modules"
ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
ModulePath "/usr/lib64/xorg/modules"
ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Mouse0"
Driver         "mouse"
Option         "Protocol" "auto"
```

```
Option          "Device" "/dev/psaux"
Option          "Emulate3Buttons" "no"
Option          "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Keyboard0"
Driver          "kbd"
EndSection
Section "Monitor"
Identifier     "Monitor0"
VendorName     "Unknown"
ModelName      "Unknown"
EndSection
Section "Device"
Identifier     "Device0"
Driver          "amdgpu"
VendorName     "AMD"
BoardName      "Radeon MxGPU V520"
BusID           "PCI:0:30:0"
EndSection
Section "Extensions"
Option          "DPMS" "Disable"
EndSection
Section "Screen"
Identifier     "Screen0"
Device          "Device0"
Monitor         "Monitor0"
DefaultDepth   24
Option          "AllowEmptyInitialConfiguration" "True"
SubSection "Display"
    Virtual    3840 2160
    Depth      32
EndSubSection
EndSection
```

To set up an interactive desktop on Amazon Linux 2

1. Install the EPEL repository.

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install epel -y
   ```

2. Install the MATE desktop.

```
[ec2-user ~]$ sudo amazon-linux-extras install mate-desktop1.x -y
[ec2-user ~]$ sudo yum groupinstall "MATE Desktop" -y
[ec2-user ~]$ sudo systemctl disable firewalld
```

3. Copy the `xorg.conf` file to `/etc/X11/xorg.conf`.
4. Reboot the instance.

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

5. (Optional) Install the Amazon DCV server to use Amazon DCV as a high-performance display protocol, and then connect to a Amazon DCV session using your preferred client.

To set up an interactive desktop on Ubuntu

1. Install the MATE desktop.

```
$ sudo apt install xorg-dev ubuntu-mate-desktop -y
$ sudo apt purge ifupdown -y
```

2. Copy the `xorg.conf` file to `/etc/X11/xorg.conf`.
3. Reboot the instance.

```
$ sudo reboot
```

4. Install the AMF encoder for the appropriate version of Ubuntu.

```
$ sudo apt install ./amdgpu-pro-20.20-*/amf-amdgpu-pro_20.20-*_amd64.deb
```

5. (Optional) Install the Amazon DCV server to use Amazon DCV as a high-performance display protocol, and then connect to a Amazon DCV session using your preferred client.
6. After the DCV installation give the DCV User video permissions:

```
$ sudo usermod -aG video dcv
```

To set up an interactive desktop on CentOS

1. Install the EPEL repository.

```
$ sudo yum update -y
$ sudo yum install epel-release -y
```

2. Install the MATE desktop.

```
$ sudo yum groupinstall "MATE Desktop" -y
$ sudo systemctl disable firewalld
```

3. Copy the `xorg.conf` file to `/etc/X11/xorg.conf`.
4. Reboot the instance.

```
$ sudo reboot
```

5. (Optional) Install the Amazon DCV server to use Amazon DCV as a high-performance display protocol, and then connect to a Amazon DCV session using your preferred client.

Get started with GPU accelerated instances

The latest generation of GPU accelerated instance types, such as those shown in the following list deliver the highest performance capabilities for deep learning and high performance computing (HPC) applications. Select the instance type link to learn more about its capabilities.

- P6 family
- P5 family

For a complete list of instance type specifications for accelerated instance types, see Accelerated computing in the *Amazon EC2 Instance Types* reference.

Software configuration

The easiest way to get started with the latest generation GPU accelerated instance types is to launch an instance from an AWS Deep Learning AMI that's preconfigured with all of the required software. For the latest AWS Deep Learning AMIs for use with GPU accelerated instance types, see P6 Supported DLAMI in the *AWS Deep Learning AMIs Developer Guide*.

If you need to build a custom AMI to launch instances that host deep learning or HPC applications, we recommend that you install the following minimum software versions on top of your base image.

Instance type	NVIDIA driver	CUDA	NVIDIA GDRCopy	EFA installer	NCCL	EFA K8s [1]
P5	530	12.1	2.3	1.24.1	2.18.3	0.4.4
P5.4xlarge	530	12.1	2.3	1.43.1 [2]	2.18.3	0.4.4
P5e	550	12.1	2.3	1.24.1	2.18.3	0.5.5
P5en	550	12.1	2.3	1.24.1	2.18.3	0.5.6
P6-B200	570	12.8	2.5	1.4.1	2.26.2-1	0.5.10

[1] The **EFA K8s** column contains the minimum recommended version for `aws-efa-k8s-device-plugin`.

[2] There is compatibility issue that affects `P5.4xlarge` instances when GPU-to-GPU communication uses Elastic Fabric Adapter (EFA) and the NVIDIA Collective Communications Library (NCCL). To mitigate the issue, set the environment variable `FI_HMEM_DISABLE_P2P` to `1`, and ensure that you install EFA version 1.43.1 or newer.

> **Note**
>
> If you use version 1.4.1 of the EFA installer, the `aws-ofi-nccl plugin` comes with it. For earlier versions of the EFA installer, use `aws-ofi-nccl plugin` version `1.7.2-aws` or later.

We also recommend that you configure the instance to not use deeper C-states. For more information, see High performance and low latency by limiting deeper C-states in the *Amazon Linux 2 User Guide*. The latest AWS Deep Learning Base GPU AMIs are preconfigured to not use deeper C-states.

For networking and Elastic Fabric Adapter (EFA) configuration see Maximize network bandwidth on Amazon EC2 instances with multiple network cards.

Amazon EC2 Mac instances

EC2 Mac instances are ideal for developing, building, testing, and signing applications for Apple platforms, such as iPhone, iPad, Mac, Vision Pro, Apple Watch, Apple TV, and Safari. You can connect to your Mac instance using SSH or Apple Remote Desktop (ARD).

> **Note**
>
> The **unit of billing** is the **dedicated host**. The instances running on that host have no additional charge.

Amazon EC2 Mac instances natively support the macOS operating system.

- EC2 x86 Mac instances (`mac1.metal`) are built on 2018 Mac mini hardware powered by 3.2 GHz Intel eighth-generation (Coffee Lake) Core i7 processors.
- EC2 M1 Mac instances (`mac2.metal`) are built on 2020 Mac mini hardware powered by Apple silicon M1 processors.
- EC2 M1 Ultra Mac instances (`mac2-m1ultra.metal`) are built on 2022 Mac Studio hardware powered by Apple silicon M1 Ultra processors.
- EC2 M2 Mac instances (`mac2-m2.metal`) are built on 2023 Mac mini hardware powered by Apple silicon M2 processors.
- EC2 M2 Pro Mac instances (`mac2-m2pro.metal`) are built on 2023 Mac mini hardware powered by Apple silicon M2 Pro processors.

Amazon EC2 Mac Dedicated Hosts support Dedicated Host auto recovery and reboot-based host maintenance.

Contents

- Launch a Mac instance using the AWS Management Console or the AWS CLI
- Connect to your Mac instance using SSH or a GUI
- Update the operating system and software on Amazon EC2 Mac instances
- Increase the size of an EBS volume on your Mac instance
- Stop or terminate your Amazon EC2 Mac instance
- Configure System Integrity Protection for Amazon EC2 Mac instances
- Find supported macOS versions for your Amazon EC2 Mac Dedicated Host
- Subscribe to macOS AMI notifications
- Retrieve macOS AMI IDs using AWS Systems Manager Parameter Store API
- Amazon EC2 macOS AMIs release notes

Considerations

The following considerations apply to Mac instances:

- Mac instances are available only as bare metal instances on Dedicated Hosts, with a minimum allocation period of 24 hours before you can release the Dedicated Host. You can launch one Mac instance per Dedicated Host. You can share the Dedicated Host with the AWS accounts or organizational units within your AWS organization, or the entire AWS organization.
- Mac instances are available in different AWS Regions. For a list of Mac instance availability in each of the AWS Regions, see Amazon EC2 instance types by Region.
- Mac instances are available only as On-Demand Instances. They are not available as Spot Instances or Reserved Instances. You can save money on Mac instances by purchasing a Savings Plan.
- The compatibility of different Mac instance types with specific macOS Amazon Machine Images (AMIs) varies. For more information, see Amazon EC2 macOS AMIs release notes.
- EBS hotplug is supported.
- AWS does not manage or support the internal SSD on the Apple hardware. We strongly recommend that you use Amazon EBS volumes instead. EBS volumes provide the same elasticity, availability, and durability benefits on Mac instances as they do on any other EC2 instance.
- We recommend using an Amazon EBS volume with 10,000 IOPS and 400 MiB/s throughput with Mac instances for optimal performance. For more information, see Amazon EBS volume types in the *Amazon EBS User Guide*.

- Mac instances support Amazon EC2 Auto Scaling.
- On x86 Mac instances, automatic software updates are disabled. We recommend that you apply updates and test them on your instance before you put the instance into production. For more information, see Update the operating system and software on Amazon EC2 Mac instances.
- When you stop or terminate a Mac instance, a scrubbing workflow is performed on the Dedicated Host. For more information, see Stop or terminate your Amazon EC2 Mac instance.
- **Important**

  Apple Intelligence features are not available when booting Mac hardware from an external volume. As EC2 Mac instances boot from external EBS volumes by default, they do not support Apple Intelligence features.
- **Warning**

  Do not use FileVault. If you enable FileVault, the host fails to boot because the partitions are locked. If data encryption is required, use Amazon EBS encryption to avoid boot issues and performance impact. With Amazon EBS encryption, encryption operations occur on the host servers, ensuring the security of both data-at-rest and data-in-transit between the instances and their attached EBS storage. For more information, see Amazon EBS encryption in the *Amazon EBS User Guide*.

Instance readiness

After you launch a Mac instance, you'll need to wait until the instance is ready before you can connect to it. For an AWS vended AMI with a x86 Mac instance or a Apple silicon Mac instance, the launch time can range from approximately 6 minutes to 20 minutes. Depending on the chosen Amazon EBS volume sizes, the inclusion of additional scripts to *user data*, or additional loaded software on a custom macOS AMI, the launch time might increase.

You can use a small shell script, like the one below, to poll the describe-instance-status API to know when the instance is ready to be connected to. In the following command, replace the example instance ID with your own.

```
for i in $(seq 1 200); do aws ec2 describe-instance-status --instance-ids=i-1234567890abcdef0 \
    --query='InstanceStatuses[0].InstanceStatus.Status'; sleep 5; done;
```

EC2 macOS AMIs

Amazon EC2 macOS is designed to provide a stable, secure, and high-performance environment for developer workloads running on Amazon EC2 Mac instances. EC2 macOS AMIs includes packages that enable easy integration with AWS, such as launch configuration tools and popular AWS libraries and tools.

For more information about EC2 macOS AMIs, see Amazon EC2 macOS AMIs release notes.

AWS provides updated EC2 macOS AMIs on a regular basis that include updates to packages owned by AWS and the latest fully-tested macOS version. Additionally, AWS provides updated AMIs with the latest minor version updates or major version updates as soon as they can be fully tested and vetted. If you do not need to preserve data or customizations to your Mac instances, you can get the latest updates by launching a new instance using the current AMI and then terminating the previous instance. Otherwise, you can choose which updates to apply to your Mac instances.

For information about how to subscribe to macOS AMI notifications, see Subscribe to macOS AMI notifications.

EC2 macOS Init

EC2 macOS Init is used to initialize EC2 Mac instances at launch. It uses priority groups to run logical groups of tasks at the same time.

The launchd plist file is `/Library/LaunchDaemons/com.amazon.ec2.macos-init.plist`. The files for EC2 macOS Init are located in `/usr/local/aws/ec2-macos-init`.

For more information, see https://github.com/aws/ec2-macos-init.

Amazon EC2 System Monitor for macOS

Amazon EC2 System Monitor for macOS provides CPU utilization metrics to Amazon CloudWatch. It sends these metrics to CloudWatch over a custom serial device in 1-minute periods. You can enable or disable this agent as follows. It is enabled by default.

```
sudo setup-ec2monitoring [enable | disable]
```

> **Note**
>
> Amazon EC2 System Monitor for macOS is not currently supported on Apple silicon Mac instances.

Related resources

For information about pricing, see Pricing.

For more information about Mac instances, see Amazon EC2 Mac Instances.

For more information about hardware specifications and network performance of Mac instances, see General purpose instances.

Launch a Mac instance using the AWS Management Console or the AWS CLI

EC2 Mac instances require a Dedicated Host. You first need to allocate a host to your account, and then launch the instance onto the host.

You can launch a Mac instance using the AWS Management Console or the AWS CLI.

Launch a Mac instance using the console

To launch a Mac instance onto a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Allocate the Dedicated Host, as follows:
   a. In the navigation pane, choose **Dedicated Hosts**.
   b. Choose **Allocate Dedicated Host** and then do the following:
      i. For **Instance family**, choose **mac1**, **mac2**, **mac2-m2**, **mac2-m2pro**, or **mac2-m1ultra**. If the instance family doesn't appear in the list, it's not supported in the currently selected Region.
      ii. For **Instance type**, choose **mac1.metal**, **mac2.metal**, **mac2-m2.metal**, **mac2-m2pro.metal**, or **mac2-m1ultra.metal** based on the instance family chosen.
      iii. For **Availability Zone**, choose the Availability Zone for the Dedicated Host.
      iv. For **Quantity**, keep **1**.
      v. Choose **Allocate**.

3. Launch the instance on the host, as follows:

   a. Select the Dedicated Host that you created and then do the following:

      i. Choose **Actions**, **Launch instance(s) onto host**.
      ii. Under **Application and OS Images (Amazon Machine Image)**, select a macOS AMI.
      iii. Under **Instance type**, select the appropriate instance type (**mac1.metal**, **mac2.metal**, **mac2-m2.metal**, **mac2-m2pro.metal**, or **mac2-m1ultra.metal**).
      iv. Under **Advanced details**, verify that **Tenancy**, **Tenancy host by**, and **Tenancy host ID** are preconfigured based on the Dedicated Host you created. Update **Tenancy affinity** as needed.
      v. Complete the wizard, specifying EBS volumes, security groups, and key pairs as needed.
      vi. In the **Summary** panel, choose **Launch instance**.

   b. A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console. The initial state of an instance is `pending`. The instance is ready when its state changes to `running` and it passes status checks.

Launch a Mac instance using the AWS CLI

Allocate the Dedicated Host

Use the following [allocate-hosts](#) command to allocate a Dedicated Host for your Mac instance, replacing the `instance-type` with either `mac1.metal`, `mac2.metal`, `mac2-m2.metal`, `mac2-m2pro.metal`, or `mac2-m1ultra.metal`, and the `region` and `availability-zone` with the appropriate ones for your environment.

```
aws ec2 allocate-hosts --region us-east-1 --instance-type mac1.metal --availability-zone us-east-1b --auto-placement "on" --quantity 1
```

Launch the instance on the host

Use the following [run-instances](#) command to launch a Mac instance, again replacing the `instance-type` with either `mac1.metal`, `mac2.metal`, `mac2-m2.metal`, `mac2-m2pro.metal`, or `mac2-m1ultra.metal`, and the `region` and `availability-zone` with the ones used previously.

```
aws ec2 run-instances --region us-east-1 --instance-type mac1.metal --placement
 Tenancy=host --image-id ami_id --key-name my-key-pair
```

The initial state of an instance is `pending`. The instance is ready when its state changes to `running` and it passes status checks. Use the following [describe-instance-status](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-status.html) command to display status information for your instance.

```
aws ec2 describe-instance-status --instance-ids i-017f8354e2dc69c4f
```

The following is example output for an instance that is running and has passed status checks.

```
{
    "InstanceStatuses": [
        {
            "AvailabilityZone": "us-east-1b",
            "InstanceId": "i-017f8354e2dc69c4f",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "InstanceStatus": {
                "Details": [
                    {
                        "Name": "reachability",
                        "Status": "passed"
                    }
                ],
                "Status": "ok"
            },
            "SystemStatus": {
                "Details": [
                    {
                        "Name": "reachability",
                        "Status": "passed"
                    }
                ],
                "Status": "ok"
            }
        }
    ]
}
```

Connect to your Mac instance using SSH or a GUI

You can connect to your Mac instance using SSH or a graphical user interface (GUI).

Multiple users can access the OS simultaneously. Typically there is a 1:1 user:GUI session, due to the built-in Screen Sharing service on port 5900. Using SSH within macOS supports multiple sessions up until the "Max Sessions" limit in the `sshd_config` file.

Connect to your instance using SSH

Amazon EC2 Mac instances do not allow remote root SSH by default. The ec2-user account is configured to log in remotely using SSH. The ec2-user account also has **sudo** privileges. After you connect to your instance, you can add other users.

To support connecting to your instance using SSH, launch the instance using a key pair and a security group that allows SSH access, and ensure that the instance has internet connectivity. You provide the `.pem` file for the key pair when you connect to the instance.

Use the following procedure to connect to your Mac instance using an SSH client. If you receive an error while attempting to connect to your instance, see Troubleshoot issues connecting to your Amazon EC2 Linux instance.

To connect to your instance using SSH

1. Verify that your local computer has an SSH client installed by entering **ssh** at the command line. If your computer doesn't recognize the command, search for an SSH client for your operating system and install it.
2. Get the public DNS name of your instance. Using the Amazon EC2 console, you can find the public DNS name on both the **Details** and the **Networking** tabs. Using the AWS CLI, you can find the public DNS name using the describe-instances command.
3. Locate the `.pem` file for the key pair that you specified when you launched the instance.
4. Connect to your instance using the following **ssh** command, specifying the public DNS name of the instance and the `.pem` file.

```
ssh -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

Password authentication is disabled to prevent brute-force password attacks. Before you make changes to the SSH configuration, open `/usr/local/aws/ec2-macos-init/init.toml` and set `secureSSHDConfig` to `false`.

Connect to your instance's graphical user interface (GUI)

Use the following procedure to connect to your instance's GUI using VNC, Apple Remote Desktop (ARD), or the Apple Screen Sharing application (included with macOS).

> **Note**
>
> macOS 10.14 and later only allows control if Screen Sharing is enabled through System Preferences.

To connect to your instance using ARD client or VNC client

1. Verify that your local computer has an ARD client or a VNC client that supports ARD installed. On macOS, you can leverage the built-in Screen Sharing application. Otherwise, search for ARD for your operating system and install it.
2. From your local computer, connect to your instance using SSH.
3. Set up a password for the ec2-user account using the **passwd** command as follows.

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

4. Install and start macOS Screen Sharing using the following command.

```
[ec2-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/com.apple.screensharing.plist
```

5. Disconnect from your instance by typing **exit** and pressing Enter.
6. From your computer, connect to your instance using the following **ssh** command. In addition to the options shown in the previous section, use the **-L** option to enable port forwarding and forward all traffic on local port 5900 to the ARD server on the instance.

```
ssh -L 5900:localhost:5900 -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

7. From your local computer, use the ARD client or VNC client that supports ARD to connect to `localhost:5900`. For example, use the Screen Sharing application on macOS as follows:

a. Open **Finder** and select **Go**.

b. Select **Connect to Server**.

c. In the **Server Address** field, enter `vnc://localhost:5900`.

d. Log in as prompted, using **`ec2-user`** as the username and the password that you created for the ec2-user account.

Modify macOS screen resolution on Mac instances

After you connect to your EC2 Mac instance using ARD or a VNC client that supports ARD, you can modify the screen resolution of your macOS environment using any of the publicly available macOS tools or utilities, such as displayplacer.

To modify the screen resolution using displayplacer

1. Install displayplacer.

```
[ec2-user ~]$ brew tap jakehilborn/jakehilborn && brew install displayplacer
```

2. Show the current screen information and possible screen resolutions.

```
[ec2-user ~]$ displayplacer list
```

3. Apply the desired screen resolution.

```
[ec2-user ~]$ displayplacer "id:<screenID> res:<width>x<height> origin:(0,0)
 degree:0"
```

For example:

```
RES="2560x1600"
displayplacer "id:69784AF1-CD7D-B79B-E5D4-60D937407F68 res:${RES} scaling:off
 origin:(0,0) degree:0"
```

Update the operating system and software on Amazon EC2 Mac instances

The following topic explains how to update the operating system and software on Apple silicon Mac instances (Mac2, Mac2-m1ultra, Mac2-m2, Mac2-m2pro) and x86 Mac instances (Mac1).

> **Warning**
>
> Installation of beta or preview macOS versions is only available on Apple silicon Mac instances. Amazon EC2 doesn't qualify beta or preview macOS versions and doesn't ensure instances will remain functional after an update to a pre-production macOS version. Attempting to install beta or preview macOS versions on Amazon EC2 x86 Mac instances will lead to degradation of your Amazon EC2 Mac Dedicated Host when you stop or terminate your instances, and will prevent you from starting or launching a new instance on that host.

> **Note**
>
> If you perform an in-place macOS update before AWS releases an official AMI, the update applies to the selected host only. If you have other hosts, or if you launch new hosts, you must perform the same update process on those hosts as well. Each macOS version requires a minimum firmware version on the underlying Apple Mac hardware. The in-place update only updates the firmware on the selected host and doesn't transfer to other existing or new hosts. To check which macOS versions are compatible with your Amazon EC2 Mac Dedicated Host, see Find supported macOS versions for your Amazon EC2 Mac Dedicated Host.

Update software on Apple silicon Mac instances

Prerequisites

Due to an update in the network driver configuration, ENA driver version 1.0.2 isn't compatible with macOS 13.3 and later. If you want to install any beta, preview, or production macOS version 13.3 or later and have not installed the latest ENA driver, use the following procedure to install a new version of the driver.

To install a new version of the ENA driver

1. In a Terminal window, connect to your Apple silicon Mac instance using SSH.
2. Update Homebrew and download the ENA application into the `Applications` file using the following command.

   ```
   [ec2-user ~]$ brew update
   ```

   ```
   [ec2-user ~]$ brew install amazon-ena-ethernet-dext
   ```

3. Disconnect from your instance by typing **exit** and pressing return.
4. Use the VNC client to activate the ENA application.

   a. Setup the VNC client using Connect to your instance's graphical user interface (GUI).

   b. Once you have connected to your instance using the Screen Sharing application, go to the **Applications** folder and open the ENA application.

   c. Choose **Activate**

   d. To confirm the driver was activated correctly, run the following command in the Terminal window. The output of the command shows that the old driver is in the terminating state and the new driver is in the activated state.

   ```
   systemextensionsctl list;
   ```

   e. After you restart the instance, only the new driver will be present.

Perform the software update

On Apple silicon Mac instances, you must complete several steps to perform an in-place operating system update. This includes delegating ownership of the Amazon EBS root volume to the EBS root volume administrative user. You can choose to do this either automatically using an Amazon EC2 API, or you can do it manually by running the commands on your instance.

Automated volume ownership delegation (Recommended)

Considerations

- It can take between 30 and 90 minutes for the volume ownership delegation task to complete. During this time, the instance is unreachable.

- The following macOS versions are supported:
  - **Mac2 | Mac2-m1ultra** — macOS Ventura (version 13.0 or later)
  - **Mac2-m2 | Mac2-m2pro** — macOS Ventura (version 13.2 or later)
- Instances must have only one bootable volume, and each attached volume can have only one additional admin user.

Step 1: Set a password and enable the secure token for the EBS root volume administrative user

You must set a password and enable the secure token for the Amazon EBS root volume administrative user (`ec2-user`).

> **Note**
>
> The password and secure token are set the first time you connect to an Apple silicon Mac instance using the GUI. If you previously connected to the instance using the GUI, you **do not** need to perform these steps.

1. Connect to the instance using SSH.
2. Set the password for the `ec2-user` user.

   ```
   $ sudo /usr/bin/dscl . -passwd /Users/ec2-user
   ```

3. Enable the secure token for the `ec2-user` user. For `-oldPassword`, specify the same password from the previous step. For `-newPassword`, specify a different password. The following command assumes that you have your old and new passowords saved in `.txt` files.

   ```
   $ sysadminctl -oldPassword `cat old_password.txt` -newPassword `cat
    new_password.txt`
   ```

4. Verify that the secure token is enabled.

   ```
   $ sysadminctl -secureTokenStatus ec2-user
   ```

Step 2: Delegate ownership of the Amazon EBS root volume to the EBS root volume administrative user

To delegate ownership, you must create a volume ownership delegation task.

1. Use the create-delegate-mac-volume-ownership-task command to create the task. For `--instance-id`, specify the ID of the instance. For `--mac-credentials`, specify the following credentials:

   - **Internal disk administrative user**
     - **Username** — Only the default administrative user (`aws-managed-user`) is supported and it is used by default. You can't specify a different administrative user.
     - **Password** — If you did not change the default password for `aws-managed-user`, specify the default password, which is *blank*. Otherwise, specify your password.
   - **Amazon EBS root volume administrative user**
     - **Username** — If you did not change the default administrative user, specify `ec2-user`. Otherwise, specify the username for your administrative user.
     - **Password** — Specify the password that you set for root volume admin user in Step 1 above.

   ```
   aws ec2 create-delegate-mac-volume-ownership-task \
   --instance-id i-1234567890abcdef0 \
   --mac-credentials file://mac-credentials.json
   ```

   The following is the contents of the `mac-credentials.json` file referenced in the preceding examples.

   ```
   {
     "internalDiskPassword":"internal-disk-admin_password",
     "rootVolumeUsername":"root-volume-admin_username",
     "rootVolumepassword":"root-volume-admin_password"
   }
   ```

2. Wait for the volume ownership delegation task to complete and for the instance to return to a healthy state. Use the describe-mac-modification-tasks command. For `--mac-modification-task-id`, specify the ID of the volume ownership delegation task from the previous step.

```
aws ec2 describe-mac-modification-tasks \
--mac-modification-task-id task-id
```

3. After the volume ownership delegation task completes, continue to Step 3.

Step 3: Update the software

After you have delegated ownership of the Amazon EBS root volume, follow the steps described in Update software on x86 Mac instances (below) to update the software.

Manual volume ownership delegation

As you work through this procedure, you create two passwords. One password is for the Amazon EBS root volume administrative user (`ec2-user`), and the other password is for the internal disk administrative user (`aws-managed-user`). Remember these passwords since you will use them as you work through the procedure.

> **Note**
>
> With this procedure on macOS Big Sur, you can only perform minor updates such as updating from macOS Big Sur 11.7.3 to macOS Big Sur 11.7.4. For macOS Monterey or above, you can perform major software updates.

To access the internal disk

1. From your local computer, in the Terminal, connect to your Apple silicon Mac instance using SSH with the following command. For more information, see Connect to your instance using SSH.

```
ssh -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

2. Install and start macOS Screen Sharing using the following command.

```
[ec2-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/
com.apple.screensharing.plist
```

3. Set a password for `ec2-user` with the following command. Remember the password as you will use it later.

```
[ec2-user ~]$ sudo /usr/bin/dscl . -passwd /Users/ec2-user
```

4. Disconnect from the instance by typing **exit** and pressing return.
5. From your local computer, in the Terminal, reconnect to your instance with an SSH tunnel to the VNC port using the following command.

```
ssh -i /path/key-pair-name.pem -L 5900:localhost:5900 ec2-user@instance-public-dns-name
```

> **Note**
>
> Do not exit this SSH session until the following VNC connection and GUI steps are completed. When the instance is restarted, the connection will close automatically.

6. From your local computer, connect to `localhost:5900` using the following steps:
   a. Open **Finder** and select **Go**.
   b. Select **Connect to Server**.
   c. In the **Server Address** field, enter `vnc://localhost:5900`.
7. In the macOS window, connect to the remote session of the Apple silicon Mac instance as `ec2-user` with the password you created in Step 3.
8. Access the internal disk, named **InternalDisk**, using one of the following options.
   a. For macOS Ventura or above: Open **System Settings**, select **General** in the left pane, then **Startup Disk** at the lower right of the pane.
   b. For macOS Monterey or below: Open **System Preferences**, select **Startup Disk**, then unlock the pane by choosing the lock icon in the lower left of the window.

> **Troubleshooting tip**
>
> If you need to mount the internal disk, run the following command in the Terminal.
>
> ```
> APFSVolumeName="InternalDisk" ; SSDContainer=$(diskutil list | grep
>  "Physical Store disk0" -B 3 | grep "/dev/disk" | awk {'print $1'} ) ;
>  diskutil apfs addVolume $SSDContainer APFS $APFSVolumeName
> ```

9. Choose the internal disk, named **InternalDisk**, and select **Restart**. Select **Restart** again when prompted.

> **Important**
>
> If the internal disk is named **Macintosh HD** instead of **InternalDisk**, your instance needs to be stopped and restarted so the dedicated host can be updated. For more information, see [Stop or terminate your Amazon EC2 Mac instance](#).

Use the following procedure to delegate ownership to the administrative user. When you reconnect to your instance with SSH, you boot from the internal disk using the special administrative user (`aws-managed-user`). The initial password for `aws-managed-user` is blank, so you need to overwrite it on your first connection. Then, you need to repeat the steps to install and start macOS Screen Sharing since the boot volume has changed.

To delegate ownership to the administrator on an Amazon EBS volume

1. From your local computer, in the Terminal, connect to your Apple silicon Mac instance using the following command.

```
ssh -i /path/key-pair-name.pem aws-managed-user@instance-public-dns-name
```

2. When you receive the warning `WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!`, use one of the following commands to resolve this issue.

   a. Clear out the known hosts using the following command. Then, repeat the previous step.

   ```
   rm ~/.ssh/known_hosts
   ```

   b. Add the following to the SSH command in the previous step.

   ```
   -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
   ```

3. Set the password for `aws-managed-user` with the following command. The `aws-managed-user` initial password is blank, so you need to overwrite it on your first connection.

a.
```
[aws-managed-user ~]$ sudo /usr/bin/dscl . -passwd /Users/aws-managed-
user password
```

b. When you receive the prompt, `Permission denied. Please enter user's old password:`, press enter.

> **Troubleshooting tip**
>
> If you get the error `passwd: DS error: eDSAuthFailed`, use the following command.
>
> ```
> [aws-managed-user ~]$ sudo passwd aws-managed-user
> ```

4. Install and start macOS Screen Sharing using the following command.

```
[aws-managed-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/
com.apple.screensharing.plist
```

5. Disconnect from the instance by typing **exit** and pressing return.
6. From your local computer, in the Terminal, reconnect to your instance with an SSH tunnel to the VNC port using the following command.

```
ssh -i /path/key-pair-name.pem -L 5900:localhost:5900 aws-managed-user@instance-
public-dns-name
```

7. From your local computer, connect to `localhost:5900` using the following steps:

   a. Open **Finder** and select **Go**.

   b. Select **Connect to Server**.

   c. In the **Server Address** field, enter `vnc://localhost:5900`.

8. In the macOS window, connect to the remote session of the Apple silicon Mac instance as `aws-managed-user` with the password you created in Step 3.

> **Note**
>
> When prompted to sign in with your Apple ID, select **Set Up Later**.

9. Access the Amazon EBS volume using one of the following options.

   a. For macOS Ventura or later: Open **System Settings**, select **General** in the left pane, then **Startup Disk** at the lower right of the pane.

   b. For macOS Monterey or earlier: Open **System Preferences**, select **Startup Disk**, then unlock the pane using the lock icon in the lower left of the window.

   > **Note**
   >
   > Until the reboot takes place, when prompted for an administrator password, use the password you set above for `aws-managed-user`. This password might be different from the one you set for `ec2-user` or the default administrator account on your instance. The following instructions specify when to use your instance's administrator password.

10. Select the Amazon EBS volume (the volume not named **InternalDisk** in the **Startup Disk** window) and choose **Restart**.

    > **Note**
    >
    > If you have multiple bootable Amazon EBS volumes attached to your Apple silicon Mac instance, be sure to use a unique name for each volume.

11. Confirm the restart, then choose **Authorize Users** when prompted.

12. On the **Authorize user on this volume** pane, verify that the administrative user (`ec2-user` by default) is selected, then select **Authorize**.

13. Enter the `ec2-user` password you created in Step 3 of the previous procedure, then select **Continue**.

14. Enter the password for the special administrative user (`aws-managed-user`) when prompted.

15. From your local computer, in the Terminal, reconnect to your instance using SSH with username `ec2-user`.

> **Troubleshooting tip**
>
> If you get the warning `WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!`, run the following command and reconnect to your instance using SSH.
>
> ```
> rm ~/.ssh/known_hosts
> ```

16. To perform the software update, use the commands under Update software on x86 Mac instances.

Update software on x86 Mac instances

On x86 Mac instances, you can install operating system updates from Apple using the `softwareupdate` command.

To install operating system updates from Apple on x86 Mac instances

1. List the packages with available updates using the following command.

   ```
   [ec2-user ~]$ softwareupdate --list
   ```

2. Install all updates or only specific updates. To install specific updates, use the following command.

   ```
   [ec2-user ~]$ sudo softwareupdate --install label
   ```

   To install all updates instead, use the following command.

   ```
   [ec2-user ~]$ sudo softwareupdate --install --all --restart
   ```

System administrators can use AWS Systems Manager to roll out pre-approved operating system updates on x86 Mac instances. For more information, see the AWS Systems Manager User Guide.

You can use Homebrew to install updates to packages in the EC2 macOS AMIs, so that you have the latest version of these packages on your instances. You can also use Homebrew to install and run common macOS applications on Amazon EC2 macOS. For more information, see the Homebrew Documentation.

To install updates using Homebrew

1. Update Homebrew using the following command.

```
[ec2-user ~]$ brew update
```

2. List the packages with available updates using the following command.

```
[ec2-user ~]$ brew outdated
```

3. Install all updates or only specific updates. To install specific updates, use the following command.

```
[ec2-user ~]$ brew upgrade package name
```

To install all updates instead, use the following command.

```
[ec2-user ~]$ brew upgrade
```

Increase the size of an EBS volume on your Mac instance

You can increase the size of your Amazon EBS volumes on your Mac instance. For more information, see Amazon EBS Elastic Volumes in the *Amazon EBS User Guide*.

After you increase the size of the volume, you must increase the size of your APFS container as follows.

Make increased disk space available for use

1. Determine if a restart is required. If you resized an existing EBS volume on a running Mac instance, you must reboot the instance to make the new size available. If disk space modification was done during launch time, a reboot will not be required.

View current status of disk sizes:

```
[ec2-user ~]$  diskutil list external physical
/dev/disk0 (external, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:                  GUID_partition_scheme            *322.1 GB    disk0
   1:                  EFI EFI                           209.7 MB    disk0s1
```

```
   2:                  Apple_APFS Container disk2        321.9 GB     disk0s2
```

2. Copy and paste the following command.

```
[ec2-user ~]$ PDISK=$(diskutil list physical external | head -n1 | cut -d" " -f1)
APFSCONT=$(diskutil list physical external | grep "Apple_APFS" | tr -s " " | cut -
d" " -f8)
yes | sudo diskutil repairDisk $PDISK
```

3. Copy and paste the following command.

```
[ec2-user ~]$ sudo diskutil apfs resizeContainer $APFSCONT 0
```

Stop or terminate your Amazon EC2 Mac instance

When you stop a Mac instance, the instance remains in the `stopping` state for about 15 minutes before it enters the `stopped` state.

When you stop or terminate a Mac instance, Amazon EC2 performs a scrubbing workflow on the underlying Dedicated Host to erase the internal SSD, to clear the persistent NVRAM variables, and to update to the latest device firmware. This ensures that Mac instances provide the same security and data privacy as other EC2 Nitro instances. It also allows you to run the latest macOS AMIs. During the scrubbing workflow, the Dedicated Host temporarily enters the pending state. On x86 Mac instances, the scrubbing workflow might take up to 50 minutes to complete. On Apple silicon Mac instances, the scrubbing workflow might take up to 110 minutes to complete. Additionally, on x86 Mac instances, if the device firmware needs to be updated, the scrubbing workflow might take up to 3 hours to complete.

You can't start the stopped Mac instance or launch a new Mac instance until after the scrubbing workflow completes, at which point the Dedicated Host enters the `available` state.

Metering and billing is paused when the Dedicated Host enters the `pending` state. You are not charged for the duration of the scrubbing workflow.

Release the Dedicated Host for your Mac instance

When you are finished with your Mac instance, you can release the Dedicated Host. Before you can release the Dedicated Host, you must stop or terminate the Mac instance. You cannot release the host until the allocation period exceeds the 24-hour minimum.

To release the Dedicated Host

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 **Instance state**, then choose either **Stop instance** or **Terminate instance**.
4. In the navigation pane, choose **Dedicated Hosts**.
5. Select the Dedicated Host and choose **Actions**, **Release host**.
6. When prompted for confirmation, choose **Release**.

Configure System Integrity Protection for Amazon EC2 Mac instances

You can configure System Integrity Protection (SIP) settings for x86 Mac instances and Apple silicon Mac instances. SIP is a critical macOS security feature that helps to prevent unauthorized code execution and system-level modifications. For more information, see About System Integrity Protection.

You can either enable or disable SIP completely, or you can selectively enable or disable specific SIP settings. It is recommended that you disable SIP only temporarily to perform necessary tasks, and then reenable it as soon as possible. Leaving SIP disabled could leave your instance vulnerable to malicious code.

SIP configuration is supported in all AWS Regions where Amazon EC2 Mac instances are supported.

Topics

Considerations

- The following Amazon EC2 Mac instance types and macOS versions are supported:
  - **Mac1 | Mac2 | Mac2-m1ultra** — macOS Ventura (version 13.0 or later)

- **Mac2-m2 | Mac2-m2pro** — macOS Ventura (version 13.2 or later)

> **Note**
>
> Beta and preview versions of macOS are not supported.

- You can specify a custom SIP configuration to selectively enable or disable individual SIP settings. If you implement a custom configuration, connect to the instance and verify the settings to ensure that your requirements are properly implemented and functioning as intended.

  SIP configurations might change with macOS updates. We recommend that you review custom SIP settings after any macOS version upgrade to ensure continued compatibility and proper functionality of your security configurations.
- For x86 Mac instances, SIP settings are applied at the instance level. Any root volume attached to the instance will automatically inherit the configured SIP settings.

  For Apple silicon Mac instances, SIP settings are applied at the volume level. Root volumes attached to the instance do not inherit the SIP settings. If you attach another root volume, you must reconfigure the SIP settings to the required state.
- It can take up to 90 mins for SIP configuration tasks to complete. The instance remains unreachable while the SIP configuration task in progress.
- SIP configurations do not transfer to snapshots or AMIs that you subsequently create from the instance.
- Apple silicon Mac instances must have only one bootable volume, and each attached volume can have only one additional admin user.

Default SIP configurations

The following table lists the default SIP configuration for x86 Mac instances and Apple silicon Mac instances.

	Apple silicon Mac instances	x86 Mac instances
Apple Internal	Enabled	Disabled

	Apple silicon Mac instances	x86 Mac instances
Filesystem Protections	Enabled	Disabled
Base System	Enabled	Enabled
Debugging Restrictions	Enabled	Enabled
Dtrace Restrictions	Enabled	Enabled
Kext Signing	Enabled	Enabled
Nvram Protections	Enabled	Enabled

Check your SIP configuration

We recommend that you check your SIP configuration before and after making changes to ensure that it is configured as expected.

To check the SIP configuration for an Amazon EC2 Mac instance

Connect to the instance using SSH, and then run the following command at the command line.

```
$ csrutil status
```

The following is example output.

```
System Integrity Protection status: enabled.

Configuration:
    Apple Internal: enabled
    Kext Signing: disabled
    Filesystem Protections: enabled
    Debugging Restrictions: enabled
```

```
    DTrace Restrictions: enabled
    NVRAM Protections: enabled
    BaseSystem Verification: disabled
```

Prerequisites for Apple silicon Mac instances

Before you can configure the SIP settings for Apple silicon Mac instances, you must set a password and enable the secure token for the Amazon EBS root volume administrative user (`ec2-user`).

> **Note**
>
> The password and secure token are set the first time you connect to an Apple silicon Mac instance using the GUI. If you previously connected to the instance using the GUI, or if you are using an x86 Mac instance, you **do not** need to perform these steps.

To set a password and enable the secure token for the EBS root volume administrative user

1. Connect to the instance using SSH.
2. Set the password for the `ec2-user` user.

   ```
   $ sudo /usr/bin/dscl . -passwd /Users/ec2-user
   ```

3. Enable the secure token for the `ec2-user` user. For `-oldPassword`, specify the same password from the previous step. For `-newPassword`, specify a different password. The following command assumes that you have your old and new passowords saved in `.txt` files.

   ```
   $ sysadminctl -oldPassword `cat old_password.txt` -newPassword `cat
    new_password.txt`
   ```

4. Verify that the secure token is enabled.

   ```
   $ sysadminctl -secureTokenStatus ec2-user
   ```

Configure SIP settings

When you configure the SIP settings for your instance, you can either enable or disable all SIP settings, or you can specify a custom configuration that selectively enables or disables specific SIP settings.

> **Note**
>
> If you implement a custom configuration, connect to the instance and verify the settings to ensure that your requirements are properly implemented and functioning as intended. SIP configurations might change with macOS updates. We recommend that you review custom SIP settings after any macOS version upgrade to ensure continued compatibility and proper functionality of your security configurations.

To configure the SIP settings for your instance, you must create a SIP configuration task. The SIP configuration task specifies the SIP settings for your instance.

When you create a SIP configuration for an Apple silicon Mac instance, you must specify the following credentials:

- **Internal disk administrative user**
  - Username — Only the default administrative user (`aws-managed-user`) is supported and it is used by default. You can't specify a different administrative user.
  - Password — If you did not change the default password for `aws-managed-user`, specify the default password, which is *blank*. Otherwise, specify your password.
- **Amazon EBS root volume administrative user**
  - Username — If you did not change the default administrative user, specify `ec2-user`. Otherwise, specify the username for your administrative user.
  - Password — You must always specify the password.

Use the following methods to create a SIP configuration task.

Console

To create a SIP configuration task using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, choose **Instances** and then select the Amazon EC2 Mac instance.
3. In the **Security** tab, choose **Modify Mac, Modify System Integrity Protection**.
4. To enable all SIP settings, select **Enable SIP**. To disable all SIP settings, clear **Enable SIP**.

5. To specify a custom configuration that selectively enables or disables specific SIP settings, select **Specify a custom SIP configuration**, and then select the SIP settings to enable, or clear the SIP settings to disable.
6. Specify the credentials for the root volume user and internal disk owner.
7. Choose **Create SIP modification task**.

AWS CLI

To create a SIP configuration task using the AWS CLI

Use the create-mac-system-integrity-protection-modification-task command.

Enable or disable all SIP settings

To completely enable or disable all SIP settings, use only the `--mac-system-integrity-protection-status` parameter.

The following example command enables all SIP settings.

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status enabled \
--mac-credentials file://mac-credentials.json
```

Specify a custom SIP configuration

To specify a custom SIP configuration that selectively enables or disable specific SIP settings, specify the `--mac-system-integrity-protection-status` and `--mac-system-integrity-protection-configuration` parameters. In this case, use `mac-system-integrity-protection-status` to specify the overall SIP status, and use `mac-system-integrity-protection-configuration` to selectively enable or disable individual SIP settings.

The following example command creates a SIP configuration task to enable all SIP settings, except `NvramProtections` and `FilesystemProtections`.

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status enabled \
```

```
--mac-system-integrity-protection-configuration
 "NvramProtections=disabled, FilesystemProtections=disabled" \
--mac-credentials file://mac-credentials.json
```

The following example command creates a SIP configuration task to disable all SIP settings, except `DtraceRestrictions`.

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status disabled \
--mac-system-integrity-protection-configuration "DtraceRestrictions=enabled" \
--mac-credentials file://mac-credentials.json
```

Contents of the `mac-credentials.json` file

The following is the contents of the `mac-credentials.json` file referenced in the preceding examples.

```
{
  "internalDiskPassword":"internal-disk-admin_password",
  "rootVolumeUsername":"root-volume-admin_username",
  "rootVolumepassword":"root-volume-admin_password"
}
```

Check SIP configuration task status

Use one of the following methods to check the state of SIP configuration tasks.

Console

To view SIP configuration tasks using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, choose **Instances** and then select the Amazon EC2 Mac instance.
3. In the **Security** tab, scroll down to the **Mac modification tasks** section.

AWS CLI

To check the state of SIP configuration tasks using the AWS CLI

Use the describe-mac-modification-tasks command.

Find supported macOS versions for your Amazon EC2 Mac Dedicated Host

You can view the latest macOS versions supported by your Amazon EC2 Mac Dedicated Host. With this functionality, you can validate whether your Dedicated Host can support instance launches with your preferred macOS versions.

Each macOS version requires a minimum firmware version on the underlying Apple Mac to successfully boot. The Apple Mac firmware version can become outdated if an allocated Mac Dedicated Host has remained idle for an extended period of time or if it has a long running instance on it.

To ensure supportability for the latest macOS versions, you can stop or terminate instances on your allocated Mac Dedicated Host. This triggers the host scrubbing workflow and updates the firmware on the underlying Apple Mac to support the latest macOS versions. A Dedicated Host with a long running instance will automatically be updated when you stop or terminate a running instance.

For more information about the scrubbing workflow, see Stop or terminate your Amazon EC2 Mac instance.

For more information about launching Mac instances, see Launch a Mac instance using the AWS Management Console or the AWS CLI.

You can view information about the latest macOS versions supported on your allocated Dedicated Host using the Amazon EC2 console or the AWS CLI.

Console

To view Dedicated Host firmware information 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. On the **Dedicated Hosts details** page, under **Latest supported macOS versions**, you can see the latest macOS versions that the host can support.

AWS CLI

To view Dedicated Host firmware information using the AWS CLI

Use the [describe-mac-hosts](#) command, replacing `region` with the appropriate AWS Region.

```
$ aws ec2 describe-mac-hosts --region us-east-1
  {
      "MacHosts": [
          {
              "HostId": "h-07879acf49EXAMPLE",
              "MacOSLatestSupportedVersions": [
                  "14.3",
                  "13.6.4",
                  "12.7.3"
              ]
          }
      ]
  }
```

Subscribe to macOS AMI notifications

To be notified when new AMIs are released or when bridgeOS has been updated, subscribe for notifications using Amazon SNS.

For more information about EC2 macOS AMIs, see [Amazon EC2 macOS AMIs release notes](#).

To subscribe to macOS AMI notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must use 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-1:898855652048:amazon-ec2-macos-ami-updates`**
    - **`arn:aws:sns:us-east-1:898855652048:amazon-ec2-bridgeos-updates`**

    b. For **Protocol**, choose one of the following:

- **Email:**

  For **Endpoint**, type an email address that you can use to receive the notifications. After you create your subscription you'll receive a confirmation message with the subject line `AWS Notification - Subscription Confirmation`. Open the email and choose **Confirm subscription** to complete your subscription

- **SMS:**

  For **Endpoint**, type a phone number that you can use to receive the notifications.

- **AWS Lambda, Amazon SQS, Amazon Data Firehose** (*Notifications come in JSON format*):

  For **Endpoint**, enter the ARN for the Lambda function, SQS queue, or Firehose stream you can use to receive the notifications.

c. Choose **Create subscription**.

Whenever macOS AMIs are released, we send notifications to the subscribers of the `amazon-ec2-macos-ami-updates` topic. Whenever bridgeOS is updated, we send notifications to the subscribers of the `amazon-ec2-bridgeos-updates` topic. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from macOS AMI notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must use 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**.

Retrieve macOS AMI IDs using AWS Systems Manager Parameter Store API

You must specify an AMI when you launch an instance. An AMI is specific to an AWS Region, operating system, and processor architecture. You can view all of the macOS AMIs in an AWS Region and retrieve the latest macOS AMI by querying the AWS Systems Manager Parameter Store API. Using these public parameters, you don't need to manually look up macOS AMI IDs. Public parameters are available for both x86 and ARM64 macOS AMIs, and can be integrated with your existing AWS CloudFormation templates.

Required permissions

To perform this action, the IAM principal must have permissions to call the `ssm:GetParameter` API action.

To view a list of all macOS AMIs in the current AWS Region using the AWS CLI

Use the following get-parameters-by-path command to view a list of all macOS AMIs in the current Region.

```
aws ssm get-parameters-by-path --path /aws/service/ec2-macos --recursive --query
 "Parameters[].Name"
```

To retrieve the AMI ID of the latest major macOS AMI using the AWS CLI

Use the following get-parameter command with the sub-parameter `image_id`. In the following example, replace `sonoma` with a macOS supported major version, `x86_64_mac` with the processor, and `region-code` with a supported AWS Region for which you want the latest macOS AMI ID.

```
aws ssm get-parameter --name /aws/service/ec2-macos/sonoma/x86_64_mac/latest/image_id
 --region region-code
```

For more information, see Calling AMI public parameters for macOS in the *AWS Systems Manager User Guide*.

Amazon EC2 macOS AMIs release notes

The following information provides details about the packages included by default in the EC2 macOS AMIs and summarizes the changes for each EC2 macOS AMI release.

For information about how to subscribe to macOS AMI notifications, see Subscribe to macOS AMI notifications.

Mac instances can run one of the following operating systems:

- macOS Mojave (version 10.14) (x86 Mac instances only)
- macOS Catalina (version 10.15) (x86 Mac instances only)

- macOS Big Sur (version 11) (x86 and M1 Mac instances)
- macOS Monterey (version 12) (x86 and M1 Mac instances)
- macOS Ventura (version 13) (all Mac instances, M2 and M2 Pro Mac instances support macOS Ventura version 13.2 or later)
- macOS Sonoma (version 14) (all Mac instances)
- macOS Sequoia (version 15) (all Mac instances)

Approve Local Network Privacy policies for macOS Sequoia

macOS Sequoia (version 15) has a new Local Network Privacy feature that impacts users of local IP-based services, including Amazon EC2 Instance Metadata Service (IMDS).

> **Important**
>
> To make sure that you have uninterrupted access to local IP-based services, use the following steps to approve the Local Network Privacy policies.

To approve Local Network Privacy policies

1. Connect to your instance's graphical user interface (GUI).
2. Follow the prompts on the screen to approve the Local Network Privacy policies.
3. After you have approved the policies, create an AMI of your EC2 Mac instance. For more information, see Create an Amazon EBS-backed AMI.

Any EC2 Mac instances that are launched from the newly created AMI will retain the Local Network Privacy permissions.

Default packages included in Amazon EC2 macOS AMIs

The following table describes packages that are included by default in the EC2 macOS AMIs.

Packages	Release notes
EC2 macOS Init	https://github.com/aws/ec2-macos-init/tags

Packages	Release notes
EC2 macOS Utils	https://github.com/aws/ec2-macos-utils/tags
Amazon SSM Agent	https://github.com/aws/amazon-ssm-agent/releases
AWS Command Line Interface (AWS CLI) version 2	https://raw.githubusercontent.com/aws/aws-cli/v2/CHANGELOG.rst
Command Line Tools for Xcode	https://developer.apple.com/documentation/xcode-release-notes
Homebrew	https://github.com/Homebrew/brew/releases
EC2 Instance Connect	https://github.com/aws/aws-ec2-instance-connect-config/releases
Safari	https://developer.apple.com/documentation/safari-release-notes

Amazon EC2 macOS AMI updates

The following table describes changes included in the EC2 macOS AMI releases. Note that some changes apply to all EC2 macOS AMIs, whereas others apply to only a subset of these AMIs.

EC2 macOS AMI updates

Release	Changes
2025.09.04	**All AMIs** • Updated `awscli` to 2.28.19 • Updated Homebrew to 4.6.7 **Release of macOS Ventura 13.7.8** • Security content of macOS Ventura 13.7.8

Release	Changes
	Release of macOS Sonoma 14.7.8 • Security content of macOS Sonoma 14.7.8 **Release of macOS Sequoia 15.6.1** • Security content of macOS Sequoia 15.6.1

2025.08.05	**All AMIs**
	• Updated `awscli` to 2.28.0
	• Updated Homebrew to 4.5.13
	
	Release of macOS Ventura 13.7.7
	• Security content of macOS Ventura 13.7.7
	• Updated Safari to 18.6
	    • Security content of Safari 18.6
	
	Release of macOS Sonoma 14.7.7
	• Security content of macOS Sonoma 14.7.7
	• Updated Safari to 18.6
	    • Security content of Safari 18.6
	
	Release of macOS Sequoia 15.6
	• Security content of macOS Sequoia 15.6
	• Updated AWS ENA Ethernet to 2.0.0

Release	Changes

2025.06.27	**All AMIs**
	
	• Updated `awscli` to 2.27.40
	• Updated Homebrew to 4.5.7
	• Migrated x86_64 images to AWS ENA Ethernet dext
	
	Release of macOS Ventura 13.7.6
	
	• Security content of macOS Ventura 13.7.6
	• Updated Safari to 18.5
	  • Security content of Safari 18.5
	
	Release of macOS Sonoma 14.7.6
	
	• Security content of macOS Sonoma 14.7.6
	• Updated Safari to 18.5
	  • Security content of Safari 18.5
	
	Release of macOS Sequoia 15.5
	
	• Security content of macOS Sequoia 15.5
	• Updated Command Line Tools to 16.4
	• Updated AWS ENA Ethernet to 1.0.9

Release	Changes

2025.05.21	**All AMIs**
	• Updated `awscli` to 2.27.7
	• Updated `ec2-macos-init` to 1.5.11
	• Updated `ec2-macos-utils` to 1.0.5
	• Updated Homebrew to 4.5.1
	
	Release of macOS Ventura 13.7.5
	• Security content of macOS Ventura 13.7.5
	• Updated Safari to 18.4
	  ◦ Security content of Safari 18.4
	
	Release of macOS Sonoma 14.7.5
	• Security content of macOS Sonoma 14.7.5
	• Updated Safari to 18.4
	  ◦ Security content of Safari 18.4
	
	Release of macOS Sequoia 15.4.1
	• Security content of macOS Sequoia 15.4.1
	• Updated Command Line Tools to 16.3

Release	Changes

2025.05.05	**All AMIs**
	• Updated `awscli` to 2.27.1
	• Updated `ec2-macos-init` to 1.5.11
	• Updated `ec2-macos-system-monitor` to 1.3.1
	• Updated `ec2-macos-utils` to 1.0.5
	• Updated Homebrew to 4.4.32
	
	Release of macOS Ventura 13.7.5
	• Security content of macOS Ventura 13.7.5
	• Updated Safari to 18.4
	  ◦ Security content of Safari 18.4
	
	Release of macOS Sonoma 14.7.5
	• Security content of macOS Sonoma 14.7.5
	• Updated Safari to 18.4
	  ◦ Security content of Safari 18.4
	
	Release of macOS Sequoia 15.4.1
	• Security content of macOS Sequoia 15.4.1

Release	Changes
2025.03.18	**All AMIs** • Updated `awscli` to 2.24.2 • Updated Homebrew to 4.4.20 **Release of macOS Sequoia 15.3.1** • Security content of macOS Sequoia 15.3.1 **Release of macOS Sonoma 14.7.4** • Security content of macOS Sonoma 14.7.4 • Updated Safari to 18.3 **Release of macOS Ventura 13.7.4** • Security content of macOS Ventura 13.7.4 • Updated Safari to 18.3

Release	Changes
2025.01.24	**All AMIs** • Updated `awscli` to 2.22.33 • Updated Homebrew to 4.4.15 **Release of macOS Sequoia 15.2** • Security content of macOS Sequoia 15.2 • Updated Command Line Tools to 16.2 **Release of macOS Sonoma 14.7.2** • Security content of macOS Sonoma 14.7.2 • Updated Safari to 18.2 • Updated Command Line Tools to 16.2 **Release of macOS Ventura 13.7.2** • Security content of macOS Ventura 13.7.2 • Updated Safari to 18.2

Release	Changes
2024.12.20	**All AMIs** • Updated Homebrew to 4.4.8 • Updated `aws-cli` to 2.22.5 • Updated `amazon-ssm-agent` to 3.3.987.0 **Release of macOS Sequoia 15.1.1** • Security content of macOS Sequoia 15.1.1 **Release of macOS Sonoma 14.7.1** • Security content of macOS Sonoma 14.7.1 • Updated Safari to 18.1.1 **Release of macOS Ventura 13.7.1** • Security content of macOS Ventura 13.7.1 • Updated Safari to 18.1.1

Release	Changes

2024.10.28	**All AMIs**
	
	• Updated Homebrew to 4.4.2
	• Updated `aws-cli` to 2.18.13
	• Updated `amazon-ssm-agent` to 3.3.987.0
	• Updated `ec2-macos-init` to 1.5.10
	• Updated `ec2-macos-utils` to 1.0.4
	
	Release of macOS Sequoia 15.0
	
	• Security content of macOS Sequoia 15
	
	Release of macOS Sonoma 14.7
	
	• Security content of macOS Sonoma 14.7.
	• Updated Command Line Tools to 16.0
	• Updated Safari to 18.0.1
	    • Security content of Safari 18
	
	Release of macOS Ventura 13.7
	
	• Security content of macOS Ventura 13.7
	• Updated Safari to 18.0.1
	    • Security content of Safari 18

<table>
<tr><th>Release</th><th>Changes</th></tr>
<tr><td>2024.08.20</td><td>All AMIs
Updated Homebrew to 4.3.14
Updated <code>aws-cli</code> to 2.17.29
Release of macOS Sonoma 14.6.1
No published CVE entries.
Release of macOS Ventura 13.6.9
No published CVE entries.
Updated Safari to 17.6
Security content of Safari 17.6
Release of macOS Monterey 12.7.6
Security content of macOS Monterey 12.7.6
Updated Safari to 17.6
Security content of Safari 17.6</td></tr>
</table>

Release	Changes

2024.06.07	**All AMIs**
	
	• Updated Homebrew to 4.3.1-1
	• Updated `aws-cli` to 2.15.56
	• Updated `amazon-ssm-agent` to 3.3.380.0-1
	
	Release of macOS Sonoma 14.5
	
	• Security content of macOS Sonoma 14.5
	
	Release of macOS Ventura 13.6.7
	
	• Security content of macOS Ventura 13.6.7
	• Updated Safari to 17.5
	    • Security content of Safari 17.5
	
	Release of macOS Monterey 12.7.5
	
	• Security content of macOS Monterey 12.7.5
	• Updated Safari to 17.5
	    • Security content of Safari 17.5

Release	Changes

2024.04.12	**All AMIs**
	
	• Updated Homebrew to 4.2.16-1
	• Updated `aws-cli` to 2.15.36
	
	Release of macOS Sonoma 14.4.1
	
	• Security content of macOS Sonoma 14.4.1
	
	Release of macOS Ventura 13.6.6
	
	• Security content of macOS Ventura 13.6.6
	• Updated Safari to 17.4.1
	  ◦ Security content of Safari 17.4.1
	
	For macOS Monterey
	
	• Updated Safari to 17.4.1
	  ◦ Security content of Safari 17.4.1

Amazon EBS-optimized instance types

Amazon EBS–optimized instances use an optimized configuration stack and provide additional, dedicated bandwidth 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.

When attached to an EBS–optimized instance, General Purpose SSD (`gp2` and `gp3`) volumes are designed to deliver at least 90 percent of their provisioned IOPS performance 99 percent of the time in a given year, and Provisioned IOPS SSD (`io1` and `io2`) volumes are designed to deliver at least 90 percent of their provisioned IOPS performance 99.9 percent of the time in a given year. Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`) deliver at least 90 percent of their expected throughput performance 99 percent of the time in a given year. Non-compliant periods are approximately uniformly distributed, targeting 99 percent of expected total throughput each hour. For more information, see [Amazon EBS volume types] in the *Amazon EBS User Guide*.

Some instance types are **EBS-optimized by default**, and there is no need to enable it and no effect if you attempt to disable it. Other instance types optionally **support EBS optimization** and you can enable it during or after launch for an additional hourly fee. Some instance types do not support EBS optimization.

For detailed instance type specifications and features, see the Amazon EC2 Instance Types Guide.

Considerations

- An instance's EBS performance is bounded by the instance type's performance limits, or the aggregated performance of its attached volumes, whichever is smaller. To achieve maximum EBS performance, an instance must have attached volumes that provide a combined performance equal to or greater than the maximum instance performance. For example, to achieve `80,000` IOPS for `r6i.16xlarge`, the instance must have at least 5 `gp3` volumes provisioned with `16,000` IOPS each (5 volumes x `16,000` IOPS = `80,000` IOPS). We recommend that you choose an instance type that provides more dedicated Amazon EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.
- The maximum number of Amazon EBS volumes that you can attach to an instance depends on the instance type and instance size. For more information, see Amazon EBS volume limits for Amazon EC2 instances.
- The maximum IOPS and throughput limits are interdependent. Depending on your I/O size, you might reach one limit before the other, which can affect overall performance. For optimal results, consider both limits when planning your workload.

EBS-optimized by default

The following instance types are EBS–optimized by default. There is no need to enable EBS optimization and no effect if you disable EBS optimization.

Instances

- General purpose
- Compute optimized
- Memory optimized

- Storage optimized
- Accelerated computing
- High-performance computing

General purpose

> **Note**
>
> M8g, M8gd, M8i, M8i-flex instance types support configurable bandwidth weightings. With these instance types, you can optimize an instance's bandwidth for either networking performance or Amazon EBS performance. The following table shows the default Amazon EBS bandwidth performance for these instance types. For more information, see EC2 instance bandwidth weighting configuration.

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
a1.medium [1]	300	3500	37.50	437.50	2500	20000
a1.large [1]	525	3500	65.62	437.50	4000	20000
a1.xlarge [1]	800	3500	100.00	437.50	6000	20000
a1.2xlarge [1]	1750	3500	218.75	437.50	10000	20000
a1.4xlarge [2]	3500		437.5		20000	
a1.metal [2]	3500		437.5		20000	
m4.large [2]	450		56.25		3600	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m4.xlarge [2]	750		93.75		6000	
m4.2xlarge [2]	1000		125.0		8000	
m4.4xlarge [2]	2000		250.0		16000	
m4.10xlarge [2]	4000		500.0		32000	
m4.16xlarge [2]	10000		1250.0		65000	
m5.large [1]	650	4750	81.25	593.75	3600	18750
m5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5.4xlarge [2]	4750		593.75		18750	
m5.8xlarge [2]	6800		850.0		30000	
m5.12xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5.16xlarge [2]	13600		1700.0		60000	
m5.24xlarge [2]	19000		2375.0		80000	
m5.metal [2]	19000		2375.0		80000	
m5a.large [1]	650	2880	81.25	360.00	3600	16000
m5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5a.4xlarge [2]	2880		360.0		16000	
m5a.8xlarge [2]	4750		593.75		20000	
m5a.12xlarge [2]	6780		847.5		30000	
m5a.16xlarge [2]	9500		1187.5		40000	
m5a.24xlarge [2]	13750		1718.75		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5ad.large [1]	650	2880	81.25	360.00	3600	16000
m5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5ad.4xlarge [2]	2880		360.0		16000	
m5ad.8xlarge [2]	4750		593.75		20000	
m5ad.12xlarge [2]	6780		847.5		30000	
m5ad.16xlarge [2]	9500		1187.5		40000	
m5ad.24xlarge [2]	13750		1718.75		60000	
m5d.large [1]	650	4750	81.25	593.75	3600	18750
m5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5d.4xlarge [2]	4750		593.75		18750	
m5d.8xlarge [2]	6800		850.0		30000	
m5d.12xlarge [2]	9500		1187.5		40000	
m5d.16xlarge [2]	13600		1700.0		60000	
m5d.24xlarge [2]	19000		2375.0		80000	
m5d.metal [2]	19000		2375.0		80000	
m5dn.large [1]	650	4750	81.25	593.75	3600	18750
m5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5dn.4xlarge [2]	4750		593.75		18750	
m5dn.8xlarge [2]	6800		850.0		30000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5dn.12xlarge [2]	9500		1187.5		40000	
m5dn.16xlarge [2]	13600		1700.0		60000	
m5dn.24xlarge [2]	19000		2375.0		80000	
m5dn.metal [2]	19000		2375.0		80000	
m5n.large [1]	650	4750	81.25	593.75	3600	18750
m5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5n.4xlarge [2]	4750		593.75		18750	
m5n.8xlarge [2]	6800		850.0		30000	
m5n.12xlarge [2]	9500		1187.5		40000	
m5n.16xlarge [2]	13600		1700.0		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5n.24xlarge [2]	19000		2375.0		80000	
m5n.metal [2]	19000		2375.0		80000	
m5zn.large [1]	800	3170	100.00	396.25	3333	13333
m5zn.xlarge [1]	1564	3170	195.50	396.25	6667	13333
m5zn.2xlarge [2]	3170		396.25		13333	
m5zn.3xlarge [2]	4750		593.75		20000	
m5zn.6xlarge [2]	9500		1187.5		40000	
m5zn.12xlarge [2]	19000		2375.0		80000	
m5zn.metal [2]	19000		2375.0		80000	
m6a.large [1]	650	10000	81.25	1250.00	3600	40000
m6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6a.8xlarge [2]	10000		1250.0		40000	
m6a.12xlarge [2]	15000		1875.0		60000	
m6a.16xlarge [2]	20000		2500.0		80000	
m6a.24xlarge [2]	30000		3750.0		120000	
m6a.32xlarge [2]	40000		5000.0		160000	
m6a.48xlarge [2]	40000		5000.0		240000	
m6a.metal [2]	40000		5000.0		240000	
m6g.medium [1]	315	4750	39.38	593.75	2500	20000
m6g.large [1]	630	4750	78.75	593.75	3600	20000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6g.4xlarge [2]	4750		593.75		20000	
m6g.8xlarge [2]	9500		1187.5		40000	
m6g.12xlarge [2]	14250		1781.25		50000	
m6g.16xlarge [2]	19000		2375.0		80000	
m6g.metal [2]	19000		2375.0		80000	
m6gd.medium [1]	315	4750	39.38	593.75	2500	20000
m6gd.large [1]	630	4750	78.75	593.75	3600	20000
m6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6gd.4xlarge [2]	4750		593.75		20000	
m6gd.8xlarge [2]	9500		1187.5		40000	
m6gd.12xlarge [2]	14250		1781.25		50000	
m6gd.16xlarge [2]	19000		2375.0		80000	
m6gd.metal [2]	19000		2375.0		80000	
m6i.large [1]	650	10000	81.25	1250.00	3600	40000
m6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6i.8xlarge [2]	10000		1250.0		40000	
m6i.12xlarge [2]	15000		1875.0		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6i.16xlarge [2]	20000		2500.0		80000	
m6i.24xlarge [2]	30000		3750.0		120000	
m6i.32xlarge [2]	40000		5000.0		160000	
m6i.metal [2]	40000		5000.0		160000	
m6id.large [1]	650	10000	81.25	1250.00	3600	40000
m6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6id.8xlarge [2]	10000		1250.0		40000	
m6id.12xlarge [2]	15000		1875.0		60000	
m6id.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6id.24xlarge [2]	30000		3750.0		120000	
m6id.32xlarge [2]	40000		5000.0		160000	
m6id.metal [2]	40000		5000.0		160000	
m6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
m6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
m6idn.8xlarge [2]	25000		3125.0		100000	
m6idn.12xlarge [2]	37500		4687.5		150000	
m6idn.16xlarge [2]	50000		6250.0		200000	
m6idn.24xlarge [2]	75000		9375.0		300000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6idn.32xlarge [2]	100000		12500.0		400000	
m6idn.metal [2]	100000		12500.0		400000	
m6in.large [1]	1562	25000	195.31	3125.00	6250	100000
m6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
m6in.8xlarge [2]	25000		3125.0		100000	
m6in.12xlarge [2]	37500		4687.5		150000	
m6in.16xlarge [2]	50000		6250.0		200000	
m6in.24xlarge [2]	75000		9375.0		300000	
m6in.32xlarge [2]	100000		12500.0		400000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6in.metal [2]	100000		12500.0		400000	
m7a.medium [1]	325	10000	40.62	1250.00	2500	40000
m7a.large [1]	650	10000	81.25	1250.00	3600	40000
m7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7a.8xlarge [2]	10000		1250.0		40000	
m7a.12xlarge [2]	15000		1875.0		60000	
m7a.16xlarge [2]	20000		2500.0		80000	
m7a.24xlarge [2]	30000		3750.0		120000	
m7a.32xlarge [2]	40000		5000.0		160000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m7a.48xlarge [2]	40000		5000.0		240000	
m7a.metal-48xl [2]	40000		5000.0		240000	
m7g.medium [1]	315	10000	39.38	1250.00	2500	40000
m7g.large [1]	630	10000	78.75	1250.00	3600	40000
m7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7g.8xlarge [2]	10000		1250.0		40000	
m7g.12xlarge [2]	15000		1875.0		60000	
m7g.16xlarge [2]	20000		2500.0		80000	
m7g.metal [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m7gd.large [1]	630	10000	78.75	1250.00	3600	40000
m7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7gd.8xlarge [2]	10000		1250.0		40000	
m7gd.12xlarge [2]	15000		1875.0		60000	
m7gd.16xlarge [2]	20000		2500.0		80000	
m7gd.metal [2]	20000		2500.0		80000	
m7i.large [1]	650	10000	81.25	1250.00	3600	40000
m7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i.8xlarge [2]	10000		1250.0		40000	
m7i.12xlarge [2]	15000		1875.0		60000	
m7i.16xlarge [2]	20000		2500.0		80000	
m7i.24xlarge [2]	30000		3750.0		120000	
m7i.48xlarge [2]	40000		5000.0		240000	
m7i.metal-24xl [2]	30000		3750.0		120000	
m7i.metal-48xl [2]	40000		5000.0		240000	
m7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
m7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
m8g.medium [1]	315	10000	39.38	1250.00	2500	40000
m8g.large [1]	630	10000	78.75	1250.00	3600	40000
m8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8g.8xlarge [2]	10000		1250.0		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m8g.12xlarge [2]	15000		1875.0		60000	
m8g.16xlarge [2]	20000		2500.0		80000	
m8g.24xlarge [2]	30000		3750.0		120000	
m8g.48xlarge [2]	40000		5000.0		240000	
m8g.metal-24xl [2]	30000		3750.0		120000	
m8g.metal-48xl [2]	40000		5000.0		240000	
m8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m8gd.large [1]	630	10000	78.75	1250.00	3600	40000
m8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m8gd.8xlarge [2]	10000		1250.0		40000	
m8gd.12xlarge [2]	15000		1875.0		60000	
m8gd.16xlarge [2]	20000		2500.0		80000	
m8gd.24xlarge [2]	30000		3750.0		120000	
m8gd.48xlarge [2]	40000		5000.0		240000	
m8gd.metal-24xl [2]	30000		3750.0		120000	
m8gd.metal-48xl [2]	40000		5000.0		240000	
m8i.large [1]	650	10000	81.25	1250.00	3600	40000
m8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m8i.8xlarge [2]	10000		1250.0		40000	
m8i.12xlarge [2]	15000		1875.0		60000	
m8i.16xlarge [2]	20000		2500.0		80000	
m8i.24xlarge [2]	30000		3750.0		120000	
m8i.32xlarge [2]	40000		5000.0		160000	
m8i.48xlarge [2]	60000		7500.0		240000	
m8i.96xlarge [2]	80000		10000.0		480000	
m8i.metal-48xl [2]	60000		7500.0		240000	
m8i.metal-96xl [2]	80000		10000.0		480000	
m8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
m8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
mac1.metal [2]	14000		1750.0		80000	
mac2.metal [2]	10000		1250.0		55000	
mac2-m1ultra.metal [2]	10000		1250.0		55000	
mac2-m2.metal [2]	8000		1000.0		55000	
mac2-m2pro.metal [2]	8000		1000.0		55000	
t3.nano [1]	43	2085	5.38	260.62	250	11800

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
t3.micro [1]	87	2085	10.88	260.62	500	11800
t3.small [1]	174	2085	21.75	260.62	1000	11800
t3.medium [1]	347	2085	43.38	260.62	2000	11800
t3.large [1]	695	2780	86.88	347.50	4000	15700
t3.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.nano [1]	45	2085	5.62	260.62	250	11800
t3a.micro [1]	90	2085	11.25	260.62	500	11800
t3a.small [1]	175	2085	21.88	260.62	1000	11800
t3a.medium [1]	350	2085	43.75	260.62	2000	11800
t3a.large [1]	695	2780	86.88	347.50	4000	15700
t3a.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.nano [1]	43	2085	5.38	260.62	250	11800
t4g.micro [1]	87	2085	10.88	260.62	500	11800

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
t4g.small [1]	174	2085	21.75	260.62	1000	11800
t4g.medium [1]	347	2085	43.38	260.62	2000	11800
t4g.large [1]	695	2780	86.88	347.50	4000	15700
t4g.xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.2xlarge [1]	695	2780	86.88	347.50	4000	15700

[1] These instances can sustain the maximum performance for 30 minutes at least once every 24 hours, after which they revert to their baseline performance.

[2] These instances can sustain their stated performance indefinitely. If your workload requires sustained maximum performance for longer than 30 minutes, use one of these instances.

Compute optimized

> **Note**
>
> C8g, C8gd instance types support configurable bandwidth weightings. With these instance types, you can optimize an instance's bandwidth for either networking performance or Amazon EBS performance. The following table shows the default Amazon EBS bandwidth performance for these instance types. For more information, see [EC2 instance bandwidth weighting configuration](#).

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c4.large [2]	500		62.5		4000	
c4.xlarge [2]	750		93.75		6000	
c4.2xlarge [2]	1000		125.0		8000	
c4.4xlarge [2]	2000		250.0		16000	
c4.8xlarge [2]	4000		500.0		32000	
c5.large [1]	650	4750	81.25	593.75	4000	20000
c5.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5.4xlarge [2]	4750		593.75		20000	
c5.9xlarge [2]	9500		1187.5		40000	
c5.12xlarge [2]	9500		1187.5		40000	
c5.18xlarge [2]	19000		2375.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c5.24xlarge [2]	19000		2375.0		80000	
c5.metal [2]	19000		2375.0		80000	
c5a.large [1]	200	3170	25.00	396.25	800	13300
c5a.xlarge [1]	400	3170	50.00	396.25	1600	13300
c5a.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5a.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5a.8xlarge [2]	3170		396.25		13300	
c5a.12xlarge [2]	4750		593.75		20000	
c5a.16xlarge [2]	6300		787.5		26700	
c5a.24xlarge [2]	9500		1187.5		40000	
c5ad.large [1]	200	3170	25.00	396.25	800	13300
c5ad.xlarge [1]	400	3170	50.00	396.25	1600	13300

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c5ad.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5ad.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5ad.8xlarge [2]	3170		396.25		13300	
c5ad.12xlarge [2]	4750		593.75		20000	
c5ad.16xlarge [2]	6300		787.5		26700	
c5ad.24xlarge [2]	9500		1187.5		40000	
c5d.large [1]	650	4750	81.25	593.75	4000	20000
c5d.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5d.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5d.4xlarge [2]	4750		593.75		20000	
c5d.9xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c5d.12xlarge [2]	9500		1187.5		40000	
c5d.18xlarge [2]	19000		2375.0		80000	
c5d.24xlarge [2]	19000		2375.0		80000	
c5d.metal [2]	19000		2375.0		80000	
c5n.large [1]	650	4750	81.25	593.75	4000	20000
c5n.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5n.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5n.4xlarge [2]	4750		593.75		20000	
c5n.9xlarge [2]	9500		1187.5		40000	
c5n.18xlarge [2]	19000		2375.0		80000	
c5n.metal [2]	19000		2375.0		80000	
c6a.large [1]	650	10000	81.25	1250.00	3600	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6a.8xlarge [2]	10000		1250.0		40000	
c6a.12xlarge [2]	15000		1875.0		60000	
c6a.16xlarge [2]	20000		2500.0		80000	
c6a.24xlarge [2]	30000		3750.0		120000	
c6a.32xlarge [2]	40000		5000.0		160000	
c6a.48xlarge [2]	40000		5000.0		240000	
c6a.metal [2]	40000		5000.0		240000	
c6g.medium [1]	315	4750	39.38	593.75	2500	20000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6g.large [1]	630	4750	78.75	593.75	3600	20000
c6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
c6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6g.4xlarge [2]	4750		593.75		20000	
c6g.8xlarge [2]	9500		1187.5		40000	
c6g.12xlarge [2]	14250		1781.25		50000	
c6g.16xlarge [2]	19000		2375.0		80000	
c6g.metal [2]	19000		2375.0		80000	
c6gd.medium [1]	315	4750	39.38	593.75	2500	20000
c6gd.large [1]	630	4750	78.75	593.75	3600	20000
c6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6gd.4xlarge [2]	4750		593.75		20000	
c6gd.8xlarge [2]	9500		1187.5		40000	
c6gd.12xlarge [2]	14250		1781.25		50000	
c6gd.16xlarge [2]	19000		2375.0		80000	
c6gd.metal [2]	19000		2375.0		80000	
c6gn.medium [1]	760	9500	95.00	1187.50	2500	40000
c6gn.large [1]	1235	9500	154.38	1187.50	5000	40000
c6gn.xlarge [1]	2375	9500	296.88	1187.50	10000	40000
c6gn.2xlarge [1]	4750	9500	593.75	1187.50	20000	40000
c6gn.4xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6gn.8xlarge [2]	19000		2375.0		80000	
c6gn.12xlarge [2]	28500		3562.5		120000	
c6gn.16xlarge [2]	38000		4750.0		160000	
c6i.large [1]	650	10000	81.25	1250.00	3600	40000
c6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6i.8xlarge [2]	10000		1250.0		40000	
c6i.12xlarge [2]	15000		1875.0		60000	
c6i.16xlarge [2]	20000		2500.0		80000	
c6i.24xlarge [2]	30000		3750.0		120000	
c6i.32xlarge [2]	40000		5000.0		160000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6i.metal [2]	40000		5000.0		160000	
c6id.large [1]	650	10000	81.25	1250.00	3600	40000
c6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6id.8xlarge [2]	10000		1250.0		40000	
c6id.12xlarge [2]	15000		1875.0		60000	
c6id.16xlarge [2]	20000		2500.0		80000	
c6id.24xlarge [2]	30000		3750.0		120000	
c6id.32xlarge [2]	40000		5000.0		160000	
c6id.metal [2]	40000		5000.0		160000	
c6in.large [1]	1562	25000	195.31	3125.00	6250	100000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
c6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
c6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
c6in.8xlarge [2]	25000		3125.0		100000	
c6in.12xlarge [2]	37500		4687.5		150000	
c6in.16xlarge [2]	50000		6250.0		200000	
c6in.24xlarge [2]	75000		9375.0		300000	
c6in.32xlarge [2]	100000		12500.0		400000	
c6in.metal [2]	100000		12500.0		400000	
c7a.medium [1]	325	10000	40.62	1250.00	2500	40000
c7a.large [1]	650	10000	81.25	1250.00	3600	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7a.8xlarge [2]	10000		1250.0		40000	
c7a.12xlarge [2]	15000		1875.0		60000	
c7a.16xlarge [2]	20000		2500.0		80000	
c7a.24xlarge [2]	30000		3750.0		120000	
c7a.32xlarge [2]	40000		5000.0		160000	
c7a.48xlarge [2]	40000		5000.0		240000	
c7a.metal-48xl [2]	40000		5000.0		240000	
c7g.medium [1]	315	10000	39.38	1250.00	2500	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c7g.large [1]	630	10000	78.75	1250.00	3600	40000
c7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7g.8xlarge [2]	10000		1250.0		40000	
c7g.12xlarge [2]	15000		1875.0		60000	
c7g.16xlarge [2]	20000		2500.0		80000	
c7g.metal [2]	20000		2500.0		80000	
c7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c7gd.large [1]	630	10000	78.75	1250.00	3600	40000
c7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7gd.8xlarge [2]	10000		1250.0		40000	
c7gd.12xlarge [2]	15000		1875.0		60000	
c7gd.16xlarge [2]	20000		2500.0		80000	
c7gd.metal [2]	20000		2500.0		80000	
c7gn.medium [1]	521	10000	65.12	1250.00	2083	40000
c7gn.large [1]	1042	10000	130.25	1250.00	4167	40000
c7gn.xlarge [1]	2083	10000	260.38	1250.00	8333	40000
c7gn.2xlarge [1]	4167	10000	520.88	1250.00	16667	40000
c7gn.4xlarge [1]	8333	10000	1041.62	1250.00	33333	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c7gn.8xlarge [1]	16667	20000	2083.38	2500.00	66667	80000
c7gn.12xlarge [1]	25000	30000	3125.00	3750.00	100000	120000
c7gn.16xlarge [1]	33333	40000	4166.62	5000.00	133333	160000
c7gn.metal [1]	33333	40000	4166.62	5000.00	133333	160000
c7i.large [1]	650	10000	81.25	1250.00	3600	40000
c7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i.8xlarge [2]	10000		1250.0		40000	
c7i.12xlarge [2]	15000		1875.0		60000	
c7i.16xlarge [2]	20000		2500.0		80000	
c7i.24xlarge [2]	30000		3750.0		120000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c7i.48xlarge [2]	40000		5000.0		240000	
c7i.metal-24xl [2]	30000		3750.0		120000	
c7i.metal-48xl [2]	40000		5000.0		240000	
c7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
c7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
c7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
c7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
c8g.medium [1]	315	10000	39.38	1250.00	2500	40000

<table>
<tr><th>Instance size</th><th>Baseline bandwidth (Mbps)</th><th>Maximum bandwidth (Mbps)</th><th>Baseline throughput (MB/s, 128 KiB I/O)</th><th>Maximum throughput (MB/s, 128 KiB I/O)</th><th>Baseline IOPS (16 KiB I/O)</th><th>Maximum IOPS (16 KiB I/O)</th></tr>
<tr><td>c8g.large [1]</td><td>630</td><td>10000</td><td>78.75</td><td>1250.00</td><td>3600</td><td>40000</td></tr>
<tr><td>c8g.xlarge [1]</td><td>1250</td><td>10000</td><td>156.25</td><td>1250.00</td><td>6000</td><td>40000</td></tr>
<tr><td>c8g.2xlarge [1]</td><td>2500</td><td>10000</td><td>312.50</td><td>1250.00</td><td>12000</td><td>40000</td></tr>
<tr><td>c8g.4xlarge [1]</td><td>5000</td><td>10000</td><td>625.00</td><td>1250.00</td><td>20000</td><td>40000</td></tr>
<tr><td>c8g.8xlarge [2]</td><td colspan="2">10000</td><td colspan="2">1250.0</td><td colspan="2">40000</td></tr>
<tr><td>c8g.12xlarge [2]</td><td colspan="2">15000</td><td colspan="2">1875.0</td><td colspan="2">60000</td></tr>
<tr><td>c8g.16xlarge [2]</td><td colspan="2">20000</td><td colspan="2">2500.0</td><td colspan="2">80000</td></tr>
<tr><td>c8g.24xlarge [2]</td><td colspan="2">30000</td><td colspan="2">3750.0</td><td colspan="2">120000</td></tr>
<tr><td>c8g.48xlarge [2]</td><td colspan="2">40000</td><td colspan="2">5000.0</td><td colspan="2">240000</td></tr>
<tr><td>c8g.metal-24xl [2]</td><td colspan="2">30000</td><td colspan="2">3750.0</td><td colspan="2">120000</td></tr>
<tr><td>c8g.metal-48xl [2]</td><td colspan="2">40000</td><td colspan="2">5000.0</td><td colspan="2">240000</td></tr>
</table>

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c8gd.large [1]	630	10000	78.75	1250.00	3600	40000
c8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8gd.8xlarge [2]	10000		1250.0		40000	
c8gd.12xlarge [2]	15000		1875.0		60000	
c8gd.16xlarge [2]	20000		2500.0		80000	
c8gd.24xlarge [2]	30000		3750.0		120000	
c8gd.48xlarge [2]	40000		5000.0		240000	
c8gd.metal-24xl [2]	30000		3750.0		120000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c8gd.metal-48xl [2]	40000		5000.0		240000	
c8gn.medium [1]	760	10000	95.00	1250.00	2500	40000
c8gn.large [1]	1250	10000	156.25	1250.00	5000	40000
c8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
c8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8gn.4xlarge [2]	10000		1250.0		40000	
c8gn.8xlarge [2]	20000		2500.0		80000	
c8gn.12xlarge [2]	30000		3750.0		120000	
c8gn.16xlarge [2]	40000		5000.0		160000	
c8gn.24xlarge [2]	60000		7500.0		240000	
c8gn.48xlarge [2]	60000		7500.0		240000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c8gn.metal-24xl [2]	60000		7500.0		240000	
c8gn.metal-48xl [2]	60000		7500.0		240000	

[1] These instances can sustain the maximum performance for 30 minutes at least once every 24 hours, after which they revert to their baseline performance.

[2] These instances can sustain their stated performance indefinitely. If your workload requires sustained maximum performance for longer than 30 minutes, use one of these instances.

Memory optimized

> **Note**
>
> - R8g, R8gd, R8i, R8i-flex, X8g instance types support configurable bandwidth weightings. With these instance types, you can optimize an instance's bandwidth for either networking performance or Amazon EBS performance. The following table shows the default Amazon EBS bandwidth performance for these instance types. For more information, see EC2 instance bandwidth weighting configuration.
> - For maximum IOPS performance with U7i instances, we recommend that you use io2 BlockExpress volumes.

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r4.large [2]	425		53.125		3000	
r4.xlarge [2]	850		106.25		6000	
r4.2xlarge [2]	1700		212.5		12000	
r4.4xlarge [2]	3500		437.5		18750	
r4.8xlarge [2]	7000		875.0		37500	
r4.16xlarge [2]	14000		1750.0		75000	
r5.large [1]	650	4750	81.25	593.75	3600	18750
r5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5.4xlarge [2]	4750		593.75		18750	
r5.8xlarge [2]	6800		850.0		30000	
r5.12xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5.16xlarge [2]	13600		1700.0		60000	
r5.24xlarge [2]	19000		2375.0		80000	
r5.metal [2]	19000		2375.0		80000	
r5a.large [1]	650	2880	81.25	360.00	3600	16000
r5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5a.4xlarge [2]	2880		360.0		16000	
r5a.8xlarge [2]	4750		593.75		20000	
r5a.12xlarge [2]	6780		847.5		30000	
r5a.16xlarge [2]	9500		1187.5		40000	
r5a.24xlarge [2]	13570		1696.25		60000	
r5ad.large [1]	650	2880	81.25	360.00	3600	16000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5ad.4xlarge [2]	2880		360.0		16000	
r5ad.8xlarge [2]	4750		593.75		20000	
r5ad.12xlarge [2]	6780		847.5		30000	
r5ad.16xlarge [2]	9500		1187.5		40000	
r5ad.24xlarge [2]	13570		1696.25		60000	
r5b.large [1]	1250	10000	156.25	1250.00	5417	43333
r5b.xlarge [1]	2500	10000	312.50	1250.00	10833	43333
r5b.2xlarge [1]	5000	10000	625.00	1250.00	21667	43333
r5b.4xlarge [2]	10000		1250.0		43333	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5b.8xlarge [2]	20000		2500.0		86667	
r5b.12xlarge [2]	30000		3750.0		130000	
r5b.16xlarge [2]	40000		5000.0		173333	
r5b.24xlarge [2]	60000		7500.0		260000	
r5b.metal [2]	60000		7500.0		260000	
r5d.large [1]	650	4750	81.25	593.75	3600	18750
r5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5d.4xlarge [2]	4750		593.75		18750	
r5d.8xlarge [2]	6800		850.0		30000	
r5d.12xlarge [2]	9500		1187.5		40000	
r5d.16xlarge [2]	13600		1700.0		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5d.24xlarge [2]	19000		2375.0		80000	
r5d.metal [2]	19000		2375.0		80000	
r5dn.large [1]	650	4750	81.25	593.75	3600	18750
r5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5dn.4xlarge [2]	4750		593.75		18750	
r5dn.8xlarge [2]	6800		850.0		30000	
r5dn.12xlarge [2]	9500		1187.5		40000	
r5dn.16xlarge [2]	13600		1700.0		60000	
r5dn.24xlarge [2]	19000		2375.0		80000	
r5dn.metal [2]	19000		2375.0		80000	
r5n.large [1]	650	4750	81.25	593.75	3600	18750

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5n.4xlarge [2]	4750		593.75		18750	
r5n.8xlarge [2]	6800		850.0		30000	
r5n.12xlarge [2]	9500		1187.5		40000	
r5n.16xlarge [2]	13600		1700.0		60000	
r5n.24xlarge [2]	19000		2375.0		80000	
r5n.metal [2]	19000		2375.0		80000	
r6a.large [1]	650	10000	81.25	1250.00	3600	40000
r6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r6a.8xlarge [2]	10000		1250.0		40000	
r6a.12xlarge [2]	15000		1875.0		60000	
r6a.16xlarge [2]	20000		2500.0		80000	
r6a.24xlarge [2]	30000		3750.0		120000	
r6a.32xlarge [2]	40000		5000.0		160000	
r6a.48xlarge [2]	40000		5000.0		240000	
r6a.metal [2]	40000		5000.0		240000	
r6g.medium [1]	315	4750	39.38	593.75	2500	20000
r6g.large [1]	630	4750	78.75	593.75	3600	20000
r6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
r6g.4xlarge [2]	4750		593.75		20000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r6g.8xlarge [2]	9500		1187.5		40000	
r6g.12xlarge [2]	14250		1781.25		50000	
r6g.16xlarge [2]	19000		2375.0		80000	
r6g.metal [2]	19000		2375.0		80000	
r6gd.medium [1]	315	4750	39.38	593.75	2500	20000
r6gd.large [1]	630	4750	78.75	593.75	3600	20000
r6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
r6gd.4xlarge [2]	4750		593.75		20000	
r6gd.8xlarge [2]	9500		1187.5		40000	
r6gd.12xlarge [2]	14250		1781.25		50000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r6gd.16xlarge [2]	19000		2375.0		80000	
r6gd.metal [2]	19000		2375.0		80000	
r6i.large [1]	650	10000	81.25	1250.00	3600	40000
r6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6i.8xlarge [2]	10000		1250.0		40000	
r6i.12xlarge [2]	15000		1875.0		60000	
r6i.16xlarge [2]	20000		2500.0		80000	
r6i.24xlarge [2]	30000		3750.0		120000	
r6i.32xlarge [2]	40000		5000.0		160000	
r6i.metal [2]	40000		5000.0		160000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
r6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
r6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6idn.8xlarge [2]	25000		3125.0		100000	
r6idn.12xlarge [2]	37500		4687.5		150000	
r6idn.16xlarge [2]	50000		6250.0		200000	
r6idn.24xlarge [2]	75000		9375.0		300000	
r6idn.32xlarge [2]	100000		12500.0		400000	
r6idn.metal [2]	100000		12500.0		400000	
r6in.large [1]	1562	25000	195.31	3125.00	6250	100000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
r6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6in.8xlarge [2]	25000		3125.0		100000	
r6in.12xlarge [2]	37500		4687.5		150000	
r6in.16xlarge [2]	50000		6250.0		200000	
r6in.24xlarge [2]	75000		9375.0		300000	
r6in.32xlarge [2]	100000		12500.0		400000	
r6in.metal [2]	100000		12500.0		400000	
r6id.large [1]	650	10000	81.25	1250.00	3600	40000
r6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6id.8xlarge [2]	10000		1250.0		40000	
r6id.12xlarge [2]	15000		1875.0		60000	
r6id.16xlarge [2]	20000		2500.0		80000	
r6id.24xlarge [2]	30000		3750.0		120000	
r6id.32xlarge [2]	40000		5000.0		160000	
r6id.metal [2]	40000		5000.0		160000	
r7a.medium [1]	325	10000	40.62	1250.00	2500	40000
r7a.large [1]	650	10000	81.25	1250.00	3600	40000
r7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7a.8xlarge [2]	10000		1250.0		40000	
r7a.12xlarge [2]	15000		1875.0		60000	
r7a.16xlarge [2]	20000		2500.0		80000	
r7a.24xlarge [2]	30000		3750.0		120000	
r7a.32xlarge [2]	40000		5000.0		160000	
r7a.48xlarge [2]	40000		5000.0		240000	
r7a.metal-48xl [2]	40000		5000.0		240000	
r7g.medium [1]	315	10000	39.38	1250.00	2500	40000
r7g.large [1]	630	10000	78.75	1250.00	3600	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7g.8xlarge [2]	10000		1250.0		40000	
r7g.12xlarge [2]	15000		1875.0		60000	
r7g.16xlarge [2]	20000		2500.0		80000	
r7g.metal [2]	20000		2500.0		80000	
r7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r7gd.large [1]	630	10000	78.75	1250.00	3600	40000
r7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7gd.8xlarge [2]	10000		1250.0		40000	
r7gd.12xlarge [2]	15000		1875.0		60000	
r7gd.16xlarge [2]	20000		2500.0		80000	
r7gd.metal [2]	20000		2500.0		80000	
r7i.large [1]	650	10000	81.25	1250.00	3600	40000
r7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7i.8xlarge [2]	10000		1250.0		40000	
r7i.12xlarge [2]	15000		1875.0		60000	
r7i.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7i.24xlarge [2]	30000		3750.0		120000	
r7i.48xlarge [2]	40000		5000.0		240000	
r7i.metal-24xl [2]	30000		3750.0		120000	
r7i.metal-48xl [2]	40000		5000.0		240000	
r7iz.large [1]	792	10000	99.00	1250.00	3600	40000
r7iz.xlarge [1]	1584	10000	198.00	1250.00	6667	40000
r7iz.2xlarge [1]	3168	10000	396.00	1250.00	13333	40000
r7iz.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7iz.8xlarge [2]	10000		1250.0		40000	
r7iz.12xlarge [2]	19000		2375.0		76000	
r7iz.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7iz.32xlarge [2]	40000		5000.0		160000	
r7iz.metal-16xl [2]	20000		2500.0		80000	
r7iz.metal-32xl [2]	40000		5000.0		160000	
r8g.medium [1]	315	10000	39.38	1250.00	2500	40000
r8g.large [1]	630	10000	78.75	1250.00	3600	40000
r8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8g.8xlarge [2]	10000		1250.0		40000	
r8g.12xlarge [2]	15000		1875.0		60000	
r8g.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r8g.24xlarge [2]	30000		3750.0		120000	
r8g.48xlarge [2]	40000		5000.0		240000	
r8g.metal-24xl [2]	30000		3750.0		120000	
r8g.metal-48xl [2]	40000		5000.0		240000	
r8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r8gd.large [1]	630	10000	78.75	1250.00	3600	40000
r8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8gd.8xlarge [2]	10000		1250.0		40000	
r8gd.12xlarge [2]	15000		1875.0		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r8gd.16xlarge [2]	20000		2500.0		80000	
r8gd.24xlarge [2]	30000		3750.0		120000	
r8gd.48xlarge [2]	40000		5000.0		240000	
r8gd.metal-24xl [2]	30000		3750.0		120000	
r8gd.metal-48xl [2]	40000		5000.0		240000	
r8i.large [1]	650	10000	81.25	1250.00	3600	40000
r8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8i.8xlarge [2]	10000		1250.0		40000	
r8i.12xlarge [2]	15000		1875.0		60000	
r8i.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r8i.24xlarge [2]	30000		3750.0		120000	
r8i.32xlarge [2]	40000		5000.0		160000	
r8i.48xlarge [2]	60000		7500.0		240000	
r8i.96xlarge [2]	80000		10000.0		480000	
r8i.metal-48xl [2]	60000		7500.0		240000	
r8i.metal-96xl [2]	80000		10000.0		480000	
r8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
r8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
r8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
r8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
u-3tb1.56xlarge [2]	19000		2375.0		80000	
u-6tb1.56xlarge [2]	38000		4750.0		160000	
u-6tb1.112xlarge [2]	38000		4750.0		160000	
u-6tb1.metal [2]	38000		4750.0		160000	
u-9tb1.112xlarge [2]	38000		4750.0		160000	
u-9tb1.metal [2]	38000		4750.0		160000	
u-12tb1.112xlarge [2]	38000		4750.0		160000	
u-12tb1.metal [2]	38000		4750.0		160000	
u-18tb1.112xlarge [2]	38000		4750.0		160000	

<table>
<tr><th>Instance size</th><th>Baseline bandwidth (Mbps)</th><th>Maximum bandwidth (Mbps)</th><th>Baseline throughput (MB/s, 128 KiB I/O)</th><th>Maximum throughput (MB/s, 128 KiB I/O)</th><th>Baseline IOPS (16 KiB I/O)</th><th>Maximum IOPS (16 KiB I/O)</th></tr>
<tr><td>u-18tb1.metal [2]</td><td colspan="2">38000</td><td colspan="2">4750.0</td><td colspan="2">160000</td></tr>
<tr><td>u-24tb1.112xlarge [2]</td><td colspan="2">38000</td><td colspan="2">4750.0</td><td colspan="2">160000</td></tr>
<tr><td>u-24tb1.metal [2]</td><td colspan="2">38000</td><td colspan="2">4750.0</td><td colspan="2">160000</td></tr>
<tr><td>u7i-6tb.112xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">560000</td></tr>
<tr><td>u7i-8tb.112xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">560000</td></tr>
<tr><td>u7i-12tb.224xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">560000</td></tr>
<tr><td>u7in-16tb.224xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">560000</td></tr>
<tr><td>u7in-24tb.224xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">560000</td></tr>
<tr><td>u7in-32tb.224xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">560000</td></tr>
</table>

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
u7inh-32tb.480xlarge [2]	160000		20000.0		840000	
x1.16xlarge [2]	7000		875.0		40000	
x1.32xlarge [2]	14000		1750.0		80000	
x1e.xlarge [2]	500		62.5		3700	
x1e.2xlarge [2]	1000		125.0		7400	
x1e.4xlarge [2]	1750		218.75		10000	
x1e.8xlarge [2]	3500		437.5		20000	
x1e.16xlarge [2]	7000		875.0		40000	
x1e.32xlarge [2]	14000		1750.0		80000	
x2gd.medium [1]	315	4750	39.38	593.75	2500	20000
x2gd.large [1]	630	4750	78.75	593.75	3600	20000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
x2gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
x2gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
x2gd.4xlarge [2]	4750		593.75		20000	
x2gd.8xlarge [2]	9500		1187.5		40000	
x2gd.12xlarge [2]	14250		1781.25		60000	
x2gd.16xlarge [2]	19000		2375.0		80000	
x2gd.metal [2]	19000		2375.0		80000	
x2idn.16xlarge [2]	40000		5000.0		173333	
x2idn.24xlarge [2]	60000		7500.0		260000	
x2idn.32xlarge [2]	80000		10000.0		260000	
x2idn.metal [2]	80000		10000.0		260000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
x2iedn.xlarge [1]	2500	20000	312.50	2500.00	8125	65000
x2iedn.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000
x2iedn.4xlarge [1]	10000	20000	1250.00	2500.00	32500	65000
x2iedn.8xlarge [2]	20000		2500.0		65000	
x2iedn.16xlarge [2]	40000		5000.0		130000	
x2iedn.24xlarge [2]	60000		7500.0		195000	
x2iedn.32xlarge [2]	80000		10000.0		260000	
x2iedn.metal [2]	80000		10000.0		260000	
x2iezn.2xlarge [2]	3170		396.25		13333	
x2iezn.4xlarge [2]	4750		593.75		20000	
x2iezn.6xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
x2iezn.8xlarge [2]	12000		1500.0		55000	
x2iezn.12xlarge [2]	19000		2375.0		80000	
x2iezn.metal [2]	19000		2375.0		80000	
x8g.medium [1]	315	10000	39.38	1250.00	2500	40000
x8g.large [1]	630	10000	78.75	1250.00	3600	40000
x8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
x8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
x8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
x8g.8xlarge [2]	10000		1250.0		40000	
x8g.12xlarge [2]	15000		1875.0		60000	
x8g.16xlarge [2]	20000		2500.0		80000	

<table>
<tr><th>Instance size</th><th>Baseline bandwidth (Mbps)</th><th>Maximum bandwidth (Mbps)</th><th>Baseline throughput (MB/s, 128 KiB I/O)</th><th>Maximum throughput (MB/s, 128 KiB I/O)</th><th>Baseline IOPS (16 KiB I/O)</th><th>Maximum IOPS (16 KiB I/O)</th></tr>
<tr><td>x8g.24xlarge [2]</td><td colspan="2">30000</td><td colspan="2">3750.0</td><td colspan="2">120000</td></tr>
<tr><td>x8g.48xlarge [2]</td><td colspan="2">40000</td><td colspan="2">5000.0</td><td colspan="2">240000</td></tr>
<tr><td>x8g.metal-24xl [2]</td><td colspan="2">30000</td><td colspan="2">3750.0</td><td colspan="2">120000</td></tr>
<tr><td>x8g.metal-48xl [2]</td><td colspan="2">40000</td><td colspan="2">5000.0</td><td colspan="2">240000</td></tr>
<tr><td>z1d.large [1]</td><td>800</td><td>3170</td><td>100.00</td><td>396.25</td><td>3333</td><td>13333</td></tr>
<tr><td>z1d.xlarge [1]</td><td>1580</td><td>3170</td><td>197.50</td><td>396.25</td><td>6667</td><td>13333</td></tr>
<tr><td>z1d.2xlarge [2]</td><td colspan="2">3170</td><td colspan="2">396.25</td><td colspan="2">13333</td></tr>
<tr><td>z1d.3xlarge [2]</td><td colspan="2">4750</td><td colspan="2">593.75</td><td colspan="2">20000</td></tr>
<tr><td>z1d.6xlarge [2]</td><td colspan="2">9500</td><td colspan="2">1187.5</td><td colspan="2">40000</td></tr>
<tr><td>z1d.12xlarge [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
<tr><td>z1d.metal [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
</table>

[1] These instances can sustain the maximum performance for 30 minutes at least once every 24 hours, after which they revert to their baseline performance.

[2] These instances can sustain their stated performance indefinitely. If your workload requires sustained maximum performance for longer than 30 minutes, use one of these instances.

Storage optimized

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
d2.xlarge [2]	750		93.75		6000	
d2.2xlarge [2]	1000		125.0		8000	
d2.4xlarge [2]	2000		250.0		16000	
d2.8xlarge [2]	4000		500.0		32000	
d3.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3.4xlarge [2]	2800		350.0		15000	
d3.8xlarge [2]	5000		625.0		30000	
d3en.xlarge [1]	850	2800	106.25	350.00	5000	15000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
d3en.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3en.4xlarge [2]	2800		350.0		15000	
d3en.6xlarge [2]	4000		500.0		25000	
d3en.8xlarge [2]	5000		625.0		30000	
d3en.12xlarge [2]	7000		875.0		40000	
h1.2xlarge [2]	1750		218.75		12000	
h1.4xlarge [2]	3500		437.5		20000	
h1.8xlarge [2]	7000		875.0		40000	
h1.16xlarge [2]	14000		1750.0		80000	
i3.large [2]	425		53.125		3000	
i3.xlarge [2]	850		106.25		6000	
i3.2xlarge [2]	1700		212.5		12000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i3.4xlarge [2]	3500		437.5		16000	
i3.8xlarge [2]	7000		875.0		32500	
i3.16xlarge [2]	14000		1750.0		65000	
i3.metal [2]	19000		2375.0		80000	
i3en.large [1]	576	4750	72.10	593.75	3000	20000
i3en.xlarge [1]	1153	4750	144.20	593.75	6000	20000
i3en.2xlarge [1]	2307	4750	288.39	593.75	12000	20000
i3en.3xlarge [1]	3800	4750	475.00	593.75	15000	20000
i3en.6xlarge [2]	4750		593.75		20000	
i3en.12xlarge [2]	9500		1187.5		40000	
i3en.24xlarge [2]	19000		2375.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i3en.metal [2]	19000		2375.0		80000	
i4g.large [1]	625	10000	78.12	1250.00	2500	40000
i4g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4g.8xlarge [2]	10000		1250.0		40000	
i4g.16xlarge [2]	20000		2500.0		80000	
i4i.large [1]	625	10000	78.12	1250.00	2500	40000
i4i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4i.8xlarge [2]	10000		1250.0		40000	
i4i.12xlarge [2]	15000		1875.0		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i4i.16xlarge [2]	20000		2500.0		80000	
i4i.24xlarge [2]	30000		3750.0		120000	
i4i.32xlarge [2]	40000		5000.0		160000	
i4i.metal [2]	40000		5000.0		160000	
i7i.large [1]	625	10000	78.12	1250.00	2500	40000
i7i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i7i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i7i.8xlarge [2]	10000		1250.0		40000	
i7i.12xlarge [2]	15000		1875.0		60000	
i7i.16xlarge [2]	20000		2500.0		80000	
i7i.24xlarge [2]	30000		3750.0		120000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i7i.48xlarge [2]	60000		7500.0		240000	
i7i.metal-24xl [2]	30000		3750.0		120000	
i7i.metal-48xl [2]	60000		7500.0		240000	
i7ie.large [1]	625	10000	78.12	1250.00	2500	40000
i7ie.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i7ie.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i7ie.3xlarge [1]	3750	10000	468.75	1250.00	15000	40000
i7ie.6xlarge [1]	7500	10000	937.50	1250.00	30000	40000
i7ie.12xlarge [2]	15000		1875.0		60000	
i7ie.18xlarge [2]	22500		2812.5		90000	
i7ie.24xlarge [2]	30000		3750.0		120000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i7ie.48xlarge [2]	60000		7500.0		240000	
i7ie.metal-24xl [2]	30000		3750.0		120000	
i7ie.metal-48xl [2]	60000		7500.0		240000	
i8g.large [1]	625	10000	78.12	1250.00	2500	40000
i8g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i8g.8xlarge [2]	10000		1250.0		40000	
i8g.12xlarge [2]	15000		1875.0		60000	
i8g.16xlarge [2]	20000		2500.0		80000	
i8g.24xlarge [2]	30000		3750.0		120000	
i8g.48xlarge [2]	60000		7500.0		240000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i8g.metal-24xl [2]	30000		3750.0		120000	
i8ge.large [1]	625	10000	78.12	1250.00	2500	40000
i8ge.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8ge.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8ge.3xlarge [1]	3750	10000	468.75	1250.00	15000	40000
i8ge.6xlarge [1]	7500	10000	937.50	1250.00	30000	40000
i8ge.12xlarge [2]	15000		1875.0		60000	
i8ge.18xlarge [2]	22500		2812.5		90000	
i8ge.24xlarge [2]	30000		3750.0		120000	
i8ge.48xlarge [2]	60000		7500.0		240000	
i8ge.metal-24xl [2]	30000		3750.0		120000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
i8ge.metal-48xl [2]	60000		7500.0		240000	
im4gn.large [1]	1250	10000	156.25	1250.00	5000	40000
im4gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
im4gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
im4gn.4xlarge [2]	10000		1250.0		40000	
im4gn.8xlarge [2]	20000		2500.0		80000	
im4gn.16xlarge [2]	40000		5000.0		160000	
is4gen.medium [1]	625	10000	78.12	1250.00	2500	40000
is4gen.large [1]	1250	10000	156.25	1250.00	5000	40000
is4gen.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
is4gen.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000

<table>
<tr><th>Instance size</th><th>Baseline bandwidth (Mbps)</th><th>Maximum bandwidth (Mbps)</th><th>Baseline throughput (MB/s, 128 KiB I/O)</th><th>Maximum throughput (MB/s, 128 KiB I/O)</th><th>Baseline IOPS (16 KiB I/O)</th><th>Maximum IOPS (16 KiB I/O)</th></tr>
<tr><td>is4gen.4xlarge [2]</td><td colspan="2">10000</td><td colspan="2">1250.0</td><td colspan="2">40000</td></tr>
<tr><td>is4gen.8xlarge [2]</td><td colspan="2">20000</td><td colspan="2">2500.0</td><td colspan="2">80000</td></tr>
</table>

[1] These instances can sustain the maximum performance for 30 minutes at least once every 24 hours, after which they revert to their baseline performance.

[2] These instances can sustain their stated performance indefinitely. If your workload requires sustained maximum performance for longer than 30 minutes, use one of these instances.

Accelerated computing

<table>
<tr><th>Instance size</th><th>Baseline bandwidth (Mbps)</th><th>Maximum bandwidth (Mbps)</th><th>Baseline throughput (MB/s, 128 KiB I/O)</th><th>Maximum throughput (MB/s, 128 KiB I/O)</th><th>Baseline IOPS (16 KiB I/O)</th><th>Maximum IOPS (16 KiB I/O)</th></tr>
<tr><td>dl1.24xlarge [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
<tr><td>dl2q.24xlarge [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
<tr><td>f1.2xlarge [2]</td><td colspan="2">1700</td><td colspan="2">212.5</td><td colspan="2">12000</td></tr>
<tr><td>f1.4xlarge [2]</td><td colspan="2">3500</td><td colspan="2">437.5</td><td colspan="2">44000</td></tr>
</table>

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
f1.16xlarge [2]	14000		1750.0		75000	
f2.6xlarge [2]	7500		937.5		30000	
f2.12xlarge [2]	15000		1875.0		60000	
f2.48xlarge [2]	60000		7500.0		240000	
g3.4xlarge [2]	3500		437.5		20000	
g3.8xlarge [2]	7000		875.0		40000	
g3.16xlarge [2]	14000		1750.0		80000	
g4ad.xlarge [1]	400	3170	50.00	396.25	1700	13333
g4ad.2xlarge [1]	800	3170	100.00	396.25	3400	13333
g4ad.4xlarge [1]	1580	3170	197.50	396.25	6700	13333
g4ad.8xlarge [2]	3170		396.25		13333	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
g4ad.16xlarge [2]	6300		787.5		26667	
g4dn.xlarge [1]	950	3500	118.75	437.50	3000	20000
g4dn.2xlarge [1]	1150	3500	143.75	437.50	6000	20000
g4dn.4xlarge [2]	4750		593.75		20000	
g4dn.8xlarge [2]	9500		1187.5		40000	
g4dn.12xlarge [2]	9500		1187.5		40000	
g4dn.16xlarge [2]	9500		1187.5		40000	
g4dn.metal [2]	19000		2375.0		80000	
g5.xlarge [1]	700	3500	87.50	437.50	3000	15000
g5.2xlarge [1]	850	3500	106.25	437.50	3500	15000
g5.4xlarge [2]	4750		593.75		20000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
g5.8xlarge [2]	16000		2000.0		65000	
g5.12xlarge [2]	16000		2000.0		65000	
g5.16xlarge [2]	16000		2000.0		65000	
g5.24xlarge [2]	19000		2375.0		80000	
g5.48xlarge [2]	19000		2375.0		80000	
g5g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
g5g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
g5g.4xlarge [2]	4750		593.75		20000	
g5g.8xlarge [2]	9500		1187.5		40000	
g5g.16xlarge [2]	19000		2375.0		80000	
g5g.metal [2]	19000		2375.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
g6.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6.4xlarge [2]	8000		1000.0		32000	
g6.8xlarge [2]	16000		2000.0		64000	
g6.12xlarge [2]	20000		2500.0		80000	
g6.16xlarge [2]	20000		2500.0		80000	
g6.24xlarge [2]	30000		3750.0		120000	
g6.48xlarge [2]	60000		7500.0		240000	
g6e.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6e.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6e.4xlarge [2]	8000		1000.0		32000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
g6e.8xlarge [2]	16000		2000.0		64000	
g6e.12xlarge [2]	20000		2500.0		80000	
g6e.16xlarge [2]	20000		2500.0		80000	
g6e.24xlarge [2]	30000		3750.0		120000	
g6e.48xlarge [2]	60000		7500.0		240000	
g6f.large [1]	936	5000	117.00	625.00	3750	20000
g6f.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6f.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6f.4xlarge [2]	6000		750.0		24000	
gr6.4xlarge [2]	8000		1000.0		32000	
gr6.8xlarge [2]	16000		2000.0		64000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
gr6f.4xlarge [2]	8000		1000.0		32000	
inf1.xlarge [1]	1190	4750	148.75	593.75	4000	20000
inf1.2xlarge [1]	1190	4750	148.75	593.75	6000	20000
inf1.6xlarge [2]	4750		593.75		20000	
inf1.24xlarge [2]	19000		2375.0		80000	
inf2.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
inf2.8xlarge [2]	10000		1250.0		40000	
inf2.24xlarge [2]	30000		3750.0		120000	
inf2.48xlarge [2]	60000		7500.0		240000	
p3.2xlarge [2]	1750		218.75		10000	
p3.8xlarge [2]	7000		875.0		40000	

<table>
<tr><th>Instance size</th><th>Baseline bandwidth (Mbps)</th><th>Maximum bandwidth (Mbps)</th><th>Baseline throughput (MB/s, 128 KiB I/O)</th><th>Maximum throughput (MB/s, 128 KiB I/O)</th><th>Baseline IOPS (16 KiB I/O)</th><th>Maximum IOPS (16 KiB I/O)</th></tr>
<tr><td>p3.16xlarge [2]</td><td colspan="2">14000</td><td colspan="2">1750.0</td><td colspan="2">80000</td></tr>
<tr><td>p3dn.24xlarge [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
<tr><td>p4d.24xlarge [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
<tr><td>p4de.24xlarge [2]</td><td colspan="2">19000</td><td colspan="2">2375.0</td><td colspan="2">80000</td></tr>
<tr><td>p5.4xlarge [2]</td><td colspan="2">10000</td><td colspan="2">1250.0</td><td colspan="2">32500</td></tr>
<tr><td>p5.48xlarge [2]</td><td colspan="2">80000</td><td colspan="2">10000.0</td><td colspan="2">260000</td></tr>
<tr><td>p5e.48xlarge [2]</td><td colspan="2">80000</td><td colspan="2">10000.0</td><td colspan="2">260000</td></tr>
<tr><td>p5en.48xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">400000</td></tr>
<tr><td>p6-b200.48xlarge [2]</td><td colspan="2">100000</td><td colspan="2">12500.0</td><td colspan="2">400000</td></tr>
<tr><td>p6e-gb200.36xlarge [2]</td><td colspan="2">60000</td><td colspan="2">7500.0</td><td colspan="2">240000</td></tr>
<tr><td>trn1.2xlarge [1]</td><td>5000</td><td>20000</td><td>625.00</td><td>2500.00</td><td>16250</td><td>65000</td></tr>
</table>

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
trn1.32xlarge [2]	80000		10000.0		260000	
trn1n.32xlarge [2]	80000		10000.0		260000	
trn2.48xlarge [2]	80000		10000.0		260000	
trn2u.48xlarge [2]	80000		10000.0		260000	
vt1.3xlarge [1]	2375	4750	296.88	593.75	10000	20000
vt1.6xlarge [2]	4750		593.75		20000	
vt1.24xlarge [2]	19000		2375.0		80000	

[1] These instances can sustain the maximum performance for 30 minutes at least once every 24 hours, after which they revert to their baseline performance.

[2] These instances can sustain their stated performance indefinitely. If your workload requires sustained maximum performance for longer than 30 minutes, use one of these instances.

High-performance computing

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
hpc6a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc6id.32xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.12xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.24xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.96xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.4xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.8xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.16xlarge [1]	87	2085	10.88	260.62	500	11000

[1] These instances can sustain the maximum performance for 30 minutes at least once every 24 hours, after which they revert to their baseline performance.

[2] These instances can sustain their stated performance indefinitely. If your workload requires sustained maximum performance for longer than 30 minutes, use one of these instances.

EBS optimization supported

The following instance types support EBS optimization but EBS optimization is not enabled by default. You must enable EBS optimization, at an additional hourly fee, during or after launch to achieve the level of EBS performance described.

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
c1.xlarge	1000	125.0	8000
c3.xlarge	500	62.5	4000
c3.2xlarge	1000	125.0	8000
c3.4xlarge	2000	250.0	16000
i2.xlarge	500	62.5	4000
i2.2xlarge	1000	125.0	8000
i2.4xlarge	2000	250.0	16000
m1.large	500	62.5	4000
m1.xlarge	1000	125.0	8000
m2.2xlarge	500	62.5	4000
m2.4xlarge	1000	125.0	8000
m3.xlarge	500	62.5	4000
m3.2xlarge	1000	125.0	8000
r3.xlarge	500	62.5	4000
r3.2xlarge	1000	125.0	8000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
r3.4xlarge	2000	250.0	16000

Note

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 the maximum Amazon EBS optimized performance

An instance's EBS performance is bounded by the instance type's performance limits, or the aggregated performance of its attached volumes, whichever is smaller. To achieve maximum EBS performance, an instance must have attached volumes that provide a combined performance equal to or greater than the maximum instance performance. For example, to achieve `80,000` IOPS for `r6i.16xlarge`, the instance must have at least 5 gp3 volumes provisioned with `16,000` IOPS each (5 volumes x `16,000` IOPS = `80,000` IOPS). We recommend that you choose an instance type 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.

Important

When using configurable bandwidth weightings, the EBS bandwidth limits for your instance might change. For instances with the `VPC-1` weighting configuration, which increases networking bandwidth, you might experience lower than expected IOPS for EBS volumes due to reaching the EBS bandwidth limit before the IOPS limit. This is particularly noticeable with larger I/O sizes. Always test your specific workload to ensure it meets your performance requirements with your selected bandwidth weighting. For more information, see EC2 instance bandwidth weighting configuration.

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.

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, Inf1, M5a, M5ad, R5a, R5ad, T3, T3a, and Z1d instances launched after February 26, 2020 provide the maximum EBS optimized performance. To get the maximum performance from an instance launched before February 26, 2020, stop and start it.
- C5, C5d, C5n, M5, M5d, M5n, M5dn, R5, R5d, R5n, R5dn, and P3dn instances launched after December 3, 2019 provide the maximum EBS optimized performance. 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 maximum EBS optimized performance. 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.

Find EBS-optimized EC2 instance types

You can view the instances types that support EBS optimization in each Region.

Console

To find instance types that are EBS-optimized by default

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instance Types**.
3. Add the filter **EBS optimization support = default**.

4. (Optional) Click the **Preferences** icon and then turn on the relevant columns, such as **EBS Maximum IOPS** and **EBS Baseline IOPS**.
5. (Optional) Add filters to further scope to specific instance types of interest.

AWS CLI

To find instance types that are EBS-optimized by default

Use the following [describe-instance-types](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-types.html) command.

```
aws ec2 describe-instance-types \
    --filters Name=ebs-info.ebs-optimized-support,Values=default  \
    --query 'InstanceTypes[].{InstanceType:InstanceType, \
        "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,\
        MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops,\
        "MaxThroughput(MB/s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
    --output=table
```

To find instance types that optionally support EBS optimization

Use the following [describe-instance-types](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-types.html) command.

```
aws ec2 describe-instance-types \
    --filters Name=ebs-info.ebs-optimized-support,Values=supported \
    --query 'InstanceTypes[].{InstanceType:InstanceType, \
        "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps, \
        MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops, \
        "MaxThroughput(MB/s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
    --output=table
```

The following is example output for `eu-west-1`.

```
--------------------------------------------------------------------------
|                          DescribeInstanceTypes                         |
+--------------+----------------------+----------+-----------------------+
| InstanceType | MaxBandwidth(Mb/s)   | MaxIOPS  |  MaxThroughput(MB/s)  |
+--------------+----------------------+----------+-----------------------+
|  i2.2xlarge  |  1000                |  8000    |  125.0                |
|  m2.4xlarge  |  1000                |  8000    |  125.0                |
|  m2.2xlarge  |  500                 |  4000    |  62.5                 |
|  c1.xlarge   |  1000                |  8000    |  125.0                |
```

```
|  i2.xlarge   |  500                   |  4000    |  62.5                   |
|  m3.xlarge   |  500                   |  4000    |  62.5                   |
|  m1.xlarge   |  1000                  |  8000    |  125.0                  |
|  r3.4xlarge  |  2000                  |  16000   |  250.0                  |
|  r3.2xlarge  |  1000                  |  8000    |  125.0                  |
|  c3.xlarge   |  500                   |  4000    |  62.5                   |
|  m3.2xlarge  |  1000                  |  8000    |  125.0                  |
|  r3.xlarge   |  500                   |  4000    |  62.5                   |
|  i2.4xlarge  |  2000                  |  16000   |  250.0                  |
|  c3.4xlarge  |  2000                  |  16000   |  250.0                  |
|  c3.2xlarge  |  1000                  |  8000    |  125.0                  |
|  m1.large    |  500                   |  4000    |  62.5                   |
+--------------+-----------------------+----------+-----------------------+
```

PowerShell

To find instance types that are EBS-optimized by default

Use the Get-EC2InstanceType cmdlet.

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="default"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

To find instance types that optionally support EBS optimization

Use the Get-EC2InstanceType cmdlet.

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="supported"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

The following is example output for `eu-west-1`.

```
InstanceType MaxBandwidth(Mb/s) MaxIOPS MaxThroughput (MB/s)
------------ ------------------ ------- --------------------
m2.4xlarge                 1000    8000              125.000
i2.2xlarge                 1000    8000              125.000
c1.xlarge                  1000    8000              125.000
m2.2xlarge                  500    4000               62.500
r3.2xlarge                 1000    8000              125.000
m3.xlarge                   500    4000               62.500
r3.4xlarge                 2000   16000              250.000
m1.xlarge                  1000    8000              125.000
i2.xlarge                   500    4000               62.500
c3.xlarge                   500    4000               62.500
c3.4xlarge                 2000   16000              250.000
c3.2xlarge                 1000    8000              125.000
i2.4xlarge                 2000   16000              250.000
r3.xlarge                   500    4000               62.500
m3.2xlarge                 1000    8000              125.000
m1.large                    500    4000               62.500
```

Enable EBS optimization for an Amazon EC2 instance

You can manually enable EBS optimization only for previous generation instances types that optionally support EBS optimization. If you enable EBS optimization for these instance types, there is additional hourly fee

Prerequisites

- Verify that the instance type requires that you enable EBS optimization. For more information, see EBS optimization supported.
- To you enable EBS optimization after launch, you must stop the 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.

Console

To enable Amazon EBS optimization during launch

In the Launch instances wizard, select the required instance type. Expand the **Advanced details** section, then for **EBS-optimized instance**, select **Enable**.

If the selected instance type does not support Amazon EBS optimization, the drop-down is disabled. If the instance type is Amazon EBS-optimized by default, Enable is already selected.

To enable Amazon EBS optimization after launch

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. Stop the instance. Choose **Actions**, **Instance state**, **Stop instance**.
4. With the instance still selected, choose **Actions**, **Instance settings**, **Change instance type**.
5. Select **EBS-optimized** and then choose **Apply**.

   If the instance type is Amazon EBS-optimized by default, or if it does not support Amazon EBS optimization, the checkbox is disabled.
6. Restart the instance. Choose **Instance state**, **Start instance**.

AWS CLI

To enable Amazon EBS optimization during launch

Use the run-instances command with the `--ebs-optimized` option.

To enable Amazon EBS optimization after launch

1. If the instance is running, stop it by using the stop-instances command.

   ```
   aws ec2 stop-instances --instance-ids i-1234567890abcdef0
   ```

2. Enable EBS optimization by using the modify-instance-attribute command with the `--ebs-optimized` option.

   ```
   aws ec2 modify-instance-attribute \
       --instance-id i-1234567890abcdef0 \
       --ebs-optimized
   ```

PowerShell

To enable Amazon EBS optimization during launch

Use the New-EC2Instance cmdlet with the `-EbsOptimized` option.

To enable Amazon EBS optimization after launch

1. If the instance is running, stop it by using the Stop-EC2Instance cmdlet.

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

2. Enable EBS optimization by using the Edit-EC2InstanceAttribute cmdlet with the `-EbsOptimized` option.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -EbsOptimized $true
```

CPU options for Amazon EC2 instances

Many Amazon EC2 instances support simultaneous multithreading (SMT), 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.

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, to optimize your instance for specific workloads or business needs, you can specify the following CPU options during and after instance launch:

- **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 SMT by specifying a single thread per CPU core. You might do this for certain workloads, such as high performance computing (HPC) workloads.

Considerations

- You can't modify the number of threads per core for T2, C7a, M7a, R7a, and Apple silicon Mac instances, and instances based on the AWS Graviton processor.
- The number of instances that you can run is based on the default vCPUs for the instance types used. How we calculate the vCPUs consumed by an instance is not affected by changing its CPU options.

Pricing

There is no additional or reduced charge for specifying CPU options. You're charged the same as instances that are launched with the default CPU options.

Contents

Rules for specifying CPU options for an Amazon EC2 instance

To specify the CPU options for your instance, be aware of the following rules:

- You can't specify CPU options for bare metal instances.
- You can specify CPU options both during and after instance launch.
- To configure CPU options, you must specify both the number of CPU cores and threads per core in the request. For example requests, see Specify CPU options for an Amazon EC2 instance.
- 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 Supported CPU options for Amazon EC2 instance types.
- To disable simultaneous multithreading (SMT), also referred to as hyper-threading, specify one thread per core.
- In the console, when you change the instance type of an existing instance, Amazon EC2 applies the CPU option settings from the existing instance to the new instance, if possible. If the new

instance type doesn't support those settings, the CPU options are reset to **Use default CPU options**. This option uses the default number of vCPUs for the new instance type.

To update settings for the new instance, select **Specify CPU options** under **Advanced details** in the **Change instance type** view.

- The specified CPU options persist after you stop, start, or reboot an instance.

Supported CPU options for Amazon EC2 instance types

The following tables list the instance types that support specifying CPU options.

Contents

General purpose instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m2.xlarge`	2	2	1	1, 2	1
`m2.2xlarge`	4	4	1	1, 2, 3, 4	1
`m2.4xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`m3.large`	2	1	2	1	1, 2
`m3.xlarge`	4	2	2	1, 2	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m4.large	2	1	2	1	1, 2
m4.xlarge	4	2	2	1, 2	1, 2
m4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m4.10xlarge	40	20	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	1, 2
m4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5.large	2	1	2	1	1, 2
m5.xlarge	4	2	2	2	1, 2
m5.2xlarge	8	4	2	2, 4	1, 2
m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5a.large	2	1	2	1	1, 2
m5a.xlarge	4	2	2	2	1, 2
m5a.2xlarge	8	4	2	2, 4	1, 2
m5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5a.12xlarge	48	24	2	6, 12, 18, 24	1, 2
m5a.16xlarge	64	32	2	8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
m5ad.large	2	1	2	1	1, 2
m5ad.xlarge	4	2	2	2	1, 2
m5ad.2xlarge	8	4	2	2, 4	1, 2
m5ad.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5ad.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
m5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2
m5ad.16xlarge	64	32	2	8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5ad.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
m5d.large	2	1	2	1	1, 2
m5d.xlarge	4	2	2	2	1, 2
m5d.2xlarge	8	4	2	2, 4	1, 2
m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m5d.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`m5dn.large`	2	1	2	1	1, 2
`m5dn.xlarge`	4	2	2	1, 2	1, 2
`m5dn.2xlarge`	8	4	2	2, 4	1, 2
`m5dn.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`m5dn.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`m5dn.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`m5dn.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5dn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5n.large	2	1	2	1	1, 2
m5n.xlarge	4	2	2	1, 2	1, 2
m5n.2xlarge	8	4	2	2, 4	1, 2
m5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5n.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5n.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5n.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5n.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5zn.large	2	1	2	1	1, 2
m5zn.xlarge	4	2	2	1, 2	1, 2
m5zn.2xlarge	8	4	2	2, 4	1, 2
m5zn.3xlarge	12	6	2	2, 4, 6	1, 2
m5zn.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
m5zn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m6a.large	2	1	2	1	1, 2
m6a.xlarge	4	2	2	1, 2	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m6a.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m6a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
m6a.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
m6a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2
m6a.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
m6a.32xlarge	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
m6a.48xlarge	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
m6g.large	2	2	1	1, 2	1
m6g.xlarge	4	4	1	1, 2, 3, 4	1
m6g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m6g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m6g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m6gd.large	2	2	1	1, 2	1
m6gd.xlarge	4	4	1	1, 2, 3, 4	1
m6gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m6gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m6gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m6i.large	2	1	2	1	1, 2
m6i.xlarge	4	2	2	1, 2	1, 2
m6i.2xlarge	8	4	2	2, 4	1, 2
m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6id.large	2	1	2	1	1, 2
m6id.xlarge	4	2	2	1, 2	1, 2
m6id.2xlarge	8	4	2	2, 4	1, 2
m6id.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m6id.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m6id.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m6id.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m6id.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6id.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
m6idn.large	2	1	2	1	1, 2
m6idn.xlarge	4	2	2	1, 2	1, 2
m6idn.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m6idn.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m6idn.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6idn.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
m6idn.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
m6idn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6idn.32xlarge	128	64	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	1, 2
m6in.large	2	1	2	1	1, 2
m6in.xlarge	4	2	2	1, 2	1, 2
m6in.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m6in.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
m6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m6in.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
`m6in.16xlarge`	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
`m6in.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6in.32xlarge	128	64	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	1, 2
m7a.large	2	2	1	1, 2	1
m7a.xlarge	4	4	1	1, 2, 3, 4	1
m7a.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m7a.4xlarge	16	16	1	1, 2, 4, 6, 8, 10, 12, 14, 16	1
m7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
m7a.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7a.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32, 40, 48, 56, 64	1
m7a.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36, 48, 60, 72, 84, 96	1
m7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1
m7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	1
m7g.large	2	2	1	1, 2	1
m7g.xlarge	4	4	1	1, 2, 3, 4	1
m7g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m7g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m7g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m7gd.large	2	2	1	1, 2	1
m7gd.xlarge	4	4	1	1, 2, 3, 4	1
m7gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m7gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
m7gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
m7i.large	2	1	2	1	1, 2
m7i.xlarge	4	2	2	1, 2	1, 2
m7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
m7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
m7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
m7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m7i.24xlarge`	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
`m7i.48xlarge`	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m7i-flex.large`	2	1	2	1	1, 2
`m7i-flex.xlarge`	4	2	2	1, 2	1, 2
`m7i-flex.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`m7i-flex.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`m7i-flex.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`m7i-flex.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m7i-flex.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
m8g.large	2	2	1	1, 2	1
m8g.xlarge	4	4	1	1, 2, 3, 4	1
m8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m8g.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8g.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
m8gd.large	2	2	1	1, 2	1
m8gd.xlarge	4	4	1	1, 2, 3, 4	1
m8gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
m8gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
m8gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8gd.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8gd.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
`m8i.large`	2	1	2	1	1, 2
`m8i.xlarge`	4	2	2	1, 2	1, 2
`m8i.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`m8i.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`m8i.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`m8i.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
m8i.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
m8i.32xlarge	128	64	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	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m8i.48xlarge	192	96	2	6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96	1, 2
m8i.96xlarge	384	192	2	12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 78, 84, 90, 96, 102, 108, 114, 120, 126, 132, 138, 144, 150, 156, 162, 168, 174, 180, 186, 192	1, 2
m8i-flex.large	2	1	2	1	1, 2
m8i-flex.xlarge	4	2	2	1, 2	1, 2
m8i-flex.2xlarge	8	4	2	1, 2, 3, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`m8i-flex.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`m8i-flex.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`m8i-flex.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
`m8i-flex.16xlarge`	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
`t3.nano`	2	1	2	1	1, 2
`t3.micro`	2	1	2	1	1, 2
`t3.small`	2	1	2	1	1, 2
`t3.medium`	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
t3.large	2	1	2	1	1, 2
t3.xlarge	4	2	2	2	1, 2
t3.2xlarge	8	4	2	2, 4	1, 2
t3a.nano	2	1	2	1	1, 2
t3a.micro	2	1	2	1	1, 2
t3a.small	2	1	2	1	1, 2
t3a.medium	2	1	2	1	1, 2
t3a.large	2	1	2	1	1, 2
t3a.xlarge	4	2	2	2	1, 2
t3a.2xlarge	8	4	2	2, 4	1, 2
t4g.nano	2	2	1	1, 2	1
t4g.micro	2	2	1	1, 2	1
t4g.small	2	2	1	1, 2	1
t4g.medium	2	2	1	1, 2	1
t4g.large	2	2	1	1, 2	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`t4g.xlarge`	4	4	1	1, 2, 3, 4	1
`t4g.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Compute optimized instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c3.large`	2	1	2	1	1, 2
`c3.xlarge`	4	2	2	1, 2	1, 2
`c3.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`c3.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`c3.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`c4.large`	2	1	2	1	1, 2
`c4.xlarge`	4	2	2	1, 2	1, 2
`c4.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`c4.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c4.8xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5.large`	2	1	2	1	1, 2
`c5.xlarge`	4	2	2	2	1, 2
`c5.2xlarge`	8	4	2	2, 4	1, 2
`c5.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c5.9xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5.12xlarge`	48	24	2	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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c5.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
c5a.large	2	1	2	1	1, 2
c5a.xlarge	4	2	2	1, 2	1, 2
c5a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5a.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5a.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5a.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5a.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c5a.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2
c5ad.large	2	1	2	1	1, 2
c5ad.xlarge	4	2	2	1, 2	1, 2
c5ad.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5ad.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5ad.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5ad.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5ad.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
c5ad.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c5d.large`	2	1	2	1	1, 2
`c5d.xlarge`	4	2	2	2	1, 2
`c5d.2xlarge`	8	4	2	2, 4	1, 2
`c5d.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c5d.9xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5d.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`c5d.18xlarge`	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c5d.24xlarge`	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`c5n.large`	2	1	2	1	1, 2
`c5n.xlarge`	4	2	2	2	1, 2
`c5n.2xlarge`	8	4	2	2, 4	1, 2
`c5n.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c5n.9xlarge`	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
`c5n.18xlarge`	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
`c6a.large`	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6a.xlarge`	4	2	2	1, 2	1, 2
`c6a.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`c6a.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`c6a.8xlarge`	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
`c6a.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
`c6a.16xlarge`	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2
`c6a.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
`c6a.32xlarge`	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
`c6a.48xlarge`	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
`c6g.large`	2	2	1	1, 2	1
`c6g.xlarge`	4	4	1	1, 2, 3, 4	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6g.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`c6g.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`c6g.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6g.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6g.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
`c6gd.large`	2	2	1	1, 2	1
`c6gd.xlarge`	4	4	1	1, 2, 3, 4	1
`c6gd.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c6gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c6gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c6gn.medium	1	1	1	1	1
c6gn.large	2	2	1	1, 2	1
c6gn.xlarge	4	4	1	1, 2, 3, 4	1
c6gn.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6gn.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c6gn.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c6gn.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6gn.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
`c6i.large`	2	1	2	1	1, 2
`c6i.xlarge`	4	2	2	1, 2	1, 2
`c6i.2xlarge`	8	4	2	2, 4	1, 2
`c6i.4xlarge`	16	8	2	2, 4, 6, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
c6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
c6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
c6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6id.large`	2	1	2	1	1, 2
`c6id.xlarge`	4	2	2	1, 2	1, 2
`c6id.2xlarge`	8	4	2	2, 4	1, 2
`c6id.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`c6id.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`c6id.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`c6id.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`c6id.24xlarge`	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6id.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
c6in.large	2	1	2	1	1, 2
c6in.xlarge	4	2	2	1, 2	1, 2
c6in.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c6in.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c6in.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
`c6in.16xlarge`	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
`c6in.24xlarge`	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6in.32xlarge	128	64	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	1, 2
c7a.large	2	2	1	1, 2	1
c7a.xlarge	4	4	1	1, 2, 3, 4	1
c7a.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c7a.4xlarge	16	16	1	1, 2, 4, 6, 8, 10, 12, 14, 16	1
c7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
c7a.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7a.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32, 40, 48, 56, 64	1
c7a.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36, 48, 60, 72, 84, 96	1
c7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1
c7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	1
c7g.large	2	2	1	1, 2	1
c7g.xlarge	4	4	1	1, 2, 3, 4	1
c7g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c7g.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`c7g.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
`c7g.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c7g.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
`c7gd.large`	2	2	1	1, 2	1
`c7gd.xlarge`	4	4	1	1, 2, 3, 4	1
`c7gd.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c7gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c7gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c7gn.large	2	2	1	1, 2	1
c7gn.xlarge	4	4	1	1, 2, 3, 4	1
c7gn.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7gn.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c7gn.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
c7gn.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7gn.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
c7i.large	2	1	2	1	1, 2
c7i.xlarge	4	2	2	1, 2	1, 2
c7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
c7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
c7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c7i.24xlarge`	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
`c7i.48xlarge`	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7i-flex.large	2	1	2	1	1, 2
c7i-flex.xlarge	4	2	2	1, 2	1, 2
c7i-flex.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c7i-flex.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c7i-flex.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
c7i-flex.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c7i-flex.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
c8g.large	2	2	1	1, 2	1
c8g.xlarge	4	4	1	1, 2, 3, 4	1
c8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
c8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
c8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c8g.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8g.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
`c8gd.large`	2	2	1	1, 2	1
`c8gd.xlarge`	4	4	1	1, 2, 3, 4	1
`c8gd.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`c8gd.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`c8gd.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c8gd.24xlarge`	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8gd.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
`c8gn.large`	2	2	1	1, 2	1
`c8gn.xlarge`	4	4	1	1, 2, 3, 4	1
`c8gn.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`c8gn.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`c8gn.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8gn.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`c8gn.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8gn.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c8gn.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	

Memory optimized instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r3.large`	2	1	2	1	1, 2
`r3.xlarge`	4	2	2	1, 2	1, 2
`r3.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`r3.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`r3.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`r4.large`	2	1	2	1	1, 2
`r4.xlarge`	4	2	2	1, 2	1, 2
`r4.2xlarge`	8	4	2	1, 2, 3, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5.large	2	1	2	1	1, 2
r5.xlarge	4	2	2	2	1, 2
r5.2xlarge	8	4	2	2, 4	1, 2
r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5a.large	2	1	2	1	1, 2
r5a.xlarge	4	2	2	2	1, 2
r5a.2xlarge	8	4	2	2, 4	1, 2
r5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
r5a.12xlarge	48	24	2	6, 12, 18, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r5a.16xlarge	64	32	2	8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
r5ad.large	2	1	2	1	1, 2
r5ad.xlarge	4	2	2	2	1, 2
r5ad.2xlarge	8	4	2	2, 4	1, 2
r5ad.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5ad.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
r5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2
r5ad.16xlarge	64	32	2	8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5ad.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
r5b.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r5b.xlarge	4	2	2	1, 2	1, 2
r5b.2xlarge	8	4	2	2, 4	1, 2
r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5b.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5b.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5b.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5d.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r5d.xlarge	4	2	2	2	1, 2
r5d.2xlarge	8	4	2	2, 4	1, 2
r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5dn.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r5dn.xlarge`	4	2	2	1, 2	1, 2
`r5dn.2xlarge`	8	4	2	2, 4	1, 2
`r5dn.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`r5dn.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`r5dn.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`r5dn.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`r5dn.24xlarge`	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`r5n.large`	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r5n.xlarge	4	2	2	1, 2	1, 2
r5n.2xlarge	8	4	2	2, 4	1, 2
r5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5n.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5n.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5n.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5n.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r6a.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6a.xlarge	4	2	2	1, 2	1, 2
r6a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r6a.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r6a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
r6a.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
r6a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2
r6a.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
r6a.32xlarge	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
r6a.48xlarge	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
r6g.large	2	2	1	1, 2	1
r6g.xlarge	4	4	1	1, 2, 3, 4	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r6g.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`r6g.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`r6g.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r6g.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r6gd.large	2	2	1	1, 2	1
r6gd.xlarge	4	4	1	1, 2, 3, 4	1
r6gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r6gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
r6gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6gd.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r6i.large	2	1	2	1	1, 2
r6i.xlarge	4	2	2	1, 2	1, 2
r6i.2xlarge	8	4	2	2, 4	1, 2
r6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6idn.large	2	1	2	1	1, 2
r6idn.xlarge	4	2	2	1, 2	1, 2
r6idn.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r6idn.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r6idn.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r6idn.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6idn.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
r6idn.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
r6idn.32xlarge	128	64	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	1, 2
r6in.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6in.xlarge	4	2	2	1, 2	1, 2
r6in.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r6in.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r6in.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
r6in.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r6in.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
`r6in.32xlarge`	128	64	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	1, 2
`r6id.large`	2	1	2	1	1, 2
`r6id.xlarge`	4	2	2	1, 2	1, 2
`r6id.2xlarge`	8	4	2	2, 4	1, 2
`r6id.4xlarge`	16	8	2	2, 4, 6, 8	1, 2
`r6id.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r6id.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r6id.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r6id.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r6id.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
r7a.large	2	2	1	1, 2	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r7a.xlarge`	4	4	1	1, 2, 3, 4	1
`r7a.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`r7a.4xlarge`	16	16	1	1, 2, 4, 6, 8, 10, 12, 14, 16	1
`r7a.8xlarge`	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
`r7a.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1
`r7a.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 16, 24, 32, 40, 48, 56, 64	1
`r7a.24xlarge`	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36, 48, 60, 72, 84, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1
r7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	1
r7g.large	2	2	1	1, 2	1
r7g.xlarge	4	4	1	1, 2, 3, 4	1
r7g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r7g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
r7g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
r7gd.large	2	2	1	1, 2	1
r7gd.xlarge	4	4	1	1, 2, 3, 4	1
r7gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r7gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
r7gd.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r7gd.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
`r7i.large`	2	1	2	1	1, 2
`r7i.xlarge`	4	2	2	1, 2	1, 2
`r7i.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`r7i.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r7i.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`r7i.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
`r7i.16xlarge`	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7i.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
r7i.48xlarge	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7iz.large	2	1	2	1	1, 2
r7iz.xlarge	4	2	2	1, 2	1, 2
r7iz.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r7iz.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r7iz.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r7iz.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r7iz.16xlarge`	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
`r7iz.32xlarge`	128	64	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	1, 2
`r8g.large`	2	2	1	1, 2	1
`r8g.xlarge`	4	4	1	1, 2, 3, 4	1
`r8g.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`r8g.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r8g.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
`r8g.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8g.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8g.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
r8gd.large	2	2	1	1, 2	1
r8gd.xlarge	4	4	1	1, 2, 3, 4	1
r8gd.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
r8gd.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
r8gd.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r8gd.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r8gd.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8gd.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8gd.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
r8i.large	2	1	2	1	1, 2
r8i.xlarge	4	2	2	1, 2	1, 2
r8i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r8i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r8i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r8i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
r8i.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
r8i.32xlarge	128	64	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	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r8i.48xlarge	192	96	2	6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96	1, 2
r8i.96xlarge	384	192	2	12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 78, 84, 90, 96, 102, 108, 114, 120, 126, 132, 138, 144, 150, 156, 162, 168, 174, 180, 186, 192	1, 2
r8i-flex.large	2	1	2	1	1, 2
r8i-flex.xlarge	4	2	2	1, 2	1, 2
r8i-flex.2xlarge	8	4	2	1, 2, 3, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`r8i-flex.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`r8i-flex.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`r8i-flex.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
`r8i-flex.16xlarge`	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u-3tb1.56xlarge`	224	112	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112	1, 2
`u-6tb1.56xlarge`	224	224	1	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u-6tb1.112xlarge`	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2
`u-9tb1.112xlarge`	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u-12tb1.112xlarge`	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2
`u-18tb1.112xlarge`	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
u-24tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
u7i-6tb.112xlarge	448	224	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u7i-8tb.112xlarge`	448	224	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152, 156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 204, 208, 212, 216, 220, 224	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u7i-12tb.224xlarge`	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
u7in-16tb.224xlarge	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u7in-24tb.224xlarge`	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
u7in-32tb.224xlarge	896	448	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
u7inh-32tb.480xlarge	1920	960	2	32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240, 256, 272, 288, 304, 320, 336, 352, 368, 384, 400, 416, 432, 448, 464, 480, 496, 512, 528, 544, 560, 576, 592, 608, 624, 640, 656, 672, 688, 704, 720, 736, 752, 768, 784, 800, 816, 832, 848, 864, 880, 896, 912, 928, 944, 960	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
x1e.xlarge	4	2	2	1, 2	1, 2
x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`x1e.32xlarge`	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
`x2gd.large`	2	2	1	1, 2	1
`x2gd.xlarge`	4	4	1	1, 2, 3, 4	1
`x2gd.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`x2gd.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`x2gd.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`x2gd.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`x2gd.16xlarge`	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
`x2idn.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
x2idn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
x2idn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
x2iedn.xlarge	4	2	2	1, 2	1, 2
x2iedn.2xlarge	8	4	2	2, 4	1, 2
x2iedn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
x2iedn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
x2iedn.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x2iedn.24 xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
x2iedn.32 xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
x2iezn.2x large	8	4	2	2, 4	1, 2
x2iezn.4x large	16	8	2	2, 4, 6, 8	1, 2
x2iezn.6x large	24	12	2	2, 4, 6, 8, 10, 12	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`x2iezn.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`x2iezn.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`x8g.large`	2	2	1	1, 2	1
`x8g.xlarge`	4	4	1	1, 2, 3, 4	1
`x8g.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`x8g.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
`x8g.8xlarge`	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`x8g.12xlarge`	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
x8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`x8g.24xlarge`	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
x8g.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
`z1d.large`	2	1	2	1	1, 2
`z1d.xlarge`	4	2	2	1, 2	1, 2
`z1d.2xlarge`	8	4	2	2, 4	1, 2
`z1d.3xlarge`	12	6	2	2, 4, 6	1, 2
`z1d.6xlarge`	24	12	2	2, 4, 6, 8, 10, 12	1, 2
`z1d.12xlarge`	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Storage optimized instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`d2.xlarge`	4	2	2	1, 2	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
d2.2xlarge	8	4	2	1, 2, 3, 4	1, 2
d2.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
d2.8xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
d3.xlarge	4	2	2	1, 2	1, 2
d3.2xlarge	8	4	2	2, 4	1, 2
d3.4xlarge	16	8	2	2, 4, 6, 8	1, 2
d3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
d3en.xlarge	4	2	2	1, 2	1, 2
d3en.2xlarge	8	4	2	2, 4	1, 2
d3en.4xlarge	16	8	2	2, 4, 6, 8	1, 2
d3en.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
d3en.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`d3en.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`h1.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`h1.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`h1.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`h1.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`i2.xlarge`	4	2	2	1, 2	1, 2
`i2.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`i2.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`i2.8xlarge`	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
`i3.large`	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i3.xlarge	4	2	2	1, 2	1, 2
i3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
i3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
i3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
i3en.large	2	1	2	1	1, 2
i3en.xlarge	4	2	2	1, 2	1, 2
i3en.2xlarge	8	4	2	2, 4	1, 2
i3en.3xlarge	12	6	2	2, 4, 6	1, 2
i3en.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`i3en.12xlarge`	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
`i3en.24xlarge`	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`i4g.large`	2	2	1	1, 2	1
`i4g.xlarge`	4	4	1	1, 2, 3, 4	1
`i4g.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`i4g.4xlarge`	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i4g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
i4g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
i4i.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`i4i.xlarge`	4	2	2	1, 2	1, 2
`i4i.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`i4i.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`i4i.8xlarge`	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
`i4i.12xlarge`	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
`i4i.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i4i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
i4i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
i7i.large	2	1	2	1	1, 2
i7i.xlarge	4	2	2	1, 2	1, 2
i7i.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i7i.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i7i.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
i7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
i7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`i7i.24xlarge`	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
`i7i.48xlarge`	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i7ie.large	2	1	2	1	1, 2
i7ie.xlarge	4	2	2	1, 2	1, 2
i7ie.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i7ie.3xlarge	12	6	2	1, 2, 3, 4, 5, 6	1, 2
i7ie.6xlarge	24	12	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	1, 2
i7ie.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i7ie.18xlarge	72	36	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36	1, 2
i7ie.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i7ie.48xlarge	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
i8g.large	2	2	1	1, 2	1
i8g.xlarge	4	4	1	1, 2, 3, 4	1
i8g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
i8g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i8g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
i8g.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i8g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`i8g.24xlarge`	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i8g.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
`i8ge.large`	2	2	1	1, 2	1
`i8ge.xlarge`	4	4	1	1, 2, 3, 4	1
`i8ge.2xlarge`	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
`i8ge.3xlarge`	12	12	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	1
`i8ge.6xlarge`	24	24	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i8ge.12xlarge	48	48	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`i8ge.18xlarge`	72	72	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i8ge.24xlarge	96	96	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
i8ge.48xlarge	192	192	1	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192	
im4gn.large	2	2	1	1, 2	1
im4gn.xlarge	4	4	1	1, 2, 3, 4	1
im4gn.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
im4gn.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
im4gn.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
im4gn.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
is4gen.medium	1	1	1	1	1
is4gen.large	2	2	1	1, 2	1
is4gen.xlarge	4	4	1	1, 2, 3, 4	1
is4gen.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
is4gen.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1
is4gen.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1

Accelerated computing instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
dl1.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`dl2q.24xlarge`	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
`f1.2xlarge`	8	4	2	1, 2, 3, 4	1, 2
`f1.4xlarge`	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
`f1.16xlarge`	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
`f2.6xlarge`	24	12	2	1, 2, 3, 6, 9, 12	1, 2
`f2.12xlarge`	48	24	2	1, 2, 3, 6, 9, 12, 15, 18, 21, 24	1, 2
`f2.48xlarge`	192	96	2	4, 6, 8, 10, 12, 24, 36, 48, 60, 72, 84, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
g3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
g3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g4ad.xlarge	4	2	2	2	1, 2
g4ad.2xlarge	8	4	2	2, 4	1, 2
g4ad.4xlarge	16	8	2	2, 4, 8	1, 2
g4ad.8xlarge	32	16	2	2, 4, 8, 16	1, 2
g4ad.16xlarge	64	32	2	2, 4, 8, 16, 32	1, 2
g4dn.xlarge	4	2	2	2	1, 2
g4dn.2xlarge	8	4	2	2, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
g4dn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
g4dn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
g4dn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
g4dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g5g.xlarge	4	4	1	1, 2, 3, 4	1
g5g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
g5g.4xlarge	16	16	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
g5g.8xlarge	32	32	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1
g5g.16xlarge	64	64	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	1
g6.xlarge	4	2	2	1, 2	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
g6.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g6.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g6.8xlarge	32	16	2	1, 2, 4, 6, 8, 10, 12, 14, 16	1, 2
g6.12xlarge	48	24	2	1, 2, 3, 6, 9, 12, 15, 18, 21, 24	1, 2
g6.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
g6.24xlarge	96	48	2	1, 2, 3, 4, 5, 6, 12, 18, 24, 30, 36, 42, 48	1, 2
g6.48xlarge	192	96	2	4, 6, 8, 10, 12, 24, 36, 48, 60, 72, 84, 96	1, 2
g6e.xlarge	4	2	2	1, 2	1, 2
g6e.2xlarge	8	4	2	1, 2, 3, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
g6e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g6e.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
g6e.12xlarge	48	24	2	3, 6, 9, 12, 15, 18, 21, 24	1, 2
g6e.16xlarge	64	32	2	4, 8, 12, 16, 20, 24, 28, 32	1, 2
g6e.24xlarge	96	48	2	6, 12, 18, 24, 30, 36, 42, 48	1, 2
g6e.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2
g6f.large	2	1	2	1	1, 2
g6f.xlarge	4	2	2	1, 2	1, 2
g6f.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g6f.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
gr6.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
gr6.8xlarge	32	16	2	1, 2, 4, 6, 8, 10, 12, 14, 16	1, 2
gr6f.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
inf1.xlarge	4	2	2	2	1, 2
inf1.2xlarge	8	4	2	2, 4	1, 2
inf1.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
inf1.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
inf2.xlarge	4	2	2	1, 2	1, 2
inf2.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
inf2.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
inf2.48xlarge	192	96	2	4, 8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
p3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
p3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
p3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p3dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
p4d.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
p4de.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
p5.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
p5.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2
p5e.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
p5en.48xlarge	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
p6-b200.48xlarge	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
p6e-gb200 .36xlarge	144	144	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100,	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144	
`trn1.2xlarge`	8	4	2	2, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
trn1.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
trn1n.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`trn2.48xlarge`	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`trn2u.48xlarge`	192	96	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, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2
`vt1.3xlarge`	12	6	2	6	1, 2
`vt1.6xlarge`	24	12	2	6, 12	1, 2
`vt1.24xlarge`	96	48	2	6, 12, 48	1, 2

High-performance computing instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`hpc6id.32xlarge`	64	64	1	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	1

Specify CPU options for an Amazon EC2 instance

You can specify CPU options during or after instance launch.

Tasks

- Disable simultaneous multithreading
- Specify a custom number of vCPUs at launch
- Specify a custom number of vCPUs in a launch template
- Change CPU options for your EC2 instance

Disable simultaneous multithreading

To disable simultaneous multithreading (SMT), also known as hyper-threading, specify 1 thread per core.

Console

To disable SMT during instance launch

1. Follow the Launch an EC2 instance using the launch instance wizard in the console procedure and configure your instance as needed.
2. Expand **Advanced details**, and select the **Specify CPU options** checkbox.
3. For **Core count**, choose the number of required CPU cores. In this example, to specify the default CPU core count for an `r5.4xlarge` instance, choose 8.
4. To disable SMT, for **Threads per core**, choose **1**.
5. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

AWS CLI

To disable SMT during instance launch

Use the run-instances AWS CLI command and specify a value of 1 for `ThreadsPerCore` for the `--cpu-options` parameter. For `CoreCount`, specify the number of CPU cores. In this example, to specify the default CPU core count for an `r7i.4xlarge` instance, specify a value of 8.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=8,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

To disable SMT during instance launch

Use the New-EC2Instance command and specify a value of 1 for `ThreadsPerCore` for the `-CpuOptions` parameter. For `CoreCount`, specify the number of CPU cores. In this example, to specify the default CPU core count for an `r7i.4xlarge` instance, specify a value of 8.

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=8; ThreadsPerCore=1} `
```

```
    -KeyName 'my-key-pair'
```

> **Note**
>
> To disable SMT for an existing instance, follow the process shown in Change CPU options for your EC2 instance, and change the number of threads that run per core to 1.

Specify a custom number of vCPUs at launch

You can customize the number of CPU cores and threads per core when you launch an instance from the EC2 console or AWS CLI. The examples in this section use an `r5.4xlarge` instance type, which has the following default settings:

- CPU cores: 8
- Threads per core: 2

Instances launch with the maximum number of vCPUs available for the instance type by default. For this instance type, that's 16 total vCPUs (8 cores running 2 threads each). For more information about this instance type, see Memory optimized instances.

The following example launches an `r5.4xlarge` instance with 4 vCPUs.

Console

To specify a custom number of vCPUs during instance launch

1. Follow the Launch an EC2 instance using the launch instance wizard in the console procedure and configure your instance as needed.
2. Expand **Advanced details**, and select the **Specify CPU options** checkbox.
3. To get 4 vCPUs, specify 2 CPU cores and 2 threads per core, as follows:
   - For **Core count**, choose **2**.
   - For **Threads per core**, choose **2**.
4. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

AWS CLI

To specify a custom number of vCPUs during instance launch

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 2 CPU cores and 2 threads per core to get 4 vCPUs.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=2,ThreadsPerCore=2" \
    --key-name my-key-pair
```

Alternatively, specify 4 CPU cores and 1 thread per core (disable SMT) to get 4 vCPUs:

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=4,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

To specify a custom number of vCPUs during instance launch

Use the New-EC2Instance command and specify the number of CPU cores and number of threads in the `-CpuOptions` parameter. You can specify 2 CPU cores and 2 threads per core to get 4 vCPUs.

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=2; ThreadsPerCore=2} `
    -KeyName 'my-key-pair'
```

Alternatively, specify 4 CPU cores and 1 thread per core (disable SMT) to get 4 vCPUs:

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
```

```
    -CpuOptions @{CoreCount=4; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

Specify a custom number of vCPUs in a launch template

You can customize the number of CPU cores and threads per core for the instance in a launch template. The examples in this section use an `r5.4xlarge` instance type, which has the following default settings:

- CPU cores: 8
- Threads per core: 2

Instances launch with the maximum number of vCPUs available for the instance type by default. For this instance type, that's 16 total vCPUs (8 cores running 2 threads each). For more information about this instance type, see Memory optimized instances.

The following example creates a launch template that specifies the configuration for an `r5.4xlarge` instance with 4 vCPUs.

Console

To specify a custom number of vCPUs in a launch template

1. Follow the Create a launch template by specifying parameters procedure and configure your launch template as needed.
2. Expand **Advanced details**, and select the **Specify CPU options** checkbox.
3. To get 4 vCPUs, specify 2 CPU cores and 2 threads per core, as follows:
   - For **Core count**, choose **2**.
   - For **Threads per core**, choose **2**.
4. In the **Summary** panel, review your instance configuration, and then choose **Create launch template**. For more information, see Store instance launch parameters in Amazon EC2 launch templates.

AWS CLI

To specify a custom number of vCPUs in a launch template

Use the create-launch-template AWS CLI command and specify the number of CPU cores and number of threads in the `CpuOptions` parameter. You can specify 2 CPU cores and 2 threads per core to get 4 vCPUs.

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForCPUOptions \
    --version-description CPUOptionsVersion1 \
    --launch-template-data file://template-data.json
```

The following is an example JSON file that contains the launch template data, which includes the CPU options, for the instance configuration for this example.

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":2,
        "ThreadsPerCore":2
    }
}
```

Alternatively, specify 4 CPU cores and 1 thread per core (disable SMT) to get 4 vCPUs:

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
```

```
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":1
    }
}
```

PowerShell

To specify a custom number of vCPUs in a launch template

Use the [New-EC2LaunchTemplate](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2LaunchTemplate.html).

```
New-EC2LaunchTemplate `
    -LaunchTemplateName 'TemplateForCPUOptions' `
    -VersionDescription 'CPUOptionsVersion1' `
    -LaunchTemplateData (Get-Content -Path 'template-data.json' | ConvertFrom-Json)
```

Change CPU options for your EC2 instance

As your needs change over time, you might want to change the configuration of CPU options for an existing instance. Each thread that runs on your instance is known as a virtual CPU (vCPU). You can change the number of vCPUs that run for an existing instance in the Amazon EC2 console, AWS CLI, API, or SDKs. The instance state must be `Stopped` before you can make this change.

Console

To change the number of active vCPUs for an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**, and select your instance.

3. If the instance is running, you must stop it before you proceed. Choose **Instance state**, **Stop instance**.
4. To change the vCPU configuration, choose **Actions**, **Instance settings**, **Change CPU options**.
5. Choose one of the following CPU options:

   - To reset your instance to the default number of vCPUs for its instance type, choose **Use default CPU options**. The default is to run all threads for all CPU cores.
   - To configure the number of vCPUs on your instance, choose **Specify CPU options**.
6. If you chose **Specify CPU options**, the **Active vCPUs** fields are displayed.

   a. Use the first selector to configure the number of threads for each CPU core. To disable simultaneous multithreading, choose 1.
   b. Use the second selector to configure the number of CPUs for your instance.

   The following fields update dynamically as you make changes to the CPU option selectors:

   - **Active vCPUs**: The number of CPU cores multiplied by the threads per core, based on the selections that you made. For example, if you selected 2 threads and 4 cores, that would equal 8 vCPUs.
   - **Total vCPUs**: The maximum number of vCPUs for the instance type. For example, for an `m6i.4xlarge` instance type, this is 16 vCPUs (8 cores running 2 threads each).
7. Choose **Change**.

AWS CLI

To change the number of active vCPUs for an instance

Use the modify-instance-cpu-options command and specify the number of CPU cores that run in the `--core-count` parameter, and the number of threads that run per core in the `--threads-per-core` parameter.

The following examples shows two possible configurations on an `m6i.4xlarge` instance type to run 8 vCPUs on the specified instance. The default for this instance type is 16 vCPUs (8 cores running 2 threads each).

Example 1: Run 4 CPU cores with 2 threads per core, for a total of 8 vCPU.

```
aws ec2 modify-instance-cpu-options \
    --instance-id 1234567890abcdef0 \
    --core-count=4 \
    --threads-per-core=2
```

Example 2: Disable simultaneous multi-threading by changing the number of threads that run per core to 1. The resulting configuration also runs a total of 8 vCPUs (8 CPU cores with 1 thread per core).

```
aws ec2 modify-instance-cpu-options \
    --instance-id 1234567890abcdef0 \
    --core-count=8 \
    --threads-per-core=1
```

PowerShell

To change the number of active vCPUs for an instance

Use the Edit-EC2InstanceCpuOption cmdlet and specify the number of CPU cores that run in the `-CoreCount` parameter, and the number of threads that run per core in the `ThreadsPerCore` parameter.

Example 1: Run 4 CPU cores with 2 threads per core, for a total of 8 vCPU.

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 4 `
    -ThreadsPerCore 2
```

Example 2: Disable simultaneous multi-threading by changing the number of threads that run per core to 1. The resulting configuration also runs a total of 8 vCPUs (8 CPU cores with 1 thread per core).

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 8 `
    -ThreadsPerCore 1
```

View CPU threads and cores for an Amazon EC2 instance

You can view the CPU options for an existing instance by describing the instance.

Console

To view the CPU options for an instance

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**.

AWS CLI

To view the CPU options for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].CpuOptions
```

The following is example output. The `CoreCount` field indicates the number of cores for the instance. The `ThreadsPerCore` field indicates the number of threads per core.

```
[
    {
        "CoreCount": 24,
        "ThreadsPerCore": 2
    },
]
```

PowerShell

To view the CPU options for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
```

```
    -InstanceId 'i-1234567890abcdef0').Instances.CpuOptions
```

The following is example output.

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
          24        2
```

Alternatively, to view CPU information, you can connect to your instance and use one of the following system tools:

- Windows `Task Manager` on your Windows instance
- The **lscpu** command on your Linux 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*.

AMD SEV-SNP for Amazon EC2 instances

AMD Secure Encrypted Virtualization-Secure Nested Paging (AMD SEV-SNP) is a CPU feature that provides the following properties:

- **Attestation** – AMD SEV-SNP enables you to retrieve a signed attestation report that contains a cryptographic measure that can be used to validate the instance's state and identity, and that it is running on genuine AMD hardware. For more information, see Attest an Amazon EC2 instance with AMD SEV-SNP.
- **Memory encryption** – Starting with AMD EPYC (Milan), AWS Graviton2, and Intel Xeon Scalable (Ice Lake) processors, instance memory is always encrypted. Instances that are enabled for AMD SEV-SNP use an instance-specific key for their memory encryption.

Contents

- Pricing
- Find EC2 instance types that support AMD SEV-SNP
- Enable AMD SEV-SNP for an EC2 instance
- Attest an Amazon EC2 instance with AMD SEV-SNP

Concepts and terminology

Before you begin using AMD SEV-SNP, ensure that you are familiar with the following concepts and terminology.

AMD SEV-SNP attestation report

The AMD SEV-SNP attestation report is a document that an instance can request from the CPU. The AMD SEV-SNP attestation report can be used to validate the state and identity of an instance, and to verify that it is running in a sanctioned AMD environment. The report includes a launch measurement, which is a cryptographic hash of the initial boot state of an instance, including its initial instance memory contents and initial state of the vCPUs. The AMD SEV-SNP attestation report is signed with a VLEK signature that chains back to an AMD root of trust.

VLEK

The Versioned Loaded Endorsement Key (VLEK) is a versioned signing key that is certified by AMD and used by the AMD CPU to sign the AMD SEV-SNP attestation reports. VLEK signatures can be validated using certificates provided by AMD.

OVMF binary

The Open Virtual Machine Firmware (OVMF) is the early-boot code that is used to provide a UEFI environment for the instance. The early-boot code is run before the code in the AMI is booted. The OVMF also finds and runs the boot loader provided in the AMI. For more information, see the OVMF repository.

Requirements

To use AMD SEV-SNP, you must do the following:

- Use one of the following supported instance types:
  - **General purpose**: `m6a.large` | `m6a.xlarge` | `m6a.2xlarge` | `m6a.4xlarge` | `m6a.8xlarge`

  - **Compute optimized**: `c6a.large` | `c6a.xlarge` | `c6a.2xlarge` | `c6a.4xlarge` | `c6a.8xlarge` | `c6a.12xlarge` | `c6a.16xlarge`
  - **Memory optimized**: `r6a.large` | `r6a.xlarge` | `r6a.2xlarge` | `r6a.4xlarge`
- Launch your instance in a supported AWS Region. Currently, only US East (Ohio) and Europe (Ireland) are supported.
- Use an AMI with `uefi` or `uefi-preferred` boot mode and an operating system that supports AMD SEV-SNP. For more information about AMD SEV-SNP support on your operating system, refer to the respective operating system's documentation. For AWS, AMD SEV-SNP is supported on AL2023, RHEL 9.3, SLES 15 SP4, and Ubuntu 23.04 and later.

Considerations

You can only enable AMD SEV-SNP when you launch an instance. When AMD SEV-SNP is enabled for your instance launch, the following rules apply.

- After it is enabled, AMD SEV-SNP can't be disabled. It remains enabled throughout the instance lifecycle.
- You can only change the instance type to another instance type that supports AMD SEV-SNP.
- Hibernation and Nitro Enclaves aren't supported.
- Dedicated Hosts aren't supported.
- If the underlying host for your instance is scheduled for maintenance, you'll receive a scheduled event notification 14 days before the event. You must manually stop or restart your instance to move it to a new host.

Pricing

When you launch an Amazon EC2 instance with AMD SEV-SNP enabled, you are charged an additional hourly usage fee that is equivalent to 10 percent of the On-Demand hourly rate of the selected instance type.

This AMD SEV-SNP usage fee is a separate charge to your Amazon EC2 instance usage. Reserved Instances, Savings Plans, and operating system usage don't impact this fee.

If you configure a Spot Instance to launch with AMD SEV-SNP enabled, you are charged an additional hourly usage fee that is equivalent to 10 percent of the On-Demand hourly rate of the selected instance type. If the allocation strategy uses price as an input, Spot Fleet does not include this additional fee; only the Spot price is used.

Find EC2 instance types that support AMD SEV-SNP

You can find instance types that support AMD SEV-SNP. The Amazon EC2 console does not display this information for an instance type.

AWS CLI

To find the instance types that support AMD SEV-SNP

Use the following describe-instance-types command.

```
aws ec2 describe-instance-types \
    --filters Name=processor-info.supported-features,Values=amd-sev-snp \
    --query 'InstanceTypes[*].[InstanceType]' \
    --output text | sort
```

The following is example output.

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

PowerShell

To find the instance types that support AMD SEV-SNP

Use the Get-EC2InstanceType cmdlet.

```
(Get-EC2InstanceType `
    -Filter @{Name="processor-info.supported-features"; Values="amd-sev-
snp"}).InstanceType.Value | Sort-Object
```

The following is example output.

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

Enable AMD SEV-SNP for an EC2 instance

You can launch an instance with AMD SEV-SNP enabled. You can't enable AMD SEV-SNP after launch.

Launch an instance with AMD SEV-SNP enabled

You can't enable AMD SEV-SNP using the Amazon EC2 console.

AWS CLI

To launch an instance with AMD SEV-SNP enabled

Use the run-instances command with the `--cpu-options` option. For additional requirements, see AMD SEV-SNP requirements.

```
--cpu-options AmdSevSnp=enabled
```

PowerShell

To launch an instance with AMD SEV-SNP enabled

Use the New-EC2Instance cmdlet with the `-CpuOption` parameter.

```
-CpuOption @{AmdSevSnp="enabled"}
```

Check if an EC2 instance is enabled for AMD SEV-SNP

You can find instances that are enabled for AMD SEV-SNP. The Amazon EC2 console does not display this information.

AWS CLI

To check whether AMD SEV-SNP is enabled for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].CpuOptions
```

The following is example output. If `AmdSevSnp` is not present in `CpuOptions`, then AMD SEV-SNP is disabled.

```
[
    {
        "AmdSevSnp": "enabled",
        "CoreCount": 1,
        "ThreadsPerCore": 2
    }
]
```

PowerShell

To check whether AMD SEV-SNP is enabled for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.CpuOptions
```

The following is example output. If the value of `AmdSevSnp` is not present, then AMD SEV-SNP is disabled.

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
enabled   1         2
```

AWS CloudTrail

In the AWS CloudTrail event for the instance launch request, the following property indicates that AMD SEV-SNP is enabled for the instance.

```
"cpuOptions": {"AmdSevSnp": "enabled"}
```

Attest an Amazon EC2 instance with AMD SEV-SNP

Attestation is a process that allows your instance to prove its state and identity. After you enable AMD SEV-SNP for your instance, you can request an AMD SEV-SNP attestation report from the underlying processor. The AMD SEV-SNP attestation report contains a cryptographic hash, called the launch measurement, of the initial guest memory contents and initial vCPU state. The attestation report is signed with a VLEK signature that chains back to an AMD root of trust. You can use the launch measurement included in the attestation report to validate that the instance is running in a genuine AMD environment and to validate the initial boot code that was used to launch the instance.

Prerequisite

Launch an instance that is enabled for AMD SEV-SNP. For more information, see Enable AMD SEV-SNP for an EC2 instance.

Steps

- Step 1: Get the attestation report
- Step 2: Validate the attestation report signature

Step 1: Get the attestation report

In this step, you install and build the `snpguest` utility, and then use it to request the AMD SEV-SNP attestation report and certificates.

1. Connect to your instance.
2. Run the following commands to build the `snpguest` utility from the snpguest repository.

```
$ git clone https://github.com/virtee/snpguest.git
$ cd snpguest
$ cargo build -r
$ cd target/release
```

3. Generate a request for the attestation report. The utility requests the attestation report from the host, and writes it to a binary file with the provided request data.

   The following example creates a random request string, and uses it as the request file (`request-file.txt`). When the command returns the attestation report it's stored in the file path that you specify (`report.bin`). In this case, the utility stores the report in the current directory.

```
$ ./snpguest report report.bin request-file.txt --random
```

4. Request the certificates from host memory, and store them as PEM files. The following example stores the files in the same directory as the `snpguest` utility. If certificates already exist in the specified directory, those certificates are overwritten.

```
$ ./snpguest certificates PEM ./
```

Step 2: Validate the attestation report signature

The attestation report is signed with a certificate, called the Versioned Loaded Endorsement Key (VLEK), which is issued by AMD for AWS. In this step, you can validate that the VLEK certificate is issued by AMD, and that the attestation report is signed by that VLEK certificate.

1. Download the VLEK root of trust certificates from the official AMD website to the current directory.

```
$ sudo curl --proto '=https' --tlsv1.2 -sSf https://kdsintf.amd.com/vlek/v1/Milan/
cert_chain -o ./cert_chain.pem
```

2. Use `openssl` to validate that the VLEK certificate is signed by the AMD root of trust certificates.

```
$ sudo openssl verify --CAfile ./cert_chain.pem vlek.pem
```

The following is example output.

```
vlek.pem: OK
```

3. Use the `snpguest` utility to validate that the attestation report is signed by the VLEK certificate.

```
$ ./snpguest verify attestation ./ report.bin
```

The following is exmaple output.

```
Reported TCB Boot Loader from certificate matches the attestation report.
Reported TCB TEE from certificate matches the attestation report.
Reported TCB SNP from certificate matches the attestation report.
Reported TCB Microcode from certificate matches the attestation report.
VEK signed the Attestation Report!
```

Processor state control for Amazon EC2 Linux instances

C-states control the sleep levels that a core can enter when it is idle. C-states are numbered starting with C0 (the shallowest state where the core is totally awake and executing instructions) and go to C6 (the deepest idle state where a core is powered off).

P-states control the desired performance (in CPU frequency) from a core. P-states are numbered starting from P0 (the highest performance setting where the core is allowed to use Intel Turbo Boost Technology to increase frequency if possible), and they go from P1 (the P-state that requests the maximum baseline frequency) to P15 (the lowest possible frequency).

> **Note**
>
> AWS Graviton processors have built-in power saving modes and operate at a fixed frequency. Therefore, they do not provide the ability for the operating system to control C-states and P-states.

C-states and P-states

The following instance types provide the ability for an operating system to control processor C-states and P-states:

- **General purpose**: `m4.10xlarge` | `m4.16xlarge`
- **Compute optimized**: `c4.8xlarge`
- **Memory optimized**: `r4.8xlarge` | `r4.16xlarge` | `x1.16xlarge` | `x1.32xlarge` | `x1e.8xlarge` | `x1e.16xlarge` | `x1e.32xlarge`
- **Storage optimized**: `d2.8xlarge` | `i3.8xlarge` | `i3.16xlarge` | `h1.8xlarge` | `h1.16xlarge`
- **Accelerated computing**: `f1.16xlarge` | `g3.16xlarge` | `p2.16xlarge` | `p3.16xlarge`
- **Bare metal**: All bare metal instances with Intel and AMD processors

C-states only

The following instance types provide the ability for an operating system to control processor C-states:

- **General purpose**: `m5.12xlarge` | `m5.24xlarge` | `m5d.12xlarge` | `m5d.24xlarge` | `m5n.12xlarge` | `m5n.24xlarge` | `m5dn.12xlarge` | `m5dn.24xlarge` | `m5zn.6xlarge` | `m5zn.12xlarge` | `m6a.24xlarge` | `m6a.48xlarge` | `m6i.16xlarge` | `m6i.32xlarge` | `m6id.16xlarge` | `m6id.32xlarge` | `m6idn.16xlarge` | `m6in.16xlarge` | `m6in.32xlarge` | `m7a.medium` | `m7a.large` | `m7a.xlarge` | `m7a.2xlarge` | `m7a.4xlarge` | `m7a.8xlarge` | `m7a.12xlarge` | `m7a.16xlarge` | `m7a.24xlarge` | `m7a.32xlarge` | `m7a.48xlarge` | `m7i.large` | `m7i.xlarge` | `m7i.2xlarge` | `m7i.4xlarge` | `m7i.8xlarge` | `m7i.12xlarge` | `m7i.16xlarge` | `m7i.24xlarge` | `m7i.48xlarge` | `m8i.large` | `m8i.xlarge` | `m8i.2xlarge` | `m8i.4xlarge` | `m8i.8xlarge` | `m8i.12xlarge` | `m8i.16xlarge` | `m8i.24xlarge` | `m8i.32xlarge` | `m8i.48xlarge` | `m8i.96xlarge`

- **Compute optimized**: c5.9xlarge | c5.12xlarge | c5.18xlarge | c5.24xlarge | c5a.24xlarge | c5ad.24xlarge | c5d.9xlarge | c5d.12xlarge | c5d.18xlarge | c5d.24xlarge | c5n.9xlarge | c5n.18xlarge | c6a.24xlarge | c6a.32xlarge | c6a.48xlarge | c6i.16xlarge | c6i.32xlarge | c6id.24xlarge | c6id.32xlarge | c6in.32xlarge | c7a.medium | c7a.large | c7a.xlarge | c7a.2xlarge | c7a.4xlarge | c7a.8xlarge | c7a.12xlarge | c7a.16xlarge | c7a.24xlarge | c7a.32xlarge | c7a.48xlarge | c7i.large | c7i.xlarge | c7i.2xlarge | c7i.4xlarge | c7i.8xlarge | c7i.12xlarge | c7i.16xlarge | c7i.24xlarge | c7i.48xlarge
- **Memory optimized**: r5.12xlarge | r5.24xlarge | r5b.12xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge | r5dn.24xlarge | r6a.24xlarge | r6a.48xlarge | r6i.16xlarge | r6i.32xlarge | r6id.16xlarge | r6id.32xlarge | r6in.16xlarge | r6in.32xlarge | r7a.medium | r7a.large | r7a.xlarge | r7a.2xlarge | r7a.4xlarge | r7a.8xlarge | r7a.12xlarge | r7a.16xlarge | r7a.24xlarge | r7a.32xlarge | r7a.48xlarge | r7i.large | r7i.xlarge | r7i.2xlarge | r7i.4xlarge | r7i.8xlarge | r7i.12xlarge | r7i.16xlarge | r7i.24xlarge | r7i.48xlarge | r7iz.large | r7iz.xlarge | r7iz.2xlarge | r7iz.4xlarge | r7iz.8xlarge | r7iz.12xlarge | r7iz.16xlarge | r7iz.32xlarge | r8i.large | r8i.xlarge | r8i.2xlarge | r8i.4xlarge | r8i.8xlarge | r8i.12xlarge | r8i.16xlarge | r8i.24xlarge | r8i.32xlarge | r8i.48xlarge | r8i.96xlarge | u-3tb1.56xlarge | u-6tb1.56xlarge | u-6tb1.112xlarge | u-9tb1.112xlarge | u-12tb1.112xlarge | u-18tb1.112xlarge | u-24tb1.112xlarge | u7i-6tb.112xlarge | u7i-8tb.112xlarge | u7i-12tb.224xlarge | u7in-16tb.224xlarge | u7in-24tb.224xlarge | u7in-32tb.224xlarge | u7inh-32tb.480xlarge | x2idn.32xlarge | x2iedn.16xlarge | x2iedn.32xlarge | x2iezn.12xlarge | z1d.6xlarge | z1d.12xlarge
- **Storage optimized**: d3en.12xlarge | dl1.24xlarge | i3en.12xlarge | i3en.24xlarge | i4i.16xlarge | i7i.large | i7i.xlarge | i7i.2xlarge | i7i.4xlarge | i7i.8xlarge | i7i.12xlarge | i7i.16xlarge | i7i.24xlarge | i7i.48xlarge | i7ie.large | i7ie.xlarge | i7ie.2xlarge | i7ie.3xlarge | i7ie.6xlarge | i7ie.12xlarge | i7ie.18xlarge | i7ie.24xlarge | i7ie.48xlarge | r5b.12xlarge | r5b.24xlarge
- **Accelerated computing**: dl1.24xlarge | f2.6xlarge | f2.12xlarge | f2.48xlarge | g5.24xlarge | g5.48xlarge | g6.24xlarge | g6.48xlarge | g6e.12xlarge | g6e.24xlarge | g6e.48xlarge | inf1.24xlarge | p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | p5.48xlarge | p5e.48xlarge | p5en.48xlarge | p6-b200.48xlarge |

`trn1.32xlarge` | `trn2.3xlarge` | `trn2.48xlarge` | `trn2a.3xlarge` | `trn2a.48xlarge` | `trn2n.3xlarge` | `trn2n.48xlarge` | `trn2p.48xlarge` | `trn2u.48xlarge` | `vt1.24xlarge`

You might want to change the C-state or P-state settings to increase processor performance consistency, reduce latency, or tune your instance for a specific workload. The default C-state and P-state settings provide maximum performance, which is optimal for most workloads. However, if your application would benefit from reduced latency at the cost of higher single- or dual-core frequencies, or from consistent performance at lower frequencies as opposed to bursty Turbo Boost frequencies, consider experimenting with the C-state or P-state settings that are available to these instances.

For information about different processor configurations and how to monitor the effects of your configuration for Amazon Linux, see Processor state control for Amazon EC2 Amazon Linux instance in the *Amazon Linux 2 User Guide*. These procedures were written for, and apply to Amazon Linux; however, they might also work for other Linux distributions with a Linux kernel of 3.9 or newer. For more information about other Linux distributions and processor state control, see your system-specific documentation.

Amazon EC2 managed instances

An *Amazon EC2 managed instance* is an EC2 instance that is provisioned and managed by a designated service provider, such as Amazon EKS through EKS Auto Mode. Managed instances provide a simplified way for running compute workloads on Amazon EC2 by allowing you to delegate operational control of the instance to a service provider.

Delegated control is the only change introduced for managed instances. The technical specifications and billing remain the same as non-managed EC2 instances. Because managed instances allow you to delegate control to the service provider, you can benefit from the service provider’s operational expertise and best practices. When an instance is managed, the service provider is responsible for tasks such as provisioning the instance, configuring software, scaling capacity, handling instance failures and replacements, and terminating the instance.

You can’t directly modify the settings of a managed instance or terminate it. The service and specific operations are determined by the agreement between you and the service provider. However, you can add, modify, or remove tags from your managed instances, allowing you to categorize them within your AWS environment.

Contents

Billing for managed instances

An Amazon EC2 managed instance incurs the same base charge as a non-managed Amazon EC2 instance, plus a separate fee for the service provider. This additional fee is charged by the service provider managing your instance and is billed separately. It covers the cost of services provided for operating and maintaining your managed instance.

All Amazon EC2 purchasing options are available for managed instances, including On-Demand Instances, Reserved Instances, Spot Instances, and Savings Plans. By sourcing your compute directly from EC2 and then providing it to your service provider, you benefit from any existing Reserved Instances or Savings Plans applied to your account, ensuring that you're using the most cost-effective compute capacity available.

For example, when using Amazon EKS Auto Mode, you pay the standard EC2 instance rate for the underlying instances, plus an additional charge from Amazon EKS for managing the instances on your behalf. If you then decide to sign up for a Savings Plans, the EC2 instance rate is reduced by the Savings Plans, while the additional charge from Amazon EKS remains unchanged.

Identify managed instances

Managed instances are identified by a **true** value in the **Managed** field. The service provider is identified in the **Operator** field (in the console) or `Principal` field (in the CLI).

Use the following procedures to identify managed instances.

Console

To identify a managed 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 you want to check.
4. On the **Details** tab (if you selected the checkbox) or in the summary area (if you selected the instance ID), find the **Managed** field.

- A value of **true** indicates a managed instance.
- A value of **false** indicates a non-managed instance.

5. If **Managed** is set to **true**, the **Operator** field displays a value identifying the service provider responsible for managing the instance. For example, a value of **eks.amazonaws.com** identifies Amazon EKS as the service provider.

AWS CLI

To identify a managed instance

Use the describe-instances command and specify the instance ID.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].Operator
```

The following is example output. If `Managed` is `true`, the instance is a managed instance and a `Principal` is included. The principal is the service provider that manages the instance. For example, a value of `eks.amazonaws.com` identifies Amazon EKS as the service provider.

```
[
    {
        "Managed": true,
        "Principal": "eks.amazonaws.com"
    }
]
```

To find your managed instances

Use the describe-instances command and specify the `operator.managed` filter with a value of `true`. The `--query` option displays only the IDs of the managed instances.

```
aws ec2 describe-instances \
    --filters "Name=operator.managed,Values=true" \
    --query Reservations[*].Instances[].InstanceId
```

PowerShell

To identify a managed instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.Operator
```

The following is example output.

```
Managed Principal
------- ---------
True    eks.amazonaws.com
```

To find your managed instances

Use the Get-EC2Instance cmdlet. This example displays only the IDs of the managed instances.

```
(Get-EC2Instance -Filter @{Name="operator.managed";
 Values="true"}).Instances.InstanceId
```

Get started with managed instances

For guidance on using managed instances, see Automate cluster infrastructure with EKS Auto Mode in the *Amazon EKS User Guide*.

Amazon EC2 billing and purchasing options

You can use the following options to optimize your costs for Amazon EC2:

- **On-Demand Instances** – Pay, by the second, for the instances that you launch.
- **Savings Plans** – Reduce your Amazon EC2 costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years.
- **Reserved Instances** – Reduce your Amazon EC2 costs by making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years.
- **Spot Instances** – Request unused EC2 instances, which can reduce your Amazon EC2 costs significantly.
- **Dedicated Hosts** – Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
- **Dedicated Instances** – Pay, by the hour, for instances that run on single-tenant hardware.
- **Capacity Reservations** – Reserve capacity for your EC2 instances in a specific Availability Zone.

If you can't make a commitment to a specific instance configuration, but you can commit to a usage amount, purchase Savings Plans to reduce your On-Demand Instance costs. If you require a capacity reservation, purchase Reserved Instances or Capacity Reservations for a specific Availability Zone. Capacity Blocks can be used to reserve a cluster of GPU instances. Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if they can be interrupted. Dedicated Hosts or Dedicated Instances can help you address compliance requirements and reduce costs by using your existing server-bound software licenses.

For more information, see Amazon EC2 Pricing and Amazon EC2 managed instances.

Purchasing On-Demand Instances for Amazon EC2

With On-Demand Instances, you pay for compute capacity by the second with no long-term commitments. You have full control over the instance's lifecycle—you decide when to launch, stop, hibernate, start, reboot, or terminate it.

There is no long-term commitment required when you purchase On-Demand Instances. You pay only for the seconds that your On-Demand Instances are in the `running` state, with a 60-second minimum. The price per second for a running On-Demand Instance is fixed, and is listed on the Amazon EC2 Pricing, On-Demand Pricing page.

We recommend that you use On-Demand Instances for applications with short-term, irregular workloads that cannot be interrupted.

For significant savings over On-Demand Instances, use AWS Savings Plans, Spot Instances, or Reserved Instances for Amazon EC2 overview.

Contents

On-Demand Instance quotas

There are quotas for the number of running On-Demand Instances per AWS account per Region. On-Demand Instance quotas 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. Each quota type specifies the maximum number of vCPUs for one or more instance families.

Your account includes the following quotas for On-Demand Instances. Instances that are in the pending, stopping, stopped, and hibernated states do not count towards your On-Demand Instance quotas. Capacity Reservations count toward your On-Demand Instance quotas, even if they are unused.

Name	Default	Adjustable
Running On-Demand DL instances	0	Yes
Running On-Demand F instances	0	Yes
Running On-Demand G and VT instances	0	Yes
Running On-Demand HPC instances	0	Yes
Running On-Demand High Memory instances	0	Yes
Running On-Demand Inf instances	0	Yes
Running On-Demand P instances	0	Yes
Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances	5	Yes
Running On-Demand Trn instances	0	Yes
Running On-Demand X instances	0	Yes

For information about the different instance families, generations, and sizes, see the Amazon EC2 Instance Types Guide.

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 quota. For example, with a Standard instance quota 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.

Tasks

- Monitor On-Demand Instance quotas and usage
- Request a quota increase

Monitor On-Demand Instance quotas and usage

You can view and manage your On-Demand Instance quotas for each Region using the following methods.

To view your current quotas using the Service Quotas console

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/.
2. From the navigation bar, select a Region.
3. In the filter field, enter **`On-Demand`**.
4. The **Applied quota value** column displays the maximum number of vCPUs for each On-Demand Instance quota type for your account.

To view your current quotas using the AWS Trusted Advisor console

Open Service limits page in the AWS Trusted Advisor console.

To configure CloudWatch alarms

With Amazon CloudWatch metrics integration, you can monitor your EC2 usage against your quotas. You can also configure alarms to warn about approaching quotas. For more information, see Service Quotas and Amazon CloudWatch alarms in the *Service Quotas User Guide*.

Request a quota increase

Even though Amazon EC2 automatically increases your On-Demand Instance quotas based on your usage, you can request a quota increase if necessary. For example, if you intend to launch more instances than your current quota allows, you can request a quota increase by using the Service Quotas console described in Amazon EC2 service quotas.

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 User Guide*.

Reserved Instances for Amazon EC2 overview

> **Important**
>
> We recommend Savings Plans over Reserved Instances. Saving Plans are the easiest and most flexible way to save money on your AWS compute costs and offer lower prices (up to 72% off On-Demand pricing), just like Reserved Instances. However, Savings Plans are different to Reserved Instances. With Reserved Instances, you make a commitment to a specific instance configuration, whereas with Savings Plans, you have the flexibility to use the instance configurations that best meet your needs. To use Savings Plans, you make a commitment to a consistent usage amount, measured in USD per hour. For more information, see the AWS Savings Plans User Guide.

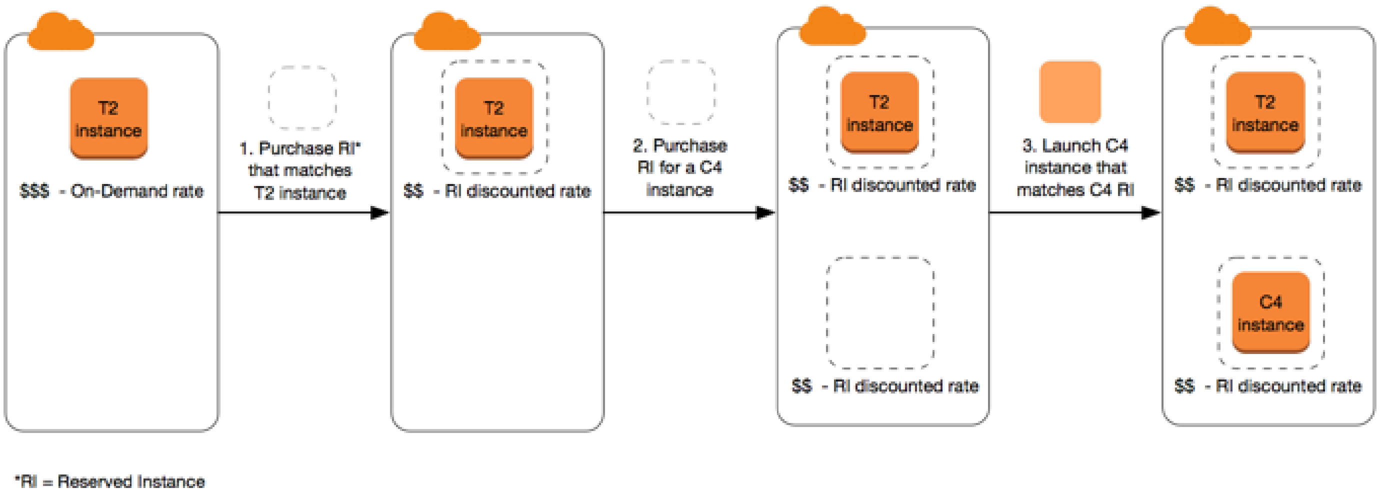
Reserved Instances provide you with significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. Reserved Instances are not physical instances, but rather a billing discount applied to the use of On-Demand Instances in your account. These On-Demand Instances must match certain attributes, such as instance type and Region, in order to benefit from the billing discount.

Reserved Instances topics

- Reserved Instance example scenario
- Key variables that determine Reserved Instance pricing
- Regional and zonal Reserved Instances (scope)
- Types of Reserved Instances (offering classes)
- How Reserved Instance discounts are applied
- Use your Reserved Instances
- How billing works with Reserved Instances
- Buy Reserved Instances for Amazon EC2
- Sell Reserved Instances for Amazon EC2 in the Reserved Instance Marketplace
- Modify Reserved Instances
- Exchange Convertible Reserved Instances
- Reserved Instance quotas

Reserved Instance example scenario

The following diagram shows a basic scenario 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 Amazon EC2 Dedicated Instances.
- **Platform**: The operating system; for example, Windows or Linux/Unix. For more information, see Choosing a platform.

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.

> **Important**
>
> After you purchase a Reserved Instance, you cannot cancel your purchase. However, you might be able to modify, exchange, or sell your Reserved Instance if your needs change.

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 Reserved Instances for Amazon EC2 in the Reserved Instance Marketplace.

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).

> **Important**
>
> After you purchase a Reserved Instance, you cannot cancel your purchase. However, you might be able to modify, exchange, or sell your Reserved Instance if your needs change.

For more information, see the Amazon EC2 Reserved Instances Pricing page.

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 and Amazon EC2 Reserved Instances Pricing.

For more information about how to specify the scope of a Reserved Instance, see RI Attributes, specifically the **Availability Zone** bullet.

Differences between regional and zonal Reserved Instances

The following table highlights some key differences between regional Reserved Instances and zonal Reserved Instances:

	Regional Reserved Instances	**Zonal Reserved Instances**
Ability to reserve capacity	A regional Reserved Instance does *not* reserve capacity.	A zonal Reserved Instance reserves capacity in the specified Availability Zone.
Availability Zone flexibility	The Reserved Instance discount applies to instance usage in any Availability Zone in the specified Region.	No Availability Zone flexibility—the Reserved Instance discount applies to instance usage in the specified Availability Zone only.
Instance size flexibility	The Reserved Instance discount applies to instance usage within the instance family, regardless of size. Only supported on Amazon Linux/Unix Reserved Instances with default tenancy. For more information, see Instance size flexibility determined by normalization factor.	No instance size flexibility—the Reserved Instance discount applies to instance usage for the specified instance type and size only.

	Regional Reserved Instances	Zonal Reserved Instances
Queuing a purchase	You can queue purchases for regional Reserved Instances.	You can't queue purchases for zonal Reserved Instances.

For more information and examples, see How Reserved Instance discounts are applied.

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.

	Standard Reserved Instance	Convertible Reserved Instance
Modify Reserved Instances	Some attributes can be modified. For more informati on, see Modify Reserved Instances.	Some attributes can be modified. For more informati on, see Modify Reserved Instances.
Exchange Reserved Instances	Can't be exchanged.	Can be exchanged during the term for another Convertib le Reserved Instance with new attributes, including instance family, instance type, platform, scope, or tenancy. For more informati

	Standard Reserved Instance	Convertible Reserved Instance
		on, see Exchange Convertible Reserved Instances.
Sell in the Reserved Instance Marketplace	Can be sold in the Reserved Instance Marketplace.	Can't be sold in the Reserved Instance Marketplace.
Buy in the Reserved Instance Marketplace	Can be bought in the Reserved Instance Marketpla ce.	Can't be bought in the Reserved Instance Marketpla ce.

How Reserved Instance discounts are applied

Reserved Instances are not physical instances, but rather a billing discount that is applied to the running On-Demand Instances in your account. The On-Demand Instances must match certain specifications of the Reserved Instances in order to benefit from the billing discount.

If you purchase a Reserved Instance and you already have a running On-Demand Instance that matches the specifications of the Reserved Instance, the billing discount is applied immediately and automatically. You do not have to restart your instances. If you do not have an eligible running On-Demand Instance, launch an On-Demand Instance with the same specifications as your Reserved Instance. For more information, see Use your Reserved Instances.

The offering class (Standard or Convertible) of the Reserved Instance does not affect how the billing discount is applied.

Topics

- How zonal Reserved Instances are applied
- How regional Reserved Instances are applied
- Instance size flexibility
- Examples of applying Reserved Instances

How zonal Reserved Instances are applied

A Reserved Instance that is purchased to reserve capacity in a specific Availability Zone is called a zonal Reserved Instance.

- The Reserved Instance discount applies to matching instance usage in that Availability Zone.
- The attributes (tenancy, platform, Availability Zone, instance type, and instance size) of the running instances must match that of the Reserved Instances.

For example, if you purchase two `c4.xlarge` default tenancy Linux/Unix Standard Reserved Instances for 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.

How regional Reserved Instances are applied

A Reserved Instance that is purchased for a Region is called a regional Reserved Instance, and provides Availability Zone and instance size flexibility.

- The Reserved Instance discount applies to instance usage in any Availability Zone in that Region.
- The Reserved Instance discount applies to instance usage within the instance family, regardless of size—this is known as instance size flexibility.

Instance size flexibility

With instance size flexibility, the Reserved Instance discount applies to instance usage for instances that have the same family. The Reserved Instance is applied from the smallest to the largest instance size within the instance family based on the normalization factor. For an example of how the Reserved Instance discount is applied, see Scenario 2: Reserved Instances in a single account using the normalization factor.

Limitations

- **Supported:** Instance size flexibility is only supported for Regional Reserved Instances.
- **Not supported:** Instance size flexibility is *not supported* for the following Reserved Instances:
  - Reserved Instances that are purchased for a specific Availability Zone (zonal Reserved Instances)
  - Reserved Instances for G4ad, G4dn, G5, G5g, G6, G6e, G6f, Gr6, Gr6f, hpc7a, P5, Inf1, Inf2, u7i-6tb, and u7i-8tb instances
  - Reserved Instances for Windows Server, Windows Server with SQL Standard, Windows Server with SQL Server Enterprise, Windows Server with SQL Server Web, RHEL, and SUSE Linux Enterprise Server

- Reserved Instances with dedicated tenancy

Instance size flexibility determined by normalization factor

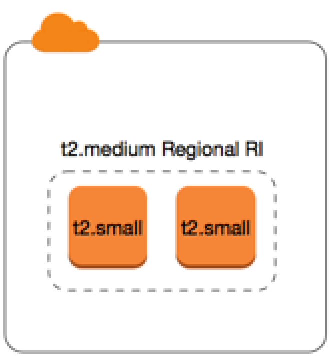
Instance size flexibility is determined by the normalization factor of the instance size. The discount applies either fully or partially to running instances of the same instance family, depending on the instance size of the reservation, in any Availability Zone in the Region. The only attributes that must be matched are the instance family, tenancy, and platform.

The following table lists the different sizes within an instance family, and the corresponding normalization factor. This scale is used to apply the discounted rate of Reserved Instances to the normalized usage of the instance family.

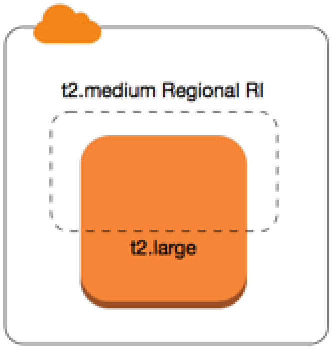
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80

Instance size	Normalization factor
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
112xlarge	896

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.

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>
<tr><th>Instance size</th><th>Normalization factor</th></tr>
<tr><td>a1.metal</td><td>32</td></tr>
<tr><td>m5zn.metal | x2iezn.metal z1d.metal</td><td>96</td></tr>
<tr><td>c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal</td><td>128</td></tr>
<tr><td>c5n.metal</td><td>144</td></tr>
</table>

Instance size	Normalization factor
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

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.

- Scenario 1: Reserved Instances in a single account
- Scenario 2: Reserved Instances in a single account using the normalization factor
- Scenario 3: Regional Reserved Instances in linked accounts

- [Scenario 4: Zonal Reserved Instances in a linked account]

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 full billing benefit to the 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.

Scenario 2: Reserved Instances in a single account using the normalization factor

You are running the following On-Demand Instances in account A:

- 2 x `m3.xlarge` Amazon Linux, default tenancy instances in Availability Zone us-east-1a
- 2 x `m3.large` Amazon Linux, default tenancy instances in Availability Zone us-east-1b

You purchase the following Reserved Instance in account A:

- 1 x `m3.2xlarge` Amazon Linux, default tenancy Reserved Instance in Region us-east-1

The Reserved Instance benefits are applied in the following way:

- The `m3.2xlarge` 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. It applies first to the `m3.large` instances and then to the `m3.xlarge` instances, because it applies from the smallest to the largest instance size within the instance family based on the normalization factor.

  An `m3.large` instance is equivalent to 4 normalized units/hour.

  An `m3.xlarge` instance is equivalent to 8 normalized units/hour.

  An `m3.2xlarge` instance is equivalent to 16 normalized units/hour.

  The benefit is applied as follows:

  The `m3.2xlarge` regional Reserved Instance provides full benefit to 2 x `m3.large` usage, because together these instances account for 8 normalized units/hour. This leaves 8 normalized units/hour to apply to the `m3.xlarge` instances.

With the remaining 8 normalized units/hour, the `m3.2xlarge` regional Reserved Instance provides full benefit to 1 x `m3.xlarge` usage, because each `m3.xlarge` instance is equivalent to 8 normalized units/hour. The remaining `m3.xlarge` usage is charged at the On-Demand rate.

Scenario 3: 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. 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.

Scenario 4: 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 Understanding your reservations in the AWS Cost and Usage Report.

> **Note**
>
> Zonal Reserved Instances reserve capacity only for the owning account and cannot be shared with other AWS accounts. If you need to share capacity with other AWS accounts, use Reserve compute capacity with EC2 On-Demand Capacity Reservations.

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 On-Demand Instance to take advantage of the billing benefit of a Reserved Instance, ensure that you specify the following information when you configure your On-Demand Instance:

Platform

You must specify an Amazon Machine Image (AMI) that matches the platform (product description) of your Reserved Instance. For example, if you specified `Linux/UNIX` for your Reserved Instance, you can launch an instance from an Amazon Linux AMI or an Ubuntu AMI.

Instance type

If you purchased a zonal Reserved Instance, you must specify the same instance type as your Reserved Instance; for example, `t3.large`. For more information, see How zonal Reserved Instances are applied.

If you purchased a regional Reserved Instance, you must specify an instance type from the same instance family as the instance type of your Reserved Instance. For example, if you specified `t3.xlarge` for your Reserved Instance, you must launch your instance from the T3 family, but you can specify any size, for example, `t3.medium`. For more information, see How regional Reserved Instances are applied.

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 the instance into any Availability Zone in the Region that you specified for the Reserved Instance.

Tenancy

The tenancy (`dedicated` or `shared`) of the instance must match the tenancy of your Reserved Instance. For more information, see Amazon EC2 Dedicated Instances.

For examples of how Reserved Instances are applied to your running On-Demand Instances, see How Reserved Instance discounts are applied. For more information, see Why aren't my Amazon EC2 Reserved Instances applying to my AWS billing in the way that I expected?

You can use various methods to launch the On-Demand Instances that use your Reserved Instance discount. For information about the different launch methods, see Launch an Amazon EC2 instance. You can also use Amazon EC2 Auto Scaling to launch an instance. For more information, see the Amazon EC2 Auto Scaling User Guide.

How billing works with Reserved Instances

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 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.

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 the standard pricing. For information, see AWS Free Tier.

Contents

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 Amazon EC2 instance state changes.

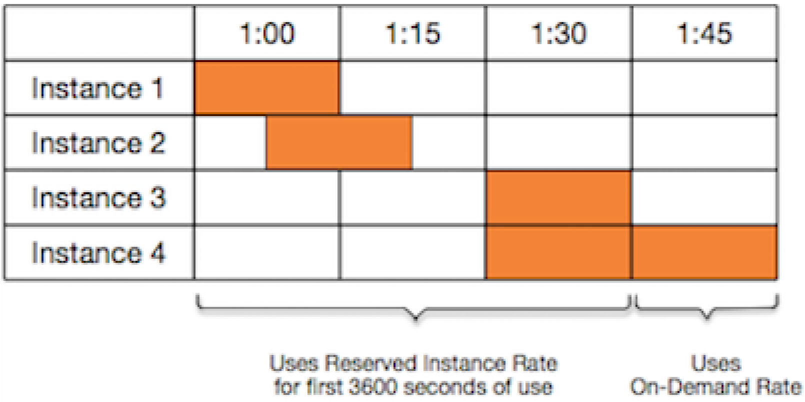
A Reserved Instance billing benefit can be applied to a running instance on a per-second basis. Per-second billing is available for instances using an open-source Linux distribution, such as Amazon Linux and Ubuntu. Per-hour billing is used for commercial Linux distributions, such as Red Hat Enterprise Linux and SUSE Linux Enterprise Server.

A Reserved Instance billing benefit can apply to a maximum of 3600 seconds (one hour) of instance usage per clock-hour. You can run multiple instances concurrently, but can only receive the benefit of the Reserved Instance discount for a total of 3600 seconds per clock-hour; instance usage that exceeds 3600 seconds in a clock-hour is billed at the On-Demand rate.

For example, if you purchase one `m4.xlarge` Reserved Instance and run four `m4.xlarge` instances concurrently for one hour, one instance is charged at one hour of Reserved Instance usage and the other three instances are charged at three hours of On-Demand usage.

However, if you purchase one `m4.xlarge` Reserved Instance and run four `m4.xlarge` instances for 15 minutes (900 seconds) each within the same hour, the total running time for the instances is one hour, which results in one hour of Reserved Instance usage and 0 hours of On-Demand usage.

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 Understanding your reservations.

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.

> **Note**
>
> Zonal Reserved Instances reserve capacity only for the owning account and cannot be shared with other AWS accounts. If you need to share capacity with other AWS accounts, use Reserve compute capacity with EC2 On-Demand Capacity Reservations.

Reserved Instance discount pricing tiers

If your account qualifies for a discount pricing tier, it automatically receives discounts on upfront and instance usage fees for Reserved Instance purchases that you make within that tier level from that point on. To qualify for a discount, the list value of your Reserved Instances in the Region must be $500,000 USD or more.

The following rules apply:

- Pricing tiers and related discounts apply only to purchases of Amazon EC2 Standard Reserved Instances.
- Pricing tiers do not apply to Reserved Instances for Windows with SQL Server Standard, SQL Server Web, and SQL Server Enterprise.
- Pricing tiers do not apply to Reserved Instances for Linux with SQL Server of any kind.
- 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

You can determine the pricing tier for your account by calculating the list value for all of your Reserved Instances in a Region. Multiply the hourly recurring price for each reservation by the total number of hours for the term and add the undiscounted upfront price (also known as the fixed price) at the time of purchase. Because the list value is based on undiscounted (public) pricing, it is not affected if you qualify for a volume discount or if the price drops after you buy your Reserved Instances.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

For example, for a 1-year Partial Upfront `t2.small` Reserved Instance, assume the upfront price is $60.00 and the hourly rate is $0.007. This provides a list value of $121.32.

```
121.32 = 60.00 + (0.007 * 8760)
```

To view the fixed price values for Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Reserved Instances**.
3. To display the **Upfront price** column, choose settings ( ) in the top-right corner, turn on **Upfront price**, and choose **Confirm**.

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)

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 Buy Reserved Instances for Amazon EC2.

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.

If your list value falls below the price point for that discount pricing tier—for example, if some of your Reserved Instances expire—future purchases of Reserved Instances in the Region are not discounted. However, you continue to get the discount applied against any Reserved Instances that were originally purchased within the discount pricing tier.

When you buy Reserved Instances, one of four possible scenarios occurs:

- **No discount**—Your purchase within a Region is still below the discount threshold.
- **Partial discount**—Your purchase within a Region crosses the threshold of the first discount tier. No discount is applied to one or more reservations and the discounted rate is applied to the remaining reservations.
- **Full discount—**Your entire purchase within a Region falls within one discount tier and is discounted appropriately.
- **Two discount rates**—Your purchase within a Region crosses from a lower discount tier to a higher discount tier. You are charged two different rates: one or more reservations at the lower discounted rate, and the remaining reservations at the higher discounted rate.

Crossing pricing tiers

If your purchase crosses into a discounted pricing tier, you see multiple entries for that purchase: one for that part of the purchase charged at the regular price, and another for that part of the purchase charged at the applicable discounted rate.

The Reserved Instance service generates several Reserved Instance IDs because your purchase crossed from an undiscounted tier, or from one discounted tier to another. There is an ID for each set of reservations in a tier. Consequently, the ID returned by your purchase CLI command or API action is different from the actual ID of the new Reserved Instances.

Consolidated billing for pricing tiers

A consolidated billing account aggregates the list value of member accounts within a Region. When the list value of all active Reserved Instances for the consolidated billing account reaches a discount pricing tier, any Reserved Instances purchased after this point by any member of the consolidated billing account are charged at the discounted rate (as long as the list value for that consolidated account stays above the discount pricing tier threshold). For more information, see Reserved Instances and consolidated billing.

Buy Reserved Instances for Amazon EC2

To buy a Reserved Instance for Amazon EC2, you can use the Amazon EC2 console, a command line tool, or an SDK to 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. When you are purchasing a third-party seller’s Reserved Instance from the Amazon EC2 Reserved Instance Marketplace, if there are offerings similar to your choice but at a lower upfront price, AWS sells you the offerings at the lower upfront 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. You can queue a purchase for a future date, and cancel the queued purchase before its scheduled time.

To purchase and modify Reserved Instances, ensure that your user has the appropriate permissions, such as the ability to describe Availability Zones. For information, see the section called “Work with Reserved Instances” (API) or the section called “Work with Reserved Instances” (console).

Tasks

- Choosing a platform
- Queue your purchase
- Buy Standard Reserved Instances

- Buy Convertible Reserved Instances
- Buy from the Reserved Instance Marketplace
- Cancel a queued purchase
- Renew a Reserved Instance

Choosing a platform

Amazon EC2 supports the following platforms for Reserved Instances:

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux
- Red Hat Enterprise Linux with HA
- Windows
- Windows with SQL Server Standard
- Windows with SQL Server Web
- Windows with SQL Server Enterprise

Considerations

- If you bring your existing subscription (BYOS) for **Red Hat Enterprise Linux**, **SUSE Linux**, or **Ubuntu Pro**, you must choose an offering for the **Linux/Unix** platform.
- Reserved Instances are not supported on instances running **macOS** or Ubuntu Pro (EC2 subscription-included, i.e., not BYOS). For saving with On-Demand instance pricing, we recommend that you use macOS and Ubuntu Pro (EC2 subscription-included) instances with Savings Plans. For more information, see Savings Plans User Guide.

To ensure that an instance runs in a specific Reserved Instance, the platform of the Reserved Instance must match the platform of the AMI used to launch the instance. For Linux AMIs, it is important to check whether the AMI platform uses the general value **Linux/UNIX** or a more specific value like **SUSE Linux**.

Console

To check the AMI platform

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select the AMI.
4. On the **Details** tab, note the value of **Platform details**.

AWS CLI

To check the AMI platform

Use the describe-images command and check the value of `PlatformDetails`.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

The following is example output.

```
[
    "Linux/UNIX"
]
```

PowerShell

To check the AMI platform

Use the Get-EC2Image cmdlet and check the value of `PlatformDetails`.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

The following is example output.

```
PlatformDetails
---------------
Linux/UNIX
```

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 set a date for your queued purchases in the Amazon EC2 console, and the purchase is queued until 00:00 UTC on this date. To specify a different time for the queued purchase, use an AWS SDK or command line tool.

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.

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.

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 need to restart your instances. If you do not have a suitable running instance, launch an instance and ensure that you match the exact criteria that you specified for your Reserved Instance. For more information, see Use your Reserved Instances.

For examples of how Reserved Instances are applied to your running instances, see How Reserved Instance discounts are applied.

Console

To buy Standard Reserved Instances

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.
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.

    If the status goes to `Retired`, AWS might not have received your payment.

AWS CLI

To buy a Standard Reserved Instance

1. Find available Reserved Instances using the describe-reserved-instances-offerings command. Specify `standard` for the `--offering-class` option to return only Standard Reserved Instances. You can apply additional criteria to narrow your results. For example, use the following command to purchase a regional `t2.large` Reserved Instance with a default tenancy for `Linux/UNIX` for a 1-year term only.

```
aws ec2 describe-reserved-instances-offerings \
    --instance-type t2.large \
    --offering-class standard \
    --product-description "Linux/UNIX" \
    --instance-tenancy default \
    --filters Name=duration,Values=31536000 \
              Name=scope,Values=Region
```

   To find Reserved Instances on the Reserved Instance Marketplace only, use the `marketplace` filter and do not specify a duration in the request, as the term 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 option 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 \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

To buy a Standard Reserved Instance

1. Find available Reserved Instances using the Get-EC2ReservedInstancesOffering cmdlet. Specify `standard` for the `-OfferingClass` parameter to return only Standard Reserved Instances. You can apply additional criteria to narrow your results. For example, use the following command to purchase a regional `t2.large` Reserved Instance with a default tenancy for `Linux/UNIX` for a 1-year term only.

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="duration"; Values="31536000"} `
             @{Name="scope"; Values="Region"
```

To find Reserved Instances on the Reserved Instance Marketplace only, use the `marketplace` filter and do not specify a duration in the request, as the term might be shorter than a 1– or 3-year term.

```
Get-EC2ReservedInstancesOffering `
    -InstanceType t2.large `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
```

```
    -InstanceTenancy default `
    -Filters @{Name="marketplace"; Values="true"}
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. Use the New-EC2ReservedInstance cmdlet 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.

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. Use the Get-EC2ReservedInstance cmdlet to get the status of your Reserved Instance.

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

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.

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.

For examples of how Reserved Instances are applied to your running instances, see How Reserved Instance discounts are applied.

Console

To buy Convertible Reserved Instances

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering class**, choose **Convertible** to display Convertible Reserved Instances.
4. To purchase a capacity reservation, 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.

    If the status goes to `Retired`, AWS might not have received your payment.

AWS CLI

To buy a Convertible Reserved Instance

1. Find available Reserved Instances using the describe-reserved-instances-offerings command. Specify `convertible` for the `--offering-class` option to return only Convertible Reserved Instances. You can apply additional criteria to narrow your results. For example, use the following command 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 option 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 \
       --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
   ```

```
    --query ReservedInstances[].State
```

PowerShell

To buy a Convertible Reserved Instance

1. Find available Reserved Instances using the Get-EC2ReservedInstancesOffering cmdlet. Specify `convertible` for the `-OfferingClass` parameter to return only Convertible Reserved Instances. You can apply additional criteria to narrow your results. For example, use the following command to purchase a regional `t2.large` Reserved Instance with a default tenancy for `Linux/UNIX`.

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "convertible" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="scope"; Values="Region"
```

   When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. Use the New-EC2ReservedInstance cmdlet to buy your Reserved Instance. You must specify the Reserved Instance offering ID that you obtained the previous step and you must specify the number of instances for the reservation.

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

   By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. Use the Get-EC2ReservedInstance cmdlet to get the status of your Reserved Instance.

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

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.

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 Support.

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.

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**.

AWS CLI

To cancel a queued purchase

Use the delete-queued-reserved-instances command.

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327
```

PowerShell

To cancel a queued purchase

Use the Remove-EC2QueuedReservedInstance cmdlet.

```
Remove-EC2QueuedReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327
```

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.

You must renew a Reserved Instance using the Amazon EC2 console.

To renew a 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 the Reserved Instance to renew.
4. Choose **Actions**, **Renew Reserved Instances**.
5. To complete the order, choose **Order all**, and then **Close**.

Sell Reserved Instances for Amazon EC2 in the Reserved Instance Marketplace

The Amazon EC2 Reserved Instance Marketplace is a platform that facilitates the sale of unused Standard Reserved Instances from AWS customers and third-party sellers. These Reserved Instances can vary in term lengths and pricing options. You might want to sell your Reserved Instances when you no longer need them, such as when you move your instances to a new AWS Region, change to a different instance type, finish projects before the Reserved Instance term expires, your business needs change, or you have excess 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 to purchase a third-party seller’s Reserved Instance via the Reserved Instance Marketplace, 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.

Contents

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.

The following limitations and restrictions apply when selling Reserved Instances:

- Only Amazon EC2 Standard regional and zonal Reserved Instances can be sold in the Reserved Instance Marketplace.
- Amazon EC2 Convertible Reserved Instances can't be sold in the Reserved Instance Marketplace.
- Reserved Instances for other AWS services, such as Amazon RDS and Amazon ElastiCache, cannot be sold in the Reserved Instance Marketplace.
- There must be at least one month remaining in the term of the Standard Reserved Instance.
- You can't 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 as long as they have been active in your account for at least 30 days. Additionally, if there is an upfront payment on a Reserved Instance, it can only be sold after AWS has received the upfront payment.
- You can't sell a Reserved Instance in the Reserved Instance Marketplace if you purchased it using a volume discount.

- 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. You can also modify your Reserved Instances before listing them. For information, see Modify Reserved Instances.
- AWS charges a service fee of 12 percent of the total upfront price of each Standard Reserved Instance you sell in the Reserved Instance Marketplace. The upfront price is the price the seller is charging for the Standard Reserved Instance.
- When you register as a seller, the bank you specify must have a US address. For more information, see Additional seller requirements for paid products in the *AWS Marketplace Seller Guide*.
- Amazon Web Services India Private Limited (AWS India) 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 AWS India 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.
- **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.

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 Form 1099-K FAQs on the IRS website.

Price your Reserved Instances

When setting the price for your Reserved Instances, consider the following:

- **Upfront price** – The upfront price is the only price that you can specify for the Reserved Instance that you're selling. The upfront price is the one-time price that the buyer pays when they purchase each Reserved Instance.

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.

The minimum allowed price in the Reserved Instance Marketplace is $0.00.

- **Limits –** The following limits for selling Reserved Instances apply to the *lifetime* of your AWS account. They are not annual limits and they can't be increased.
  - **You can sell up to $50,000 in Reserved Instances**.
  - **You can sell up to 5,000 Reserved Instances**.
- **Can't modify –** You cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.
- **Can cancel –** 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.

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.

Console

To list a Reserved Instance in the Reserved Instance Marketplace

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 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. Select the Reserved Instance that you've listed and choose the **My Listings** tab near the bottom of the page.

AWS CLI

To manage Reserved Instances in the Reserved Instance Marketplace

1. Get a list of your Reserved Instances by using the describe-reserved-instances command. Note that ID of the Reserved Instance that you want to list.
2. Use the create-reserved-instances-listing command. 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-listing command.

PowerShell

To manage Reserved Instances in the Reserved Instance Marketplace

1. Get a list of your Reserved Instances by using the Get-EC2ReservedInstance cmdlet. Note that ID of the Reserved Instance that you want to list.
2. Use the New-EC2ReservedInstancesListing cmdlet. You must specify the ID of the Reserved Instance, the number of instances, and the pricing schedule.
3. To view your listing, use the Get-EC2ReservedInstancesListing cmdlet.
4. To cancel your listing, use the Stop-EC2ReservedInstancesListing cmdlet.

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.

Console

To track the status of a Reserved Instance listing

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation page, choose **Reserved Instances**.
3. On the **My Listings** tab, find the value of **Listing State**. The tab also contains information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and canceled.

AWS CLI

To track the status of a Reserved Instance listing

Use the describe-reserved-instances-listings command with the appropriate filter to obtain information about your listings.

```
aws ec2 describe-reserved-instances-listings
```

PowerShell

To track the status of a Reserved Instance listing

Use the Get-EC2ReservedInstancesListing cmdlet.

```
Get-EC2ReservedInstancesListing
```

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 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 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, instance size (within the same instance family and generation), 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.

You can modify all or a subset of your Reserved Instances. You can separate your original Reserved Instances into two or more new Reserved Instances. For example, if you have a reservation for 10 instances in `us-east-1a` and decide to move 5 instances to `us-east-1b`, the modification request results in two new reservations: one for 5 instances in `us-east-1a` and the other for 5 instances in `us-east-1b`.

You can also *merge* two or more Reserved Instances into a single Reserved Instance. For example, if you have four `t2.small` Reserved Instances of one instance each, you can merge them to create one `t2.large` Reserved Instance. For more information, see [Support for modifying instance sizes](#).

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

- Submit modification requests
- Troubleshoot modification requests

Requirements and restrictions for modification

You can modify these attributes as follows.

Modifiable attribute	Supported platforms	Limitations and considerations
Change **Availability Zones** within the same Region	Linux and Windows	-
Change the **scope** from Availability Zone to Region and vice versa	Linux and Windows	A zonal Reserved Instance is scoped to an Availability Zone and reserves capacity in that Availability Zone. If you change the scope from Availability Zone to Region (in other words, from zonal to regional), you lose the capacity reservation benefit. A regional Reserved Instance is scoped to a Region. Your Reserved Instance discount can apply to instances running in any Availability Zone in that Region. Furthermore, the Reserved Instance discount applies to instance usage across all sizes in the selected instance family. If you change the scope from Region to Availability Zone (in other

Modifiable attribute	Supported platforms	Limitations and considera tions
		words, from regional to zonal), you lose Availability Zone flexibility and instance size flexibility (if applicable). For more information, see [How Reserved Instance discounts are applied].
Change the **instance size** within the same instance family and generation	Linux/UNIX only Instance size flexibility is not available for Reserved Instances on the other platforms, which include Linux with SQL Server Standard, Linux with SQL Server Web, Linux with SQL Server Enterprise, Red Hat Enterprise Linux, SUSE Linux, Windows, Windows with SQL Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web.	The reservation must use default tenancy. Some instance families are not supported, because there are no other sizes available. For more information, see [Support for modifying instance sizes]

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
- There must be a match between the instance size footprint of the original reservation and the new configuration. For more information, see Support for modifying instance sizes.
- 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
- For modifying instance size, the Reserved Instance must support instance size flexibility. For the list of Reserved Instances that don't support instance size flexibility, see Instance size flexibility.

Support for modifying instance sizes

You can modify the instance size of a Reserved Instance if the following requirements are met.

Requirements

- The platform is Linux/UNIX.
- You must select another instance size in the same instance family (indicated by a letter, for example, T) and generation (indicated by a number, for example, 2).

  For example, you can modify a Reserved Instance from `t2.small` to `t2.large` because they're both in the same T2 family and generation. But you can't modify a Reserved Instance from T2 to M2 or from T2 to T3, because in both these examples, the target instance family and generation are not the same as the original Reserved Instance.
- You can modify the instance size of a Reserved Instance only if it supports instance size flexibility. For the list of Reserved Instances that don't support instance size flexibility, see Instance size flexibility.
- You can't modify the instance size of Reserved Instances for `t1.micro` instances, because `t1.micro` has only one size.
- The original and new Reserved Instance must have the same instance size footprint.

Contents

Instance size footprint

Each Reserved Instance has an *instance size footprint*, which is determined by the normalization factor of the instance size and the number of instances in the reservation. When you modify the instance sizes in an Reserved Instance, the footprint of the new configuration must match that of the original configuration, otherwise the modification request is not processed.

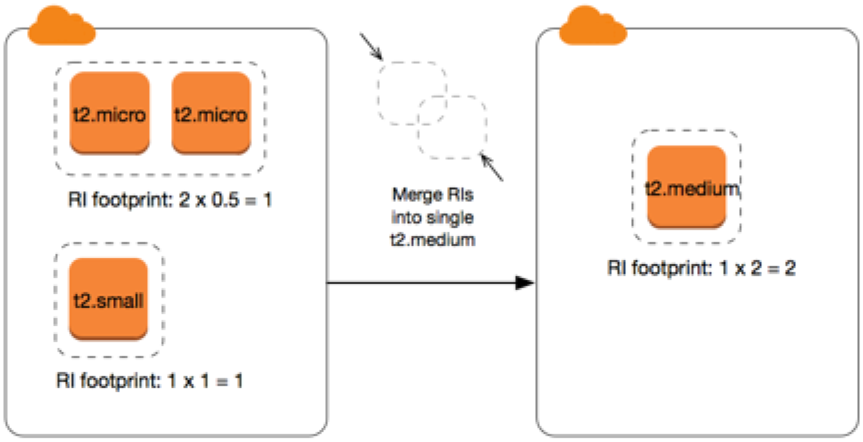
To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. In the Amazon EC2 console, the normalization factor is measured in units. The following table describes the normalization factor for the instance sizes in an instance family. For example, `t2.medium` has a normalization factor of 2, so a reservation for four `t2.medium` instances has a footprint of 8 units.

Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80

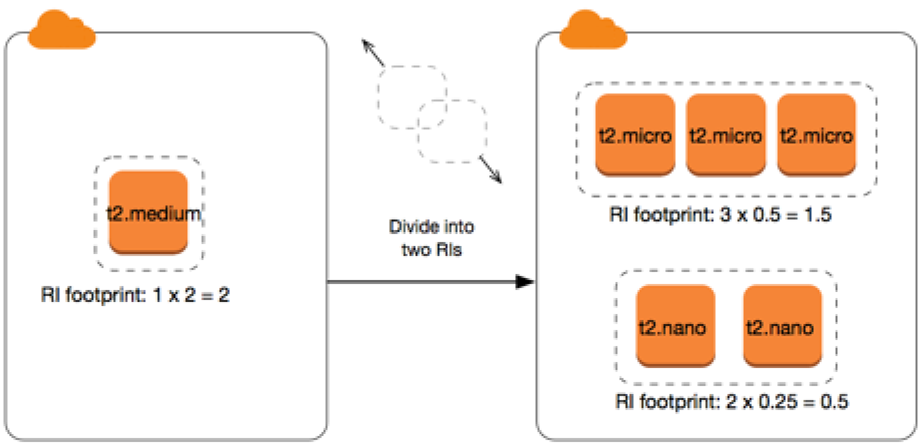
Instance size	Normalization factor
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
112xlarge	896

You can allocate your reservations into different instance sizes across the same instance family as long as the instance size footprint of your reservation remains the same. For example, you can divide a reservation for one `t2.large` (1 @ 4 units) instance into four `t2.small` (4 @ 1 unit) instances. Similarly, you can combine a reservation for four `t2.small` instances into one `t2.large` instance. However, you cannot change your reservation for two `t2.small` instances into one `t2.large` instance because the footprint of the new reservation (4 units) is larger than the footprint of the original reservation (2 units).

In the following example, you have a reservation with two `t2.micro` instances (1 unit) and a reservation with one `t2.small` instance (1 unit). If you merge both of these reservations to a single reservation with one `t2.medium` instance (2 units), the footprint of the new reservation equals the footprint of the combined reservations.



You can also modify a reservation to divide it into two or more reservations. In the following example, you have a reservation with a `t2.medium` instance (2 units). You can divide the reservation into two reservations, one with two `t2.nano` instances (.5 units) and the other with three `t2.micro` instances (1.5 units).



Normalization factors for bare metal instances

You can modify a reservation with `metal` instances using other sizes within the same instance family. Similarly, you can modify a reservation with instances other than bare metal instances using the `metal` size within the same instance family. Generally, a bare metal instance is the same size as the largest available instance size within the same instance family. For example, an `i3.metal` instance is the same size as an `i3.16xlarge` instance, so they have the same normalization factor.

The following table describes the normalization factor for the bare metal instance sizes in the instance families that have bare metal instances. The normalization factor for `metal` instances depends on the instance family, unlike the other instance sizes.

Instance size	Normalization factor
a1.metal	32
m5zn.metal | x2iezn.metal z1d.metal	96
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

For example, an `i3.metal` instance has a normalization factor of 128. If you purchase an `i3.metal` default tenancy Amazon Linux/Unix Reserved Instance, you can divide the reservation as follows:

- An `i3.16xlarge` is the same size as an `i3.metal` instance, so its normalization factor is 128 (128/1). The reservation for one `i3.metal` instance can be modified into one `i3.16xlarge` instance.
- An `i3.8xlarge` is half the size of an `i3.metal` instance, so its normalization factor is 64 (128/2). The reservation for one `i3.metal` instance can be divided into two `i3.8xlarge` instances.
- An `i3.4xlarge` is a quarter the size of an `i3.metal` instance, so its normalization factor is 32 (128/4). The reservation for one `i3.metal` instance can be divided into four `i3.4xlarge` instances.

Submit modification requests

Before you modify your Reserved Instances, ensure that you have read the applicable restrictions. Before you modify the instance size, calculate the total instance size footprint of the original reservations that you want to modify and ensure that it matches the total instance size footprint of your new configurations.

Console

To modify your Reserved Instances

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**.

   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.
   - **Instance type**: Select the required instance type. The combined configurations must equal the instance size footprint of your original configurations.
   - **Count**: Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose **Add**, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.
4. 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

AWS CLI

To modify your Reserved Instances

Use the modify-reserved-instances command. You can provide the configuration details in a JSON file.

```
aws ec2 modify-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 \
    --target-configurations file://configuration.json
```

To get the status of your modification request

Use the describe-reserved-instances-modifications command. The status is `processing`, `fulfilled`, or `failed`.

```
aws ec2 describe-reserved-instances-modifications \
    --reserved-instances-modification-ids rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687
 \
    --query ReservedInstancesModifications[].Status
```

PowerShell

To modify your Reserved Instances

Use the Edit-EC2ReservedInstance cmdlet. You can provide the configuration details in an object of type `Amazon.EC2.Model.ReservedInstancesConfiguration`.

```
Edit-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342f5bd7c02 `
    -TargetConfiguration $configuration
```

To get the status of your modification request

Use the Get-EC2ReservedInstancesModification cmdlet. The status is `processing`, `fulfilled`, or `failed`.

```
Get-EC2ReservedInstancesModification `
    -ReservedInstancesModificationId rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687 | `
    Select Status
```

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 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 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 .

Contents

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
- Have at least 24 hours remaining before it expires

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. For more information about modifying your Reserved Instances, see Modify Reserved Instances.
- 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.
- When Amazon EC2 exchanges a Convertible Reserved Instance, it retires the associated reservation, and transfers the end date to the new reservation. After the exchange, Amazon EC2 sets both the end date for the old reservation and the start date for the new reservation equal to the date of the exchange. For example, if you exchange a three-year reservation that had 16 months left in its term, the new reservation is a 16-month reservation with the same end date as the reservation from the Convertible Reserved Instance that you exchanged.

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 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.

```
$35/$10 = 3.5
```

You can exchange your Convertible Reserved Instance for three $10 Convertible Reserved Instances. It's not possible to purchase half reservations; therefore you must purchase an additional Convertible Reserved Instance to cover the remainder:

```
3.5 = 3 whole Convertible Reserved Instances + 1 additional Convertible Reserved
 Instance
```

The fourth Convertible Reserved Instance has the same end date as the other three. If you are exchanging Partial or All Upfront Convertible Reserved Instances, you pay the true-up cost for the fourth reservation. If the remaining upfront cost of your Convertible Reserved Instances is $500, and the new reservation would normally cost $600 on a prorated basis, you are charged $100.

```
$600 prorated upfront cost of new reservations - $500 remaining upfront cost of old
 reservations = $100 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 old Convertible Reserved Instances, or the highest of the Convertible Reserved Instances. The expiration date for the new Convertible Reserved Instance is the expiration date that's furthest in the future.

For example, you have the following Convertible Reserved Instances in your account:

Reserved Instance ID	Term	Expiration date
aaaa1111	1-year	2018-12-31
bbbb2222	1-year	2018-07-31
cccc3333	3-year	2018-06-30
dddd4444	3-year	2019-12-31

- You can merge aaaa1111 and bbbb2222 and exchange them for a 1-year Convertible Reserved Instance. You cannot exchange them for a 3-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2018-12-31.
- You can merge bbbb2222 and cccc3333 and exchange them for a 3-year Convertible Reserved Instance. You cannot exchange them for a 1-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2018-07-31.
- You can merge cccc3333 and dddd4444 and exchange them for a 3-year Convertible Reserved Instance. You cannot exchange them for a 1-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2019-12-31.

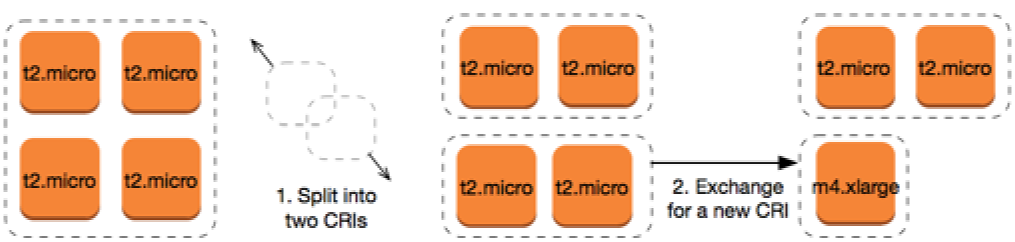
Exchange a portion of a Convertible Reserved Instance

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:

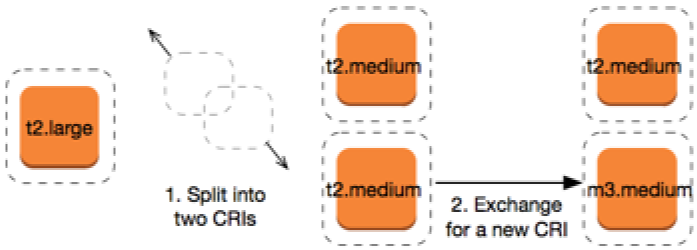
1. Modify the `t2.micro` Convertible Reserved Instance by splitting it into two `t2.micro` Convertible Reserved Instances with two instances each.
2. Exchange one of the new `t2.micro` Convertible Reserved Instances for an `m4.xlarge` Convertible Reserved Instance.



Example Example: Convertible Reserved Instance with a single instance

In this example, you have a `t2.large` Convertible Reserved Instance. To change it to a smaller `t2.medium` instance and a `m3.medium` instance:

1. Modify the `t2.large` Convertible Reserved Instance by splitting it into two `t2.medium` Convertible Reserved Instances. A single `t2.large` instance has the same instance size footprint as two `t2.medium` instances.
2. Exchange one of the new `t2.medium` Convertible Reserved Instances for an `m3.medium` Convertible Reserved Instance.



For more information, see Support for modifying instance sizes and Submit exchange requests.

Submit exchange requests

You can exchange your Convertible Reserved Instances. Reserved Instances that are exchanged are retired.

Console

To exchange Convertible Reserved Instances

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**.

AWS CLI

To exchange a Convertible Reserved Instance

1. Find a new Convertible Reserved Instance that meets your needs by using the describe-reserved-instances-offerings command.
2. Get a quote for the exchange by using the get-reserved-instances-exchange-quote command. This includes the number of Reserved Instances you get from the exchange, and the true-up cost for the exchange:
3. Perform the exchange by using the accept-reserved-instances-exchange-quote command.

PowerShell

To exchange a Convertible Reserved Instance

1. Find a new Convertible Reserved Instance that meets your needs by using the Get-EC2ReservedInstancesOffering cmdlet.
2. Get a quote for the exchange by using the GetEC2-ReservedInstancesExchangeQuote cmdlet. This includes the number of Reserved Instances you get from the exchange, and the true-up cost for the exchange:
3. Perform the exchange by using the Approve-EC2ReservedInstancesExchangeQuote cmdlet

Reserved Instance quotas

You can purchase new Reserved Instances each month. The number of new Reserved Instances that you can purchase each month is determined by your monthly quota, as follows:

Quota description	Default quota
New regional Reserved Instances	20 per Region per month
New zonal Reserved Instances	20 per Availability Zone per month

For example, in a Region with three Availability Zones, the default quota is 80 new Reserved Instances per month, calculated as follows:

- 20 regional Reserved Instances for the Region

- Plus 60 zonal Reserved Instances (20 for each of the three Availability Zones)

Instances in the `running` state count toward your quota. Instances that are in the `pending`, `stopping`, `stopped`, and `hibernated` states do not count towards your quota.

View the number of Reserved Instances you have purchased

The number of Reserved Instances that you purchase is indicated by the **Instance count** field (console) or the `InstanceCount` parameter (AWS CLI). When you purchase new Reserved Instances, the quota is measured against the total instance count. For example, if you purchase a single Reserved Instance configuration with an instance count of 10, the purchase counts towards your quota as 10, not 1.

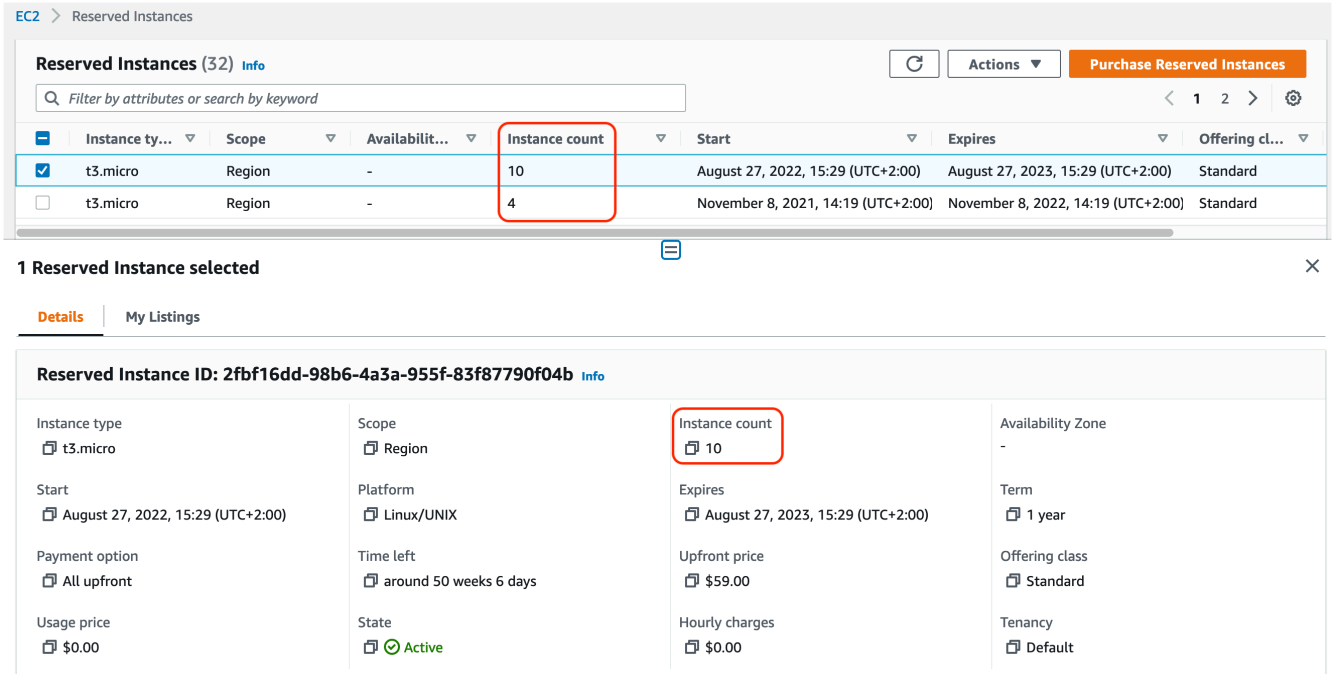
You can view how many Reserved Instances you have purchased by using the Amazon EC2 or the AWS CLI.

Console

To view the number of Reserved Instances you have purchased

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Reserved Instances**.
3. Select a Reserved Instance configuration from the table, and check the **Instance count** field.

   In the following screenshot, the selected line represents a single Reserved Instance configuration for a `t3.micro` instance type. The **Instance count** column in the table view and the **Instance count** field in the detail view (outlined in the screenshot) indicate that there are 10 Reserved Instances for this configuration.



AWS CLI

To view the number of Reserved Instances you have purchased

Use the describe-reserved-instances command and specify the ID of the Reserved Instance configuration.

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --output table
```

The following is example output. The `InstanceCount` field indicates that there are 10 Reserved Instances for this configuration.

```
-------------------------------------------------------------------
|                    DescribeReservedInstances                    |
+-----------------------------------------------------------------+
||                      ReservedInstances                        ||
|+----------------------+----------------------------------------+|
||  CurrencyCode        |  USD                                   ||
||  Duration            |  31536000                              ||
||  End                 |  2023-08-27T13:29:44+00:00             ||
||  FixedPrice          |  59.0                                  ||
```

```
||  InstanceCount       |  10                                   ||
||  InstanceTenancy     |  default                              ||
||  InstanceType        |  t3.micro                             ||
||  OfferingClass       |  standard                             ||
||  OfferingType        |  All Upfront                          ||
||  ProductDescription  |  Linux/UNIX                           ||
||  ReservedInstancesId |  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ||
||  Scope               |  Region                               ||
||  Start               |  2022-08-27T13:29:45.938000+00:00     ||
||  State               |  active                               ||
||  UsagePrice          |  0.0                                  ||
|+----------------------+---------------------------------------+|
|||                     RecurringCharges                        |||
||+----------------------------------+--------------------------+||
|||  Amount                          |  0.0                     |||
|||  Frequency                       |  Hourly                  |||
||+----------------------------------+--------------------------+||
```

PowerShell

To view the number of Reserved Instances you have purchased

Use the Get-EC2ReservedInstance Cmdlet and specify the ID of the Reserved Instance configuration.

```
Get-EC2ReservedInstance -ReservedInstancesId a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

The following is example output. The `InstanceCount` field indicates that there are 10 Reserved Instances for this configuration.

```
AvailabilityZone    :
CurrencyCode        : USD
Duration            : 31536000
End                 : 1/12/2017 8:57:08 PM
FixedPrice          : 0
InstanceCount       : 10
InstanceTenancy     : default
InstanceType        : t3.medium
OfferingClass       : standard
OfferingType        : All Upfront
ProductDescription  : Windows
RecurringCharges    : {}
```

```
ReservedInstancesId : a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
Scope               : Region
Start               : 10/12/2016 4:00:00 PM
State               : active
Tags                : {}
UsagePrice          : 0
```

Considerations

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 a 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.

View your Reserved Instance quotas and request a quota increase

The Amazon EC2 console provides quota information. You can also request an increase in your quotas. For more information, see View your current quotas and Request an increase.

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.

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.

For a comparison of the different purchasing options for EC2 instances, see Amazon EC2 billing and purchasing options.

Concepts

Before you get started with Spot Instances, you should be familiar with the following concepts:

- *Spot capacity pool* – A set of unused EC2 instances with the same instance type (for example, `m5.large`) and Availability Zone.
- *Spot price* – The current price of a Spot Instance per hour.
- *Spot Instance request* – Requests a Spot Instance. When capacity is available, Amazon EC2 fulfills your request. A Spot Instance request is either *one-time* or *persistent*. Amazon EC2 automatically resubmits a persistent Spot Instance request after the Spot Instance associated with the request is interrupted.
- *EC2 instance rebalance recommendation* – Amazon EC2 emits an instance rebalance recommendation signal to notify you that a Spot Instance is at an elevated risk of interruption. This signal provides an opportunity to proactively rebalance your workloads across existing or new Spot Instances without having to wait for the two-minute Spot Instance interruption notice.
- *Spot Instance interruption* – Amazon EC2 terminates, stops, or hibernates your Spot Instance when Amazon EC2 needs the capacity back. Amazon EC2 provides a Spot Instance interruption notice, which gives the instance a two-minute warning before it is interrupted.

Differences between Spot Instances and On-Demand Instances

The following table lists the key differences between Spot Instances and On-Demand Instances.

	Spot Instances	On-Demand Instances

	Spot Instances	On-Demand Instances
Launch time	Can only be launched immediately if the Spot Instance request is active and capacity is available.	Can only be launched immediately if you make a manual launch request and capacity is available.
Available capacity	If capacity is not available, the Spot 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 long-term supply and 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, Amazon EC2 can interrupt an individual Spot Instance if capacity is no longer available.	You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).

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. Your Spot Instances run until you terminate them, capacity is no longer available, 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.

Spot Instances are not covered by Savings Plans. If you have a Savings Plans, it does not provide additional savings on top of the savings that you already get from using Spot Instances. Furthermore, your spend on Spot Instances does not apply the commitments in your Compute Savings Plans.

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. For more information, see View Spot Instance pricing history.

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.

View billing

Your bill provides details about your service usage. For more information, see Viewing your bill in the *AWS Billing User Guide*.

Best practices for Amazon EC2 Spot

Amazon EC2 provides access to spare EC2 compute capacity in the AWS Cloud through Spot Instances at savings of up to 90% 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 notice, if Amazon EC2 needs to reclaim the capacity. To ensure the best experience with Spot Instances, it's important to understand and apply best practices for their use.

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. We do not recommend Spot Instances for workloads that are intolerant of occasional periods when the entire target capacity is not completely available. While following Spot best practices to be flexible about instance types and Availability Zones provides the best chance for high availability, there are no guarantees that capacity will be available, because surges in demand for On-Demand Instances can disrupt workloads on Spot Instances.

We strongly discourage using Spot Instances for these workloads or attempting to fail over to On-Demand Instances to handle interruptions or periods of unavailability. Failing over to On-Demand Instances can inadvertently drive interruptions for your other Spot Instances. In addition if Spot Instances for a combination of instance type and Availability Zone get interrupted, it might become difficult for you to get On-Demand Instances with that same combination.

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
- Be flexible about instance types and Availability Zones
- Use attribute-based instance type selection
- Use Spot placement scores to identify optimal Regions and Availability Zones
- Use EC2 Auto Scaling groups or EC2 Fleet to manage your aggregate capacity
- Use the price and capacity optimized allocation strategy
- Use integrated AWS services to manage your Spot Instances
- Which is the best Spot request method to use?

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 signal that notifies you when a Spot Instance is at an 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 signal by using the Capacity Rebalancing feature in Auto Scaling groups and EC2 Fleet.

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. 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 and Spot Instance interruptions.

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 attribute-based instance type selection

With attribute-based instance type selection, you can specify instance attributes—such as vCPUs, memory, and storage—for the workload you want to run. EC2 Auto Scaling or EC2 Fleet will then automatically identify and launch instances that match your specified attributes. This removes the effort required to manually select specific instance types, which requires an in-depth understanding of each instance type's offering.

Moreover, attribute-based instance type selection enables you to automatically use newly released instance types as they become available. This ensures seamless access to an increasingly broad range of Spot Instance capacity.

Attribute-based instance type selection is ideal for workloads and frameworks that can be flexible about the instance types they run on, such as High Performance Computing (HPC) and big data workloads.

For more information, see Create mixed instances group using attribute-based instance type selection in the *Amazon EC2 Auto Scaling User Guide* and Specify attributes for instance type selection for EC2 Fleet or Spot Fleet in this guide.

Use Spot placement scores to identify optimal Regions and Availability Zones

Spot Instances are unused EC2 capacity, and this capacity fluctuates based on EC2 supply and demand. As a result, you might not always get the exact Spot capacity that you require in a specific location at a specific time. To mitigate this unpredictability, you can use the Spot placement score feature. This feature provides recommendations for Regions or Availability Zones that are more likely to have sufficient capacity to meet your Spot capacity needs without requiring you to launch Spot Instances in those locations first.

Spot placement score is best used for workloads that can be flexible about the instance types and the Region or Availability Zone they can use. All you need to do is specify the Spot capacity that you need, your instance type requirements, and whether you want a recommendations for Regions or Availability Zones. In return, you receive a score ranging from 1 to 10 for each Region or Availability Zone, indicating the likelihood of successfully provisioning your requested Spot capacity in that location. A score of 10 indicates that your Spot request is highly likely to succeed.

It's important to note that a Spot placement score is a point-in-time recommendation, because capacity can vary over time. It does not guarantee available capacity or predict the risk of interruption.

You can use the Spot placement score feature in the Amazon EC2 console, AWS CLI, or an SDK. For more information, see Spot placement score.

Use EC2 Auto Scaling groups or EC2 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 EC2 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 an EC2 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 an EC2 Fleet in this user guide.

Use the price and 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 `price-capacity-optimized` strategy because this strategy automatically provisions instances from the most-available Spot capacity pools that also have the lowest possible price. You can also take advantage of the `price-capacity-optimized` allocation strategy in EC2 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, see Allocation strategies for multiple instance types in the *Amazon EC2 Auto Scaling User Guide* and When workloads have a high cost of interruption in this user guide.

Use integrated AWS services to manage your Spot Instances

Other AWS services integrate with Spot to reduce overall compute costs without the need to manage the individual instances or fleets. We recommend that you consider the following solutions for your applicable workloads: Amazon EMR, Amazon Elastic Container Service, AWS Batch, Amazon Elastic Kubernetes Service, Amazon SageMaker AI, AWS Elastic Beanstalk, and Amazon GameLift Servers. To learn more about Spot best practices with these services, see the Amazon EC2 Spot Instances Workshops Website.

Which is the best Spot request method to use?

Use the following table to determine which API to use when requesting Spot Instances.

API	When to use?	Use case	Should I use this API?
[CreateAutoScalingGroup]()	• You need multiple instances with either a single configuration or a mixed configuration. • You want to automate the lifecycle management through a configurable API.	Create an Auto Scaling group that manages the lifecycle of your instances while maintaining the desired number of instances. Supports horizontal scaling (adding more instances) between specified minimum and maximum limits.	Yes
[CreateFleet]()	• You need multiple instances with either a single configuration or a mixed configuration. • You want to self-manage your instance lifecycle. • If you don't need auto scaling, we recommend that you use an	Create a fleet of both On-Demand Instances and Spot Instances in a single request, with multiple launch specifications that vary by instance type, AMI, Availability Zone, or subnet. The Spot Instance allocation strategy defaults to `lowest-price` per unit, but you can change it to `price-capacity-`	Yes – in `instant` mode if you don't need auto scaling

API	When to use?	Use case	Should I use this API?
	`instant` type fleet.	`optimized` , `capacity-optimized` , or `diversified` .	
RunInstances	• You're already using the RunInstances API to launch On-Demand Instances, and you simply want to change to launching Spot Instances by changing a single parameter. • You do not need multiple instances with different instance types.	Launch a specified number of instances using an AMI and one instance type.	No – because RunInstances does not allow mixed instance types in a single request

API	When to use?	Use case	Should I use this API?
RequestSpotFleet	• We strongly discourage using the RequestSpotFleet API because it is a legacy API with no planned investment. • If you want to manage your instance lifecycle, use the CreateFleet API. • If you don't want to manage your instance lifecycle, use the CreateAutoScalingGroup API.	DO NOT USE. RequestSpotFleet is legacy API with no planned investment.	No
RequestSpotInstances	• We strongly discourage using the RequestSpotInstances API because it is a legacy API with no planned investment.	DO NOT USE. RequestSpotInstances is legacy API with no planned investment.	No

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 by using the launch instance wizard in the Amazon EC2 console or the run-instances command. For more information, see Manage your Spot Instances.
- 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.
- 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.

Your Spot Instance launches if there is available capacity. Your Spot Instance runs until you stop or terminate it, or until Amazon EC2 interrupts it (known as a *Spot Instance interruption*). Amazon EC2 can stop, terminate, or hibernate a Spot Instance when it interrupts it.

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 or when the supply of Spot Instances decreases. When Amazon EC2 interrupts a Spot Instance, it provides a Spot Instance interruption notice, which gives the instance a two-minute warning before Amazon EC2 interrupts it. You can't enable termination protection for Spot Instances. For more information, see Spot Instance interruptions.

Contents

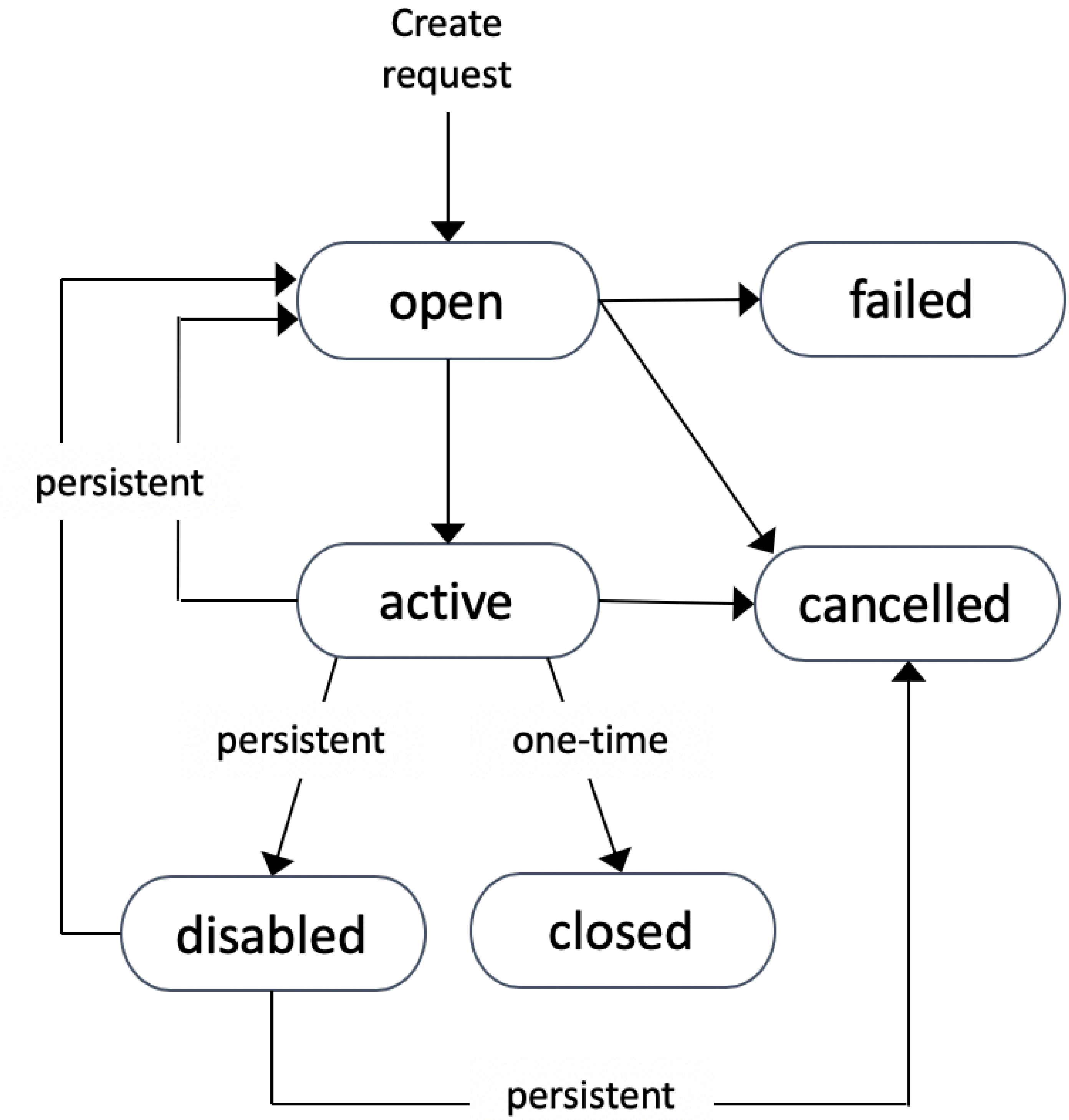
- [Launch on single-tenant hardware]

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 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 capacity is not available, your Spot Instance is interrupted. After your instance is interrupted, when 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. 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, Start a Spot Instance, and Terminate a Spot Instance.

You can track the status of your Spot Instance requests, as well as the status of the Spot Instances launched, through the status. For more information, see Get the status of a Spot Instance request.

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, 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 Amazon EC2 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.

- [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.

Launch burstable performance instances

The T instance types are burstable performance instances. If you launch your Spot Instances using a burstable performance instance type, 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 to avoid paying higher costs. If you launch burstable performance Spot Instances in Unlimited mode 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.

T2 instances, when configured in Standard mode, get launch credits. T2 instances are the only burstable performance instances that get launch credits. 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 for T2 Spot Instances, or use an instance type with dedicated CPU.

Launch on single-tenant hardware

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 Amazon EC2 Dedicated Instances and the Amazon EC2 Dedicated Instances.

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 Manage your Spot Instances.
- Request a Spot Instance in a VPC with an instance tenancy of `dedicated`. For more information, see Launch Dedicated Instances into a VPC with default tenancy. You can't 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.

View 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 your Spot request is fulfilled, your Spot Instances launch at the current Spot price, not exceeding the On-Demand price. You can view the Spot price history for the last 90 days, filtering by instance type, operating system, and Availability Zone.

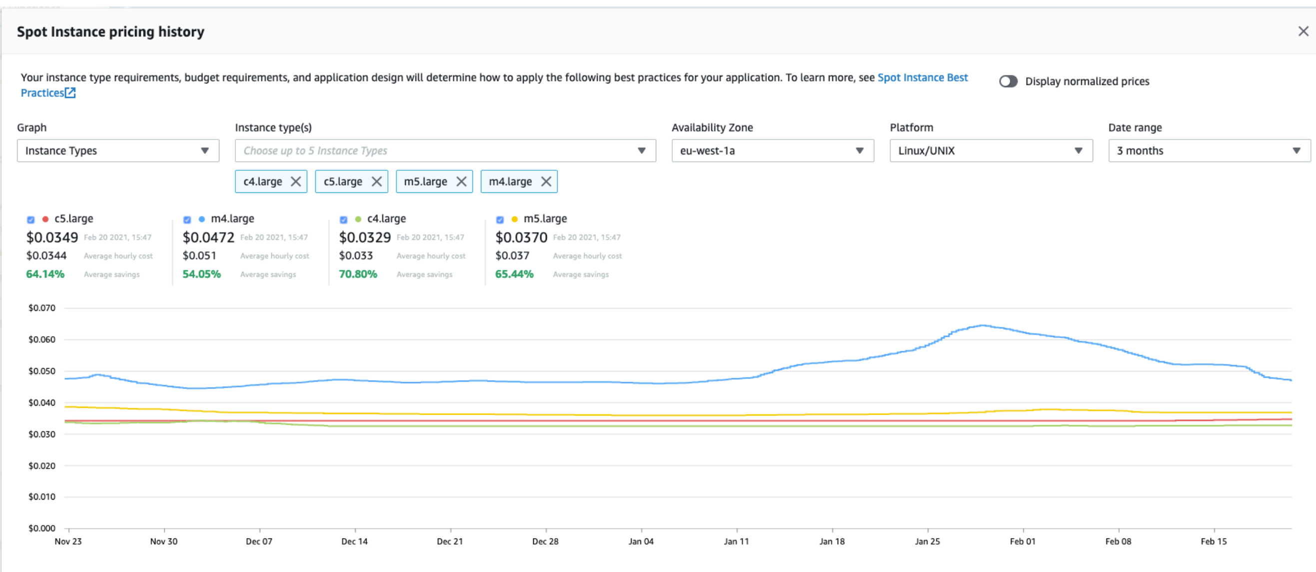
For the *current* Spot Instance prices, see Amazon EC2 Spot Instances Pricing.

Console

To view the Spot price history

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. Hover (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**.

AWS CLI

To view the Spot price history

Use the following describe-spot-price-history command.

```
aws ec2 describe-spot-price-history \
    --instance-types c6i.xlarge \
    --product-descriptions "Linux/UNIX" \
    --start-time 2025-04-01T00:00:00 \
    --end-time 2025-04-02T00:00:0
```

PowerShell

To view the Spot price history

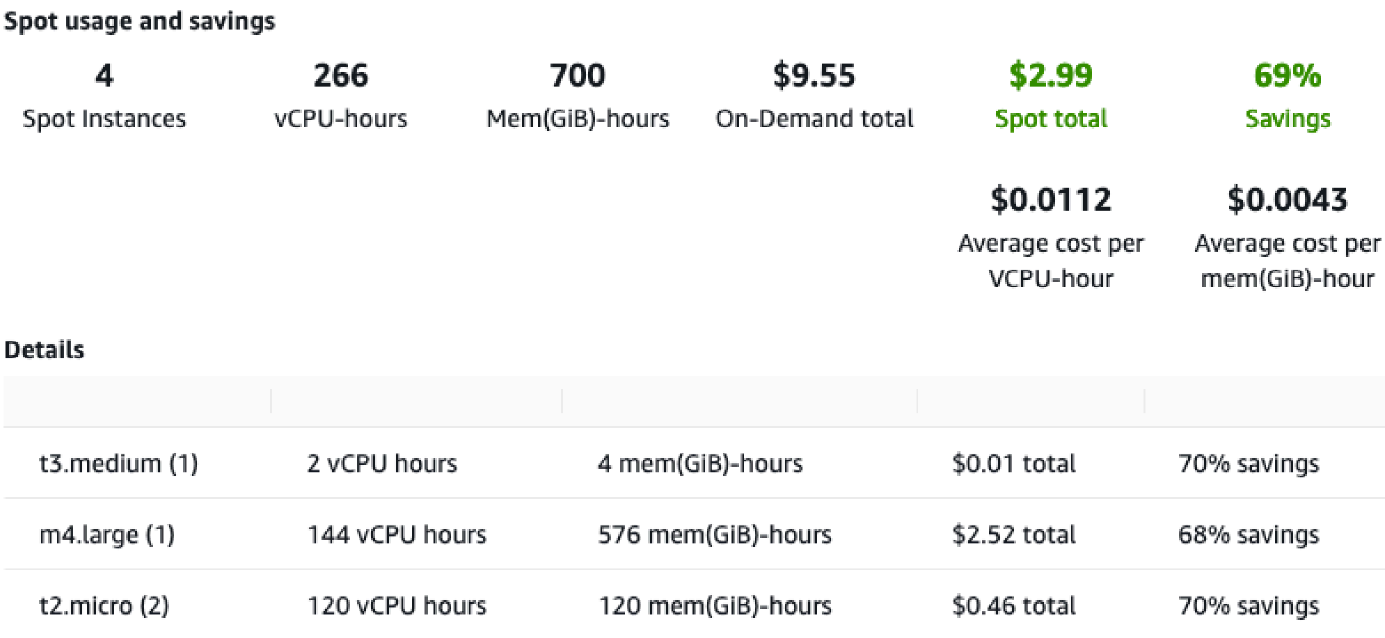
Use the Get-EC2SpotPriceHistory cmdlet.

```
Get-EC2SpotPriceHistory `
    -InstanceType c6i.xlarge `
    -ProductDescription "Linux/UNIX" `
    -UtcStartTime 2025-04-01T00:00:00 `
    -UtcEndTime 2025-04-02T00:00:0
```

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.



You can view the following usage and savings information:

- **Spot Instances** – The number of Spot Instances launched and terminated by the Spot Fleet. When viewing the savings summary, the number represents all your running Spot Instances.
- **vCPU-hours** – The number of vCPU hours used across all the Spot Instances for the selected time frame.
- **Mem(GiB)-hours** – The number of GiB hours used across all the Spot Instances for the selected time frame.
- **On-Demand total** – The total amount you would've paid for the selected time frame had you launched these instances as On-Demand Instances.
- **Spot total** – The total amount to pay for the selected time frame.
- **Savings** – The percentage that you are saving by not paying the On-Demand price.
- **Average cost per vCPU-hour** – The average hourly cost of using the vCPUs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per vCPU-hour** = **Spot total / vCPU-hours**.
- **Average cost per mem(GiB)-hour** – The average hourly cost of using the GiBs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per mem(GiB)-hour** = **Spot total / Mem(GiB)-hours**.

- **Details** table – The different instance types (the number of instances per instance type is in parentheses) that comprise the Spot Fleet. When viewing the savings summary, these comprise all your running Spot Instances.

Savings information can only be viewed using the Amazon EC2 console.

To view the savings information for a Spot Fleet

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 checkbox 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

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**.

Create a Spot Instance request

To use Spot Instances, you create a Spot Instance request that includes the desired number of instances, the instance type, and the Availability Zone. If capacity is available, Amazon EC2 fulfills your request immediately. Otherwise, Amazon EC2 waits until your request can be fulfilled or until you cancel the request.

You can use the launch instance wizard in the Amazon EC2 console or the run-instances command to request a Spot Instance in the same way that you can launch an On-Demand Instance. This method is only recommended for the following reasons:

- You're already using the launch instance wizard or run-instances command to launch On-Demand Instances, and you simply want to change to launching Spot Instances by changing a single parameter.
- You do not need multiple instances with different instance types.

This method is generally not recommended for launching Spot Instances because you can't specify multiple instance types, and you can't launch Spot Instances and On-Demand Instances in the same request. For the preferred methods for launching Spot Instances, which include launching a *fleet* that includes Spot Instances and On-Demand Instances with multiple instance types, see Which is the best Spot request method to use?

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 Get the status of a Spot Instance request.

Console

To create a Spot Instance request

Steps 1–9 are the same steps you'd use to launch an On-Demand Instance. At Step 10, you configure the Spot Instance request.

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. (Optional) Under **Name and tags**, you can name your instance, and tag the Spot Instance request, the instance, the volumes, and the elastic graphics. For information about tags, see Tag your Amazon EC2 resources.

    a. For **Name**, enter a descriptive name for your instance.

       The instance name is a tag, where the key is **Name**, and the value is the name that you specify. If you don't specify a name, the instance can be identified by its ID, which is automatically generated when you launch the instance.

    b. To tag the Spot Instance request, the instance, the volumes, and the elastic graphics, choose **Add additional tags**. Choose **Add tag**, and then enter a key and value, and select the resource type to tag. Choose **Add tag** again for each additional tag to add.

5. Under **Application and OS Images (Amazon Machine Image)**, choose the operating system (OS) for your instance, and then select an AMI. For more information, see Application and OS Images (Amazon Machine Image).
6. Under **Instance type**, select the instance type that meets your requirements for the hardware configuration and size of your instance. For more information, see Instance type.
7. Under **Key pair (login)**, choose an existing key pair, or choose **Create new key pair** to create a new one. For more information, see Amazon EC2 key pairs and Amazon EC2 instances.

   > **Important**
   >
   > If you choose the **Proceed without key pair (Not recommended)** 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.

8. Under **Network settings**, use the default settings, or choose **Edit** to configure the network settings as necessary.

   Security groups form part of the network settings, and define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance.

   For more information, see Network settings.
9. The AMI you selected includes one or more volumes of storage, including the root device volume. Under **Configure storage**, you can specify additional volumes to attach to the instance by choosing **Add new volume**. For more information, see Configure storage.
10. Under **Advanced details**, configure the Spot Instance request as follows:

    a. Under **Purchasing option**, select the **Request Spot Instances** checkbox.

    b. You can either keep the default configuration for the Spot Instance request, or choose **Customize** (at the right) to specify custom settings for your Spot Instance request.

       When you choose **Customize**, the following fields appear.

       i. **Maximum price**: You can request Spot Instances at the Spot price, capped at the On-Demand price, or you can specify the maximum amount you're willing to pay.

> **Warning**
>
> If you specify a maximum price, your instances will be interrupted more frequently than if you choose **No maximum price**.
>
> If you specify a maximum price, it must be more than USD $0.001. Specifying a value below USD $0.001 will result in a failed launch.

- **No maximum price**: Your Spot Instance will launch at the current Spot price. The price will never exceed the On-Demand price. (Recommended)
- **Set your maximum price (per instance/hour)**: You can specify the maximum amount you're willing pay.
  - If you specify a maximum price that is less than the current Spot price, your Spot Instance will not launch.
  - If you specify a maximum price that is more than the current Spot price, your Spot Instance will launch and be charged at the current Spot price. After your Spot Instance is running, if the Spot price rises above your maximum price, Amazon EC2 interrupts your Spot Instance.
  - Regardless of the maximum price you specify, you will always be charged the current Spot price.

To review Spot price trends, see View Spot Instance pricing history.

ii. **Request type**: The Spot Instance request type that you choose determines what happens if your Spot Instance is interrupted.

- **One-time**: Amazon EC2 places a one-time request for your Spot Instance. If your Spot Instance is interrupted, the request is not resubmitted.
- **Persistent request**: Amazon EC2 places a persistent request for your Spot Instance. If your Spot Instance is interrupted, the request is resubmitted to replenish the interrupted Spot Instance.

If you do not specify a value, the default is a one-time request.

iii. **Valid to**: The expiration date of a *persistent* Spot Instance request.

This field is not supported for one-time requests. A *one-time* request remains active until all the instances in the request launch or you cancel the request.

- **No request expiry date**: The request remains active until you cancel it.
- **Set your request expiry date**: The persistent request remains active until the date that you specify, or until you cancel it.

iv. **Interruption behavior**: The behavior that you choose determines what happens when a Spot Instance is interrupted.

- For persistent requests, valid values are **Stop** and **Hibernate**. When an instance is stopped, charges for EBS volume storage apply.

> **Note**
>
> Spot Instances now use the same hibernation functionality as On-Demand Instances. To enable hibernation, you can either choose **Hibernate** here, or you can choose **Enable** from the **Stop - Hibernate behavior** field, which appears lower down in the launch instance wizard. For the hibernation prerequisites, see Prerequisites for EC2 instance hibernation.

- For one-time requests, only **Terminate** is valid.

If you do not specify a value, the default is **Terminate**, which is not valid for a persistent Spot Instance request. If you keep the default and try to launch a persistent Spot Instance request, you'll get an error.

For more information, see Behavior of Spot Instance interruptions.

11. On the **Summary** panel, for **Number of instances**, enter the number of instances to launch.

> **Note**
>
> Amazon EC2 creates a separate request for each Spot Instance.

12. On the **Summary** panel, review the details of your instance, and make any necessary changes. After you submit your Spot Instance request, you can't change the parameters

of the request. You can navigate directly to a section in the launch instance wizard by choosing its link in the **Summary** panel. For more information, see Summary.

13. When you're ready to launch your instance, choose **Launch instance**.

    If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

AWS CLI

To create a Spot Instance request using run-instances

Use the run-instances command and specify the Spot Instance options in the `--instance-market-options` parameter as follows.

```
--instance-market-options file://spot-options.json
```

The following is the data structure to specify in the JSON file. You can also specify `ValidUntil` and `InstanceInterruptionBehavior`. If you do not specify a field in the data structure, the default value is used.

The following example creates a `persistent` request.

```
{
  "MarketType": "spot",
  "SpotOptions": {
    "SpotInstanceType": "persistent"
  }
}
```

To create a Spot Instance request using request-spot-instances

> **Note**
>
> We strongly discourage using the request-spot-instances command to request a Spot Instance because it is a legacy API with no planned investment. For more information, see Which is the best Spot request method to use?

Use the request-spot-instances command to create a one-time request.

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json
```

Use the request-spot-instances command to create a persistent request.

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "persistent" \
    --launch-specification file://specification.json
```

For example launch specification files to use with these commands, see Spot Instance request example launch specifications. If you download a launch specification file from the Spot Requests console, you must use the request-spot-fleet command instead (the Spot Requests console specifies a Spot Instance request using a Spot Fleet).

PowerShell

To create a Spot Instance request

Use the New-EC2Instance cmdlet and specify the Spot Instance options using the `-InstanceMarketOption` parameter.

```
-InstanceMarketOptions $marketOptions
```

Create the data structure for the Spot Instance options as follows.

```
$spotOptions = New-Object Amazon.EC2.Model.SpotMarketOptions
$spotOptions.SpotInstanceType="persistent"
$marketOptions = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketOptions.MarketType = "spot"
$marketOptions.SpotOptions = $spotOptions
```

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 Manage your Spot Instances.

> **Important**
>
> We strongly discourage using the request-spot-instances command to request a Spot Instance because it is a legacy API with no planned investment. For more information, see Which is the best Spot request method to use?

Examples

- Example 1: Launch Spot Instances
- Example 2: Launch Spot Instances in the specified Availability Zone
- Example 3: Launch Spot Instances in the specified subnet
- Example 4: Launch a Dedicated Spot Instance

Example 1: Launch Spot Instances

The following example does not include an Availability Zone or subnet. Amazon EC2 selects an Availability Zone for you. Amazon EC2 launches the instances in the default subnet of the selected Availability Zone.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Example 2: Launch Spot Instances in the specified Availability Zone

The following example includes an Availability Zone. Amazon EC2 launches the instances in the default subnet of the specified Availability Zone.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
```

```
  "InstanceType": "m5.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Example 3: Launch Spot Instances in the specified subnet

The following example includes a subnet. Amazon EC2 launches the instances in the specified subnet. If the VPC is a nondefault VPC, the instance does not receive a public IPv4 address by default.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.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 the previous code block.

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "InstanceType": "m5.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
      "Groups": [ "sg-1a2b3c4d5e6f7g8h9" ],
      "AssociatePublicIpAddress": true
```

```
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Example 4: Launch a Dedicated Spot Instance

The following example requests Spot Instance with a tenancy of `dedicated`. A Dedicated Spot Instance must be launched in a VPC.

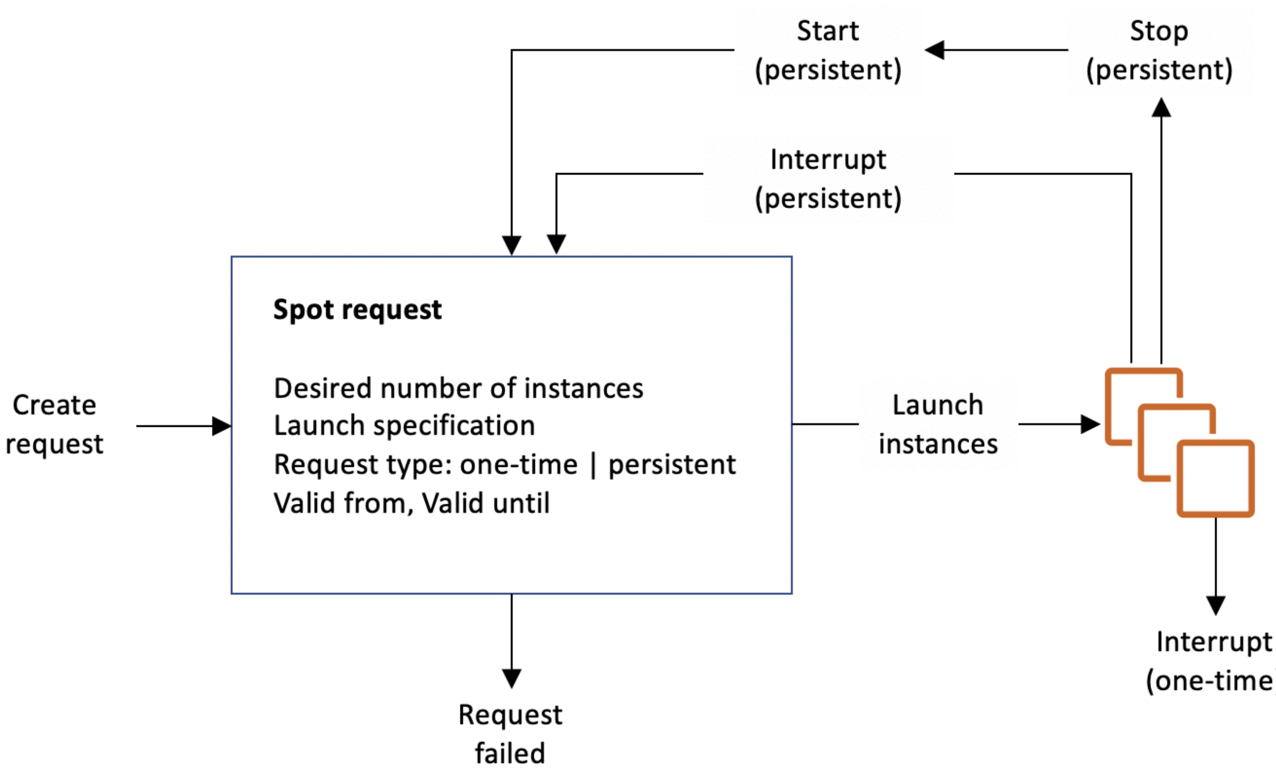
```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "c5.8xlarge",
  "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
  "Placement": {
    "Tenancy": "dedicated"
  }
}
```

Get the status of a Spot Instance request

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.

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.



Contents

Get request status information

You can get status information for your Spot Instance request.

Console

To get request status information

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests** and select the Spot request.
3. To check the status, on the **Description** tab, check the **Status** field.

AWS CLI

To get request status information

Use the following describe-spot-instance-requests command.

```
aws ec2 describe-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

To get request status information

Use the Get-EC2SpotInstanceRequest cmdlet.

```
Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

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, see Quotas for Amazon EBS in the *Amazon EBS User Guide*.

`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 Spot Instance Request Fulfillment event

When a Spot Instance request is fulfilled, Amazon EC2 sends an EC2 Spot Instance Request Fulfillment event to Amazon EventBridge. You can create a rule to take an action whenever this event occurs, such as invoking a Lambda function or notifying an Amazon SNS topic.

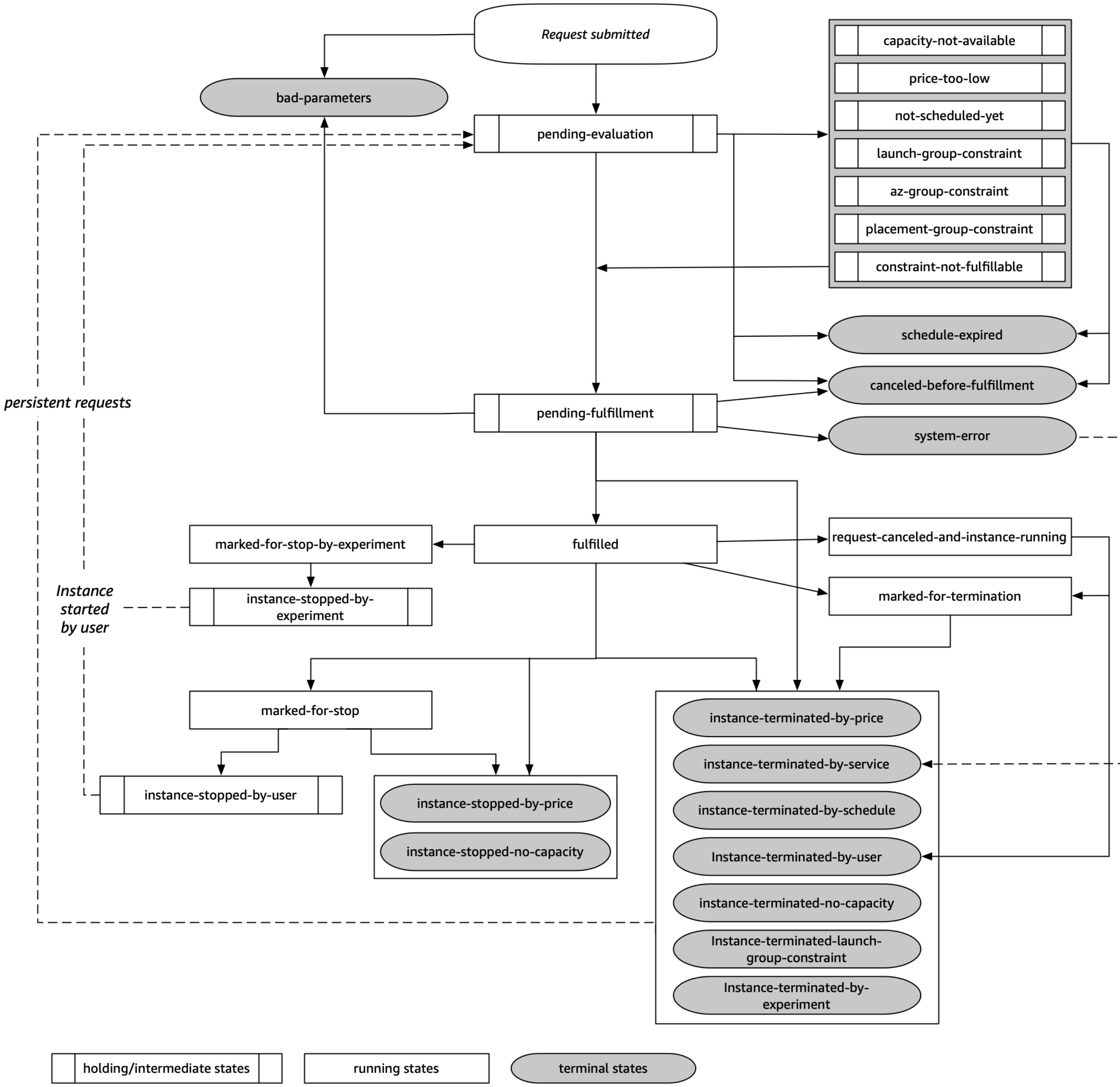
The following is example data for this event.


```
{
    "version": "0",
    "id": "01234567-1234-0123-1234-012345678901",
    "detail-type": "EC2 Spot Instance Request Fulfillment",
    "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": {
        "spot-instance-request-id": "sir-0e54a519c9EXAMPLE",
        "instance-id": "i-1234567890abcdef0"
    }
}
```


For more information, see the [Amazon EventBridge User Guide].

State changes for a Spot request

The following diagram shows you the paths that your Spot request can follow throughout its lifecycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot Instance.



Pending evaluation

As soon as you create a Spot Instance request, it goes into the `pending-evaluation` state unless one or more request parameters are not valid (`bad-parameters`).

Status code	Request state	Instance state
`pending-evaluation`	`open`	Not applicable
`bad-parameters`	`closed`	Not applicable

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 there is no capacity, your request stays in a holding state until there is available capacity. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

In the event of an outage of one of the Availability Zones, there is a chance that the spare EC2 capacity available for Spot Instance requests in other Availability Zones can be affected.

Status code	Request state	Instance state
`capacity-not-available`	`open`	Not applicable
`price-too-low`	`open`	Not applicable
`not-scheduled-yet`	`open`	Not applicable
`launch-group-constraint`	`open`	Not applicable
`az-group-constraint`	`open`	Not applicable
`placement-group-constraint`	`open`	Not applicable
`constraint-not-fulfillable`	`open`	Not applicable

Pending evaluation/fulfillment-terminal

Your Spot Instance request can go to a `terminal` state if you create a request that is valid only during a specific time period and this time period expires before your request reaches the pending fulfillment phase. It might also happen if you cancel the request, or if a system error occurs.

Status code	Request state	Instance state
`schedule-expired`	`cancelled`	Not applicable
`canceled-before-fulfillment` [1]	`cancelled`	Not applicable
`bad-parameters`	`failed`	Not applicable
`system-error`	`closed`	Not applicable

[1] If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met, 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.

Status code	Request state	Instance state
`pending-fulfillment`	`open`	Not applicable

Fulfilled

When all the specifications for your Spot Instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot Instances, which can take a few minutes. If a Spot Instance is hibernated or stopped when interrupted, it remains in this state until the request can be fulfilled again or the request is canceled.

Status code	Request state	Instance state
`fulfilled`	`active`	`pending` → `running`
`fulfilled`	`active`	`stopped` → `running`

If you stop a Spot Instance, your Spot request goes into the `marked-for-stop` or `instance-stopped-by-user` state until the Spot Instance can be started again or the request is cancelled.

Status code	Request state	Instance state
`marked-for-stop`	`active`	`stopping`
`instance-stopped-by-user` [1]	`disabled` or `cancelled` [2]	`stopped`

[1] 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.

[2] 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 there is available capacity for your instance type, and you don't terminate the instance. If Amazon EC2 must terminate your Spot Instances, the Spot request goes into a terminal state. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot Instances.

Status code	Request state	Instance state
request-canceled-and-instance-running	cancelled	running
marked-for-stop	active	running
marked-for-termination	active	running
instance-stopped-by-price	disabled	stopped
instance-stopped-by-user	disabled	stopped
instance-stopped-no-capacity	disabled	stopped
instance-terminated-by-price	closed (one-time), open (persistent)	terminated
instance-terminated-by-schedule	closed	terminated
instance-terminated-by-service	cancelled	terminated
instance-terminated-by-user	closed or cancelled [1]	terminated
instance-terminated-no-capacity	closed (one-time), open (persistent)	running †
instance-terminated-no-capacity	closed (one-time), open (persistent)	terminated

Status code	Request state	Instance state
`instance-terminated-launch-group-constraint`	`closed` (one-time), `open` (persistent)	`terminated`

[1] 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`.

† When Amazon EC2 interrupts a Spot Instance if it needs the capacity back *and* the instance is configured to *terminate* on interruption, the status is immediately set to `instance-terminated-no-capacity` (it is not set to `marked-for-termination`). However, the instance remains in the `running` state for 2 minutes to reflect the 2-minute period when the instance receives the Spot Instance interruption notice. After 2 minutes, the instance state is set to `terminated`.

Interruption experiments

You can use AWS Fault Injection Service to initiate a Spot Instance interruption so that you can test how the applications on your Spot Instances respond. If AWS FIS stops a Spot Instance, your Spot request enters the `marked-for-stop-by-experiment` state and then the `instance-stopped-by-experiment` state. If AWS FIS terminates a Spot Instance, your Spot request enters the `instance-terminated-by-experiment` state. For more information, see the section called “Initiate an interruption”.

Status code	Request state	Instance state
`marked-for-stop-by-experiment`	`active`	`running`
`instance-stopped-by-experiment`	`disabled`	`stopped`
`instance-terminated-by-experiment`	`closed`	`terminated`

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.

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.

Contents

Prerequisites

Grant the user the permission to tag resources. For more information about IAM policies and example policies, see Example: Tag resources.

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 a 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 a user the permission to tag resources when using request-spot-instances.

To grant a 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 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 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"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagSpotInstanceRequests",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
```

```
    ]
}
```

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 it 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.

To grant a 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 user permission to create a Spot Instance request.
- The `ec2:CreateTags` action. This grants the user permission to create tags.

- For `Resource`, specify `spot-instances-request`. This allows users to tag only the Spot Instance request.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotInstanceRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:RequestSpotInstances",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-instances-
request/*"
        }
    ]
}
```

Tag a new Spot Instance request

In the AWS CLI and PowerShell examples, configure the Spot Instance request as follows:

- 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.

Console

To tag a new Spot Instance request

1. Follow the Manage your Spot Instances 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 Manage your Spot Instances.

AWS CLI

To tag a new Spot Instance request

Use the request-spot-instances command with the `--tag-specification` option.

The tag specification adds two tags to the Spot Instance request: `Environment=Production` and `Cost-Center=123`.

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json \
    --tag-specification 'ResourceType=spot-instances-
request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]'
```

PowerShell

To tag a new Spot Instance request

Use the Request-EC2SpotInstance cmdlet with the `-TagSpecification` parameter.

```
-TagSpecification $tagspec
```

The tag specification is defined as follows. It adds two tags to the Spot Instance request: `Environment=Production` and `Cost-Center=123`.

```
$tag1 = @{Key="Environment"; Value="Production"}
$tag2 = @{Key="Cost-Center"; Value="123"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "spot-instances-request"
$tagspec.Tags = @($tag1,$tag2)
```

Tag an existing Spot Instance request

Console

To tag an existing Spot Instance request

After you have created a Spot Instance request, you can add tags to the Spot Instance request using the console.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Instance request.
4. 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 tags using the console.

AWS CLI

To tag an existing Spot Instance request or Spot Instance

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 `purpose=test`.

```
aws ec2 create-tags \
    --resources sir-0e54a519c9EXAMPLE i-1234567890abcdef0 \
    --tags Key=purpose,Value=test
```

PowerShell

To tag an existing Spot Instance request or Spot Instance

Use the New-EC2Tag cmdlet. The following example adds the tag `purpose=test` to the existing Spot Instance request and the Spot Instance.

```
New-EC2Tag `
    -Resource sir-0e54a519c9EXAMPLE, i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="test"}
```

View Spot Instance request tags

Console

To view Spot Instance request tags

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 Instance request and choose the **Tags** tab.

AWS CLI

To describe Spot Instance request tags

Use the describe-spot-instance-requests command to view the configuration of the specified Spot Instance request, which includes any tags that were specified for the request.

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].Tags"
```

The following is example output.

```
[
    [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Department",
            "Value": "101"
        }
    ]
]
```

PowerShell

To describe Spot Instance request tags

Use the Get-EC2SpotInstanceRequest cmdlet.

```
(Get-EC2SpotInstanceRequest `
    -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).Tags
```

The following is example output.

```
Key         Value
---         -----
Environment Production
Department  101
```

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.

Console

To cancel a Spot Instance request

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Select the Spot Instance request.
4. Choose **Actions**, **Cancel request**.
5. (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**.

AWS CLI

To cancel a Spot Instance request

Use the following cancel-spot-instance-requests command.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

To cancel a Spot Instance request

Use the Stop-EC2SpotInstanceRequest cmdlet.

```
Stop-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

Manage your Spot Instances

Amazon EC2 launches a Spot Instance when capacity is available. A Spot Instance runs until it is interrupted or you terminate it yourself.

Contents

Find your Spot Instances

A Spot Instance appears in the **Instances** page in the console, along with On-Demand Instances. Use the following procedure to find your Spot Instances.

Console

To find your Spot Instances

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

2. In the navigation pane, choose **Instances**.
3. To find all Spot Instances, in the search pane, choose **Instance lifecycle=spot**.
4. To verify that an instance is a Spot Instance, select the instance, choose the **Details** tab, and check the value of **Lifecycle**. The value for a Spot Instance is `spot` and the value for an On-Demand Instance is `normal`.

AWS CLI

To find your Spot Instances

Use the following describe-instances command.

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

To determine whether an instance is a Spot Instance

Use the following describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[*].InstanceLifecycle" \
    --output text
```

If the output is `spot`, the instance is a Spot Instance. If there is no output, the instance is an On-Demand Instance.

PowerShell

To find your Spot Instances

Use the Get-EC2Instance cmdlet.

```
Get-EC2Instance -Filter @{Name="instance-lifecycle"; Values="spot"}
```

To determine whether an instance is a Spot Instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.InstanceLifecycle
```

If the output is `Spot`, the instance is a Spot Instance. If there is no output, the instance is an On-Demand Instance.

Find instances launched by a specific request

Use the following procedure to find the Spot Instances launched from a specific Spot Instance or Spot Fleet request.

Console

To find the Spot Instances for a request

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**. The list contains both Spot Instance requests and Spot Fleet requests.
3. If a Spot Instance request is 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**.
4. For a Spot Fleet, **Capacity** indicates how much of the requested capacity is fulfilled. To view the IDs of the instances in a Spot Fleet, choose the fleet ID to open its details page and locate the **Instances** pane.

AWS CLI

To find the Spot Instances for a request

Use the following describe-spot-instance-requests command.

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

The following is example output:

```
[
    {
        "ID": "i-1234567890abcdef0"
```

```
    },
    {
        "ID": "i-0598c7d356eba48d7"
    }
]
```

PowerShell

To find the Spot Instances for a request

Use the Get-EC2SpotInstanceRequest cmdlet.

```
(Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).InstanceId
```

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.

Console

To stop a Spot Instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the Spot Instance. If you didn't save the instance ID of the Spot Instance, see the section called "Find your Spot Instances".
4. Choose **Instance state**, **Stop instance**.
5. When prompted for confirmation, choose **Stop**.

AWS CLI

To stop a Spot Instance

Use the stop-instances command to manually stop your Spot Instances.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To stop a Spot Instance

Use the Stop-EC2Instance cmdlet.

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

Start a Spot Instance

You can start a Spot Instance that you previously stopped.

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.

The steps for starting a Spot Instance are similar to the steps for starting an On-Demand Instance.

Console

To start a Spot Instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the Spot Instance. If you didn't save the instance ID of the Spot Instance, see the section called “Find your Spot Instances”.
4. Choose **Instance state**, **Start instance**.

AWS CLI

To start a Spot Instance

Use the start-instances command to manually start your Spot Instances.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To start a Spot Instance

Use the Start-EC2Instance cmdlet.

```
Start-EC2Instance -InstanceId 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 more information, see Cancel a Spot Instance request.

Console

To manually terminate a Spot Instance

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the navigation pane, choose **Instances**.
4. Select the Spot Instance. If you didn't save the instance ID of the Spot Instance, see the section called "Find your Spot Instances".
5. Choose **Instance state**, **Terminate (delete) instance**.
6. Choose **Terminate (delete)** when prompted for confirmation.

AWS CLI

To manually terminate a Spot Instance

Use the terminate-instances command to manually terminate your Spot Instances.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

PowerShell

To manually terminate a Spot Instance

Use the Remove-EC2Instance cmdlet.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

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. The following are the possible reasons that Amazon EC2 might interrupt your Spot Instances:

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.

Price

The Spot price is higher than your maximum price.

You can specify the maximum price in your Spot request. However, if you specify a maximum price, your instances will be interrupted more frequently than if you do not specify it.

Constraints

If your Spot request includes a constraint such as a launch group or an Availability Zone group, the Spot Instances are terminated as a group when the constraint can no longer be met.

When Amazon EC2 interrupts a Spot Instance, it either terminates, stops, or hibernates the instance, depending on the interruption behavior that you specified when you created the Spot request.

Contents

Behavior of Spot Instance interruptions

When you create a Spot request, you can specify the interruption behavior. The following are the possible interruption behaviors:

- Stop
- Hibernate
- Terminate

The default behavior is that Amazon EC2 terminates Spot Instances when they are interrupted.

Stop interrupted Spot Instances

You can specify that Amazon EC2 stops your Spot Instances when they are interrupted. The Spot Instance request type must be `persistent`. You can't specify a launch group in the Spot Instance request. For EC2 Fleet or Spot Fleet, the request type must be `maintain`.

Considerations

- Only Amazon EC2 can restart an interrupted stopped Spot Instance.
- For a Spot Instance launched by a `persistent` Spot Instance request: Amazon EC2 restarts the stopped instance when capacity is available in the same Availability Zone and for the same instance type as the stopped instance (the same launch specification must be used).
- 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 Amazon EC2 attempts to start the Spot Instance, the instance will fail to start and Amazon EC2 will terminate 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, Amazon EC2 terminates any associated Spot Instances that are stopped.
- While an interrupted 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. For more information about how you're charged when a Spot Instance is interrupted, see Billing for interrupted Spot Instances.
- Make sure that you are familiar with the implications of stopping an instance. For information about what happens when an instance is stopped, see Differences between instance states.

Hibernate interrupted Spot Instances

You can specify that Amazon EC2 hibernates your Spot Instances when they are interrupted. For more information, see Hibernate your Amazon EC2 instance.

Amazon EC2 now offers the same hibernation experience for Spot Instances as is currently available for On-Demand Instances. It offers more extensive support, where the following is now supported for Spot Instance hibernation:

- More supported AMIs
- More supported instance families
- User-initiated hibernation

Terminate interrupted Spot Instances

When Amazon EC2 interrupts a Spot Instance, it terminates the instance by default, unless you specify a different interruption behavior, such as stop or hibernate. For more information, see Terminate Amazon EC2 instances.

Prepare for Spot Instance interruptions

Demand for Spot Instances can vary significantly from moment to moment, and the availability of Spot Instances can also vary significantly depending on how many unused EC2 instances are available. It is always possible that your Spot Instance might be interrupted. Therefore, you must ensure that your application is prepared for a Spot Instance interruption.

We recommend that you follow these best practices so that you're prepared for a Spot Instance interruption.

- Create your Spot request using an Auto Scaling group. If your Spot Instances are interrupted, the Auto Scaling group will automatically launch replacement instances. For more information, see Auto Scaling groups with multiple instance types and purchase options in the *Amazon EC2 Auto Scaling User Guide*.
- 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 startup.
- Data on instance store volumes is lost when the instance is stopped or terminated. Back up any important data on instance store volumes to a more persistent storage, such as Amazon S3, Amazon EBS, or Amazon DynamoDB.
- Store important data regularly in a place that isn't affected if 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.
- Use the two-minute Spot Instance interruption notices to monitor the status of your Spot Instances. For more information, see Spot Instance interruption notices.
- 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 this by running the application using an On-Demand Instance and then terminating the On-Demand Instance yourself.
- Run a controlled fault injection experiment with AWS Fault Injection Service to test how your application responds when your Spot Instance is interrupted. For more information, see the Tutorial: Test Spot Instance interruptions using AWS FIS in the *AWS Fault Injection Service User Guide*.

Initiate a Spot Instance interruption

You can select a Spot Instance request or a Spot Fleet request in the Amazon EC2 console and initiate a Spot Instance interruption so that you can test how the applications on your Spot

Instances handle being interrupted. When you initiate a Spot Instance interruption, Amazon EC2 notifies you that your Spot Instance will be interrupted in two minutes, and then, after two minutes, the instance is interrupted.

The underlying service that performs the Spot Instance interruption is AWS Fault Injection Service (AWS FIS). For information about AWS FIS, see AWS Fault Injection Service.

> **Note**
>
> Interruption behaviors are `terminate`, `stop`, and `hibernate`. If you set the interruption behavior to `hibernate`, when you initiate a Spot Instance interruption, the hibernation process will begin immediately.

Initiating a Spot Instance interruption is supported in all AWS Regions except Asia Pacific (Jakarta), Asia Pacific (Osaka), China (Beijing), China (Ningxia), and Middle East (UAE).

Contents

Initiate a Spot Instance interruption

You can use the EC2 console to quickly initiate a Spot Instance interruption. When you select a Spot Instance request, you can initiate the interruption of one Spot Instance. When you select a Spot Fleet request, you can initiate the interruption of multiple Spot Instances at once.

For more advanced experiments to test Spot Instance interruptions, you can create your own experiments using the AWS FIS console.

To initiate the interruption of one Spot Instance in a Spot Instance request using the EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation pane, choose **Spot Requests**.
3. Select a Spot Instance request, and then choose **Actions**, **Initiate interruption**. You can't select multiple Spot Instance requests to initiate an interruption.

4. In the **Initiate Spot Instance interruption** dialog box, under **Service access**, either use the default role, or choose an existing role. To choose an existing role, choose **Use an existing service role**, and then, for **IAM role**, select the role to use.
5. When you're ready to initiate the Spot Instance interruption, choose **Initiate interruption.**

To initiate the interruption of one or more Spot Instances in a Spot Fleet request using the EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation pane, choose **Spot Requests**.
3. Select a Spot Fleet request, and then choose **Actions**, **Initiate interruption**. You can't select multiple Spot Fleet requests to initiate an interruption.
4. In the **Specify number of Spot Instances** dialog box, for **Number of instances to interrupt**, enter the number of Spot Instances to interrupt, and then choose **Confirm**.

   > **Note**
   >
   > The number can't exceed the number of Spot Instances in the fleet or your quota for the number of Spot Instances that AWS FIS can interrupt per experiment.

5. In the **Initiate Spot Instance interruption** dialog box, under **Service access**, either use the default role, or choose an existing role. To choose an existing role, choose **Use an existing service role**, and then, for **IAM role**, select the role to use.
6. When you're ready to initiate the Spot Instance interruption, choose **Initiate interruption.**

To create more advanced experiments to test Spot Instance interruptions using the AWS FIS console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation pane, choose **Spot Requests**.
3. Choose **Actions**, **Create advanced experiments**.

   The AWS FIS console opens. For more information, see Tutorial: Test Spot Instance interruptions using AWS FIS in the *AWS Fault Injection Service User Guide*.

Verify the Spot Instance interruption

After you initiate the interruption, the following occurs:

- The Spot Instance receives an instance rebalance recommendation.
- A Spot Instance interruption notice is issued two minutes before AWS FIS interrupts your instance.
- After two minutes, the Spot Instance is interrupted.
- A Spot Instance that was stopped by AWS FIS remains stopped until you restart it.

To verify that the instance was interrupted after you initiated the interruption

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation pane, open **Spot Requests** and **Instances** in separate browser tabs or windows.
3. For **Spot Requests**, select the Spot Instance request or Spot Fleet request. The initial status is `fulfilled`. After the instance is interrupted, the status changes as follows, depending on the interruption behavior:
   - `terminate` – The status changes to `instance-terminated-by-experiment`.
   - `stop` – The status changes to `marked-for-stop-by-experiment` and then `instance-stopped-by-experiment`.
4. For **Instances**, select the Spot Instance. The initial status is `Running`. Two minutes after you receive the Spot Instance interruption notice, the status changes as follows, depending on the interruption behavior:
   - `stop` – The status changes to `Stopping` and then `Stopped`.
   - `terminate` – The status changes to `Shutting-down` and then `Terminated`.

Quotas

Your AWS account has the following default quota for the number of Spot Instances that AWS FIS can interrupt per experiment.

Name	Default	Adjustable	Description
		Yes	

Name	Default	Adjustable	Description
Target SpotInstances for aws:ec2:send-spot-instance-interruptions	Each supported Region: 5		The maximum number of Spot Instances that aws:ec2:send-spot-instance-interruptions can target when you identify targets using tags, per experiment.

You can request a quota increase. For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

To view all the quotas for AWS FIS, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **AWS Fault Injection Service**. You can also view all the quotas for AWS Fault Injection Service in the *AWS Fault Injection Service User Guide*.

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.

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. We recommend that you check for these interruption notices every 5 seconds.

The interruption notices are made available as an EventBridge event and as items in the instance metadata on the Spot Instance. Interruption notices are emitted on a best effort basis.

EC2 Spot Instance Interruption Warning event

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 EventBridge. For more information about EventBridge events, see the Amazon

EventBridge 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`, or `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-2a:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

> **Note**
>
> The ARN format of the Spot Instance interruption event is `arn:aws:ec2:`*`availability-zone`*`:instance/`*`instance-id`*. This format differs from the EC2 resource ARN format.

instance-action

The `instance-action` item specifies the action and the approximate time, in UTC, when the action will occur.

If your Spot Instance is marked to be stopped or terminated by Amazon EC2, the `instance-action` item is present in your instance metadata. Otherwise, it is not present. You can retrieve the `instance-action` using Instance Metadata Service Version 2 (IMDSv2) as follows.

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/spot/instance-action
```

Windows

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/instance-action
```

The following example output indicates the time at which this instance will be stopped.

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

The following example output indicates the time at which this instance will be terminated.

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

If Amazon EC2 is not preparing to stop or terminate the instance, or if you terminated the instance yourself, `instance-action` is not present in the instance metadata and you receive an HTTP 404 error when you try to retrieve it.

termination-time

The `termination-time` item specifies the approximate time in UTC when the instance will receive the shutdown signal.

> **Note**
>
> This item is maintained for backward compatibility; you should use `instance-action` instead.

If your Spot Instance is marked for termination by Amazon EC2 (either due to a Spot Instance interruption where the interruption behavior is set to `terminate`, or due to the cancellation of a persistent Spot Instance request), the `termination-time` item is present in your instance metadata. Otherwise, it is not present. You can retrieve the `termination-time` using IMDSv2 as follows.

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo termination_scheduled; fi
```

Windows

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/termination-time
```

The following is example output.

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance (either because there is no Spot Instance interruption or because your interruption behavior is set to `stop` or `hibernate`), or if you terminated the Spot Instance yourself, the `termination-time` item is either not present in the instance metadata (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

When you describe your EC2 instances, the results include your Spot Instances. The instance lifecycle of a Spot Instance is `spot`. The instance state of a Spot Instance is either `stopped` or `terminated`, depending on the interruption behavior that you configured. For a hibernated Spot Instance, the instance state is `stopped`.

For additional details about the reason for the interruption, check the Spot request status code. For more information, see the section called “Get the status of a Spot Instance request”.

Console

To find an interrupted Spot Instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Apply the following filter: **Instance lifecycle=spot**.
4. Apply the **Instance state=stopped** or **Instance state=terminated** filter depending on the interruption behavior that you configured.
5. For each Spot Instance, on the **Details** tab, under **Instance details**, find **State transition message**. The following codes indicate that the Spot Instance was interrupted.
   - `Server.SpotInstanceShutdown`
   - `Server.SpotInstanceTermination`

AWS CLI

To find interrupted Spot Instances

Use the describe-instances command with the `--filters` option. To list only the instance IDs in the output, include the `--query` option.

If the interruption behavior is to terminate the Spot Instances, use the following example:

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=terminated \
              Name=state-reason-code,Values=Server.SpotInstanceTermination \
    --query "Reservations[*].Instances[*].InstanceId"
```

If the interruption behavior is to stop the Spot Instances, use the following example:

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=stopped \
              Name=state-reason-code,Values=Server.SpotInstanceShutdown \
    --query "Reservations[*].Instances[*].InstanceId"
```

PowerShell

To find interrupted Spot Instances

Use the Get-EC2Instance cmdlet.

If the interruption behavior is to terminate the Spot Instances, use the following example:

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="terminated"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

If the interruption behavior is to stop the Spot Instances, use the following example:

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="stopped"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

Determine whether Amazon EC2 terminated a Spot Instance

A Spot Instance runs until Amazon EC2 terminates it in response to a Spot Instance interruption, or until you terminate it yourself. For more information, see the section called “Interruption behavior”.

After a Spot Instance is terminated, you can use AWS CloudTrail to see whether Amazon EC2 terminated it. If the CloudTrail log includes a `BidEvictedEvent`, this indicates that Amazon EC2 terminated the Spot Instance. If instead you see a `TerminateInstances` event, this indicates that a user terminated the Spot Instance.

Alternatively, if you want to receive notification that Amazon EC2 is going to interrupt your Spot Instance, use Amazon EventBridge to respond to the EC2 Spot Instance Interruption Warning event.

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. From the list of filters, choose **Event name**, and then in the filter field to the right, enter **BidEvictedEvent**.
4. (Optional) Select a time range.
5. If the list is not empty, choose **BidEvictedEvent** from the resulting entry to open its details page. You can find information about the Spot Instance in the **Event record** pane, including the ID of the Spot Instance. The following is an example of the event record.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "ec2.amazonaws.com"
    },
    "eventTime": "2016-08-16T22:30:00Z",
    "eventSource": "ec2.amazonaws.com",
    "userAgent": "ec2.amazonaws.com",
    "sourceIPAddress": "ec2.amazonaws.com",
    "eventName": "BidEvictedEvent",
    "awsRegion": "us-east-2",
    "eventID": "d27a6096-807b-4bd0-8c20-a33a83375054",
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123456789012",
    "RequestParameters": null,
    "ResponseElements": null,
    "serviceEventDetails": {
        "instanceIdSet": [
          "i-1eb2ac8eEXAMPLE"
        ]
    }
}
```

6. If you did not find an entry for the `BidEvictedEvent` event, enter **TerminateInstances** as the event name. For more information about the event record for `TerminateInstances`, see the section called "Amazon EC2 API event examples".

Billing for interrupted Spot Instances

When a Spot Instance is interrupted, you're charged for instance and EBS volume usage, and you might incur other charges, as follows.

Instance usage

Who interrupts the Spot Instance	Operating system	Interrupted in the first hour	Interrupted in any hour after the first hour
If **you** stop or terminate the Spot Instance	Windows and Linux (excluding SUSE)	Charged for the seconds used	Charged for the seconds used
	SUSE	Charged for the full hour even if you used a partial hour	Charged for the full hours used, and charged a full hour for the interrupted partial hour
If the **Amazon EC2** interrupts the Spot Instance	Windows and Linux (excluding SUSE)	No charge	Charged for the seconds used
	SUSE	No charge	Charged for the full hours used, but no charge for the inter rupted partial hour

EBS volume usage

While an interrupted 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.

EC2 instance rebalance recommendations

An EC2 instance *rebalance recommendation* is a 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, 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.

Rebalance recommendations are made available as a EventBridge event and as an item in the instance metadata on the Spot Instance. Events are emitted on a best effort basis.

> **Note**
>
> Rebalance recommendations are only supported for Spot Instances that are launched after November 5, 2020 00:00 UTC.

Contents

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 Use Capacity Rebalancing to handle Amazon EC2 Spot interruptions in the *Amazon EC2 Auto Scaling User Guide*, and Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances 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
- Use instance metadata

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.

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 for a rebalance recommendation event

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Choose **Create rule**.
3. For **Define rule detail**, do the following:

   a. 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.

   b. For **Event bus**, choose **default**. When an AWS service in your account generates an event, it always goes to your account's default event bus.

   c. For **Rule type**, choose **Rule with an event pattern**.

   d. Choose **Next**.

4. For **Build event pattern**, do the following:

   a. For **Event source**, choose **AWS events or EventBridge partner events**.

   b. For **Event pattern**, for this example you'll specify the following event pattern to match the `EC2 Instance Rebalance Recommendation` event, and then choose **Save**.

```
{
 "source": ["aws.ec2"],
 "detail-type": ["EC2 Instance Rebalance Recommendation"]
}
```

   To add the event pattern, you can either use a template by choosing **Event pattern form**, or specify your own pattern by choosing **Custom pattern (JSON editor)**, as follows:

      i. To use a template to create the event pattern, do the following:

         A. Choose **Event pattern form**.

         B. For **Event source**, choose **AWS services**.

         C. For **AWS Service**, choose **EC2 Spot Fleet**.

         D. For **Event type**, choose **EC2 Instance Rebalance Recommendation**.

         E. To customize the template, choose **Edit pattern** and make your changes to match the example event pattern.

      ii. (Alternative) To specify a custom event pattern, do the following:

         A. Choose **Custom pattern (JSON editor)**.

         B. In the **Event pattern** box, add the event pattern for this example.

   c. Choose **Next**.

5. For **Select target(s)**, do the following:

   a. For **Target types**, choose **AWS service**.

   b. For **Select a target**, choose **SNS topic** to send an email, text message, or mobile push notification when the event occurs.

   c. 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*.

   d. (Optional) Under **Additional settings**, you can optionally configure additional settings. For more information, see Creating Amazon EventBridge rules that react to events (step 16) in the *Amazon EventBridge User Guide*.

   e. Choose **Next**.

6. (Optional) For **Tags**, you can optionally assign one or more tags to your rule, and then choose **Next**.

7. For **Review and create**, do the following:

   a. Review the details of the rule and modify them as necessary.

   b. Choose **Create rule**.

For more information, see Amazon EventBridge rules and Amazon EventBridge 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.

IMDSv2

Linux

Run the following command from your Linux instance.

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

Run the following cmdlet from your Windows instance

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/events/recommendations/
rebalance
```

IMDSv1

Linux

Run the following command from your Linux instance.

```
curl http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

Run the following cmdlet from your Windows instance.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/events/
recommendations/rebalance
```

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:

- Use Capacity Rebalancing to handle Amazon EC2 Spot interruptions in the *Amazon EC2 Auto Scaling User Guide*
- Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances in the EC2 Fleet and Spot Fleet topic in this user guide

Spot placement score

The Spot placement score feature can recommend an AWS Region or Availability Zone based on your Spot capacity requirements. Spot capacity fluctuates, and you can't be sure that you'll always get the capacity that you need. A Spot placement score indicates how likely it is that a Spot request will succeed in a Region or Availability Zone.

> **Note**
>
> A Spot placement score does not provide any guarantees in terms of available capacity or risk of interruption. A Spot placement score serves only as a recommendation.

Use cases

You can use the Spot placement score feature for the following:

- To relocate and scale Spot compute capacity in a different Region, as needed, in response to increased capacity needs or decreased available capacity in the current Region.
- To identify the most optimal Availability Zone in which to run single-Availability Zone workloads.
- To simulate future Spot capacity needs so that you can pick an optimal Region for the expansion of your Spot-based workloads.
- To find an optimal combination of instance types to fulfill your Spot capacity needs.

Contents

- How Spot placement score works
- Required permissions for Spot placement score
- Calculate the Spot placement score

Limitations

- **Target capacity limit** – Your Spot placement score target capacity limit is based on your recent Spot usage, while accounting for potential usage growth. If you have no recent Spot usage, we provide you with a low default limit aligned with your Spot request limit.
- **Request configurations limit** – We can limit the number of new request configurations within a 24-hour period if we detect patterns not associated with the intended use of the Spot placement score feature. If you reach the limit, you can retry the request configurations that you've already used, but you can't specify new request configurations until the next 24-hour period.
- **Minimum number of instance types** – If you specify instance types, you must specify at least three different instance types, otherwise Amazon EC2 will return a low Spot placement score. Similarly, if you specify instance attributes, they must resolve to at least three different instance types. Instance types are considered different if they have a different name. For example, m5.8xlarge, m5a.8xlarge, and m5.12xlarge are all considered different.

Costs

There is no additional charge for using the Spot placement score feature.

How Spot placement score works

When you use the Spot placement score feature, you first specify your compute requirements for your Spot Instances, and then Amazon EC2 returns the top 10 Regions or Availability Zones where your Spot request is likely to succeed. Each Region or Availability Zone is scored on a scale from 1 to 10, with 10 indicating that your Spot request is highly likely to succeed, and 1 indicating that your Spot request is not likely to succeed.

To use the Spot placement score feature, follow these steps:

- Step 1: Specify your Spot requirements
- Step 2: Filter the Spot placement score response
- Step 3: Review the recommendations
- Step 4: Use the recommendations

Step 1: Specify your Spot requirements

First, you specify your desired target Spot capacity and your compute requirements, as follows:

1. **Specify the target Spot capacity, and optionally the target capacity unit.**

   You can specify your desired target Spot capacity in terms of the number of instances or vCPUs, or in terms of the amount of memory in MiB. To specify the target capacity in number of vCPUs or amount of memory, you must specify the target capacity unit as `vcpu` or `memory-mib`. Otherwise, it defaults to number of instances.

   By specifying your target capacity in terms of the number of vCPUs or the amount of memory, you can use these units when counting the total capacity. For example, if you want to use a mix of instances of different sizes, you can specify the target capacity as a total number of vCPUs. The Spot placement score feature then considers each instance type in the request by its number of vCPUs, and counts the total number of vCPUs rather than the total number of instances when totaling up the target capacity.

   For example, say you specify a total target capacity of 30 vCPUs, and your instance type list consists of c5.xlarge (4 vCPUs), m5.2xlarge (8 vCPUs), and r5.large (2 vCPUs). To achieve a total of 30 vCPUs, you could get a mix of 2 c5.xlarge (2*4 vCPUs), 2 m5.2xlarge (2*8 vCPUs), and 3 r5.large (3*2 vCPUs).

2. **Specify instance types or instance attributes.**

   You can either specify the instance types to use, or you can specify the instance attributes that you need for your compute requirements, and then let Amazon EC2 identify the instance types that have those attributes. This is known as attribute-based instance type selection.

   You can't specify both instance types and instance attributes in the same Spot placement score request.

   If you specify instance types, you must specify at least three different instance types, otherwise Amazon EC2 will return a low Spot placement score. Similarly, if you specify instance attributes, they must resolve to at least three different instance types.

For examples of different ways to specify your Spot requirements, see Example configurations.

Step 2: Filter the Spot placement score response

Amazon EC2 calculates the Spot placement score for each Region or Availability Zone, and returns either the top 10 Regions or the top 10 Availability Zones where your Spot request is likely to succeed. The default is to return a list of scored Regions. If you plan to launch all of your Spot capacity into a single Availability Zone, then it's useful to request a list of scored Availability Zones.

You can specify a Region filter to narrow down the Regions that will be returned in the response.

You can combine the Region filter and a request for scored Availability Zones. In this way, the scored Availability Zones are confined to the Regions for which you've filtered. To find the highest-scored Availability Zone in a Region, specify only that Region, and the response will return a scored list of all of the Availability Zones in that Region.

Step 3: Review the recommendations

The Spot placement score for each Region or Availability Zone is calculated based on the target capacity, the composition of the instance types, the historical and current Spot usage trends, and the time of the request. Because Spot capacity is constantly fluctuating, the same Spot placement score request can yield different scores when calculated at different times.

Regions and Availability Zones are scored on a scale from 1 to 10. A score of 10 indicates that your Spot request is highly likely—but not guaranteed—to succeed. A score of 1 indicates that your Spot request is not likely to succeed at all. The same score might be returned for different Regions or Availability Zones.

If low scores are returned, you can edit your compute requirements and recalculate the score. You can also request Spot placement score recommendations for the same compute requirements at different times of the day.

Step 4: Use the recommendations

A Spot placement score is only relevant if your Spot request has exactly the same configuration as the Spot placement score configuration (target capacity, target capacity unit, and instance types or instance attributes), and is configured to use the `capacity-optimized` allocation strategy. Otherwise, the likelihood of getting available Spot capacity will not align with the score.

While a Spot placement score serves as a guideline, and no score guarantees that your Spot request will be fully or partially fulfilled, you can use the following information to get the best results:

- **Use the same configuration** – The Spot placement score is relevant only if the Spot request configuration (target capacity, target capacity unit, and instance types or instance attributes) in your Auto Scaling group, EC2 Fleet, or Spot Fleet is the same as what you entered to get the Spot placement score.

  If you used attribute-based instance type selection in your Spot placement score request, you can use attribute-based instance type selection to configure your Auto Scaling group, EC2 Fleet, or Spot Fleet. For more information, see Create mixed instances group using attribute-based instance type selection and Specify attributes for instance type selection for EC2 Fleet or Spot Fleet.

  > **Note**
  >
  > If you specified your target capacity in terms of the number of vCPUs or the amount of memory, and you specified instance types in your Spot placement score configuration, note that you can’t currently create this configuration in your Auto Scaling group, EC2 Fleet, or Spot Fleet. Instead, you must manually set the instance weighting by using the `WeightedCapacity` parameter.

- **Use the `capacity-optimized` allocation strategy** – Any score assumes that your fleet request will be configured to use all Availability Zones (for requesting capacity across Regions) or a single Availability Zone (if requesting capacity in one Availability Zone) and the `capacity-optimized` Spot allocation strategy for your request for Spot capacity to succeed. If you use other allocation strategies, such as `lowest-price`, the likelihood of getting available Spot capacity will not align with the score.
- **Act on a score immediately** – The Spot placement score recommendation reflects the available Spot capacity at the time of the request, and the same configuration can yield different scores when calculated at different times due to Spot capacity fluctuations. While a score of 10 means that your Spot capacity request is highly likely—but not guaranteed—to succeed, for best results we recommend that you act on a score immediately. We also recommend that you get a fresh score each time you attempt a capacity request.

Required permissions for Spot placement score

By default, IAM identities (users, roles, or groups) don't have permission to use the section called “Spot placement score”. To allow IAM identities to use Spot placement score, you must create an

IAM policy that grants permission to use the `ec2:GetSpotPlacementScores` EC2 API action. You then attach the policy to the IAM identities that require this permission.

The following is an example IAM policy that grants permission to use the `ec2:GetSpotPlacementScores` EC2 API action.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:GetSpotPlacementScores",
            "Resource": "*"
        }
    ]
}
```

For information about editing an IAM policy, see Editing IAM policies in the *IAM User Guide*.

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

Calculate the Spot placement score

You can calculate a Spot placement score based on target capacity and compute requirements. For more information, see the section called “How Spot placement score works”.

Required permissions

Ensure that you have the required permissions. For more information, see the section called “Required permissions”.

Options

- Calculate using instance attributes
- Calculate using instance types
- Calculate using the AWS CLI

Looking for an automated solution? Instead of following the manual steps in this user guide, you can build a Spot placement score tracker dashboard that automatically captures and stores the scores in Amazon CloudWatch. For more information, see Guidance for Building a Spot Placement Score Tracker Dashboard on AWS.

Calculate using instance attributes

To calculate a Spot placement score by specifying instance attributes

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Choose the down arrow next to **Request Spot Instances** and choose **Calculate Spot Placement Score**.
4. Choose **Enter requirements**.
5. For **Target capacity**, enter your desired capacity in terms of the number of **instances** or **vCPUs**, or the amount of **memory (MiB)**.
6. For **Instance type requirements**, to specify your compute requirements and let Amazon EC2 identify the optimal instance types with these requirements, choose **Specify instance attributes that match your compute requirements**.
7. For **vCPUs**, enter the desired minimum and maximum number of vCPUs. To specify no limit, select **No minimum**, **No maximum**, or both.

8. For **Memory (GiB)**, enter the desired minimum and maximum amount of memory. To specify no limit, select **No minimum**, **No maximum**, or both.
9. For **CPU architecture**, select the required instance architecture.
10. (Optional) For **Additional instance attributes**, you can optionally specify one or more attributes to express your compute requirements in more detail. Each additional attribute adds a further constraint to your request. You can omit the additional attributes; when omitted, the default values are used. For a description of each attribute and their default values, see get-spot-placement-scores.
11. (Optional) To view the instance types with your specified attributes, expand **Preview matching instance types**. To exclude instance types from being used in the placement evaluation, select the instances and then choose **Exclude selected instance types**.
12. Choose **Load placement scores**, and review the results.
13. (Optional) To display the Spot placement score for specific Regions, for **Regions to evaluate**, select the Regions to evaluate, and then choose **Calculate placement scores**.
14. (Optional) To display the Spot placement score for the Availability Zones in the displayed Regions, select the **Provide placement scores per Availability Zone** checkbox. A list of scored Availability Zones is useful if you want to launch all of your Spot capacity into a single Availability Zone.
15. (Optional) To edit your compute requirements and get a new placement score, choose **Edit**, make the necessary adjustments, and then choose **Calculate placement scores**.

Calculate using instance types

To calculate a Spot placement score by specifying instance types

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Choose the down arrow next to **Request Spot Instances** and choose **Calculate Spot Placement Score**.
4. Choose **Enter requirements**.
5. For **Target capacity**, enter your desired capacity in terms of the number of **instances** or **vCPUs**, or the amount of **memory (MiB)**.
6. For **Instance type requirements**, to specify the instance types to use, choose **Manually select instance types**.

7. Choose **Select instance types**, select the instance types to use, and then choose **Select**. To quickly find instance types, you can use the filter bar to filter the instance types by different properties.
8. Choose **Load placement scores**, and review the results.
9. (Optional) To display the Spot placement score for specific Regions, for **Regions to evaluate**, select the Regions to evaluate, and then choose **Calculate placement scores**.
10. (Optional) To display the Spot placement score for the Availability Zones in the displayed Regions, select the **Provide placement scores per Availability Zone** checkbox. A list of scored Availability Zones is useful if you want to launch all of your Spot capacity into a single Availability Zone.
11. (Optional) To edit the list of instance types and get a new placement score, choose **Edit**, make the necessary adjustments, and then choose **Calculate placement scores**.

Calculate using the AWS CLI

To calculate the Spot placement score

1. (Optional) To generate all of the possible parameters that can be specified for the Spot placement score configuration, use the get-spot-placement-scores command and the `--generate-cli-skeleton` parameter.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --generate-cli-skeleton
```

The following is example output.

```
{
    "InstanceTypes": [
        ""
    ],
    "TargetCapacity": 0,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "RegionNames": [
        ""
    ],
    "InstanceRequirementsWithMetadata": {
```

```
        "ArchitectureTypes": [
            "x86_64_mac"
        ],
        "VirtualizationTypes": [
            "hvm"
        ],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 0,
                "Max": 0
            },
            "MemoryMiB": {
                "Min": 0,
                "Max": 0
            },
            "CpuManufacturers": [
                "amd"
            ],
            "MemoryGiBPerVCpu": {
                "Min": 0.0,
                "Max": 0.0
            },
            "ExcludedInstanceTypes": [
                ""
            ],
            "InstanceGenerations": [
                "previous"
            ],
            "SpotMaxPricePercentageOverLowestPrice": 0,
            "OnDemandMaxPricePercentageOverLowestPrice": 0,
            "BareMetal": "excluded",
            "BurstablePerformance": "excluded",
            "RequireHibernateSupport": true,
            "NetworkInterfaceCount": {
                "Min": 0,
                "Max": 0
            },
            "LocalStorage": "included",
            "LocalStorageTypes": [
                "hdd"
            ],
            "TotalLocalStorageGB": {
                "Min": 0.0,
                "Max": 0.0
```

```
            },
            "BaselineEbsBandwidthMbps": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorTypes": [
                "fpga"
            ],
            "AcceleratorCount": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorManufacturers": [
                "amd"
            ],
            "AcceleratorNames": [
                "vu9p"
            ],
            "AcceleratorTotalMemoryMiB": {
                "Min": 0,
                "Max": 0
            }
        }
    },
    "DryRun": true,
    "MaxResults": 0,
    "NextToken": ""
}
```

2. Create a JSON configuration file using the output from the previous step, and configure it as follows:

    a. For `TargetCapacity`, enter your desired Spot capacity in terms of the number of instances or vCPUs, or the amount of memory (MiB).

    b. For `TargetCapacityUnitType`, enter the unit for the target capacity. If you omit this parameter, it defaults to `units`.

       Valid values: `units` (which translates to number of instances) | `vcpu` | `memory-mib`

    c. For `SingleAvailabilityZone`, specify `true` for a response that returns a list of scored Availability Zones. A list of scored Availability Zones is useful if you want to launch all of your Spot capacity into a single Availability Zone. If you omit this parameter, it defaults to `false`, and the response returns a list of scored Regions.

d. (Optional) For `RegionNames`, specify the Regions to use as a filter. You must specify the Region code, for example, `us-east-1`.

   With a Region filter, the response returns only the Regions that you specify. If you specified `true` for `SingleAvailabilityZone`, the response returns only the Availability Zones in the specified Regions.

e. You can include either `InstanceTypes` or `InstanceRequirements`, but not both in the same configuration.

   Specify one of the following in your JSON configuration:

   - To specify a list of instance types, specify the instance types in the `InstanceTypes` parameter. Specify at least three different instance types. If you specify only one or two instance types, Spot placement score returns a low score. For the list of instance types, see Amazon EC2 Instance Types.
   - To specify the instance attributes so that Amazon EC2 will identify the instance types that match those attributes, specify the attributes that are located in the `InstanceRequirements` structure.

     You must provide values for `VCpuCount`, `MemoryMiB`, and `CpuManufacturers`. You can omit the other attributes; when omitted, the default values are used. For a description of each attribute and their default values, see get-spot-placement-scores.

   For example configurations, see Example configurations.

3. To get the Spot placement score for the requirements that you specified in the JSON file, use the get-spot-placement-scores command, and specify the name and path to your JSON file by using the `--cli-input-json` parameter.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

Example output if `SingleAvailabilityZone` is set to `false` or omitted (if omitted, it defaults to `false`) – a scored list of Regions is returned.

```
"SpotPlacementScores": [
    {
```

```
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

Example output if `SingleAvailabilityZone` is set to `true` – a scored list of Availability Zones is returned.

```
"SpotPlacementScores": [
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "use1-az1",
        "Score": 8
    },
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "usw2-az3",
        "Score": 6
    },
   ...
```

Example configurations

When using the AWS CLI, you can use the following example configurations.

Example configurations

- Example: Specify instance types and target capacity
- Example: Specify instance types, and target capacity in terms of memory
- Example: Specify attributes for attribute-based instance type selection
- Example: Specify attributes for attribute-based instance type selection and return a scored list of Availability Zones

Example: Specify instance types and target capacity

The following example configuration specifies three different instance types and a target Spot capacity of 500 Spot Instances.

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
    ],
    "TargetCapacity": 500
}
```

Example: Specify instance types, and target capacity in terms of memory

The following example configuration specifies three different instance types and a target Spot capacity of 500,000 MiB of memory, where the number of Spot Instances to launch must provide a total of 500,000 MiB of memory.

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
    ],
    "TargetCapacity": 500000,
    "TargetCapacityUnitType": "memory-mib"
}
```

Example: Specify attributes for attribute-based instance type selection

The following example configuration is configured for attribute-based instance type selection, and is followed by a text explanation of the example configuration.

```
{
    "TargetCapacity": 5000,
    "TargetCapacityUnitType": "vcpu",
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
```

```
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

InstanceRequirementsWithMetadata

To use attribute-based instance type selection, you must include the `InstanceRequirementsWithMetadata` structure in your configuration, and specify the desired attributes for the Spot Instances.

In the preceding example, the following required instance attributes are specified:

- `ArchitectureTypes` – The architecture type of the instance types must be `arm64`.
- `VirtualizationTypes` – The virtualization type of the instance types must be `hvm`.
- `VCpuCount` – The instance types must have a minimum of 1 and a maximum of 12 vCPUs.
- `MemoryMiB` – The instance types must have a minimum of 512 MiB of memory. By omitting the `Max` parameter, you are indicating that there is no maximum limit.

Note that there are several other optional attributes that you can specify. For the list of attributes, see get-spot-placement-scores.

TargetCapacityUnitType

The `TargetCapacityUnitType` parameter specifies the unit for the target capacity. In the example, the target capacity is `5000` and the target capacity unit type is `vcpu`, which together specify a desired target capacity of 5000 vCPUs, where the number of Spot Instances to launch must provide a total of 5000 vCPUs.

Example: Specify attributes for attribute-based instance type selection and return a scored list of Availability Zones

The following example configuration is configured for attribute-based instance type selection. By specifying `"SingleAvailabilityZone": true`, the response will return a list of scored Availability Zones.

```
{
    "TargetCapacity": 1000,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

Track your Spot Instance costs using the 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. If you don't have a Spot Instance running during a certain hour, you don't receive a data feed file for that hour.

Each hour of Spot Instance 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**
>
> You can create only one Spot Instance data feed per AWS account.

Spot Instance data feed is supported in all AWS Regions except China (Beijing), China (Ningxia), AWS GovCloud (US), and the Regions that are disabled by default.

Contents

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 **`amzn-s3-demo-bucket`** and your prefix is **`my-prefix`**, your file names are similar to the following:

```
amzn-s3-demo-bucket.s3.amazonaws.com/my-
prefix/111122223333.2023-12-09-07.001.b959dbc6.gz
```

For more information about bucket names, see Bucket naming rules in the *Amazon S3 User Guide*.

The Spot Instance data feed files are tab-delimited. Each line in the data file corresponds to one instance hour and contains the fields listed in the following table.

Field	Description
`Timestamp`	The timestamp used to determine the price charged for this instance usage.

Field	Description
`UsageType`	The type of usage and instance type being charged for. For `m1.small` Spot Instances, this field is set to `SpotUsage` . For all other instance types, this field is set to `SpotUsage:` {*instance-type*}. For example, `SpotUsage:c1.medium` .
`Operation`	The product being charged for. For Linux Spot Instances, this field is set to `RunInstances` . For Windows Spot Instances, this field is set to `RunInstances:0002` . Spot usage is grouped according to Availability Zone.
`InstanceID`	The ID of the Spot Instance that generated this instance usage.
`MyBidID`	The ID for the Spot Instance request that generated this instance usage.
`MyMaxPrice`	The maximum price specified for this Spot request.
`MarketPrice`	The Spot price at the time specified in the `Timestamp` field.
`Charge`	The price charged for this instance usage.
`Version`	The data feed version. The possible version is 1.0.

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. 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. For more information, see Logs sent to Amazon S3 in the *Amazon CloudWatch Logs User Guide*.

  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.
- If you encrypt your Amazon S3 bucket using server-side encryption with a AWS KMS key stored in AWS Key Management Service (SSE-KMS), you must use a customer managed key. For more information, see Amazon S3 bucket server-side encryption in the *Amazon CloudWatch Logs User Guide*.

Subscribe to your Spot Instance data feed

You can subscribe to your Spot Instance data feed at any time. You can't complete this task using the Amazon EC2 console.

If you get an error that the bucket does not have enough permissions, see the following article for troubleshooting information: Troubleshoot the data feed for Spot Instances.

AWS CLI

To subscribe to your data feed

Use the create-spot-datafeed-subscription command.

To specify a bucket with a prefix, use the following example:

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket \
    --prefix my-prefix
```

To specify a bucket without a prefix, use the following example:

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket
```

PowerShell

To subscribe to your data feed

Use the New-EC2SpotDatafeedSubscription cmdlet.

To specify a bucket with a prefix, use the following example:

```
New-EC2SpotDatafeedSubscription `
    -Bucket amzn-s3-demo-bucket `
    -Prefix my-prefix
```

To specify a bucket without a prefix, use the following example:

```
New-EC2SpotDatafeedSubscription `
    -Bucket amzn-s3-demo-bucket
```

View the data in your data feed

In the AWS Management Console, open AWS CloudShell. Use the following s3 sync command to get the .gz files from the S3 bucket for your data feed and store them in the folder that you specify.

```
aws s3 sync s3://amzn-s3-demo-bucket ./data-feed
```

To display the contents of a .gz file, change to the folder where you stored the contents of the S3 bucket.

```
cd data-feed
```

Use the **ls** command to view the names of the files. Use the **zcat** command with the name of the file to display the contents of the compressed file. The following is an example command.

```
zcat  111122223333.2023-12-09-07.001.b959dbc6.gz
```

The following is example output.

```
#Version: 1.0
#Fields: Timestamp UsageType Operation InstanceID MyBidID MyMaxPrice MarketPrice Charge
 Version
2023-12-09 07:13:47 UTC USE2-SpotUsage:c7a.medium       RunInstances:SV050
 i-0c3e0c0b046e050df     sir-pwq6nmfp    0.0510000000 USD        0.0142000000 USD
  0.0142000000 USD        1
```

Delete your Spot Instance data feed

When you are finished with the Spot Instance data feed, you can delete it.

AWS CLI

To delete your data feed

Use the delete-spot-datafeed-subscription command.

```
aws ec2 delete-spot-datafeed-subscription
```

PowerShell

To delete your data feed

Use the Remove-EC2SpotDatafeedSubscription cmdlet.

```
Remove-EC2SpotDatafeedSubscription
```

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 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**.
6. On the **Review** page, choose **Create role**.

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 or an encrypted Amazon EBS snapshot 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/aws-service-role/
spot.amazonaws.com/AWSServiceRoleForEC2Spot \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

Spot Instance quotas

There are quotas for the number of running Spot Instances and pending Spot Instance requests per AWS account per Region. After a pending Spot Instance request is fulfilled, the request no longer counts towards the quota because the running instance is counted towards the quota.

Spot Instance quotas 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 quota until Amazon EC2 detects the Spot Instance terminations and closes the requests.

We provide the following quota types for Spot Instances.

Name	Default	Adjustable
All DL Spot Instance Requests	0	Yes
All F Spot Instance Requests	0	Yes
All G and VT Spot Instance Requests	0	Yes
All Inf Spot Instance Requests	0	Yes
All P4, P3 and P2 Spot Instance Requests	0	Yes
All P5 Spot Instance Requests	0	Yes
All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests	5	Yes
All Trn Spot Instance Requests	0	Yes
All X Spot Instance Requests	0	Yes

Even though Amazon EC2 automatically adjusts your Spot Instance quotas based on your usage, you can request a quota increase if necessary. For example, if you intend to launch more Spot Instances than your current quota allows, you can request a quota increase. You can also request a quota increase if you submit a Spot Instance request and you receive the error `Max spot instance count exceeded`. To request a quota increase, use the Service Quotas console described in Amazon EC2 service quotas.

You can launch any combination of instance types that meet your changing application needs. For example, with an All Standard Spot Instance Requests quota of 256 vCPUs, you could request 32 `m5.2xlarge` Spot Instances (32 x 8 vCPUs) or 16 `c5.4xlarge` Spot Instances (16 x 16 vCPUs).

With Amazon CloudWatch metrics integration, you can monitor EC2 usage against your quotas. You can also configure alarms to warn about approaching quotas. For more information, see Service Quotas and Amazon CloudWatch alarms in the *Service Quotas User Guide*.

Amazon EC2 Dedicated Hosts

An Amazon EC2 Dedicated Host is a physical server that is fully dedicated for your use. You can optionally choose to share the instance capacity with other AWS accounts. For more information, see Cross-account Amazon EC2 Dedicated Host sharing.

Dedicated Hosts provide visibility and control over instance placement and they support host affinity. This means that you can launch and run instances on specific hosts, and you can ensure that instances run only on specific hosts. For more information, see Amazon EC2 Dedicated Host auto-placement and host affinity.

Dedicated Hosts provide comprehensive Bring Your Own License (BYOL) support. They allow you to use your existing per-socket, per-core, or per-VM software licenses, including Windows Server, SQL Server, SUSE Linux Enterprise Server, Red Hat Enterprise Linux, or other software licenses that are bound to VMs, sockets, or physical cores, subject to your license terms.

If you require your instances to run on dedicated hardware, but you do not need visibility or control over instance placement, and you do not need to use per-socket or per-core software licenses, you can consider using Dedicated Instances instead. Dedicated Instances and Dedicated Hosts can both be used to launch Amazon EC2 instances onto dedicated physical servers. There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, there are some key differences between them. The following table highlights some of the key differences between Dedicated Instances and Dedicated Hosts:

	Dedicated Host	**Dedicated Instance**
Dedicated physical server	Physical server with instance capacity fully dedicated to your use.	Physical server that's dedicated to a single customer account.
Instance capacity sharing	Can share instance capacity with other accounts.	Not supported
Billing	Per-host billing	Per-instance billing

	Dedicated Host	Dedicated Instance
Visibility of sockets, cores, and host ID	Provides visibility of the number of sockets and physical cores	No visibility
Host and instance affinity	Allows you to consistently deploy your instances to the same physical server over time	Not supported
Targeted instance placement	Provides additional visibility and control over how instances are placed on a physical server	Not supported
Automatic instance recovery	Supported. For more information, see Amazon EC2 Dedicated Host recovery.	Supported
Bring Your Own License (BYOL)	Supported	Partial support *
Capacity Reservations	Not supported	Supported

* Microsoft SQL Server with License Mobility through Software Assurance, and Windows Virtual Desktop Access (VDA) licenses can be used with Dedicated Instance.

For more information about Dedicated Instances, see Amazon EC2 Dedicated Instances.

Dedicated Hosts restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions:

- To run RHEL and SUSE Linux on Dedicated Hosts, you must bring your own AMIs. You can't use the RHEL and SUSE Linux AMIs that are offered by AWS or that are available on AWS Marketplace with Dedicated Hosts. For more information about how to create your own AMI, see Bring your own software licenses to Amazon EC2 Dedicated Hosts.

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.

- There is a limit on the number of running Dedicated Hosts per instance family per AWS account per Region. Quotas apply to running instances only. If your instance is pending, stopping, or stopped, it does not count towards your quota. To view the quotas for your account, or to request a quota increase, use the [Service Quotas console](#).
- Auto Scaling groups are supported when using a launch template that specifies a host resource group. For more information, see [Create a launch template using advanced settings](#) 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 can't launch Dedicated Hosts into placement groups.
- If you allocate a host for a virtualized instance type, you can't modify the instance type to a `.metal` instance type after the host is allocated. For example, if you allocate a host for the `m5.large` instance type, you can't modify the instance type to `m5.metal`.

  Similarly, if you allocate a host for a `.metal` instance type, you can't modify the instance type to a virtualized instance type after the host is allocated. For example, if you allocate a host for the `m5.metal` instance type, you can't modify the instance type to `m5.large`.

Contents

- Modify supported instance types for an existing Amazon EC2 Dedicated Host
- Modify Amazon EC2 Dedicated Host tenancy and affinity for an Amazon EC2 instance
- Release an Amazon EC2 Dedicated Host
- Migrate to Nitro-based Amazon EC2 Dedicated Hosts
- Purchase Dedicated Host Reservations for Dedicated Host billing discounts
- Cross-account Amazon EC2 Dedicated Host sharing
- Amazon EC2 Dedicated Hosts on AWS Outposts
- Amazon EC2 Dedicated Host recovery
- Host maintenance for Amazon EC2 Dedicated Host
- Monitor the state of your Amazon EC2 Dedicated Hosts
- Track Amazon EC2 Dedicated Host configuration changes using AWS Config

Amazon EC2 Dedicated Host pricing and billing

The price for a Dedicated Host varies by payment option.

Payment Options

- On-Demand Dedicated Hosts
- Dedicated Host Reservations
- Savings Plans
- Pricing for Windows Server on 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 an Amazon EC2 Dedicated Host.

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].

Note

Savings Plans are not supported with `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, and `u-24tb1.metal` Dedicated Hosts.

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 only. For more information, see Amazon EC2 Dedicated Hosts Pricing.

Amazon EC2 Dedicated Host instance capacity configurations

Dedicated Hosts support different configurations (physical cores, sockets, and VCPUs) that allow you to run instances of different families and sizes.

When you allocate a Dedicated Host in your account, you can choose a configuration that supports either a **single instance type**, or **multiple instance types** within the same instance family. The number of instances that you can run on a host depends on the configuration you choose.

Contents

Single instance type support

You can allocate a Dedicated Host that supports only one instance type. With this configuration, every instance that you launch on the Dedicated Host must be of the same instance type, which you specify when you allocate the host.

For example, you can allocate a host that supports only the `m5.4xlarge` instance type. In this case, you can run only `m5.4xlarge` instances on that host.

The number of instances that you can launch onto the host depends on the number of physical cores provided by the host, and the number of cores consumed by the specified instance type. For example, if you allocate a host for `m5.4xlarge` instances, the host provides 48 physical cores, and each `m5.4xlarge` instance consumes 8 physical cores. This means that you can launch up to 6 instances on that host (*48 physical cores / 8 cores per instance = 6 instances*).

Multiple instance type support

You can allocate a Dedicated Host that supports multiple instance types within the same instance family. This allows you to run different instance types on the same host, as long as they're in the same instance family and the host has sufficient instance capacity.

For example, you can allocate a host that supports different instance types within the R5 instance family. In this case, you can launch certain combinations of R5 instance types, such as `r5.large`, `r5.xlarge`, `r5.2xlarge`, and `r5.4xlarge`, on that host, within the host's physical core capacity.

The following instance families support Dedicated Hosts with multiple instance type support:

- **General purpose:** A1 | M5 | M5n | M6i | M7i | T3
- **Compute optimized:** C5 | C5n | C6i | C7i
- **Memory optimized:** R5 | R5n | R6i | R7i

The number of instances you can run on the host depends on the number of physical cores provided by the host, and the number of cores consumed by each instance type that you run on the host. For example, if you allocate an R5 host, which provides 48 physical cores, and you run two `r5.2xlarge` instances (*4 cores x 2 instances*) and three `r5.4xlarge` instances (*8 cores x 3 instances*), those instances consume a total of 32 cores, and you might be able to run certain combinations of R5 instances as long as they are within the remaining 16 cores.

However, for each instance family, there is a limit on the number of instances that can be run for each instance type. For example, an R5 Dedicated Host supports a maximum of 2 `r5.8xlarge` instances, which uses 32 of the physical cores. In this case, additional R5 instances of smaller types can then be used to fill the host to core capacity. For the supported number of instance types for each instance family, see the [Dedicated Hosts Configuration Table](#).

The following table shows example instance type combinations:

Instance family	Example instance type combinations
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`

Considerations

Keep the following in mind when working with Dedicated Hosts that support multiple instance types:

- Using multiple instance types on the same host is possible only within the same instance family.
- When mixing instance types, to maximize host utilization, we recommend launching larger instance types first followed by smaller instance types.
  - Depending on the combination and launch order of the instance types on a Dedicated Host, it may not be physically possible to maximize the utilization of the host. When mixing instance types on a host, some capacity might be available on the host but not usable. For example, you might see 16 vCPUs available on an r5n host but may not be able to launch a 4xlarge instance on the host even though r5n.4xlarge runs on 16 vCPUs.

> **Note**
>
> If you enable an A1 Dedicated Host for multiple instance types, you can launch only a mix of `a1.xlarge` and `a1.2xlarge` instances on that host. If you launch an `a1.medium` or `a1.large` instance on that host, you will be restricted to launching only more of that same instance type on the host. A single `a1.4xlarge` instance consumes all capacity on the host. If you require a host for either `a1.medium` or `a1.large` instances, we recommend that you allocate separate hosts for those instance types.

Burstable T3 instances on Amazon EC2 Dedicated Hosts

Dedicated Hosts support burstable performance T3 instances. T3 instances provide a cost-efficient way of using your eligible BYOL license software on dedicated hardware. The smaller vCPU footprint of T3 instances enables you to consolidate your workloads on fewer hosts and maximize your per-core license utilization.

T3 Dedicated Hosts are best suited for running BYOL software with low to moderate CPU utilization. This includes eligible per-socket, per-core, or per-VM software licenses, such as Windows Server, Windows Desktop, SQL Server, SUSE Enterprise Linux Server, Red Hat Enterprise Linux, and Oracle Database. Examples of workloads suited for T3 Dedicated Hosts are small and medium databases, virtual desktops, development and test environments, code repositories, and product prototypes. T3 Dedicated Hosts are not recommended for workloads with sustained high CPU utilization or for workloads that experience correlated CPU bursts simultaneously.

T3 instances on Dedicated Hosts use the same credit model as T3 instances on shared tenancy hardware. However, they support the `standard` credit mode only; they do not support the `unlimited` credit mode. In `standard` mode, T3 instances on Dedicated Hosts *earn*, *spend*, and *accrue* credits in the same way as burstable instances on shared tenancy hardware. They provide a baseline CPU performance with the ability to burst above the baseline level. To burst above the baseline, the instance spends credits that it has accrued in its CPU credit balance. When the accrued credits are depleted, CPU utilization is lowered to the baseline level. For more information about `standard` mode, see How standard burstable performance instances work.

T3 Dedicated Hosts support all of the features offered by Amazon EC2 Dedicated Hosts, including multiple instance sizes on a single host, Host resource groups, and BYOL.

Supported T3 instance sizes and configurations

T3 Dedicated Hosts run general purpose burstable T3 instances that share CPU resources of the host by providing a baseline CPU performance and the ability to burst to a higher level when needed. This enables T3 Dedicated Hosts, which have 48 cores, to support up to a maximum of 192 instances per host. In order to efficiently utilize the host's resources and to provide the best instance performance, the Amazon EC2 instance placement algorithm automatically calculates the supported number of instances and instance size combinations that can be launched on the host.

T3 Dedicated Hosts support multiple instance types on the same host. All T3 instance sizes are supported on Dedicated Hosts. You can run different combinations of T3 instances up to the CPU limit of the host.

The following table lists the supported instance types, summarizes the performance of each instance type, and indicates the maximum number of instances of each size that can be launched.

Insta type	vCPUs	Memory (GiB)	Baseline CPU utilization per vCPU	Network burst bandwidth (Gbps)	Amazon EBS burst bandwidth (Mbps)	Max number of instances per Dedicated Host
t3.na	2	0.5	5%	5	Up to 2,085	192
t3.m	2	1	10%	5	Up to 2,085	192
t3.sr	2	2	20%	5	Up to 2,085	192
t3.m	2	4	20%	5	Up to 2,085	192
t3.la	2	8	30%	5	2,780	96
t3.xl	4	16	40%	5	2,780	48
t3.2x e	8	32	40%	5	2,780	24

Monitor CPU utilization for T3 Dedicated Hosts

You can use the `DedicatedHostCPUUtilization` Amazon CloudWatch metric to monitor the vCPU utilization of a Dedicated Host. The metric is available in the `EC2` namespace and `Per-Host-Metrics` dimension. For more information, see [Dedicated Host metrics](#).

Bring your own software licenses to Amazon EC2 Dedicated Hosts

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 Amazon EC2 Dedicated Host configuration changes using AWS Config.

Amazon EC2 Dedicated Host auto-placement and host 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 auto-placement is **disabled** for a Dedicated Host, it accepts only host tenancy instance launches that specify its unique host ID. This is the default setting for new Dedicated Hosts.

When auto-placement is **enabled** for a Dedicated Host, it accepts any untargeted, host tenancy 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 `Default`, 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).

Allocate an Amazon EC2 Dedicated Host for use in your account

To begin using a Dedicated Host, you must first allocate it in your account. 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.

When you allocate a Dedicated Host in your account, you can choose a configuration that supports either a **single instance type**, or **multiple instance types** within the same instance family. The number of instances that you can run on the host depends on the configuration you choose. For more information see Amazon EC2 Dedicated Host instance capacity configurations.

Console

To allocate a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dedicated Hosts** and then choose **Allocate Dedicated Host**.
3. For **Instance family**, choose the instance family for the Dedicated Host.
4. Specify whether the Dedicated Host supports multiple instance 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 Amazon EC2 Dedicated Host auto-placement and host affinity.
7. To enable host recovery for the Dedicated Host, for **Host recovery**, choose **Enable**. For more information, see Amazon EC2 Dedicated Host recovery.
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**.

AWS CLI

To allocate a Dedicated Host

Use the allocate-hosts command. The following example allocates a Dedicated Host that supports multiple instance types from the m5 instance family in the us-east-1a Availability Zone. It also enables host recovery and disables auto-placement.

```
aws ec2 allocate-hosts \
    --instance-family "m5" \
    --availability-zone "us-east-1a" \
    --auto-placement "off" \
    --host-recovery "on" \
    --quantity 1
```

The following example allocates a Dedicated Host that supports *untargeted* instance launches in the specified Availability Zone, enables host recovery, and enables auto-placement.

```
aws ec2 allocate-hosts \
    --instance-type "m5.large" \
    --availability-zone "eu-west-1a" \
    --auto-placement "on" \
    --host-recovery "on" \
    --quantity 1
```

PowerShell

To allocate a Dedicated Host

Use the New-EC2Host cmdlet. The following example allocates a Dedicated Host that supports multiple instance types from the m5 instance family in the us-east-1a Availability Zone. The host also has host recovery enabled and auto-placement disabled.

```
New-EC2Host `
    -InstanceFamily m5 `
    -AvailabilityZone us-east-1a `
    -AutoPlacement Off `
    -HostRecovery On `
    -Quantity 1
```

The following example allocates a Dedicated Host that supports *untargeted* instance launches in the specified Availability Zone and enables host recovery.

```
New-EC2Host `
    -InstanceType m5.large `
```

```
    -AvailabilityZone eu-west-1a `
    -AutoPlacement On `
    -HostRecovery On `
    -Quantity 1
```

Launch Amazon EC2 instances on an Amazon EC2 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.

Considerations

- SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.
- For Dedicated Hosts that support multiple instance sizes, we recommend that you launch the larger instance sizes first, and then fill the remaining instance capacity with the smaller instance sizes as needed.
- Before you launch your instances, take note of the limitations. For more information, see [Dedicated Hosts restrictions](#).

Console

To launch an instance onto a specific Dedicated Host from the Dedicated Hosts page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, select a host and choose **Actions**, **Launch Instance(s) onto host**.
4. In the **Application and OS Images** section, select an AMI from the list.
5. In the **Instance type** section, select the instance type to launch.

> **Note**
>
> 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. In the **Key pair** section, select the key pair to associate with the instance.
7. In the **Advanced details** section, for **Tenancy affinity**, choose one of the following:

   - **Off** — Host affinity disabled. The instance launches onto the specified host, but it is not guaranteed to restart on the same Dedicated Host if stopped.
   - A Dedicated Host ID — Host affinity enabled. If stopped, the instance always restarts on this specified host if it has capacity. If the host does not have capacity, the instance can't be restarted; you must estabilish affinity with a different host.

   For more information about Affinity, see Amazon EC2 Dedicated Host auto-placement and host affinity.

   > **Note**
   >
   > The **Tenancy** and **Host** options are pre-configured based on the host that you selected.

8. Configure the remaining instance options as needed. For more information, see Reference for Amazon EC2 instance configuration parameters.
9. Choose **Launch instance**.

To launch an instance onto a Dedicated Host using the Launch Instance wizard

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, **Launch instance**.
3. In the **Application and OS Images** section, select an AMI from the list.
4. In the **Instance type** section, select the instance type to launch.
5. In the **Key pair** section, select the key pair to associate with the instance.
6. In the **Advanced details** section, do the following:

   a. For **Tenancy**, select **Dedicated Host**.

   b. For **Target host by**, select **Host ID**.

c. For **Target host ID**, select the host onto which to launch the instance.

d. For **Tenancy affinity**, choose one of the following:

- **Off** — Host affinity disabled. The instance launches onto the specified host, but it is not guaranteed to restart on the same Dedicated Host if stopped.
- A Dedicated Host ID — Host affinity enabled. If stopped, the instance always restarts on this specified host if it has capacity. If the host does not have capacity, the instance can't be restarted; you must estabilish affinity with a different host.

For more information about Affinity, see Amazon EC2 Dedicated Host auto-placement and host affinity.

7. Configure the remaining instance options as needed. For more information, see Reference for Amazon EC2 instance configuration parameters.
8. Choose **Launch instance**.

AWS CLI

To launch an instance onto a Dedicated Host

Use the run-instances command and specify the instance affinity, tenancy, and host in the `--placement` option.

```
--placement Affinity=Host,Tenancy=dedicated,HostId=h-07879acf49EXAMPLE
```

PowerShell

To launch an instance onto a Dedicated Host

Use the New-EC2Instance cmdlet and specify the instance affinity, tenancy, and host in the `-Placement` parameter.

```
-Placement_Affinity Host `
-Placement_Tenancy dedicated `
-Placement_HostId h-07879acf49EXAMPLE
```

Launch Amazon EC2 instances into a host resource group

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*.

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.

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. In the **Application and OS Images** section, select an AMI from the list.
4. In the **Instance type** section, select the instance type to launch.
5. In the **Key pair** section, select the key pair to associate with the instance.
6. In the **Advanced details** section, do the following:

    a. For **Tenancy**, select **Dedicated Host**.

    b. For **Target host by**, select **Host resource group**.

    c. For **Tenancy host resource group**, select the host resource group into which to launch the instance.

d. For **Tenancy affinity**, do one of the following:

- Select **Off** — The instance launches onto the specified host, but it is not guaranteed to restart on the same Dedicated Host if stopped.
- Select the Dedicated Host ID — If stopped, the instance always restarts on this specific host.

For more information about Affinity, see Amazon EC2 Dedicated Host auto-placement and host affinity.

7. Configure the remaining instance options as needed. For more information, see Reference for Amazon EC2 instance configuration parameters.
8. Choose **Launch instance**.

AWS CLI

To launch an instance into a host resource group

Use the run-instances command. In the `--placement` option, omit the tenancy and specify the ARN of the host resource group.

```
--placement HostResourceGroupArn=arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

PowerShell

To launch an instance into a host resource group

Use the New-EC2Instance cmdlet. In the `-Placement` parameter, omit the tenancy and specify the ARN of the host resource group.

```
-Placement_HostResourceGroupArn arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

Modify the auto-placement setting for an existing Amazon EC2 Dedicated Host

You can modify the auto-placement settings of a Dedicated Host after you have allocated it to your AWS account.

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 Amazon EC2 Dedicated Host auto-placement and host affinity.
5. Choose **Save**.

AWS CLI

To modify the auto-placement of a Dedicated Host

Use the modify-hosts command.

```
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 cmdlet.

```
Edit-EC2Host `
    -AutoPlacement 1 `
    -HostId h-012a3456b7890cdef
```

Modify supported instance types for an existing Amazon EC2 Dedicated Host

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.

If you allocate a host for a virtualized instance type, you can't modify the instance type to a `.metal` instance type after the host is allocated. For example, if you allocate a host for the `m5.large` instance type, you can't modify the instance type to `m5.metal`. Similarly, if you allocate a host for a `.metal` instance type, you can't modify the instance type to a virtualized instance type after the host is allocated. For example, if you allocate a host for the `m5.metal` instance type, you can't modify the instance type to `m5.large`.

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**.

AWS CLI

To modify the supported instance types for a Dedicated Host

Use the modify-hosts command.

The following example 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 example 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 cmdlet.

The following example modifies a Dedicated Host to support multiple instance types within the `m5` instance family.

```
Edit-EC2Host `
    -InstanceFamily m5 `
    -HostId h-012a3456b7890cdef
```

The following example modifies a Dedicated Host to support `m5.xlarge` instances only.

```
Edit-EC2Host `
    -InstanceType m5.xlarge `
    -HostId h-012a3456b7890cdef
```

Modify Amazon EC2 Dedicated Host tenancy and affinity for an Amazon EC2 instance

You can change the tenancy of an instance after you have launched it. You can also modify the affinity for your instance to target a specific host or allow it to launch on any available dedicated host with matching attributes in your account. To modify either instance tenancy or affinity, the instance must be in the `stopped` state.

The operating system details of your instance—and whether SQL Server is installed—affect what conversions are supported. For more information about the tenancy conversion paths available to your instance, see Tenancy conversion in the *License Manager User Guide*.

> **Note**
>
> For T3 instances, you must launch the instance on a Dedicated Host to use a tenancy of `host`. For T3 instances, you can't change the tenancy from `host` to `dedicated` or `default`. Attempting to make one of these unsupported tenancy changes results in an `InvalidRequest` error code.

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. With the instance selected, choose **Actions**, **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 Amazon EC2 Dedicated Instances.

- 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 Amazon EC2 Dedicated Host auto-placement and host affinity.

6. Choose **Save**.

AWS CLI

To modify instance tenancy or affinity

Use the modify-instance-placement 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 \
    --tenancy host \
    --host-id h-012a3456b7890cdef
```

PowerShell

To modify instance tenancy or affinity

Use the Edit-EC2InstancePlacement cmdlet. The following example changes the specified instance's affinity from `default` to `host`, and specifies the Dedicated Host that the instance has affinity with.

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
```

```
    -Affinity host `
    -Tenancy host `
    -HostId h-012a3456b7890cdef
```

Release an Amazon EC2 Dedicated Host

If you no longer need Dedicated Host, you can stop the instances running on the host, direct them to launch on a different host, and then *release* the host.

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.

Console

To release a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dedicated Hosts**.
3. On the **Dedicated Hosts** page, select the Dedicated Host to release.
4. Choose **Actions**, **Release host**.
5. To confirm, choose **Release**.

AWS CLI

To release a Dedicated Host

Use the release-hosts command.

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

To release a Dedicated Host

Use the Remove-EC2Host cmdlet.

```
Remove-EC2Host -HostId h-012a3456b7890cdef
```

After you release a Dedicated Host, you can't reuse the same host or host ID again, and you are no longer charged On-Demand billing rates for it. The state of the Dedicated Host is changed to `released`, and you are not able to launch any instances onto that host.

> **Note**
>
> If you have recently released Dedicated Hosts, it can take some time for them to stop counting towards your limit. During this time, you might experience `LimitExceeded` errors when trying to allocate new Dedicated Hosts. If this is the case, try allocating new hosts again after a few minutes.

The instances that were stopped are still available for use and are listed on the **Instances** page. They retain their `host` tenancy setting.

Migrate to Nitro-based Amazon EC2 Dedicated Hosts

The Nitro System is a collection of hardware and software components built by AWS that enable high performance, high availability, and high security. Nitro-based Dedicated Hosts offer improved price performance compared to Xen-based Dedicated Hosts. If you have any Xen-based Dedicated Hosts in your account, we recommend that you migrate your workloads to Nitro-based Dedicated Hosts. For more information, see AWS Nitro System.

To migrate from a Xen-based Dedicated Host to a Nitro-based Dedicated Host, you need to migrate the Xen-based instances on your Dedicated Host to Nitro-based instance types, allocate a new Nitro-based Dedicated Host, and then move your migrated Nitro-based instances to your new Nitro-based Dedicated Host.

This topic provides detailed steps for migrating from Xen-based Dedicated Hosts to Nitro-based Dedicated Hosts.

Migration steps

- Step 1: Identify your Xen-based Dedicated Hosts
- Step 2: Migrate Xen-based instances to Nitro-based instance types
- Step 3: Allocate a Nitro-based Dedicated Host
- Step 4: Move migrated instances to new Nitro-based Dedicated Host
- Step 5: Release unused Xen-based Dedicated Host

Step 1: Identify your Xen-based Dedicated Hosts

The following Dedicated Hosts are Xen-based and are eligible to be migrated to Nitro-based Dedicated Hosts.

- **General purpose:** M3 | M4
- **Compute optimized:** C3 | C4
- **Memory optimized:** R3 | R4 | X1 | X1e
- **Storage optimized:** D2 | H1 | I2 | I3
- **Accelerated computing:** F1 | G3 | P2 | P3

To check if you have Xen-based Dedicated Hosts in your account

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, choose **Dedicated Hosts**.
3. In the **Search field**, use the **Instance family** filter to search for the Xen-based Dedicated Hosts above. For example, *Instance family = m3*.

Step 2: Migrate Xen-based instances to Nitro-based instance types

Instances that run on Xen-based Dedicated Hosts are also Xen-based. You must migrate these instances to Nitro-based instance types before you can move them to Nitro-based Dedicated Hosts.

> **Important**
>
> Before you begin migrating your instances, we recommend that you back up your data. For more information, see Create multi-volume Amazon EBS snapshots from an Amazon EC2 instance.

To find instances running on your Xen-based Dedicated Hosts

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, choose **Dedicated Hosts**.
3. Select the Xen-based host you intend to migrate and then select the **Running instances** tab. The tab lists all of the instances running of the selected host.

To migrate **Linux instances**, see Amazon EC2 instance type changes.

To migrate **Windows instances**, see Migrate an EC2 Windows instance to a Nitro-based instance type.

> **Note**
>
> Ensure that you migrate your instances to an instance type that matches the Nitro-based Dedicated Host that you intend to migrate to. For example, if you intend to migrate to a M7i Dedicated Host, ensure that you migrate your instances to an M7i instance type.

Step 3: Allocate a Nitro-based Dedicated Host

To find supported Nitro-based Dedicated Hosts

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, select **Instance Types**.
3. Apply the following filters:
   - *Hypervisor = nitro*
   - *Dedicated Host support = true*

After you've found a suitable Nitro-based instance type, allocate a new Dedicated Host.

Step 4: Move migrated instances to new Nitro-based Dedicated Host

After you have allocated the Nitro-based Dedicated Host and it reaches the `available` state, you can move the instances that you previously migrated to Nitro-based instance types to the new Dedicated Host.

To move your instances to your new Nitro-based Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, **Instances**.
3. Select the instance that you migrated and choose **Actions**, **Instance settings**, **Modify instance placement**.
4. For **Target dedicated host**, select the new Nitro-based Dedicated Host, and then choose **Save**.

5. Restart the instance. Select the instance and choose **Instance state**, **Start instance**.

Step 5: Release unused Xen-based Dedicated Host

After you have migrated your workloads from the Xen-based Dedicated Host to the new Nitro-based Dedicated Host, you can release the Xen-based Dedicated Host if you no longer need it.

Purchase Dedicated Host Reservations for Dedicated Host billing discounts

Dedicated Host Reservations provide you with a discount of up to 70 percent compared to On-Demand Dedicated Host pricing. You must have active Dedicated Hosts allocated in your account before you can purchase Dedicated Host Reservations. For more information, see Dedicated Host Reservations.

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 **Find offerings** screen, do the following:
   a. For **Instance family**, select the instance family of the Dedicated Host for which to purchase the Dedicated Host Reservation.
   b. For **Payment option**, select and configure your preferred payment option.
4. Choose **Next**.
5. Select the Dedicated Hosts with which to associate the Dedicated Host Reservation, and then choose **Next**.
6. (*Optional*) Assign tags to the Dedicated Host Reservation.
7. Review your order and choose **Purchase**.

AWS CLI

To purchase reservations

1. Use the describe-host-reservation-offerings 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.

   The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

   ```
   aws ec2 describe-host-reservation-offerings \
       --filter Name=instance-family,Values=m4 \
       --max-duration 31536000
   ```

   The command returns a list of offerings that match your criteria. Note the ID of the offering to purchase.

2. Use the purchase-host-reservation 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 cmdlet to list the available offerings that match your needs. The following examples list the offerings that support instances in the `m5` instance family and have a one-year term.

   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.

```
$filter = @{Name="instance-family"; Values="m5"}
Get-EC2HostReservationOffering `
    -Filter $filter `
    -MaxDuration 31536000
```

The command returns a list of offerings that match your criteria. Note the ID of the offering to purchase.

2. Use the New-EC2HostReservation cmdlet to purchase the offering and provide the offering ID 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.

```
New-EC2HostReservation `
    -OfferingId hro-03f707bf363b6b324 `
    -HostIdSet h-013abcd2a00cbd123
```

Cross-account Amazon EC2 Dedicated Host sharing

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

- To share a Dedicated Host, you must own it in your AWS account. You can't share a Dedicated Host that has been shared with you.
- To share a Dedicated Host with your AWS organization or an organizational unit in your AWS organization, you must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the *AWS RAM User Guide*.

Limitations for sharing Dedicated Hosts

You can't share Dedicated Hosts that have been allocated for the following instance types: `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, and `u-24tb1.metal`.

Related services

AWS Resource Access Manager

Dedicated Host sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS

Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, or organizational units or an entire organization from AWS Organizations.

For more information about AWS RAM, see the AWS RAM User Guide.

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

1. Open the AWS RAM console at https://console.aws.amazon.com/ram/home.
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.

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 [Amazon EC2 Dedicated Host recovery](#).

Share an Amazon EC2 Dedicated Host across AWS accounts

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.

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.

To share a Dedicated Host that you own using the AWS RAM console

See [Create a resource share](#) in the *AWS RAM User Guide*.

AWS CLI

To share a Dedicated Host that you own

Use the create-resource-share command.

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

PowerShell

To share a Dedicated Host that you own

Use the New-RAMResourceShare cmdlet.

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

Unshare a Dedicated Host that is shared with other AWS accounts

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.

Console

To unshare a shared Dedicated Host that you own

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**.

To unshare a shared Dedicated Host that you own using the AWS RAM console

See Update a resource share in the *AWS RAM User Guide*.

AWS CLI

To unshare a shared Dedicated Host that you own

Use the disassociate-resource-share command.

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

PowerShell

To unshare a shared Dedicated Host that you own

Use the Disconnect-RAMResourceShare cmdlet.

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE"
```

View shared Amazon EC2 Dedicated Hosts in your AWS account

You can view Dedicated Host that you are sharing with other accounts, and Dedicated Hosts that are shared with you. If you own the Dedicated Host, you can see all of the instances running on the host, including instances launched by consumers. If the Dedicated Host is shared with you, you can see only the instances that you launched onto the shared host, and not those launched by other consumers.

Owners and consumers can identify shared Dedicated Hosts using one of the following methods.

Console

To identify a shared Dedicated Host

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.
3. The **Owner** column shows the AWS account ID of the Dedicated Host owner.
4. To view the instances running on the hosts, select the **Instances** tab.

AWS CLI

To identify a shared Dedicated Host

Use the describe-hosts command. The command returns the Dedicated Hosts that you own and Dedicated Hosts that are shared with you. The value of `Owner` is the account ID of the owner of the Dedicated Host. The `Instances` list describes the instances running on the host.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

PowerShell

To identify a shared Dedicated Host

Use the Get-EC2host cmdlet. The cmdlet returns the Dedicated Hosts that you own and Dedicated Hosts that are shared with you. The value of `Owner` in the response is the account ID of the owner of the Dedicated Host. The `Instances` list describes the instances running on the host.

```
Get-EC2Host -Filter @{Name="state"; Values="available"}
```

Amazon EC2 Dedicated Hosts on AWS Outposts

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to your premises. By providing local access to AWS managed infrastructure, AWS Outposts enables you 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 allocate Dedicated Hosts on Outposts that you own in your account. This makes it easier for you to bring your existing software licenses and workloads that require a dedicated physical server to AWS Outposts. You can also target specific hardware assets on an Outpost to help minimize latency between your workloads.

Dedicated Hosts allow you to use your eligible software licenses on Amazon EC2, so that you get the flexibility and cost effectiveness of using your own licenses. Other software licenses that are bound to virtual machines, sockets, or physical cores, can also be used on Dedicated Hosts, subject to their license terms. While Outposts have always been a single-tenant environments that are eligible for BYOL workloads, Dedicated Hosts allows you to limit the needed licenses to a single host as opposed to the entire Outpost deployment.

Additionally, using Dedicated Hosts on an Outpost gives you greater flexibility in instance type deployment, and more granular control over instance placement. You can target a specific host for instance launches and use host affinity to ensure that the instance always runs on that host, or you can use auto-placement to launch an instance onto any available host that has matching configurations and available capacity.

Contents

Prerequisites

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*.

Supported features

- The following instance families are supported: `C5`, `M5`, `R5`, `C5d`, `M5d`, `R5d`, `G4dn`, and `i3en`.

- Dedicated Hosts on Outposts can be configured to support multiple instance sizes. Support for multiple instance sizes is available for the following instance families: `C5`, `M5`, `R5`, `C5d`, `M5d`, and `R5d`. For more information, see Amazon EC2 Dedicated Host instance capacity configurations.
- Dedicated Hosts on Outposts support auto-placement and targeted instance launches. For more information, see Amazon EC2 Dedicated Host auto-placement and host affinity.
- Dedicated Hosts on Outposts support host affinity. For more information, see Amazon EC2 Dedicated Host auto-placement and host affinity.
- Dedicated Hosts on Outposts support sharing with AWS RAM. For more information, see Cross-account Amazon EC2 Dedicated Host sharing.

Considerations

- Dedicated Host Reservations are not supported on Outposts.
- Host resource groups and AWS License Manager are not supported on Outposts.
- Dedicated Hosts on Outposts do not support burstable T3 instances.
- Dedicated Hosts on Outposts do not support host recovery.
- Simplified automatic recovery is not supported for instances with Dedicated Host tenancy on Outposts.

Allocate an Amazon EC2 Dedicated Host on AWS Outposts

You allocate and use Dedicated Hosts on Outposts in the same way that would with Dedicated Hosts in an AWS Region.

Prerequisites

Create a subnet on the Outpost. For more information, see Create a subnet in the *AWS Outposts User Guide*.

To allocate a Dedicated Host on an Outpost, use one of the following methods:

Console

To allocate a Dedicated Host on an Outpost using the AWS Outposts console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. In the navigation pane, choose **Outposts**. Select the Outpost and then choose **Actions**, **Allocate Dedicated Host**.
3. Configure the Dedicated Host as needed. For more information, see [Allocate an Amazon EC2 Dedicated Host for use in your account](#).

> **Note**
>
> **Availability Zone** and **Outpost ARN** should be pre-populated with the Availability Zone and ARN of the selected Outpost.

4. Choose **Allocate**.

To allocate a Dedicated Host on an Outpost using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dedicated Hosts**, and then choose **Allocate Dedicated Host**.
3. For **Availability Zone**, select the Availability Zone associated with the Outpost.
4. For **Outpost ARN**, enter the ARN of the Outpost.
5. To target specific hardware assets on the Outpost, for **Target specific hardware assets on Outpost**, select **Enable**. For each hardware asset to target, choose **Add asset ID**, and then enter the ID of the hardware asset.

> **Note**
>
> The value that you specify for **Quantity** must be equal to the number of asset IDs that you specify. For example, if you specify 3 asset IDs, then Quantity must also be 3.

6. Configure the remaining Dedicated Host settings as needed. For more information, see [Allocate an Amazon EC2 Dedicated Host for use in your account](#).
7. Choose **Allocate**.

AWS CLI

To allocate a Dedicated Host on an Outpost

Use the allocate-hosts command. For `--availability-zone`, specify the Availability Zone associated with the Outpost. For `--outpost-arn`, specify the ARN of the Outpost. Optionally, for `--asset-ids`, specify the IDs of the Outpost hardware assets to target.

```
aws ec2 allocate-hosts \
    --availability-zone "us-east-1a" \
    --outpost-arn "arn:aws:outposts:us-east-1a:111122223333:outpost/
op-4fe3dc21baEXAMPLE" \
    --asset-ids asset_id \
    --instance-family "m5" \
    --auto-placement "off" \
    --quantity 1
```

PowerShell

To allocate a Dedicated Host on an Outpost

Use the New-EC2Host cmdlet. Specify the Availability Zone associated with the Outpost. Optionally, for `-AssetId`, specify the IDs of the Outpost hardware assets to target.

```
New-EC2Host `
    -AvailabilityZone "us-east-1a" `
    -OutpostArn "arn:aws:outposts:us-east-1a:111122223333:outpost/
op-4fe3dc21baEXAMPLE" `
    -AssetId asset_id `
    -InstanceFamily "m5" `
    -AutoPlacement "off" `
    -Quantity 1
```

To launch an instance onto a Dedicated Host on an Outpost

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dedicated Hosts**. Select the Dedicated Host that you allocated in the previous step and choose **Actions**, **Launch instance onto host**.
3. Configure the instance as needed and then launch the instance. For more information, see Launch Amazon EC2 instances on an Amazon EC2 Dedicated Host.

Amazon EC2 Dedicated Host recovery

Dedicated Host auto recovery restarts your instances on to a new replacement host when certain problematic conditions 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 concerning system power or network connectivity events. Other Dedicated Host issues will require manual intervention to recover from.

Contents

How Amazon EC2 Dedicated Host recovery works

Dedicated Hosts and the host resource groups recovery process use host-level health checks to assess Dedicated Host availability and to detect underlying system failures. The type of Dedicated Host failure determines if Dedicated Host auto recovery is possible. 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

> **Important**
>
> Dedicated Host auto recovery does not occur when the host is scheduled for retirement.

Dedicated Host auto recovery

When a system power or network connectivity failure is detected on your Dedicated Host, Dedicated Host auto recovery is initiated and Amazon EC2 **automatically allocates a replacement**

Dedicated Host in the same Availability Zone as the original 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
- Reservation

When 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

Additionally, the built-in integration with AWS License Manager automates the tracking and management of your licenses.

> **Note**
>
> AWS License Manager integration is supported only in Regions in which AWS License Manager is available.

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 Health Dashboard event. A second notification is sent after the host recovery has been successfully completed.

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 notification settings have been configured for AWS License Manager). 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 and Settings in License Manager in the *AWS License Manager User Guide*.

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 root volumes, the Dedicated Host is not released. Instead, it is marked for retirement and enters the `permanent-failure` state.

Scenarios without Dedicated Host auto recovery

Dedicated Host auto recovery does not occur when the host is scheduled for retirement. You will receive a retirement notification in the AWS Health Dashboard, an Amazon CloudWatch event, and the AWS account owner email address receives a message regarding the Dedicated Host failure.

Follow the remedial steps described in the retirement notification within the specified time period to manually recover the instances on the retiring host.

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.

Supported instance types

Host recovery is supported for the following instance families: A1 | C3 | C4 | C5 | C5n | C6a | C6g | C6i | Inf1 | G3 | G5g | M3 | M4 | M5 | M5n | M5zn | M6a | M6g | M6i | Mac1 | Mac2 | Mac2-m1ultra | Mac2-m2 | Mac2-m2pro | P2 | P3 | R3 | R4 | R5 | R5b | R5n | R6g | R6i | T3 | X1 | X1e | X2iezn | u-6tb1 | u-9tb1 | u-12tb1 | u-18tb1 | u-24tb1.

To recover instances that are not supported, see Manually recover instances that are not supported by Amazon EC2 Dedicated Host recovery.

> **Note**
>
> Dedicated Host auto recovery of supported metal instance types will take longer to detect and recover from than non-metal instance types.

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.

Manage Amazon EC2 Dedicated Host recovery

Dedicated Host auto recovery restarts your instances on to a new replacement host when certain problematic conditions are detected on your Dedicated Host. You can enable host recovery when you allocate the Dedicated Host or after allocation.

Use the following procedures to enable host recovery when allocating the host.

Console

To enable host recovery at allocation

When allocating a Dedicated Host using the Amazon EC2 console, for **Host recovery**, choose **Enable**. For more information, see Allocate an Amazon EC2 Dedicated Host for use in your account.

AWS CLI

To enable host recovery at allocation

Use the allocate-hosts command.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

PowerShell

To enable host recovery at allocation

Use the New-EC2Host cmdlet.

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement on `
    -HostRecovery on `
    -Quantity 1
```

Use the following procedures to manage host recovery for a Dedicated Host.

Console

To manage host recovery after allocation

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.
4. Choose **Actions**, **Modify host**.
5. For **Host recovery**, select or clear **Enable**.
6. Choose **Save**.

AWS CLI

To enable host recovery after allocation

Use the modify-hosts command.

```
aws ec2 modify-hosts \
    --host-recovery on \
    --host-ids h-012a3456b7890cdef
```

To disable host recovery after allocation

Use the modify-hosts command and specify the `host-recovery` parameter with a value of `off`.

```
aws ec2 modify-hosts \
    --host-recovery off \
    --host-ids h-012a3456b7890cdef
```

PowerShell

To enable host recovery after allocation

Use the Edit-host cmdlet.

```
Edit-EC2Host `
    -HostRecovery on `
```

```
    -HostId h-012a3456b7890cdef
```

To disable host recovery after allocation

Use the Edit-EC2Host cmdlet.

```
Edit-EC2Host `
    -HostRecovery off `
    -HostId h-012a3456b7890cdef
```

View the host recovery setting for your Amazon EC2 Dedicated Host

You can view the host recovery configuration for a Dedicated Host at any time.

Console

To view the host recovery configuration for 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, and in the **Description** tab, review the **Host Recovery** field.

AWS CLI

To view the host recovery configuration for a Dedicated Host

Use the describe-hosts command.

```
aws ec2 describe-hosts \
    --host-ids h-012a3456b7890cdef \
    --query Hosts[].HostRecovery
```

The following is example output.

```
on
```

PowerShell

To view the host recovery configuration for a Dedicated Host

Use the Get-EC2Host cmdlet.

```
(Get-EC2Host -HostId h-012a3456b7890cdef).Hosts | Select HostRecovery
```

The following is example output.

```
HostRecovery
------------
on
```

Manually recover instances that are not supported by Amazon EC2 Dedicated Host recovery

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 Amazon EC2 instances.

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.

Host maintenance for Amazon EC2 Dedicated Host

With host maintenance, in the rare event that a Dedicated Host becomes degraded, we automatically migrate instances running on it onto a healthy replacement Dedicated Host. This helps to minimize the downtime for your workload, and simplify the management of your Dedicated Hosts. Host maintenance is also performed for planned and routine Amazon EC2 maintenance.

Amazon EC2 supports two types of host maintenance:

- **Live migration host maintenance** — Instances are automatically migrated to the replacement host within 24 hours, without stopping and restarting them.
- **Reboot-based host maintenance** — Instances are scheduled for *instance reboot* scheduled events, during which they are automatically stopped and restarted on the replacement host.

Contents

Host maintenance versus host recovery

The following table shows the main differences between host recovery and host maintenance.

	Host recovery	Host maintenance
Instance reachability	Unreachable	Reachable
Dedicated Host state	`under-assessment`	`permanent-failure`
Host resource group	Supported	Not supported

For more information about host recovery, see Host recovery.

Considerations

- Host maintenance is available in all AWS Regions, except the China Regions and AWS GovCloud (US) Regions.
- Host maintenance is not supported in AWS Outposts, AWS Local Zones, and AWS Wavelength Zones.
- Host maintenance can't be turned on or off for hosts already within a host resource group. Hosts added to a host resource group retain their host maintenance setting. For more information, see Host resource groups.
- Host maintenance is not supported with the following instance types, because they have instance store-backed root volumes: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

Related services

Dedicated Host integrates with **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.

You must have sufficient licenses in your AWS account for your new Dedicated Host. The licenses associated with your degraded host are released when the host is released after the completion of the scheduled event.

Pricing

There are no additional charges for using host maintenance, but the usual Dedicated Host charges apply. For more information, see Amazon EC2 Dedicated Hosts Pricing.

As soon as host maintenance is initiated, you are no longer billed for the degraded Dedicated Host. Billing for the replacement Dedicated Host begins only after it enters the `available` state.

If the degraded Dedicated Host was billed using the On-Demand rate, the replacement Dedicated Host is also billed using the On-Demand rate. If the degraded Dedicated Host had an active Dedicated Host Reservation, it is transferred to the new Dedicated Host.

How host maintenance works for Amazon EC2 Dedicated Hosts

When a degradation is detected on a Dedicated Host that is enabled for host maintenance, we automatically allocate a replacement Dedicated Host in your account. The replacement Dedicated

Host receives a new host ID, but retains the same attributes as the original Dedicated Host, including:

- Auto placement settings
- Availability Zone
- Dedicated Host Reservation association
- Host affinity
- Host maintenance settings
- Host recovery settings
- Instance type
- Tags

After the replacement host has been allocated, we migrate the instances using either **live migration host maintenance** or **reboot-based host maintenance**, depending on the instance.

After the degraded host has no more running instances, it is permanently released from your account.

Live migration host maintenance

Instances that require live migration host maintenance are automatically migrated to the replacement host within 24 hours, without stopping and restarting them. The migrated instances retain their existing attributes, including:

- Instance ID
- Instance metadata
- Amazon EBS volume attachments
- Elastic IP addresses and private IP address
- Memory, CPU, and networking states

Some larger instance sizes might experience a slight performance decrease during the migration.

After the instances are automatically migrated to the replacement host, we send you email and AWS Health Dashboard notifications. Notifications include the IDs of the degraded and replacement hosts, information about the instances that were automatically migrated using live migration host maintenance, and information about the remaining instances.

Reboot-based host maintenance

Instances that require reboot-based host maintenance are scheduled for instance reboot scheduled events for 14 days from the date of the notification. You can continue to access your instances on the degraded Dedicated Host before the scheduled event.

You can reschedule reboot events for a date that is within 7 days of the original event date and time. For more information, see Reschedule a scheduled event for an EC2 instance.

Amazon EC2 automatically reserves capacity on the replacement host for these instances. You can't run instances in this reserved capacity.

The Amazon EC2 console shows the reserved capacity as used capacity. It could appear that the instances are running on both the degraded host and the replacement host. However, the instances will continue to run only on the degraded host until they are stopped or they are migrated into the reserved capacity on the replacement host.

At the date and time of the scheduled event, the instances are automatically stopped and restarted into the reserved capacity on the replacement host. The migrated instances retain their existing attributes, including:

- Instance ID
- Instance metadata
- Amazon EBS volume attachments
- Elastic IP addresses and private IP address

However, since the instances are stopped and restarted during the migration, they do not retain their memory, CPU, and networking states.

You can also manually stop and restart these instances at any time before the scheduled event to migrate them to the replacement host or to a different host. You might need to modify your instance's host affinity to restart it on a different host. If you stop an instance before the scheduled event, the reserved capacity on the replacement host is released and becomes available for use.

Host maintenance states

When a host becomes degraded, it enters the `permanent-failure` state. You can't launch instances on a Dedicated Host that is in the `permanent-failure` state.

After the replacement host is allocated, it remains in the `pending` state until the instances that support live migration host maintenance are automatically migrated from the degraded host, and until the scheduled events are scheduled for the remaining instances. After these tasks are completed, the replacement host enters the `available` state.

After the replacement host enters the `available` state, you can use it in the same way that you use any host in your account. However, some instance capacity on the replacement host is reserved for the instances that require reboot-based host migration. You can't launch new instances into this reserved capacity.

When the degraded host has no more running instances, it enters the `released, permanent-failure` state, and it is permanently released from your account. Note that the host and its resources remain visible in the console for a short time.

Automatic migration

Some instances can't be automatically migrated to the replacement host.

Instances with EBS-backed root volumes

For these instances, we schedule instance stop events for 28 days from the date of the notification. At the date and time of the scheduled event, the instances are stopped. We recommend that you manually stop on restart the instance on the replacement host or on a different host. You might need to modify your instance's host affinity to restart it on a different host.

Instances with instance store-backed root volumes

For these instances, we schedule instance retirement events for 28 days from the date of the notification. At the date and time of the scheduled event, the instances are permanently terminated. We recommend that you manually launch replacement instances on the replacement host and then migrate the required data to the replacement instances before the scheduled event.

The following instances have instance store-backed root volumes: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

You can continue to access your instances on the degraded Dedicated Host before the scheduled event.

Configure the host maintenance setting for an Amazon EC2 Dedicated Host

Enable host maintenance to ensure that your instances running on a Dedicated Host are automatically recovered onto a new Dedicated Host during a scheduled maintenance event.

If you disable host maintenance, you receive an email notification to evict the degraded host and manually migrate your instances to another host within 28 days. A replacement host is allocated if you have Dedicated Host reservation. After 28 days, the instances running on the degraded host are terminated, and the host is released automatically.

Console

To enable host maintenance for your 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 > **Actions** > **Modify host**.
4. Select *on* in the **Host maintenance** field.

To disable host maintenance for your 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 > **Actions** > **Modify host**.
4. Select *off* in the **Host maintenance** field.

AWS CLI

To enable host maintenance for your Dedicated Host

Use the modify-hosts command.

```
aws ec2 modify-hosts \
    --host-maintenance on \
    --host-ids h-0d123456bbf78910d
```

To disable host maintenance for your Dedicated Host

Use the modify-hosts command.

```
aws ec2 modify-hosts \
    --host-maintenance off \
    --host-ids h-0d123456bbf78910d
```

PowerShell

To enable host maintenance for your Dedicated Host

Use the Edit-EC2Host cmdlet.

```
Edit-EC2Host `
    -HostMaintenance on `
    -HostId h-0d123456bbf78910d
```

To disable host maintenance for your Dedicated Host

Use the Edit-EC2Host cmdlet.

```
Edit-EC2Host `
    -HostMaintenance off `
    -HostId h-0d123456bbf78910d
```

Monitor the state of your Amazon EC2 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 command.

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

To view the state of a Dedicated Host

Use the Get-EC2Host cmdlet.

```
Get-EC2Host -HostId h-012a3456b7890cdef
```

The following table explains the possible Dedicated Host states.

State	Description
`available`	AWS hasn't detected an issue with the Dedicated Host. No maintenance or repairs are scheduled. Instances can be launched onto this Dedicated Host.
`released`	The Dedicated Host has been released. The host ID is no longer in use. Released hosts can't be reused.
`under-assessment`	AWS is exploring a possible issue with the Dedicated Host. If action must be taken, you are notified via the AWS Management Console or email. Instances can't be launched onto a Dedicated Host in this state.
`pending`	The Dedicated Host can't be used for new instance launches. It is either being modified to support multiple instance types, or a host recovery is in progress.
`permanent-failure`	An unrecoverable failure has been detected. You receive an eviction notice through your instances and by email. Your instances might continue to run. If you stop or terminate all instances on a Dedicated Host with this state, AWS retires the host. AWS does not restart instances in this state. Instances can't be launched onto Dedicated Hosts in this state.
`released-permanent-failure`	AWS permanently releases Dedicated Hosts that have failed and no longer have running instances on them. The Dedicated Host ID is no longer available for use.

Track Amazon EC2 Dedicated Host configuration changes using AWS Config

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.
- 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.

Amazon EC2 Dedicated Instances

By default, EC2 instances run on shared tenancy hardware. This means that multiple AWS accounts might share the same physical hardware.

Dedicated Instances are EC2 instances that run on hardware that's dedicated to a single AWS account. This means that Dedicated Instances are physically isolated at the host hardware level from instances that belong to other AWS accounts, even if those accounts are linked to a single payer account. However, Dedicated Instances might share hardware with other instances from the same AWS account that are not Dedicated Instances.

Dedicated Instances provide no visibility or control over instance placement, and they do not support host affinity. If you stop and start a Dedicated Instance, it might not run on the same host. Similarly, you cannot target a specific host on which to launch or run an instance. Additionally, Dedicated Instances provide limited support for Bring Your Own License (BYOL).

If you require visibility and control over instance placement and more comprehensive BYOL support, consider using a Dedicated Host instead. Dedicated Instances and Dedicated Hosts can both be used to launch Amazon EC2 instances onto dedicated physical servers. There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, there are some key differences between them. The following table highlights some of the key differences between Dedicated Instances and Dedicated Hosts:

	Dedicated Host	Dedicated Instance
Dedicated physical server	Physical server with instance capacity fully dedicated to your use.	Physical server that's dedicated to a single customer account.
Instance capacity sharing	Can share instance capacity with other accounts.	Not supported

	Dedicated Host	Dedicated Instance
Billing	Per-host billing	Per-instance billing
Visibility of sockets, cores, and host ID	Provides visibility of the number of sockets and physical cores	No visibility
Host and instance affinity	Allows you to consistently deploy your instances to the same physical server over time	Not supported
Targeted instance placement	Provides additional visibility and control over how instances are placed on a physical server	Not supported
Automatic instance recovery	Supported. For more information, see Amazon EC2 Dedicated Host recovery.	Supported
Bring Your Own License (BYOL)	Supported	Partial support *
Capacity Reservations	Not supported	Supported

* Microsoft SQL Server with License Mobility through Software Assurance, and Windows Virtual Desktop Access (VDA) licenses can be used with Dedicated Instance.

For more information, see Amazon EC2 Dedicated Hosts.

Contents

- Launch Dedicated Instances into a VPC with default tenancy
- Change the tenancy of an EC2 instance
- Change the instance tenancy of a VPC

Dedicated Instance basics

A VPC can have a tenancy of either `default` or `dedicated`. By default, your VPCs have `default` tenancy and instances launched into a `default` tenancy VPC have `default` tenancy. To launch Dedicated Instances, do the following:

- Create a VPC with a tenancy of `dedicated`, so that all instances in the VPC run as Dedicated Instances. For more information, see Launch Dedicated Instances into a VPC with default tenancy.
- Create a VPC with a tenancy of `default` and manually specify a tenancy of `dedicated` for the instances to run as Dedicated Instances. For more information, see Launch Dedicated Instances into a VPC with default tenancy.

Supported features

Dedicated Instances support the following features and AWS service integrations:

Features

- Reserved Instances
- Automatic scaling
- Automatic recovery
- Dedicated Spot Instances
- Burstable performance instances

Reserved Instances

To reserve capacity for your Dedicated Instances, you can purchase Dedicated Reserved Instances or Capacity Reservations. For more information, see Reserved Instances for Amazon EC2 overview and Reserve compute capacity with EC2 On-Demand Capacity Reservations.

When you purchase a Dedicated Reserved Instance, you are purchasing the capacity to launch a Dedicated Instance 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 does 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

You can use Amazon EC2 Auto Scaling to launch Dedicated Instances. For more information, see Create a launch template using advanced settings in the *Amazon EC2 Auto Scaling User Guide*.

Automatic recovery

You can configure automatic recovery for a Dedicated Instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. For more information, see Automatic instance recovery.

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 Launch on single-tenant hardware.

Burstable performance instances

You can leverage the benefits of running on dedicated tenancy hardware with the section called "Burstable performance instances". 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".

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.

Dedicated Instances limitations

Keep the following in mind when using Dedicated Instances:

- Some AWS services or their features are not supported with a VPC with the instance tenancy set to `dedicated`. Refer to the respective service's documentation to confirm if there are any limitations.
- Some instance types can't be launched into a VPC with the instance tenancy set to `dedicated`. For more information about supported instance types, see Amazon EC2 Dedicated Instances.
- When you launch a Dedicated Instance backed by Amazon EBS, the EBS volume doesn't run on single-tenant hardware.

Pricing for Dedicated Instances

Pricing for Dedicated Instances is different from pricing for On-Demand Instances. For more information, see the Amazon EC2 Dedicated Instances.

Launch Dedicated Instances into a VPC with default tenancy

When you create a VPC, you have the option of specifying its instance tenancy. If you launch an instance into a VPC that has an instance tenancy of `dedicated`, it runs as a Dedicated Instance on hardware that's dedicated for your use.

For more information about launching an instance with a tenancy of `host`, see Launch Amazon EC2 instances on an Amazon EC2 Dedicated Host.

For more information about VPC tenancy options, see Create a VPC in the *Amazon VPC User Guide*.

Requirements

- Choose a supported instance type. For more information, see Amazon EC2 Dedicated Instances.

Console

To launch a Dedicated Instance into a default tenancy VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, **Launch instance**.
3. In the **Application and OS Images** section, select an AMI from the list.

4. In the **Instance type** section, select the instance type to launch.
5. In the **Key pair** section, select the key pair to associate with the instance.
6. In the **Advanced details** section, for **Tenancy**, select **Dedicated**.
7. Configure the remaining instance options as needed. For more information, see Reference for Amazon EC2 instance configuration parameters.
8. Choose **Launch instance**.

AWS CLI

To set the tenancy option for an instance during launch

Use the run-instances command and include `Tenancy` with the `--placement` option.

```
--placement Tenancy=dedicated
```

PowerShell

To set the tenancy option for an instance during launch

Use the New-EC2Instance cmdlet with the `-Placement_Tenancy` parameter.

```
-Placement_Tenancy dedicated
```

Change the tenancy of an EC2 instance

You can change the tenancy of a stopped instance after launch. The changes that you make take effect the next time the instance starts.

Alternatively, you can change the tenancy of your virtual private cloud (VPC). For more information, see the section called “Change the tenancy of a VPC”.

Limitations

- You can't change the tenancy of an instance using the AWS Management Console.
- The instance must be in the `stopped` state.
- The operating system details of your instance—and whether SQL Server is installed—affect what conversions are supported. For more information about the tenancy conversion paths available to your instance, see Tenancy conversion in the *License Manager User Guide*.

- For T3 instances, you must launch the instance on a Dedicated Host to use a tenancy of `host`. You can't change the tenancy from `host` to `dedicated` or `default`. Attempting to make one of these unsupported tenancy changes results in an `InvalidRequest` error code.

AWS CLI

To modify the tenancy value of an instance

Use the [modify-instance-placement](#) command.

```
aws ec2 modify-instance-placement \
    --instance-id i-1234567890abcdef0 \
    --tenancy dedicated
```

PowerShell

To modify the tenancy value of an instance

Use the [Edit-EC2InstancePlacement](#) cmdlet.

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
    -Tenancy Dedicated
```

Change the instance tenancy of a VPC

You can change the instance tenancy of a virtual private cloud (VPC) from `dedicated` to `default` after you create it. Modifying the instance tenancy of a VPC does not affect the tenancy of any existing instances in the VPC. The next time that you launch an instance in the VPC, it has a tenancy of `default`, unless you specify otherwise during instance launch.

Alternatively, you can change the tenancy of specific instances. For more information, see the section called "Change the tenancy of an instance".

Limitations

- You can't change the instance tenancy of a VPC from `default` to `dedicated` after it is created.
- You can't modify the instance tenancy of a VPC using the AWS Management Console.

AWS CLI

To modify the instance tenancy attribute of a VPC

Use the modify-vpc-tenancy command. The only supported tenancy value is `default`.

```
aws ec2 modify-vpc-tenancy \
    --vpc-id vpc-1234567890abcdef0 \
    --instance-tenancy default
```

PowerShell

To modify the instance tenancy attribute of a VPC

Use the Edit-EC2VpcTenancy cmdlet. The only supported tenancy value is `Default`.

```
Edit-EC2VpcTenancy `
    -VpcId vpc-1234567890abcdef0 `
    -InstanceTenancy Default
```

On-Demand Capacity Reservations and Capacity Blocks for ML

Capacity Reservations allow you to reserve compute capacity for Amazon EC2 instances in a specific Availability Zone. There are two types of Capacity Reservations serving different use cases.

Types of Capacity Reservations

- On-Demand Capacity Reservations
- Capacity Blocks for ML

The following are some common use cases for On-Demand Capacity Reservations:

- **Scaling events** – Create On-Demand Capacity Reservations ahead of your business-critical events to ensure that you can scale when you need to.
- **Regulatory requirements and disaster recovery** – Use On-Demand Capacity Reservations to satisfy regulatory requirements for high availability, and reserve capacity in a different Availability Zone or Region for disaster recovery.

The following are some common use cases for Capacity Blocks for ML:

- **Machine learning (ML) model training and fine-tuning –** Get uninterrupted access to the GPU instances that you reserved to complete ML model training and fine-tuning.
- **ML experiments and prototypes –** Run experiments and build prototypes that require GPU instances for short durations.

When to use On-Demand Capacity Reservation

Use On-Demand Capacity Reservations if you have strict capacity requirements, and your current or future business-critical workloads require capacity assurance. With On-Demand Capacity Reservations, you can ensure that you'll always have access to the Amazon EC2 capacity you've reserved for as long as you need it.

When to use Capacity Blocks for ML

Use Capacity Blocks for ML when you need to ensure that you have uninterrupted access to GPU instances for a defined period of time starting on a future date. Capacity Blocks are ideal for training and fine-tuning ML models, short experimentation runs, and handling temporary surges in inference demand in the future. With Capacity Blocks, you can ensure that you'll have access to GPU resources on a specific date to run your ML workloads.

Reserve compute capacity with EC2 On-Demand Capacity Reservations

Amazon EC2 Capacity Reservations allow you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. If you have strict capacity requirements for current or future business-critical workloads that require a certain level of long or short-term capacity assurance, we recommend that you create a Capacity Reservation to help ensure that you always have access to Amazon EC2 capacity when you need it, for as long as you need it.

You can create a Capacity Reservation at any time, and you can choose when it starts. You can request a Capacity Reservation for immediate use or you can request a Capacity Reservation for a future date.

- If you request a **Capacity Reservation for immediate use**, the Capacity Reservation becomes available for use immediately and there is no term commitment. You can modify the Capacity Reservation at any time, and you can cancel it at any time to release the reserved capacity and to stop incurring changes.
- If you request a **future-dated Capacity Reservation**, you specify the future date at which you need the Capacity Reservation to become available for use. You must also specify a commitment

duration for which you commit to keeping the requested capacity in your account after the specified date. At the requested date and time, the Capacity Reservation becomes available for use and the commitment duration starts. During the commitment duration, you can't decrease the instance count or commitment duration below your initial commitment, or cancel the Capacity Reservation. After the commitment duration elapses, you can modify the Capacity Reservation in any way or cancel it if you no longer need it.

Capacity Reservations can only be used by instances that match their attributes. By default, Capacity Reservations automatically match new instances and running instances that have matching attributes (instance type, platform, Availability Zone, and tenancy). 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 allows you to explicitly control which instances are allowed to run in that reserved capacity. You can also specify that instances will only run in a Capacity Reservation or Capacity Reservation resource group.

Important

Future-dated Capacity Reservations are for helping you launch and cover incremental instances, and not to cover existing running instances. If you need to cover existing running instances, use Capacity Reservations that start immediately instead.

All supported Amazon EC2 instances with matching attributes, that is instance type, platform, Availability Zone, and tenancy, are eligible to run in a Capacity Reservation. The Amazon EC2 instance can be launched by you (*non-managed instances*) or on your behalf by an AWS service (*managed instances*). This is particularly true for *open* Capacity Reservations, which automatically match with any running instances that have matching attributes. For example, managed instances launched on your behalf by the following services are eligible to run in Capacity Reservations that you create and manage.

- Amazon EC2 Auto Scaling
- Amazon ECS
- Amazon EKS
- Amazon EMR
- Amazon SageMaker AI
- AWS Batch

- AWS Elastic Beanstalk
- AWS ParallelCluster
- AWS Parallel Computing Service (AWS PCS)

Contents

Concepts for Amazon EC2 Capacity Reservations

The following key concepts apply to Capacity Reservations.

Topics

Start date and time

The start date and time defines when the Capacity Reservation becomes available for use. A Capacity Reservation can start **immediately**, or it can start at a **future date**.

- If you choose to start a Capacity Reservation immediately, the reserved capacity becomes available for use immediately after you create it, and billing starts as soon as the Capacity Reservation enters the active state. You do not need to enter into any term commitments. You can modify the Capacity Reservation as needed at any time to meet your needs, and you can cancel it at any time to release the capacity and to stop incurring charges.
- If you choose to start a Capacity Reservation at a future date, you specify a *future date and time* at which you will need the reserved capacity, and a *commitment duration*, which is the minimum duration for which you commit to keeping the requested Capacity Reservation in your account after it has been provisioned. At the specified future date, the Capacity Reservation becomes available for use and billing starts at that time, once the Capacity Reservation enters the active state. The commit duration starts as soon as the Capacity Reservation is provisioned in your account. During this time, you can't decrease the instance count below the committed instance count, choose an end date that is before the commitment duration, or cancel the Capacity Reservation. However, after the commitment duration lapses, you are free to modify the Capacity Reservation in any way, or cancel it to release the reserved capacity and to stop incurring charges.

End date and time

The end date and time defines when the Capacity Reservation ends and the reserved capacity is released from your account. You can configure a Capacity Reservation to **end automatically** at a specific date and time, or you can configure it to stay active indefinitely until you **manually cancel it**.

If you configure a Capacity Reservation to end automatically, the Capacity Reservation expires 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, the reserved capacity is released from your account and 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 can't restart them until you remove their Capacity Reservation targeting preference or configure them to target a different Capacity Reservation. For more information, see [Modify the Capacity Reservation settings of your instance](#).

Commitment duration

The commitment duration applies to future-dated Capacity Reservations only.

The commitment duration is a minimum duration for which you commit to having the future-dated Capacity Reservation in the active state in your account after it has been provisioned. You can keep a future-dated Capacity Reservation for longer than the commitment duration, but not shorter. The following apply during the commitment duration:

- You can't cancel a Capacity Reservation during the commitment duration.
- You can't decrease the instance count below the committed instance count, but you can increase it.
- You can't configure a Capacity Reservation to automatically end at a date or time that is within the commitment duration. You can extend the end date and time during the commitment period.

Amazon EC2 uses the commitment duration that you specify to assess whether the request can be supported. The minimum commitment duration is 14 days. While assessing a request, Amazon EC2 might determine that it can support a shorter commitment duration. In that case, Amazon EC2 will schedule the future-dated Capacity Reservation with the shorter commitment duration. This means that you are committed to keeping the Capacity Reservation in your account for a shorter period than you initially requested.

Future-dated Capacity Reservation assessment

When you request a future-dated Capacity Reservation, Amazon EC2 assesses the request to determine whether it can be supported based on capacity availability and the commitment duration you specify. The assessment is typically completed within 5 days. Amazon EC2 considers multiple factors when evaluating a request, including:

- Forecasted capacity supply
- The commitment duration
- How early you request the Capacity Reservation relative to your start date
- The size of your request

You can request a future-dated Capacity Reservation between 5 and 120 days in advance. We recommend that you make the request at least 56 days (8 weeks) in advance to improve our ability to support your request. The minimum commitment duration is 14 days and the minimum instance count is 64 vCPUs.

The Capacity Reservation remains in the `assessing` state while the request is being assessed.

If the request can be supported, the Capacity Reservation enters the `scheduled` state and it is scheduled for delivery on the requested date and time. The total instance count remains 0 during while the Capacity Reservation is in the `scheduled` state. A scheduled Capacity Reservation will become `active` and available for use at the requested date.

If a request can't be supported, the Capacity Reservation enters the `unsupported` state. Unsupported Capacity Reservations are not delivered.

You can cancel a future-dated Capacity Reservation while it is in the `assessing` state.

For more information, see [Create a future-dated Capacity Reservation].

Capacity Reservation attributes

When you create a Capacity Reservation, you must specify the following attributes:

- Availability Zone
- Instance type
- Platform (operating system type)
- Tenancy (`default` or `dedicated`)

Only instances that match these attributes can launch or run in the Capacity Reservation.

Instance matching criteria

Instance matching criteria, or instance eligibility, determines which instances are allowed to launch and run in the Capacity Reservation. A Capacity Reservation can have one of the following matching criteria:

- **Open** — The Capacity Reservation automatically matches all instances that have matching attributes (instance type, platform, and Availability Zone). New and existing instances that have matching attributes automatically run in the Capacity Reservation without any additional configuration.
- **Targeted** — The Capacity Reservation accepts only instances that have matching attributes (instance type, platform, and Availability Zone), and that explicitly target the Capacity Reservation. The instance must specifically target the Capacity Reservation to launch or run in its reserved capacity. This allows you to explicitly control which instances are allowed to run in the reserved capacity and helps you avoid unintentional reserved capacity usage.

When you request a future-dated Capacity Reservation, you can specify only targeted matching criteria. This ensures that the capacity delivered by the Capacity Reservation is incremental, or additional, to any running instances or reserved capacity that you have at the time of delivery. After the Capacity Reservation becomes active in your account, you can change the instance matching criteria to open if needed. However, keep in mind that any matching instances will automatically run in the Capacity Reservation, which could lead to unintentional capacity usage and prevent you from launching new instances for the full requested instance count.

Differences between Capacity Reservations, Reserved Instances, and Savings Plans

The following table highlights key differences between Capacity Reservations, Reserved Instances, and Savings Plans:

	Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
Term	No commitment required for immediate-use Capacity Reservati	Requires a fixed one-year or three-year commitment		

	Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
	ons. They can be created, modified, and canceled as needed. With future-dated Capacity Reservations, you specify a commitment duration for which you commit to keeping the capacity in your account. After the commitment duration elapses, you can cancel the Capacity Reservation at any time.			
Capacity benefit	Capacity reserved in a specific Availability Zone.		No capacity reserved.	
Billing discount	No billing discount. †	Provides a billing discount.		
Instance Limits	Your On-Demand Instance limits per Region apply.	Default is 20 per Availability Zone. You can request a limit increase.	Default is 20 per Region. You can request a limit increase.	No limit.

† You can combine Capacity Reservations with Savings Plans or Regional Reserved Instances to receive a discount.

For more information, see the following:

- Reserved Instances for Amazon EC2 overview
- 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 values for `platform`:

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux
- RHEL with SQL Server Standard
- RHEL with SQL Server Enterprise
- RHEL with SQL Server Web
- RHEL with HA
- RHEL with HA and SQL Server Standard
- RHEL with HA and SQL Server Enterprise
- Ubuntu Pro
- Windows
- Windows with SQL Server
- Windows with SQL Server Web
- Windows with SQL Server Standard
- Windows with SQL Server Enterprise

To ensure that an instance runs in a specific Capacity Reservation, the platform of the Capacity Reservation must match the platform of the AMI used to launch the instance. For Linux AMIs, it is important to check whether the AMI platform uses the general value **Linux/UNIX** or a more specific value like **SUSE Linux**.

Console

To check the AMI platform

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select the AMI.
4. On the **Details** tab, note the value of **Platform details**.

AWS CLI

To check the AMI platform

Use the describe-images command and check the value of `PlatformDetails`.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

The following is example output.

```
[
    "Linux/UNIX"
]
```

PowerShell

To check the AMI platform

Use the Get-EC2Image cmdlet and check the value of `PlatformDetails`.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

The following is example output.

```
PlatformDetails
---------------
Linux/UNIX
```

Quotas

The number of instances for which you are allowed to reserve capacity is based on your account's On-Demand Instance quota. You can reserve capacity for as many instances as that quota allows, minus the number of instances that are already running.

Capacity Reservations in the `assessing`, `scheduled`, `pending` , `active`, and `delayed` state count towards your On-Demand Instance quota.

Limitations

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 Shared Capacity Reservations.
- Zonal Reserved Instance billing discounts do not apply to Capacity Reservations.
- Capacity Reservations can be created in cluster placement groups. Spread and partition placement groups are not supported.
- Capacity Reservations can't be used with Dedicated Hosts. Capacity Reservations can be used with Dedicated Instances.
- [Windows instances] Capacity Reservations can't be used with Bring Your Own License (BYOL).
- [Red Hat instances] Capacity Reservations can 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.
- You can request future-dated Capacity Reservations for an instance count with a minimum of 64 vCPUs. For example, if you request a future-dated Capacity Reservation for `m5.xlarge` instances, you must request at least 16 instances (*16 * m5.xlarge = 64 vCPUs*).
- You can request future-dated Capacity Reservations for instance types in the following series only: C, G, I, M, R, and T.

Capacity Reservation pricing and billing

The topics in this section provide an overview of pricing and billing for Capacity Reservations.

Topics

- Pricing
- Billing
- Billing discounts
- Viewing your bill

Pricing

Capacity Reservations are charged at the equivalent On-Demand rate whether you run instances in reserved capacity or not. If you do not use the reservation, this shows up as unused reservation on your Amazon 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.

For more information, see Amazon EC2 Pricing.

Billing

Billing starts as soon as the Capacity Reservation is provisioned in your account, and it continues while the Capacity Reservation remains provisioned in your account. For future-dated Capacity Reservations, this means that billing starts only once the Capacity Reservation is provisioned in your account at the requested future date.

Capacity Reservations are billed at per-second granularity. This means that you are charged for partial hours. For example, if a Capacity Reservation remains provisioned in your account for `24` hours and `15` minutes, you are 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 provisioned 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.

Hour	1	2	3	4	5	6	Total cost
Unused Capacity Reservation	$0.10	$0.00	$0.00	$0.00	$0.00	$0.00	$0.10
On-demand Instance Usage	$0.00	$0.10	$0.10	$0.10	$0.10	$0.10	$0.50
Hourly cost	$0.10	$0.10	$0.10	$0.10	$0.10	$0.10	$0.60

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 for Amazon EC2 overview
- Savings Plans User Guide
- Billing and purchase options

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.

Create a Capacity Reservation

You can create a Capacity Reservation at any time to ensure that you have compute capacity available in a specific Availability Zone. A Capacity Reservation can start immediately, or it can start at a future date. The capacity becomes available for use only once the Capacity Reservation enters the `active` state.

> **Note**
>
> If you create a Capacity Reservation with `open` instance matching criteria, and you have running instances with matching attributes at the time the Capacity Reservation becomes active, those instances automatically run in the reserved capacity. To avoid this, use `targeted` instance matching criteria. For more information, see [Instance matching criteria](#).

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 request. 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 quotas](#).

Topics

Create a Capacity Reservation for immediate use

You create a Capacity Reservation for immediate use.

Console

To create a Capacity Reservation

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Capacity Reservations**, and then choose **Create Capacity Reservation**.

3. Configure the following settings in the **Instance details** section.

   a. **Instance Type** — The instance type for which to reserve capacity.

   b. **Platform** — The operating system for your instances. For more information, see Supported platforms.

   c. **Availability Zone** — The Availability Zone in which to reserve the capacity.

   d. **Tenancy** — The type of tenancy to use for the reserved capacity. Choose Default to reserve capacity on shared hardware, or Dedicated to reserve capacity on hardware that is dedicated to your account.

   e. (*Optional*) **Placement group ARN** — The ARN of the cluster placement group in which to create the Capacity Reservation. For more information, see Use Capacity Reservations with cluster placement groups.

   f. **Total instance count** — The number of instances for which to reserve capacity. If you specify a quantity that exceeds your remaining On-Demand Instance quota for the selected instance type, the request fails.

4. Configure the following settings in the **Reservation details** section:

   a. **Capacity Reservation starts** — Choose **Immediately**.

   b. **Capacity 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.

   c. **Instance eligibility** — Choose one of the following options:

      - **open** — (Default) The Capacity Reservation matches any instance that has matching attributes (instance type, platform, Availability Zone, and tenancy). 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, Availability Zone, and tenancy), and that explicitly target the reservation.

5. Choose **Create**.

AWS CLI

To create a Capacity Reservation

Use the create-capacity-reservation command.

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria open|targeted
```

PowerShell

To create a Capacity Reservation

Use the Add-EC2CapacityReservation cmdlet.

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion open|targeted
```

Create a future-dated Capacity Reservation

Request a future-dated Capacity Reservation if you need the reserved capacity to become available at a future date and time.

After you request a future-dated Capacity Reservation, the request undergoes an assessment to determine whether it can be supported. For more information, see Future-dated Capacity Reservation assessment.

Considerations

- You can request future-dated Capacity Reservations for instance types in the following series: C, G, I, M, R, and T.

- You can request future-dated Capacity Reservations for an instance count with a minimum of 64 vCPUs. For example, if you request a future-dated Capacity Reservation for `m5.xlarge` instances, you must request capacity for at least 16 instances (*16 * m5.xlarge = 64 vCPUs*).
- You can request a future-dated Capacity Reservation between 5 and 120 days in advance. However, we recommend that you request it at least 56 days (8 weeks) in advance to improve supportability.
- The minimum commitment duration is 14 days.

Console

To create a Capacity Reservation

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Capacity Reservations**, and then choose **Create Capacity Reservation**.
3. Configure the following settings in the **Instance details** section.
   a. **Instance Type** — The instance type for which to reserve capacity.
   b. **Platform** — The operating system for your instances. For more information, see Supported platforms.
   c. **Availability Zone** — The Availability Zone in which to reserve the capacity.
   d. **Tenancy** — The type of tenancy to use for the reserved capacity. Choose Default to reserve capacity on shared hardware, or Dedicated to reserve capacity on hardware that is dedicated to your account.
   e. **Total instance count** — The number of instances for which to reserve capacity. If you specify a quantity that exceeds your remaining On-Demand Instance quota for the selected instance type, the request fails.
4. Configure the following settings in the **Reservation details** section:
   a. **Capacity Reservation starts** — Choose **At a specific time**.
   b. **Start date** — Specify the date and time at which the Capacity Reservation must become availble for use. For more information, see Start date and time.
   c. **Commitment duration** — Specify the minimum duration for which you commit keeping the Capacity Reservation after it has been delivered. For more information, see Commitment duration.
   d. **Capacity Reservation ends** — Choose one of the following options:

- **When I cancel it** — Reserve the capacity until you explicitly cancel it.
- **Specific time** — Cancel the capacity reservation automatically at the specified date and time.

5. Choose **Create**.

AWS CLI

To create a Capacity Reservation

Use the create-capacity-reservation command.

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria targeted \
    --delivery-preference incremental \
    --commitment-duration commitment_in_seconds \
    --start-date YYYY-MMDDThh:mm:ss.sssZ
```

PowerShell

To create a Capacity Reservation

Use the Add-EC2CapacityReservation cmdlet.

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion targeted `
    -DeliveryPreference incremental `
    -CommitmentDuration commitment_in_seconds `
    -StartDate  YYYY-MMDDThh:mm:ss.sssZ
```

View the state of a Capacity Reservation

Amazon EC2 constantly monitors the state of your Capacity Reservations.

Due to the eventual consistency model followed by the Amazon EC2 API, after you create a Capacity Reservation, it can take up to 5 minutes for the state of the Capacity Reservation to reflect that it is `active`. During this time, the Capacity Reservation might remain in the `pending` state. However, it might already be available for use, in which case attempts to launch instances into the Capacity Reservation would succeed.

Console

To view your Capacity Reservations

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.

AWS CLI

To describe your Capacity Reservations

Use the describe-capacity-reservations command.

For example, the following command describes all Capacity Reservations.

```
aws ec2 describe-capacity-reservations
```

The following is example output.

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE",
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-16T09:03:18.000Z",
            "AvailableInstanceCount": 1,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 1,
            "State": "active",
```

```
                "Tenancy": "default",
                "EbsOptimized": true,
                "InstanceType": "a1.medium",
                "PlacementGroupArn": "arn:aws:ec2:us-east-1:123456789012:placement-
group/MyPG"
            },
            {
                "CapacityReservationId": "cr-abcdEXAMPLE9876ef",
                "EndDateType": "unlimited",
                "AvailabilityZone": "eu-west-1a",
                "InstanceMatchCriteria": "open",
                "Tags": [],
                "EphemeralStorage": false,
                "CreateDate": "2019-08-07T11:34:19.000Z",
                "AvailableInstanceCount": 3,
                "InstancePlatform": "Linux/UNIX",
                "TotalInstanceCount": 3,
                "State": "cancelled",
                "Tenancy": "default",
                "EbsOptimized": true,
                "InstanceType": "m5.large"
            }
        ]
}
```

PowerShell

To describe a Capacity Reservation

Use the Get-EC2CapacityReservation cmdlet.

```
Get-EC2CapacityReservation `
    -CapacityReservationId cr-1234abcd56EXAMPLE
```

Capacity Reservation states

Capacity Reservations have the following possible states.

State	Description
`active`	The capacity is available for use.

State	Description
`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 request parameters that are not valid, capacity constraints, or instance limit constraints. You can view a failed request for 60 minutes.
`scheduled`	(*Future-dated Capacity Reservations only*) The future-dated Capacity Reservation request was approved and the Capacity Reservation is scheduled for delivery on the requested start date.
`assessing`	(*Future-dated Capacity Reservations only*) Amazon EC2 is assessing your request for a future-dated Capacity Reservation. For more information, see Future-dated Capacity Reservation assessment.
`delayed`	(*Future-dated Capacity Reservations only*) Amazon EC2 encountered a delay in provisioning the requested future-dated Capacity Reservati on. Amazon EC2 is unable to deliver the requested capacity by the requested start date and time.
`unsupported`	(*Future-dated Capacity Reservations only*) Amazon EC2 can't support the future-dated Capacity Reservation request due to capacity constraints. You can view unsupported requests for 30 days. The Capacity Reservation will not be delivered.

Launch instances into an existing Capacity Reservation

You can only launch an instance into a Capacity Reservation that:

- Has matching attributes (instance type, platform, Availability Zone, and tenancy)

- Has sufficient available capacity
- Is in the `active` state

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.

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.

Console

To launch instances into an existing Capacity Reservation

1. Follow the procedure to launch an instance, but don't launch the instance until you've completed the following steps to specify the settings for the placement group and Capacity Reservation.
2. Expand **Advanced details** and do the following:
    a. For **Placement group**, select the cluster placement group in which to launch the instance.
    b. For **Capacity Reservation**, choose one of the following options depending on the configuration of the Capacity Reservation:
        - **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.
        - **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 resource 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.
- **Specify Capacity Reservation only –** Launches the instances into a Capacity Reservation. If a Capacity Reservation ID isn't specified, the instances launch into an open Capacity Reservation. If capacity isn't available, the instances fail to launch.
- **Specify Capacity Reservation resource group only –** Launches the instances into a Capacity Reservation in a Capacity Reservation resource group. If a Capacity Reservation resource group ARN isn't specified, the instances launch into an open Capacity Reservation. If capacity isn't available, the instances fail to launch.

3. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**.

AWS CLI

To launch an instance into an existing Capacity Reservation

Use the run-instances command and specify the `--capacity-reservation-specification` option.

The following example launches an instance into any open Capacity Reservation with matching attributes and available capacity:

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification CapacityReservationPreference=open
```

The following example launches an instance into a `targeted` Capacity Reservation:

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
```

```
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

The following example launches an instance into the specified Capacity Reservation group:

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

The following example launches an instance into a Capacity Reservation only. Because it does not specify a Capacity Reservation ID, the instance launches in any open Capacity Reservation with matching attributes and available capacity:

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only
```

The following example launches an instance into a specific Capacity Reservation only. If capacity isn't available in the specified Capacity Reservation, the instance fails to launch.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
```

```
        CapacityReservationPreference=capacity-reservations-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

PowerShell

To launch an instance into an existing Capacity Reservation

Use the New-EC2Instance cmdlet.

The following example launches an instance into any open Capacity Reservation with matching attributes and available capacity:

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "open"
```

The following example launches an instance into a `targeted` Capacity Reservation:

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

The following example launches an instance into the specified Capacity Reservation group:

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

The following example launches an instance into a Capacity Reservation only. Because it does not specify a Capacity Reservation ID, the instance launches in any open Capacity Reservation with matching attributes and available capacity:

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-reservations-only"
```

The following example launches an instance into a specific Capacity Reservation only. If capacity isn't available in the specified Capacity Reservation, the instance fails to launch.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-reservations-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

Modify an active Capacity Reservation

If you have an existing Capacity Reservation which isn't a good fit for the workload that needs the capacity, you can modify the instance quantity, instance eligibility (`open` or `targeted`), and end time (`At specific time` or `Manually`). If you specify a new instance quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the update fails.

The allowed modifications depend on the state of the Capacity Reservation:

- `assessing` or `scheduled` state — You can modify the tags only.
- `pending` state — You can't modify the Capacity Reservation in any way.
- `active` state but still within the commitment duration — You can't decrease the instance count below the committed instance count, or set an end date that is before the committed duration. All other modifications are allowed.
- `active` state with no commitment duration or elapsed commitment duration — All modifications are allowed.
- `expired`, `cancelled`, `unsupported`, or `failed` state — You can't modify the Capacity Reservation in any way.

Considerations

- You can't change the instance type, platform, Availability Zone, or tenancy after creation. 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 modify an existing Capacity Reservation by changing the instance eligibility from `targeted` to `open`, any running instances that match the attributes of the Capacity Reservation, have the `CapacityReservationPreference` parameter set to `open`, and are not yet running in a Capacity Reservation, will automatically use the modified Capacity Reservation.
- To change the instance eligibility, the Capacity Reservation must be completely idle (zero usage) because Amazon EC2 can't modify instance eligibility when instances are running inside the reservation.

Console

To modify a Capacity Reservation

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Capacity Reservations**, select the Capacity Reservation to modify, and then choose **Edit**.
3. Modify the **Total capacity**, **Capacity Reservation ends**, or **Instance eligibility** options as needed, and choose **Save**.

AWS CLI

To modify a Capacity Reservation

Use the [modify-capacity-reservation](https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-capacity-reservation.html) command. The following example modifies the specified Capacity Reservation to reserve capacity for eight instances.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --instance-count 8
```

PowerShell

To modify a Capacity Reservation

Use the Edit-EC2CapacityReservation cmdlet. The following example modifies the specified Capacity Reservation to reserve capacity for eight instances.

```
Edit-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0 `
    -InstanceCount 8
```

Modify the Capacity Reservation settings of your instance

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, Availability Zone, and tenancy) and available capacity.
- Start the instance in a specific Capacity Reservation.
- Start 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.

Console

To modify instance Capacity Reservation settings

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Instances** and select the instance to modify. Stop the instance if it is not already stopped.
3. Choose **Actions**, **Instance settings**, **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.
- **Specify Capacity Reservation only –** Launches the instances into a Capacity Reservation. If a Capacity Reservation ID isn't specified, the instances launch into an open Capacity Reservation. If capacity isn't available, the instances fail to launch.
- **Specify Capacity Reservation resource group only –** Launches the instances into a Capacity Reservation in a Capacity Reservation resource group. If a Capacity Reservation resource group ARN isn't specified, the instances launch into an open Capacity Reservation. If capacity isn't available, the instances fail to launch.

AWS CLI

To modify instance Capacity Reservation settings

Use the modify-instance-capacity-reservation-attributes command.

The following example changes the Capacity Reservation preference to none.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=none
```

The following example the target to a specific Capacity Reservation.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
    CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

The following example changes the target to 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-2:123456789012:group/my-cr-group}
```

The following example changes the Capacity Reservation preference to `capacity-reservation-only`. Because it doesn't specify a Capacity Reservation, instances launch into any open Capacity Reservation with matching attributes and available capacity.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=capacity-
reservation-only
```

The following example changes the Capacity Reservation preference to `capacity-reservation-only` and changes the target to a specific Capacity Reservation. If capacity isn't available in the specified Capacity Reservation, the instances fail to launch.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservation-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

PowerShell

To modify instance Capacity Reservation settings

Use the Edit-EC2InstanceCapacityReservationAttribute cmdlet.

The following example changes the Capacity Reservation preference to none.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "none"
```

The following example the target to a specific Capacity Reservation.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
```

```
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

The following example changes the target to a specific Capacity Reservation group.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

The following example changes the Capacity Reservation preference to `capacity-reservation-only`. Because it doesn't specify a Capacity Reservation, instances launch into any open Capacity Reservation with matching attributes and available capacity.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only"
```

The following example changes the Capacity Reservation preference to `capacity-reservation-only` and changes the target to a specific Capacity Reservation. If capacity isn't available in the specified Capacity Reservation, the instances fail to launch.

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

Move capacity between Capacity Reservations

You can move capacity from one Capacity Reservation to another to redistribute your reserved compute resources as needed. For example, if you need additional capacity in a reservation with growing usage, and you have capacity available in another reservation, then you can reallocate that capacity between the two reservations.

Prerequisites for moving capacity

As a prerequisite, the two Capacity Reservations must meet the following requirements:

- Both reservations must be in the active state.

- Both reservations must be owned by your AWS account. You cannot move capacity between reservations owned by different AWS accounts.
- Both reservations must have the same:
  - Instance type
  - Platform
  - Availability Zone
  - Tenancy
  - Placement group
  - End time

The destination Capacity Reservations instance eligibility (`open` or `targeted`), and tags, don't have to match the source reservation. The configuration of both reservations remains the same, except that the source reservation has reduced capacity and the destination reservation has increased capacity.

When you specify the quantity of instances to be moved, by default, any available capacity is moved first, followed by any eligible running instances (the used capacity in your reservation). For example, if you move 4 instances from a reservation with 5 used instances and 3 available instances, then the 3 available instances and 1 used instance will be moved.

> **Note**
>
> When you move used capacity from your reservation by specifying a **Quantity to move** that's greater than the available capacity, only the instances that were launched with their **Capacity Reservation Specification** as `open` will be moved.

Considerations

The following considerations apply when moving capacity from one reservation to another:

- The used capacity can only be moved between Capacity Reservations with `open` instance eligibility that are shared with the same set of accounts.
- When you move used capacity, the eligible instances are randomly selected. You cannot specify which running instances are moved. If a sufficient number of eligible instances are not found to fulfill the move quantity, the move operation will fail.

- If you move all of the capacity from the source reservation, the Capacity Reservation will be automatically canceled.
- **Future-dated Capacity Reservations** — You can't move capacity for a future-dated Capacity Reservation during the commitment period.

> **Note**
>
> Moving capacity from a Capacity Block isn't supported.

Move capacity

You can move capacity from a source Capacity Reservation to a destination Capacity Reservation.

Console

To move capacity

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Capacity Reservations**.
3. Select an On-Demand Capacity Reservation ID that has capacity to move.
4. Under **Actions**, **Manage capacity**, choose **Move**.
5. On the **Move capacity** page, under **Destination Capacity Reservation**, select a reservation from the list.
6. Under **Quantity to move**, use the slider or type the number of instances to move from the source Capacity Reservation to the destination Capacity Reservation.
7. Review the summary, and when you're ready, choose **Move**.

AWS CLI

To move capacity

Use the move-capacity-reservation-instances command. The following example moves 10 instances from the specified source Capacity Reservation to the specified destination Capacity Reservation.

```
aws ec2 move-capacity-reservation-instances \
```

```
    --source-capacity-reservation-id cr-1234567890abcdef0 \
    --destination-capacity-reservation-id cr-021345abcdef56789 \
    --instance-count 10
```

PowerShell

To move capacity

Use the [Move-EC2CapacityReservationInstance](#) cmdlet. The following example moves 10 instances from the specified source Capacity Reservation to the specified destination Capacity Reservation.

```
Move-EC2CapacityReservationInstance `
    -SourceCapacityReservationId cr-1234567890abcdef0 `
    -DestinationCapacityReservationId cr-021345abcdef56789 `
    -InstanceCount 10
```

Split off capacity from an existing Capacity Reservation

You can split off capacity from an existing Capacity Reservation to create a new reservation. By splitting capacity, you allocate a portion of the original reservation to a specific workload or share it with another AWS account. For example, to partially share a Capacity Reservation with another account, you can split off some of the capacity to create a smaller sized Capacity Reservation. The smaller sized Capacity Reservation can then be shared with the other accounts using AWS Resource Access Manager.

When you split capacity from an existing Capacity Reservation, a new Capacity Reservation is automatically created. The existing reservation will be unchanged, except for the reduced total capacity from the number of instances split off. Instances that are running in the existing Capacity Reservation are not affected. You can split the existing reservation into only one new Capacity Reservation.

The new Capacity Reservation will have the same configuration as the existing Capacity Reservation except for tags. By default, the new Capacity Reservation doesn't have any tags. You can specify new tags during the split operation. The new Capacity Reservation can also be modified after it is created, if necessary.

When you specify the quantity of instances to be split, by default, any available capacity is split first, followed by any eligible running instances (the used capacity in your reservation). For example: if you split 4 instances from a Capacity Reservation with 5 used instances and 3 available instances, then the 3 available instances and 1 used instance will be split into a new reservation.

Prerequisites for splitting capacity

As a prerequisite, your Capacity Reservation must meet the following requirements:

- The source reservation must be in the active state.
- The source reservation must be owned by your AWS account.

> **Note**
>
> When you split used capacity from your reservation by specifying a **Quantity to split** that's greater than the available capacity, only the instances that were launched with their **Capacity Reservation Specification** as open will be split.

Considerations

The following considerations apply when splitting off capacity from one reservation to a new one:

- The used capacity can only be split for Capacity Reservations with “open” instance eligibility that are not shared with any account.
- When you split the used capacity, the eligible instances are randomly selected. You cannot specify which running instances are split. If a sufficient number of eligible instances are not found to fulfill the split quantity, the split operation will fail.
- The maximum quantity of instances to split from an existing reservation is the size of the reservation minus one. For example, if your reservation’s total capacity is 5 instances, you can split a maximum of 4 instances into a new reservation.
- **Future-dated Capacity Reservations** – You can't split capacity for a future-dated Capacity Reservation during the commitment period.
- **Resource groups** – If the existing Capacity Reservation belongs to a resource group, the new Capacity Reservation will not be automatically added to the resource group. You can add the new Capacity Reservation to a resource group after it is created, if necessary.
- **Sharing** – If the existing Capacity Reservation is shared with a consumer account, the new Capacity Reservation will not be automatically shared with the consumer account. You can share the new Capacity Reservation after it is created, if necessary.

- **Cluster placement group** – If the existing Capacity Reservation is part of a cluster placement group, the new Capacity Reservation will be created in the same cluster placement group.

> **Note**
>
> Splitting capacity from a Capacity Block isn't supported.

Control access for splitting Capacity Reservations using tags

You can use tags to control access to Amazon EC2 resources, including splitting capacity from an existing Capacity Reservation to create a new Capacity Reservation. For more information, see [Controlling access to AWS resources using tags]() in the *IAM User Guide*.

To control access for splitting a Capacity Reservation using tags, make sure that you specify both resource and request tags in the policy statement because IAM policies are evaluated against both the source Capacity Reservation and the newly created Capacity Reservation. The following example policy includes the `ec2:ResourceTag` condition key with the tag `Owner=ExampleDepartment1` for the source Capacity Reservation and the `ec2:RequestTag` condition key with the tag `stack=production` for the newly created Capacity Reservation.

```
{
  "Statement": [
    {
      "Sid": "AllowSourceCapacityReservation",
      "Effect": "Allow",
      "Action": "ec2:CreateCapacityReservationBySplitting",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:capacity-reservation/
cr-1234567890abcdef0",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Owner": "ExampleDepartment1"
        }
      }
    },
    {
      "Sid": "AllowNewlyCreatedCapacityReservation",
      "Effect": "Allow",
      "Action": ["ec2:CreateCapacityReservationBySplitting", "ec2:CreateTags"],
```

```
      "Resource": "arn:aws:ec2:us-east-1:111122223333:capacity-reservation/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stack": "production"
        }
      }
    }
  ]
}
```

Split off capacity

You can split off capacity from an existing Capacity Reservation to create a new Capacity Reservation.

Console

To split off capacity

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Capacity Reservations**.
3. Select an On-Demand Capacity Reservation ID that has capacity to split.
4. Under **Actions**, **Manage capacity**, choose **Split**.
5. On the **Split Capacity Reservation** page, under **Quantity to split**, use the slider or type the number of instances to split from the current reservation.
6. (Optional) Add tags for the new Capacity Reservation.
7. Review the summary, and when you're ready, choose **Split**.

AWS CLI

To split off capacity

Use the `create-capacity-reservation-by-splitting` command. The following example creates a new Capacity Reservation by splitting off 10 instances from the specified Capacity Reservation.

```
aws ec2 create-capacity-reservation-by-splitting \
    --source-capacity-reservation-id cr-1234567890abdef0 \
    --instance-count 10
```

PowerShell

To split off capacity

Use the New-EC2CapacityReservationBySplitting cmdlet. The following example creates a new Capacity Reservation by splitting off 10 instances from the specified Capacity Reservation.

```
New-EC2CapacityReservationBySplitting `
    -SourceCapacityReservationId cr-1234567890abdef0 `
    -InstanceCount 10
```

Cancel a Capacity Reservation

You can cancel a Capacity Reservation that is in one of the following states:

- `assessing`
- `active` and there is no commitment duration or the commitment duration has elapsed. You can't cancel a future-dated Capacity Reservation during the commitment duration.

> **Note**
>
> You can't modify or cancel a Capacity Block. For more information, see Capacity Blocks for ML.

If a future-dated Capacity Reservation enters the `delayed` state, the commitment duration is waived, and you can cancel it as soon as it enters the `active` state.

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 Plans 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 the Capacity Reservation settings of your instance.

Console

To cancel a Capacity Reservation

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**.

AWS CLI

To cancel a Capacity Reservation

Use the cancel-capacity-reservation command.

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0
```

PowerShell

To cancel a Capacity Reservation

Use the Remove-EC2CapacityReservation cmdlet.

```
Remove-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0
```

Use Capacity Reservations with cluster placement groups

You can create Capacity Reservations in a cluster placement group to reserve Amazon EC2 compute capacity for your workloads. Cluster placement groups offer the benefit of low network latency and high network throughput.

Creating a Capacity Reservation in a cluster placement group ensures that you have access to compute capacity in your cluster placement groups when you need it, for as long as you need it. This is ideal for reserving capacity for high-performance (HPC) workloads that require compute scaling. It allows you to scale your cluster down while ensuring that the capacity remains available for your use so that you can scale back up when needed.

After you create a Capacity Reservation in a cluster placement group, you can share it with other AWS accounts. For more information, see Sharing Capacity Reservations in cluster placement groups.

Topics

- Limitations
- Work with Capacity Reservations in cluster placement groups
- Sharing Capacity Reservations in cluster placement groups

Limitations

Keep the following in mind when creating Capacity Reservations in cluster placement groups:

- If an existing Capacity Reservation is not in a placement group, you can't modify the Capacity Reservation to reserve capacity in a placement group. To reserve capacity in a placement group, you must create the Capacity Reservation in the placement group.
- After you create a Capacity Reservation in a placement group, you can't modify it to reserve capacity outside of the placement group.
- You can increase your reserved capacity in a placement group by modifying an existing Capacity Reservation in the placement group, or by creating additional Capacity Reservations in the placement group. However, you increase your chances of getting an insufficient capacity error.
- You can share Capacity Reservations only from the cluster placement group that you own. You cannot share Capacity Reservations from a cluster placement group that you do not own.
- You can't delete a cluster placement group that has `active` Capacity Reservations. You must cancel all Capacity Reservations in the cluster placement group before you can delete it.

Work with Capacity Reservations in cluster placement groups

To start using Capacity Reservations with cluster placement groups, perform the following steps.

> **Note**
>
> If you want to create a Capacity Reservation in an existing cluster placement group, skip Step 1. Then for Steps 2 and 3, specify the ARN of the existing cluster placement group.

Tasks

- Step 1: (Conditional) Create a cluster placement group for use with a Capacity Reservation
- Step 2: Create a Capacity Reservation in a cluster placement group
- Step 3: Launch instances into Capacity Reservations in a cluster placement group

Step 1: (*Conditional*) Create a cluster placement group for use with a Capacity Reservation

Perform this step only if you need to create a new cluster placement group. To use an existing cluster placement group, skip this step and then for Steps 2 and 3, use the ARN of that cluster placement group.

Console

To create a cluster placement group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Placement Groups**, and then choose **Create placement group**.
3. For **Name**, specify a descriptive name for the placement group.
4. For **Placement strategy**, choose **Cluster**.
5. Choose **Create group**.
6. In the **Placement groups** table, in the **Group ARN** column, make a note of the ARN of the cluster placement group that you created. You'll need it for the next step.

AWS CLI

To create a cluster placement group

Use the create-placement-group command.

```
aws ec2 create-placement-group \
    --group-name MyPG \
    --strategy cluster
```

Make a note of the placement group ARN returned in the output, because you'll need it for the next step.

PowerShell

To create a cluster placement group

Use the New-EC2PlacementGroup cmdlet.

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy "cluster"
```

Make a note of the placement group ARN returned in the output, because you'll need it for the next step.

Step 2: Create a Capacity Reservation in a cluster placement group

You create a Capacity Reservation in a cluster placement group in the same way that you create any Capacity Reservation. However, you must also specify the ARN of the cluster placement group in which to create the Capacity Reservation.

Considerations

- The specified cluster placement group must be in the `available` state. If the cluster placement group is in the `pending`, `deleting`, or `deleted` state, the request fails.
- The Capacity Reservation and the cluster placement group must be in the same Availability Zone. If the request to create the Capacity Reservation specifies an Availability Zone that is different from that of the cluster placement group, the request fails.
- You can create Capacity Reservations only for instance types that are supported by cluster placement groups. If you specify an unsupported instance type, the request fails.
- If you create an `open` Capacity Reservation in a cluster placement group and there are existing running instances that have matching attributes (placement group ARN, instance type, Availability Zone, platform, and tenancy), those instances automatically run in the Capacity Reservation.
- 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 workload 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 quotas.

Console

To create a Capacity Reservation

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, specify the instance type, platform, Availability Zone, Tenancy, quantity, and end date as needed.
4. For **Placement group**, select the ARN of the cluster placement group in which to create the Capacity Reservation.
5. Choose **Create**.

For more information, see Create a Capacity Reservation.

AWS CLI

To create a Capacity Reservation

Use the create-capacity-reservation command. For `--placement-group-arn`, specify the ARN of the cluster placement group in which to create the Capacity Reservation.

```
aws ec2 create-capacity-reservation \
    --instance-type instance_type \
    --instance-platform platform \
    --availability-zone-id az_id \
    --instance-count quantity \
    --placement-group-arn "placement_group_arn"
```

PowerShell

To create a Capacity Reservation

Use the Add-EC2CapacityReservation cmdlet. For `-PlacementGroupArn`, specify the ARN of the cluster placement group in which to create the Capacity Reservation.

```
Add-EC2CapacityReservation `
    -InstanceType instance_type `
    -InstancePlatform platform `
    -AvailabilityZoneId az_id `
    -InstanceCount quantity `
```

```
    -PlacementGroupArn "placement_group_arn"
```

Step 3: Launch instances into Capacity Reservations in a cluster placement group

You can launch an instance into a Capacity Reservation that is in a cluster placement group with one of the following options:

- *Specifying the ARN of the cluster placement group in which to launch the instance* – When you provide the ARN of a cluster placement group, Amazon EC2 launches the instance into that cluster placement group. You can use one of the following methods:
  - *Specifying open* – You do not have to specify the Capacity Reservation in the instance launch request. If the instance has attributes (placement group ARN, instance type, Availability Zone, platform, and tenancy) that match a Capacity Reservation in the specified placement group, the instance automatically runs in the Capacity Reservation.
  - *Specifying a Capacity Reservation* – If the Capacity Reservation accepts only targeted instance launches, you must specify the target Capacity Reservation in addition to the cluster placement group in the request.
  - *Specifying a Capacity Reservation group* – For more information, see [Using Capacity Reservation in cluster placement groups with a Capacity Reservation group](#).
- *Specifying only a Capacity Reservation group* – For more information, see [Using Capacity Reservation in cluster placement groups with a Capacity Reservation group](#).
- *Specifying only a Capacity Reservation* – You can launch instances into a Capacity Reservation in a cluster placement group.

> **Note**
>
> When you launch instances by specifying only a Capacity Reservation or only a Capacity Reservation group, the instances are launched into the Capacity Reservations that are created in the cluster placement group, but the instances are not directly attached to the cluster placement group.

Console

To launch instances into an existing Capacity Reservation

1. Follow the procedure to [launch an instance](), but don't launch the instance until you've completed the following steps to specify the settings for the placement group and Capacity Reservation.
2. Expand **Advanced details** and do the following:

   a. For **Placement group**, select the cluster placement group in which to launch the instance.

   b. For **Capacity Reservation**, choose one of the following options depending on the configuration of the Capacity Reservation:

      - **Open** – To launch the instances into any `open` Capacity Reservation in the cluster placement group that has matching attributes and sufficient capacity.
      - **Target by ID** – To launch the instances into a Capacity Reservation that accepts only targeted instance launches.
      - **Target by group** – To launch the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group.
3. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see [Launch an EC2 instance using the launch instance wizard in the console]().

AWS CLI

To launch instances into an existing Capacity Reservation

Use the [run-instances]() command. If you need to target a specific Capacity Reservation or a Capacity Reservation group, specify the `--capacity-reservation-specification` parameter. For `--placement`, specify the `GroupName` parameter and then specify the name of the placement group that you created in the previous steps.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count quantity \
    --instance-type instance_type \
```

```
    --key-name key_pair_name \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_reservation_id} \
    --placement "GroupName=cluster_placement_group_name"
```

PowerShell

To launch instances into an existing Capacity Reservation

Use the New-EC2Instance cmdlet. For `-Placement`, specify the `GroupName` parameter and then specify the name of the placement group that you created in the previous steps.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType instance_type `
    -KeyName key_pair_name `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId capacity_reservation_id `
    -Placement_GroupName cluster_placement_group_name
```

Sharing Capacity Reservations in cluster placement groups

You can share Capacity Reservations in cluster placement groups by either sharing only the Capacity Reservations, or by sharing both the Capacity Reservations and the cluster placement group in which they were created.

By sharing only the Capacity Reservation, you give consumer accounts access to that Capacity Reservation only. Consumer accounts have no visibility or access to the cluster placement group in which the Capacity Reservation is created. This gives you fine-grained control over consumer account access. Consumer accounts can't view any information about the cluster placement group, including its ARN.

When you share the cluster placement group and the Capacity Reservation, the cluster placement group is visible and accessible to consumer accounts. They can launch instances and create their own Capacity Reservations in it.

For more information, see the following resources.

- Launch instances into Capacity Reservations in a cluster placement group

- Shared Capacity Reservations
- Shared placement groups

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. For more information about resource groups, see What are resource groups? in the *AWS Resource Groups User Guide*.

You can include Capacity Reservations that you own in your account, and Capacity Reservations that are shared with you by other AWS accounts in a single resource group. You can also include Capacity Reservations that have different attributes (instance type, platform, Availability Zone, and tenancy) in a single resource group.

When you create resource groups for 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, Availability Zone, and tenancy) 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 **Only instances that specify this reservation** 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.

Tasks

- [Create a Capacity Reservation group]
- [Add a Capacity Reservation to a group]
- [Remove a Capacity Reservation from a group]
- [Delete a Capacity Reservation group]
- [Using Capacity Reservation in cluster placement groups with a Capacity Reservation group]

Create a Capacity Reservation group

You can use the following examples to create a resource group for Capacity Reservations with the following request parameters.

- `AWS::EC2::CapacityReservationPool` – Ensures that the resource group can be targeted for instance launches.
- `AWS::ResourceGroups::Generic` with `allowed-resource-types` set to `AWS::EC2::CapacityReservation` – Ensures that the resource group accepts Capacity Reservations only.

After you create a group, you can add Capacity Reservations to the group.

AWS CLI

To create a group for Capacity Reservations

Use the create-group AWS CLI command.

```
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"]}]}'
```

PowerShell

To create a group for Capacity Reservations

Use the New-RGGroup cmdlet.

```
New-RGGroup `
    -Name MyCRGroup `
    -Configuration `
        @{"Type"="AWS::EC2::CapacityReserationPool"} `
        @{"Type"="AWS::ResourceGroups::Generic"; "Parameters"=@{"allowed-resource-
types"=@{"Values"="AWS::EC2::CapacityReservations"}}}
```

Add a Capacity Reservation to a group

If you add a Capacity Reservation that is shared with you to a group, and that Capacity Reservation is unshared, it is automatically removed from the group.

AWS CLI

To add a Capacity Reservation to a group

Use the group-resources command.

The following example adds two Capacity Reservations to the specified group.

```
aws resource-groups group-resources \
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1
 \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

To add a Capacity Reservation to a group

Use the Add-RGResource cmdlet.

The following example adds two Capacity Reservations to the specified group.

```
Add-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-1234567890abcdef1", `
```

```
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

Remove a Capacity Reservation from a group

You can use the following examples to remove a Capacity Reservation from a group.

AWS CLI

To remove a Capacity Reservation from a group

Use the ungroup-resources command.

The following example removes two Capacity Reservations from the specified group.

```
aws resource-groups ungroup-resources \
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd
 \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

To remove a Capacity Reservation from a group

Use the Remove-RGResource cmdlet.

The following example removes two Capacity Reservations from the specified group.

```
Remove-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-0e154d26a16094dd", `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

Delete a Capacity Reservation group

You can use the following examples to delete a Capacity Reservation group.

AWS CLI

To delete a group

Use the delete-group command.

```
aws resource-groups delete-group --group MyCRGroup
```

PowerShell

To delete a group

Use the Remove-RGGroup cmdlet.

```
Remove-RGGroup -GroupName MyCRGroup
```

Using Capacity Reservation in cluster placement groups with a Capacity Reservation group

Amazon EC2 provides different launch methods for you to use Capacity Reservations in a cluster placement group with a Capacity Reservation group. You can choose one of the following methods to target a Capacity Reservation group based on your workload requirements:

- *Specifying the ARN of the cluster placement group and Capacity Reservation group* – This will use any available 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.

> **Note**
>
> When you launch instances using this method, the instances are placed in the specified cluster placement group.

- *Specifying only a Capacity Reservation group* – This will use all available capacity within the Capacity Reservation group by specifying only the Capacity Reservation group. While launching instances, capacity is used in the following order:
  - Capacity Reservations not associated with any cluster placement group.
  - Capacity Reservation in any cluster placement group within the Capacity Reservation group.

- If the group does not have a Capacity Reservation with matching attributes and available capacity, the instances run using On-Demand capacity and they are not placed in any cluster placement group.

> **Note**
>
> When you launch instances by specifying only a Capacity Reservation group, the instances are launched into the Capacity Reservations that are created in the cluster placement group, but the instances are not directly attached to the cluster placement group.

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 Getting started with AWS Local Zones in the *AWS Local Zones User Guide*.
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.

3. Create a subnet in the Local Zone. For **Availability Zone**, choose the Local Zone. For more information, see Create 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.

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-wl1-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 Get started with AWS Wavelength in the *AWS Wavelength Developer Guide*.
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-wl1-bos-wlz-1`. For more information, see Create a Capacity Reservation.

3. Create a subnet in the Wavelength Zone. For **Availability Zone**, choose the Wavelength Zone. For more information, see Create 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-wl1-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.

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 Shared Capacity Reservations.

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.

   > **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*.

Shared Capacity Reservations

Capacity Reservation sharing enables Capacity Reservation owners to share their reserved capacity with other AWS accounts or within an AWS organization. This enables you to create and manage Capacity Reservations centrally, and share the reserved capacity across multiple AWS accounts or within your AWS organization.

In this model, the AWS account that owns the Capacity Reservation (owner) shares it with other AWS accounts (consumers). Consumers can launch instances into Capacity Reservations that are shared with them in the same way that they launch instances into Capacity Reservations that they own in their own account. The Capacity Reservation owner is responsible for managing the Capacity Reservation and the instances that they launch into it. Owners cannot modify instances that consumers launch into Capacity Reservations that they have shared. Consumers are responsible for managing the instances that they launch into Capacity Reservations shared with them. Consumers cannot view or modify instances owned by other consumers or by the Capacity Reservation owner.

A Capacity Reservation owner can share a Capacity Reservation with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization
- Its entire AWS organization

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

1. Open the AWS RAM console at https://console.aws.amazon.com/ram/home.
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, Availability Zone, and tenancy) 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](https://console.aws.amazon.com/ram).

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are granted access to the shared Capacity Reservation if the [prequisites for sharing](#) are met. If the Capacity Reservation is shared with external accounts, they receive an invitation to join the resource share and are granted access to the shared Capacity Reservation after accepting the invitation.

Important

Before launching instances into a Capacity Reservation that is shared with you, verify that you have access to the shared Capacity Reservation by viewing it in the console or by describing it using the [describe-capacity-reservations] AWS CLI command. If you can view the shared Capacity Reservation in the console or describe it using the AWS CLI, it is available for your use and you can launch instances into it. If you attempt to launch instances into the Capacity Reservation and it is not accessible due to a sharing failure, the instances will launch into On-Demand capacity.

Console

To share a Capacity Reservation that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**.
3. Choose the Capacity Reservation to share and choose **Actions**, **Share reservation**.
4. Select the resource share to which to add the Capacity Reservation and choose **Share Capacity Reservation**.

   It could take a few minutes for consumers to get access to the shared Capacity Reservation.

To share a Capacity Reservation that you own using the AWS RAM console

See Creating a resource share in the *AWS RAM User Guide*.

AWS CLI

To share a Capacity Reservation that you own

Use the create-resource-share command.

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

To share a Capacity Reservation that you own

Use the New-RAMResourceShare cmdlet.

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

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.

Console

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*.

AWS CLI

To stop sharing a Capacity Reservation that you own

Use the disassociate-resource-share command.

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE
```

PowerShell

To stop sharing a Capacity Reservation that you own

Use the Disconnect-RAMResourceShare cmdlet.

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE"
```

Billing assignment for shared Amazon EC2 Capacity Reservations

By default, when a Capacity Reservation is shared, the owner is billed for the instances they run in the Capacity Reservation and for any available capacity, also called *unused capacity*, in the Capacity Reservation; while consumers are billed only for the instances they run in the shared Capacity Reservation.

If needed, the Capacity Reservation owner can assign the billing of any available capacity in the Capacity Reservation to any one of the accounts with which the Capacity Reservation is shared. After billing is assigned to another account, that account becomes the *billing owner* of any available capacity in the Capacity Reservation. Any charges for available capacity in the Capacity Reservation, from that point onward, are billed to the assigned account instead of the owner's account. The Capacity Reservation owner and the accounts with which the Capacity Reservation is shared continue to be billed for the instances they run in the Capacity Reservation.

> **Important**
>
> The Capacity Reservation owner remains the resource owner and they remain responsible for managing the Capacity Reservation. The account to which billing is assigned does not get any additional privileges; they can't cancel, modify, or share the Capacity Reservation in any way.

Topics

How it works

Only the Capacity Reservation owner can assign billing of a shared Capacity Reservation to another account. Billing can be assigned only to an account with which the Capacity Reservation is shared and that is consolidated under the same AWS Organizations payer account as the Capacity Reservation owner.

To assign billing of the available capacity of a Capacity Reservation to another account, the Capacity Reservation owner must initiate a request to the required account. The specified account receives the request and they must either accept or reject it within 12 hours.

- If they **accept**, they become the *billing owner* of any available capacity, also called *unused capacity*, in the Capacity Reservation. From that point onward, charges for any available capacity in the Capacity Reservation are billed to their account instead of the owner's account. After it is accepted, only the Capacity Reservation owner can revoke billing from the assigned account.
- If they **reject**, the Capacity Reservation owner remains the billing owner of the available capacity in the Capacity Reservation. Charges for any available capacity in the Capacity Reservation continue to be billed to the owner's account.

- If they **do not accept or reject** the request within 12 hours, it expires and charges for any available capacity in the Capacity Reservation continue to be billed to owner's account.

For the period that billing is assigned to another account, the `Reservation` and `UnusedBox` line items appear in the assigned account's Cost and Usage Report (CUR) instead of the owner's CUR.

The following table shows which line items appear in the CUR for the Capacity Reservation owner and consumer accounts **before** billing is assigned to another account.

Account	CUR line items before billing is assigned
Capacity Reservation owner	• `Reservation` • `BoxUsage` * • `UnusedBox`
Consumer accounts with which the Capacity Reservation is shared	• `BoxUsage` *

The following table shows which line items appear in the CUR for the Capacity Reservation owner and consumer accounts **after** billing is assigned to another account.

Account	CUR line items after billing is assigned
Capacity Reservation owner	• `BoxUsage` *
Consumer account to which billing is assigned	• `Reservation`

Account	CUR line items after billing is assigned
	• `BoxUsage` * • `UnusedBox`
Other consumer accounts with which the Capacity Reservation is shared	• `BoxUsage` *

> **Note**
>
> - * The `BoxUsage` line item appears in an account's CUR only if they have running instances in the Capacity Reservation. For more information about the CUR line items, see [Monitoring Capacity Reservations](#).
> - Use the Capacity Reservation ARN in the CUR to determine who owns the Capacity Reservation. If the ARN includes your AWS account ID, you are the Capacity Reservation owner. Otherwise, the Capacity Reservation is owned by a different account but billing is assigned to you.
> - Cost allocation tags assigned to Capacity Reservation by the owner will not appear in the consumer account's CUR. Cost allocation tags appear in the Capacity Reservation owner's CUR only.

Considerations

Keep the following in mind when assigning billing of a shared Capacity Reservation:

- You can't do partial or split billing assignments. Billing of all available capacity of a Capacity Reservation can be assigned to one account at a time.
- The available capacity of a Capacity Reservation can change over time. This will impact billing for the assigned account. For example, available capacity can increase if the Capacity Reservation owner increases the size of the Capacity Reservation, or if other consumer accounts stop or terminate their instances running in the Capacity Reservation.

- Billing can be assigned only to a consumer account that is consolidated under the same AWS Organizations payer account. Billing is automatically revoked from the consumer account if they leave the organization, or if the Capacity Reservation is no longer shared with them.
- Only the Capacity Reservation owner can cancel a pending billing assignment request and revoke billing from an assigned account after the request has been accepted.

Assign billing of a shared EC2 Capacity Reservation to another account

To assign billing of the available capacity of a shared Capacity Reservation to another account, the Capacity Reservation owner must initiate a request to the required account. In the Amazon EC2 console, this request is called a *transfer request*.

A Capacity Reservation owner can assign billing of the available capacity of Capacity Reservation to an account only if:

- The Capacity Reservation is already shared with that account.
- The account is consolidated under the same AWS Organizations payer account as the Capacity Reservation owner.

Billing is assigned to the specified account only once they accept the request.

When a Capacity Reservation owner initiates a request, an Amazon EventBridge event is sent to the requested account. For more information, see Monitor billing assignment requests for shared Capacity Reservations.

Console

To assign billing of a shared Capacity Reservation

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, select **Capacity Reservations** and then choose the shared Capacity Reservation.
3. In the **Billing of available capacity** section, choose **Assign billing**.
4. In the **Assign billing** screen, select the consumer account to which to assign billing, and then choose **Request**.

AWS CLI

To assign billing of a shared Capacity Reservation

Use the associate-capacity-reservation-billing-owner command. For `--capacity-reservation-id`, specify the ID of the shared Capacity Reservation. For `--unused-reservation-billing-owner-id`, specify the ID of the AWS account to which to assign billing.

```
aws ec2 associate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

To assign billing of a shared Capacity Reservation

Use the Register-EC2CapacityReservationBillingOwner cmdlet. For `-CapacityReservationId`, specify the ID of the shared Capacity Reservation. For `-UnusedReservationBillingOwnerId`, specify the ID of the AWS account to which to assign billing.

```
Register-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

View billing assignment requests for shared EC2 Capacity Reservations

A Capacity Reservation owner can view only the most recent billing assignment request that they initiated. And consumer accounts can view only the most recent billing assignment requests sent to them.

Requests can be viewed for 24 hours after they enter the `cancelled`, `expired`, or `revoked` state. After 24 hours, they can no longer be viewed.

Console

(Capacity Reservation owner) To view requests you initiated

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

2. In the navigation panel, select **Capacity Reservations** and then choose the shared Capacity Reservation for which to view requests.
3. The **Billing of available capacity** section shows the most recent request and its current state.

(Consumer account) To requests sent to you

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, select **Capacity Reservations**.
3. If you have pending requests, the **Pending billing assignment requests** banner appears at the top of the screen. If the banner does not appear, you do not have pending requests.

   To view the requests, choose **Review requests** in the banner.

AWS CLI

(Capacity Reservation owner) To view requests you initiated

Use the describe-capacity-reservation-billing-requests command.

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role odcr-owner
```

(Consumer account) To view requests sent to you

Use the describe-capacity-reservation-billing-requests command.

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role unused-reservation-billing-owner
```

PowerShell

(Capacity Reservation owner) To view requests you initiated

Use the Get-EC2CapacityReservationBillingRequest cmdlet.

```
Get-EC2CapacityReservationBillingRequest `
    -Role odcr-owner
```

(Consumer account) To view requests sent to you

Use the Get-EC2CapacityReservationBillingRequest cmdlet.

```
Get-EC2CapacityReservationBillingRequest `
    -Role unused-reservation-billing-owner
```

A request can be in one of the following states.

State	Description
`pending`	The request has not been accepted or rejected, but it has not yet expired.
`accepted`	The request was accepted by the specified account. Billing of available capacity of the Capacity Reservation is assigned to the consumer account.
`rejected`	The request was rejected by the consumer account.
`cancelled`	The request was cancelled by the Capacity Reservation owner while it was in the `pending` state.
`revoked`	Billing was revoked from the consumer account for one of the following reasons: • It was explicitly revoked by the Capacity Reservation owner.

State	Description
	• The Capacity Reservation is no longer shared with the consumer account. • The consumer account is no longer part of the AWS organization.
`expired`	The request expired because the consumer account did not accept or reject it within 12 hours.

Accept or reject billing of a shared EC2 Capacity Reservation

If you receive a billing assignment request for a Capacity Reservation that is shared with you, you can either accept or reject it. The request remains in the `pending` state until it is accepted or rejected.

If you accept the request, it enters the `accepted` state, and billing of any available, or *unused*, capacity of that Capacity Reservation is assigned to your account from that point onward. After you accept a request, only the Capacity Reservation owner can revoke billing from your account.

If you reject the request, it enters the `rejected` state, and billing of the available capacity of the Capacity Reservation remains assigned to the Capacity Reservation owner.

Requests expire if they are not accepted or rejected within 12 hours. If a request expires, billing of any unused capacity of the Capacity Reservation remains assigned to the Capacity Reservation owner.

When a request is accepted or rejected, an Amazon EventBridge event is sent to the Capacity Reservation owner's account. When a request expires, an Amazon EventBridge event is sent to the Capacity Reservation owner and the consumer account. For more information, see Monitor billing assignment requests for shared Capacity Reservations.

Console

To accept or reject a request

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, select **Capacity Reservations**.
3. If you have pending requests, the **Pending billing assignment requests** banner appears at the top of the screen. If the banner does not appear, you do not have pending requests.

   To view the requests, choose **Review requests** in the banner.
4. Select the request to accept or reject, and then choose either **Accept** or **Reject**.

AWS CLI

To accept a request

Use the accept-capacity-reservation-billing-ownership command. For `--capacity-reservation-id`, specify the ID of the Capacity Reservation for which to accept the request.

```
aws ec2 accept-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

To reject a request

Use the reject-capacity-reservation-billing-ownership command. For `--capacity-reservation-id`, specify the ID of the Capacity Reservation for which to reject the request.

```
aws ec2 reject-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

PowerShell

To accept a request

Use the Approve-EC2CapacityReservationBillingOwnership cmdlet.

```
Approve-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

To reject a request

Use the [Deny-EC2CapacityReservationBillingOwnership](#) cmdlet.

```
Deny-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

Cancel or revoke billing assignment requests for shared EC2 Capacity Reservations

Only the Capacity Reservation owner can cancel a `pending` billing assignment request. If a pending request is cancelled, it enters the `cancelled` state and billing of any available, or *unused*, capacity of the Capacity Reservation remains assigned to Capacity Reservation owner.

After a request is `accepted`, only the Capacity Reservation owner can revoke billing from the assigned account. If billing is revoked, the request enters the `revoked` state and billing of any available capacity of the Capacity Reservation is reassigned to Capacity Reservation owner.

When a request is cancelled or revoked, Amazon EventBridge events are sent to the Capacity Reservation owner and specified consumer account. For more information, see Monitor billing assignment requests for shared Capacity Reservations.

Console

To cancel or revoke a request

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation panel, select **Capacity Reservations** and then choose the Capacity Reservation for which to cancel or revoke the request.
3. In the **Billing of available capacity** section, choose **Cancel transfer** or **Revoke transfer**, depending on the current state of the request.

AWS CLI

To cancel or revoke a request

Use the disassociate-capacity-reservation-billing-owner command. For `--unused-reservation-billing-owner-id`, specify the ID of the AWS account to which the request was sent.

```
aws ec2 disassociate-capacity-reservation-billing-owner \
```

```
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

To cancel or revoke a request

Use the Unregister-EC2CapacityReservationBillingOwner cmdlet. For -UnusedReservationBillingOwnerId, specify the ID of the AWS account to which the request was sent.

```
Unregister-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

Monitor billing assignment requests for shared Capacity Reservations

Amazon EC2 sends Amazon EventBridge events when the state of a billing assignment request changes.

- Events are sent to the Capacity Reservation owner when a request enters the following states: `accepted` | `rejected` | `expired` | `revoked`.
- Events are sent to the requested consumer account when a request enters the following states: `pending` | `expired` | `cancelled` | `revoked`.

For more information about Amazon EventBridge, see the Amazon EventBridge User Guide.

The following is the Amazon EventBridge event pattern.

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request pending|
accepted|rejected|cancelled|revoked|expired",
   "source":"aws.ec2",
   "account":"account_id",
   "time":"state_change_timestamp",
   "region":"region",
   "resources":[
      "arn:aws:ec2:region:cr_owner_account_id:capacity-reservation/cr_id"
```

```
    ],
    "detail":{
        "capacity-reservation-id":"cr_id",
        "updateTime":timestamp,
        "ownerAccountId":"cr_owner_account_id",
        "unusedReservationChargesOwnerID":"consumer_account_id",
        "status":"pending|accepted|rejected|cancelled|revoked|expired",
        "statusMessage":"message
    }
}
```

The following is an exmaple of an event that is sent to the Capacity Reservation owner (222222222222) when a consumer account (111111111111) accepts a billing assignment request for a shared Capacity Reservation (`cr-01234567890abcdef`).

```
{
    "version":"0",
    "id":"12345678-1234-1234-1234-123456789012",
    "detail-type":"On-Demand Capacity Reservation Billing Ownership Request accepted",
    "source":"aws.ec2",
    "account":"222222222222",
    "time":"2024-09-01Thh:59:59Z",
    "region":"us-east-1",
    "resources":[
        "arn:aws:ec2:us-east-1:222222222222:capacity-reservation/cr-01234567890abcdef"
    ],
    "detail":{
        "capacity-reservation-id":"cr-01234567890abcdef",
        "updateTime":"2024-08-01Thh:59:59Z",
        "ownerAccountId":"222222222222",
        "unusedReservationChargesOwnerID":"111111111111",
        "status":"accepted",
        "statusMessage":"billing transfer status message"
    }
}
```

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.

By default, the Capacity Reservation owner is billed for instances that they run inside the Capacity Reservation and for unused reserved capacity, while consumers are billed for the instances that they run inside the shared Capacity Reservation. However, you can assign billing of the available capacity of a shared Capacity Reservation to a specific consumer account. For more information, see Billing assignment for shared Amazon EC2 Capacity Reservations.

If the Capacity Reservation owner belongs to a different payer account and the Capacity Reservation is covered by a Regional Reserved Instance or a Savings Plan, the Capacity Reservation owner continues to be billed for the Regional Reserved Instance or Savings Plan. In these cases, the Capacity Reservation owner pays for the Regional Reserved Instance or Savings Plan, and consumers are billed for the instances that the run in 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.

Capacity Reservation Fleets

An *On-Demand Capacity Reservation Fleet* is a group of Capacity Reservations.

A Capacity Reservation Fleet request contains all of the configuration information that's needed to launch a Capacity Reservation Fleet. Using a single request, you can reserve large amounts of Amazon EC2 capacity for your workload across multiple instance types, up to a target capacity that you specify.

After you create a Capacity Reservation Fleet, you can manage the Capacity Reservations in the fleet collectively by modifying or canceling the Capacity Reservation Fleet.

Topics

How Capacity Reservation Fleets work

When you create a Capacity Reservation Fleet, the Fleet attempts to create individual Capacity Reservations to meet the total target capacity that you specified in the Fleet request.

The number of instances for which the Fleet reserves capacity depends on the *total target capacity* and the *instance type weights* that you specify. The instance type for which it reserves capacity depends on the *allocation strategy* and *instance type priorities* that you use.

If there is insufficient capacity at the time the Fleet is created, and it is unable to immediately meet its total target capacity, the Fleet asynchronously attempts to create Capacity Reservations until it has reserved the requested amount of capacity.

When the Fleet reaches its total target capacity, it attempts to maintain that capacity. If a Capacity Reservation in the Fleet is cancelled, the Fleet automatically creates one or more Capacity

Reservations, depending on your Fleet configuration, to replace the lost capacity and to maintain its total target capacity.

The Capacity Reservations in the Fleet can't be managed individually. They must be managed collectively by modifying the Fleet. When you modify a Fleet, the Capacity Reservations in the Fleet are automatically updated to reflect the changes.

Currently, Capacity Reservation Fleets support the `open` instance matching criteria, and all Capacity Reservations launched by a Fleet automatically use this instance matching criteria. With this criteria, new instances and existing instances that have matching attributes (instance type, platform, Availability Zone, and tenancy) automatically run in the Capacity Reservations created by a Fleet. Capacity Reservation Fleets do not support targeted instance matching criteria.

Considerations

Keep the following in mind when working with Capacity Reservation Fleets:

- A Capacity Reservation Fleet can be created, modified, viewed, and cancelled using the AWS CLI and AWS API.
- The Capacity Reservations in a Fleet can't be managed individually. They must be managed collectively by modifying or cancelling the Fleet.
- A Capacity Reservation Fleet can't span across Regions.
- A Capacity Reservation Fleet can't span across Availability Zones.
- Capacity Reservations created by a Capacity Reservation Fleet are automatically tagged with the following AWS generated tag:
  - Key — `aws:ec2-capacity-reservation-fleet`
  - Value — *`fleet_id`*

  You can use this tag to identify Capacity Reservations that were created by a Capacity Reservation Fleet.

Pricing

There are no additional charges for using Capacity Reservation Fleets. You are billed for the individual Capacity Reservations that are created by your Capacity Reservation Fleets. For more information about how Capacity Reservations are billed, see Capacity Reservation pricing and billing.

Capacity Reservation Fleet concepts and planning

The following information describes how to plan a Capacity Reservation Fleet and describes Capacity Reservation Fleet concepts including total target capacity, allocation strategy, instance type weight, and instance type priority.

Topics

Plan a Capacity Reservation Fleet

When planning your Capacity Reservation Fleet, we recommend that you do the following:

1. Determine the amount of compute capacity that is needed by your workload.
2. Decide on the instance types and Availability Zones that you want to use.
3. Assign each instance type a priority based on your needs and preferences. For more information, see Instance type priority.
4. Create a capacity weighting system that makes sense for your workload. Assign a weight to each instance type and determine your total target capacity. For more information, see Instance type weight and Total target capacity.
5. Determine whether you need the Capacity Reservation indefinitely or only for a specific period of time.

Total target capacity

The *total target capacity* defines the total amount of compute capacity that the Capacity Reservation Fleet reserves. You specify the total target capacity when you create the Capacity Reservation Fleet. After the Fleet has been created, Amazon EC2 automatically creates Capacity Reservations to reserve capacity up to the total target capacity.

The number of instances for which the Capacity Reservation Fleet reserves capacity is determined by the total target capacity and the *instance type weight* that you specify for each instance type in the Capacity Reservation Fleet (`total target capacity/instance type weight=number of instances`).

You can assign a total target capacity based on units that are meaningful to your workload. For example, if your workload requires a certain number of vCPUs, you can assign the total target capacity based on the number of vCPUs required. If your workload requires `2048` vCPUs, specify a total target capacity of `2048` and then assign instance type weights based on the number of vCPUs provided by the instance types in the Fleet. For an example, see Instance type weight.

Allocation strategy

The allocation strategy for your Capacity Reservation Fleet determines how it fulfills your request for reserved capacity from the instance type specifications in the Capacity Reservation Fleet configuration.

Currently, only the `prioritized` allocation strategy is supported. With this strategy, the Capacity Reservation Fleet creates Capacity Reservations using the priorities that you have assigned to each of the instance type specifications in the Capacity Reservation Fleet configuration. Lower priority values indicate higher priority for use. For example, say you create a Capacity Reservation Fleet that uses the following instance types and priorities:

- `m4.16xlarge` — priority = 1
- `m5.16xlarge` — priority = 3
- `m5.24xlarge` — priority = 2

The Fleet first attempts to create Capacity Reservations for `m4.16xlarge`. If Amazon EC2 has insufficient `m4.16xlarge` capacity, the Fleet attempts to create Capacity Reservations for `m5.24xlarge`. If Amazon EC2 has insufficient `m5.24xlarge` capacity, the Fleet creates Capacity Reservations for `m5.16xlarge`.

Instance type weight

The *instance type weight* is a weight that you assign to each instance type in the Capacity Reservation Fleet. The weight determines how many units of capacity each instance of that specific instance type counts toward the Fleet's *total target capacity*.

You can assign weights based on units that are meaningful to your workload. For example, if your workload requires a certain number of vCPUs, you can assign weights based on the number of vCPUs provided by each instance type in the Capacity Reservation Fleet. In this case, if you create a Capacity Reservation Fleet using `m4.16xlarge` and `m5.24xlarge` instances, you would assign weights that correspond to the number of vCPUs for each instance as follows:

- `m4.16xlarge` — 64 vCPUs, weight = 64 units
- `m5.24xlarge` — 96 vCPUs, weight = 96 units

The instance type weight determines the number of instances for which the Capacity Reservation Fleet reserves capacity. For example, if a Capacity Reservation Fleet with a total target capacity of `384` units uses the instance types and weights in the preceding example, the Fleet could reserve capacity for 6 `m4.16xlarge` instances (384 total target capacity/64 instance type weight=6 instances), or 4 `m5.24xlarge` instances (384 / 96 = 4).

If you do not assign instance type weights, or if you assign an instance type weight of `1`, the total target capacity is based purely on instance count. For example, if a Capacity Reservation Fleet with a total target capacity of `384` units uses the instance types in the preceding example, but omits the weights or specifies a weight of `1` for both instance types, the Fleet could reserve capacity for either `384` `m4.16xlarge` instances or `384` `m5.24xlarge` instances.

Instance type priority

The *instance type priority* is a value that you assign to the instance types in the Fleet. The priorities are used to determine which of the instance types specified for the Fleet should be prioritized for use.

Lower priority values indicate a higher priority for use.

Create a Capacity Reservation Fleet

When you create a Capacity Reservation Fleet it automatically creates Capacity Reservations for the instance types specified in the Fleet request, up to the specified total target capacity. The number of instances for which the Capacity Reservation Fleet reserves capacity depends on the total target capacity and instance type weights that you specify in the request. For more information, see Instance type weight and Total target capacity.

When you create the Fleet, you must specify the instance types to use and a priority for each of those instance types. For more information, see Allocation strategy and Instance type priority.

Note

The **AWSServiceRoleForEC2CapacityReservationFleet** service-linked role is automatically created in your account the first time that you create a Capacity Reservation Fleet. For more information, see Using service-linked roles for Capacity Reservation Fleet.

Currently, Capacity Reservation Fleets support the open instance matching criteria only.

AWS CLI

To create a Capacity Reservation Fleet

Use the create-capacity-reservation-fleet command.

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 24 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-12-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

The following are the contents of instanceTypeSpecification.json.

```
[
  {
    "InstanceType": "m5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "Weight": 3.0,
    "AvailabilityZone":"us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

The following is example output.

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
```

```
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

PowerShell

To create a Capacity Reservation Fleet

Use the New-EC2CapacityReservationFleet cmdlet.

```
New-EC2CapacityReservationFleet `
    -TotalTargetCapacity 24 `
    -AllocationStrategy "prioritized" `
    -InstanceMatchCriterion "open" `
    -Tenancy "default" `
    -EndDate 2021-12-31T23:59:59.000Z `
    -InstanceTypeSpecification $specification
```

The specification is defined as follows.

```
$specification = New-Object Amazon.EC2.Model.ReservationFleetInstanceSpecification
$specification.InstanceType = "m5.xlarge"
$specification.InstancePlatform = "Linux/UNIX"
$specification.Weight = 3.0
$specification.AvailabilityZone = "us-east-1a"
$specification.EbsOptimized = $true
$specification.Priority = 1
```

Modify a Capacity Reservation Fleet

You can modify the total target capacity and date of a Capacity Reservation Fleet at any time. When you modify the total target capacity of a Capacity Reservation Fleet, the Fleet automatically creates new Capacity Reservations, or modifies or cancels existing Capacity Reservations in the Fleet to meet the new total target capacity. When you modify the end date for the Fleet, the end dates for all of the individual Capacity Reservations are updated accordingly.

Considerations

- After you modify a Fleet, its status transitions to `modifying`. You can't attempt additional modifications to a Fleet while it is in the `modifying` state.

- You can't modify the tenancy, Availability Zone, instance types, instance platforms, priorities, or weights used by a Capacity Reservation Fleet. If you need to change any of these parameters, you might need to cancel the existing Fleet and create a new one with the required parameters.
- You can't specify `--end-date` and `--remove-end-date` in the same command.

AWS CLI

To modify a Capacity Reservation Fleet

Use the [modify-capacity-reservation-fleet](#) command.

Example 1: Modify total target capacity

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

Example 2: Modify end date

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2021-07-04T23:59:59.000Z
```

Example 3: Remove end date

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --remove-end-date
```

PowerShell

To modify a Capacity Reservation Fleet

Use the [Edit-EC2CapacityReservationFleet](#) cmdlet.

Example 1: Modify total target capacity

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -TotalTargetCapacity 160
```

Example 2: Modify end date

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -EndDate 2021-07-04T23:59:59.000Z
```

Example 3: Remove end date

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -RemoveEndDate
```

Cancel a Capacity Reservation Fleet

When you no longer need a Capacity Reservation Fleet and the capacity it reserves, you can cancel it. When you cancel a Fleet, its status changes to `cancelled` and it can no longer create new Capacity Reservations. Additionally, all of the individual Capacity Reservations in the Fleet are canceled. The instances that were previously running in the reserved capacity continue to run normally in the shared capacity.

AWS CLI

To cancel a Capacity Reservation Fleet

Use the cancel-capacity-reservation-fleets command.

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

The following is example output.

```
{
    "SuccessfulFleetCancellations": [
        {
            "CurrentFleetState": "cancelling",
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
```

```
}
```

PowerShell

To cancel a Capacity Reservation Fleet

Use the Stop-EC2CapacityReservationFleet cmdlet.

```
Stop-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-abcdef01234567890
```

Example Capacity Reservation Fleet configurations

The following example creates a Capacity Reservation Fleet that uses two instance types: `m5.4xlarge` and `m5.12xlarge`.

It uses a weighting system based on the number of vCPUs provided by the specified instance types. The total target capacity is `480` vCPUs. The `m5.4xlarge` provides 16 vCPUs and gets a weight of `16`, while the `m5.12xlarge` provides 48 vCPUs and gets a weight of `48`. This weighting system configures the Capacity Reservation Fleet to reserve capacity for either 30 `m5.4xlarge` instances (480/16=30), or 10 `m5.12xlarge` instances (480/48=10).

The Fleet is configured to prioritize the `m5.12xlarge` capacity and gets priority of `1`, while the `m5.4xlarge` gets a lower priority of `2`. This means that the fleet will attempt to reserve the `m5.12xlarge` capacity first, and only attempt to reserve the `m5.4xlarge` capacity if Amazon EC2 has insufficient `m5.12xlarge` capacity.

The Fleet reserves the capacity for `Windows` instances and the reservation automatically expires on `October 31, 2021` at `23:59:59` UTC.

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 480 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-10-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

The following is the contents of `instanceTypeSpecification.json`.

```
[
    {
        "InstanceType": "m5.4xlarge",
        "InstancePlatform":"Windows",
        "Weight": 16,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 2
    },
    {
        "InstanceType": "m5.12xlarge",
        "InstancePlatform":"Windows",
        "Weight": 48,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 1
    }
]
```

Using service-linked roles for Capacity Reservation Fleet

On-Demand Capacity Reservation Fleet uses AWS Identity and Access Management (IAM) [service-linked roles](). A service-linked role is a unique type of IAM role that is linked directly to Capacity Reservation Fleet. Service-linked roles are predefined by Capacity Reservation Fleet and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Capacity Reservation Fleet easier because you don't have to manually add the necessary permissions. Capacity Reservation Fleet defines the permissions of its service-linked roles, and unless defined otherwise, only Capacity Reservation Fleet can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Capacity Reservation Fleet resources because you can't inadvertently remove permission to access the resources.

Service-linked role permissions for Capacity Reservation Fleet

Capacity Reservation Fleet uses the service-linked role named **AWSServiceRoleForEC2CapacityReservationFleet** to create, describe, modify, and cancel Capacity Reservations in a Capacity Reservation Fleet on your behalf.

The AWSServiceRoleForEC2CapacityReservationFleet service-linked role trusts the following entity to assume the role:

- `capacity-reservation-fleet.amazonaws.com`

The role uses the `AWSEC2CapacityReservationFleetRolePolicy` AWS managed policy. For more information, see AWS managed policy: AWSEC2CapacityReservationFleetRolePolicy.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Create a service-linked role for Capacity Reservation Fleet

You don't need to manually create a service-linked role. When you create a Capacity Reservation Fleet using the `create-capacity-reservation-fleet` AWS CLI command or the `CreateCapacityReservationFleet` API, the service-linked role is automatically created for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create a Capacity Reservation Fleet, Capacity Reservation Fleet creates the service-linked role for you again.

Edit a service-linked role for Capacity Reservation Fleet

Capacity Reservation Fleet does not allow you to edit the AWSServiceRoleForEC2CapacityReservationFleet service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Edit a service-linked role description in the *IAM User Guide*.

Delete a service-linked role for Capacity Reservation Fleet

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must delete the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the Capacity Reservation Fleet service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete the AWSServiceRoleForEC2CapacityReservationFleet service-linked role

1. Use the `delete-capacity-reservation-fleet` AWS CLI command or the `DeleteCapacityReservationFleet` API to delete the Capacity Reservation Fleets in your account.
2. Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForEC2CapacityReservationFleet service-linked role. For more information, see Delete a service-linked role in the *IAM User Guide*.

Supported Regions for Capacity Reservation Fleet service-linked roles

Capacity Reservation Fleet supports using service-linked roles in all of the Regions where the service is available. For more information, see AWS Regions and Endpoints.

Monitor Capacity Reservations usage with CloudWatch metrics

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.

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

- View CloudWatch metrics for Capacity Reservations

Capacity Reservation usage metrics

The `AWS/EC2CapacityReservations` namespace includes the following usage metrics you can use to monitor and maintain on-demand capacity within thresholds you specify for your reservation.

Metric	Description
`UsedInstanceCount`	The number of instances that are currently in use. Unit: Count
`AvailableInstanceCount`	The number of instances that are available. Unit: Count
`TotalInstanceCount`	The total number of instances you have reserved. Unit: Count
`InstanceUtilization`	The percentage of reserved capacity instances that are currently in use. Unit: Percent

Capacity Reservation metric dimensions

You can use the following dimensions to refine the metrics listed in the previous table within the selected Region and account.

Dimension	Description
(No dimension)	

Dimension	Description
	This dimension filters the specified metric for all Capacity Reservations.
`CapacityReservationId`	This dimension filters the specified metric for the identified Capacity Reservation.
`InstanceType`	This dimension filters the specified metric for the identified instance type.
`AvailabilityZone`	This dimension filters the specified metric for the identified Availability Zone.
`InstanceMatchCriteria`	This dimension filters the specified metric for the identified instance match criteria (`open` or `targeted`).
`InstancePlatform`	This dimension filters the specified metric data for the identified platform.
`Tenancy`	This dimension filters the specified metric for the identified tenancy.

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

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the Region. From the navigation bar, select the Region where your Capacity Reservation resides. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, choose **Metrics**.

4. For **All metrics**, choose **EC2 Capacity Reservations**.
5. Choose from the preceding metric dimensions **Across All Capacity Reservations**, **By Capacity Reservation**, **By Instance Type**, **By Availability Zone**, **By Platform**, **By Instance Match Criteria** or **By Tenancy** and metrics will be grouped by No dimension, `CapacityReservationId`, `InstanceType`, `AvailabilityZone`, `Platform`, `InstanceMatchCriteria`, and `Tenancy` respectively.
6. To sort the metrics, use the column heading. To graph a metric, select the checkbox next to the metric.

To view Capacity Reservation metrics using the AWS CLI

Use the following list-metrics command:

```
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

Monitor Capacity Reservation underutilization

You can monitor Capacity Reservation underutilization using the following:

Topics

- Amazon EventBridge events
- Email and AWS Health Dashboard notifications

Amazon EventBridge events

AWS Health sends events to Amazon EventBridge when a Capacity Reservation in your account is below 20 percent usage over certain periods. With EventBridge, you can establish rules that trigger programmatic actions in response to such events. For example, you can create a rule that automatically cancels a Capacity Reservation when its utilization drops below 20 percent utilization over a 7-day period.

Events in EventBridge 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 Amazon EventBridge event patterns in the *Amazon EventBridge User Guide*.

For more information, see the Amazon EventBridge User Guide.

This feature is not supported in AWS GovCloud (US).

Events

AWS Health sends the following events when capacity usage for a Capacity Reservation is below 20 percent.

- `AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION`

  The following is an example of an event that is generated when a newly created Capacity Reservation is below 20 percent capacity usage over a 24-hour period.


```
{
    "version": "0",
    "id": "b3e00086-f271-12a1-a36c-55e8ddaa130a",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-03-10T12:03:38Z",
    "region": "ap-south-1",
    "resources": [
        "cr-01234567890abcdef"
    ],
    "detail": {
        "eventArn": "arn:aws:health:ap-south-1::event/EC2/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_cr-01234567890abcdef-6211-4d50-9286-0c9fbc243f04",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION",
        "eventTypeCategory": "accountNotification",
        "startTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "endTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef"
            }
```

```
        ]
    }
    }
```

- `AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY`

  The following is an example of an event that is generated when one or more Capacity Reservation is below 20 percent capacity usage over a 7-day period.

```
{
    "version": "0", "id":"7439d42b-3c7f-ad50-6a88-25e2a70977e2",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-03-07T06:06:01Z",
    "region": "us-east-1",
    "resources": [
        "cr-01234567890abcdef | us-east-1b | t3.medium | Linux/UNIX | 0.0%",
        "cr-09876543210fedcba | us-east-1a | t3.medium | Linux/UNIX | 0.0%"
    ],
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/
EC2/AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY_726c1732-d6f6-4037-b9b8-
bec3c2d3ba65",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY",
        "eventTypeCategory": "accountNotification",
        "startTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "endTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef | us-east-1b | t3.medium |
 Linux/UNIX | 0.0%"
            },
            {
```

```
                "entityValue": "cr-09876543210fedcba | us-east-1a | t3.medium |
 Linux/UNIX | 0.0%"
            }
        ]
    }
}
```

Create an EventBridge rule

To receive email notifications when your Capacity Reservation utilization drops below 20 percent, create an Amazon SNS topic, and then create an EventBridge rule for the `AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION` event.

To create the Amazon SNS topic

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Topics**, and then choose **Create topic**.
3. For **Type**, choose **Standard**.
4. For **Name**, enter a name for the new topic.
5. Choose **Create topic**.
6. Choose **Create subscription**.
7. For **Protocol**, choose **Email**, and then for **Endpoint**, enter the email address that receives the notifications.
8. Choose **Create subscription**.
9. The email address entered above will receive email message with the following subject line: `AWS Notification - Subscription Confirmation`. Follow the directions to confirm your subscription.

To create the EventBridge rule

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. In the navigation pane, choose **Rules**, and then choose **Create rule**.
3. For **Name**, enter a name for the new rule.
4. For **Rule type**, choose **Rule with an event pattern**.
5. Choose **Next**.

6. In the **Event pattern**, do the following:

   a. For **Event source**, choose **AWS services**.

   b. For **AWS service**, choose **AWS Health**.

   c. For **Event type**, choose **EC2 ODCR Underutilization Notification**.

7. Choose **Next**.

8. For **Target 1**, do the following:

   a. For **Target types**, choose **AWS service**.

   b. For **Select a target**, choose **SNS topic**.

   c. For **Topic**, choose the topic that you created earlier.

9. Choose **Next** and then **Next** again.

10. Choose **Create rule**.

Email and AWS Health Dashboard notifications

AWS Health sends the following email and AWS Health Dashboard notifications when capacity utilization for Capacity Reservations in your account drops below 20 percent.

- Individual notifications for each newly created Capacity Reservation that has been below 20 percent utilization over the last 24-hour period.
- A summary notification for all Capacity Reservations that have been below 20 percent utilization over the last 7-day period.

The email notifications and AWS Health Dashboard notifications are sent to the email address associated with the AWS account that owns the Capacity Reservations. The notifications include the following information:

- The ID of the Capacity Reservation.
- The Availability Zone of the Capacity Reservation.
- The average utilization rate for the Capacity Reservation.
- The instance type and platform (operating system) of the Capacity Reservation.

Additionally, when capacity utilization for a Capacity Reservation in your account drops below 20 percent over a 24-hour and 7-day period, AWS Health sends events to EventBridge. With

EventBridge, you can create rules that activate automatic actions, such as sending email notifications or triggering AWS Lambda functions, in response to such events. For more information, see Monitor Capacity Reservation underutilization.

Monitor state changes for future-dated Capacity Reservations

Amazon EC2 sends an event to Amazon EventBridge when the state of a future-dated Capacity Reservation changes.

The following is example of this event. In this example, the future-dated Capacity Reservation entered the `scheduled` state. Note the state highlighted in the `detail-type` field.

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"EC2 Capacity Reservation Scheduled",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"yyyy-mm-ddThh:mm:ssZ",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdefg"
   ],
   "detail":{
      "capacity-reservation-id":"cr-1234567890abcdefg",
      "state":"scheduled"
   }
}
```

The possible values for the `detail-type` field are:

- `Scheduled`
- `Active`
- `Delayed`
- `Unsupported`
- `Failed`
- `Expired`

For more information about these states, see View the state of a Capacity Reservation.

You can create Amazon EventBridge events that monitor for these events and then trigger specific actions when they occur. For more information, see [Creating rules that react to events in Amazon EventBridge](#).

To create a rule that monitors for all state change events, you can use the following event pattern.

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation"
  }]
}
```

To create a rule that monitors for only specific state changes, you can use the following event pattern.

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation state"
  }]
}
```

For example, the following event pattern monitors for events that are sent when a future-dated Capacity Reservation enters the `active` state.

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation Active"
  }]
}
```

Capacity Blocks for ML

Capacity Blocks for ML allow you to reserve GPU-based accelerated computing instances on a future date to support your short duration machine learning (ML) workloads. Instances that run inside a Capacity Block are automatically placed close together inside [Amazon EC2 UltraClusters](#), for low-latency, petabit-scale, non-blocking networking.

You can also use Capacity Blocks to reserve capacity for Amazon EC2 UltraServers. UltraServers connect multiple Amazon EC2 instances within a low-latency, high-bandwidth accelerator interconnect. You can use UltraServers to handle the most compute and memory intensive AI/ML workloads in training, fine-tuning, and inference. For more information, see Amazon EC2 UltraServers.

With Capacity Blocks, you can see when GPU instance capacity is available on future dates, and you can schedule a Capacity Block to start at a time that works best for you. When you reserve a Capacity Block, you get predictable capacity assurance for GPU instances while paying only for the amount of time that you need. We recommend Capacity Blocks when you need GPUs to support your ML workloads for days or weeks at a time and don't want to pay for a reservation while your GPU instances aren't in use.

The following are some common use cases for Capacity Blocks.

- **ML model training and fine-tuning –** Get uninterrupted access to the GPU instances that you reserved to complete ML model training and fine-tuning.
- **ML experiments and prototypes –** Run experiments and build prototypes that require GPU instances for short durations.

You can reserve a Capacity Block with the following specifications:

- Reserve a start time up to 8 weeks in advance
- Set a reservation duration of one to 14 days or a multiple of 7 days, up to 182 days (Examples: 21 days, 28 days)
- Configure up to 64 instances per Capacity Block
- Configure up to 256 instances across multiple Capacity Blocks

For Amazon EC2 UltraServers, each UltraServer corresponds to one Capacity Block. You can request multiple UltraServers through a single request.

You can use Capacity Blocks to reserve `p6-b200`, `p5`, `p5e`, `p5en`, `p4d`, `p4de`, `trn1`, and `trn2` instances. You can purchase the following UltraServer types through Capacity Blocks: `P6e-GB200` and `Trn2` (in preview).

To reserve a Capacity Block, you start by specifying your capacity needs, including the instance type or UltraServer type, the number of instances or UltraServers, amount of time, earliest start date, and latest end date that you need. Then, you can see an available Capacity Block offering that meets your specifications. The Capacity Block offering includes details such as start time, Availability Zone, and reservation price. The price of a Capacity Block offering depends on available supply and demand at the time the offering was delivered. After you reserve a Capacity Block, the price doesn't change. For more information, see Capacity Blocks pricing and billing.

When you purchase a Capacity Block offering, your reservation is created for the date and number of instances that you selected. When your Capacity Block reservation begins, you can target instance launches by specifying the reservation ID in your launch requests.

You can use all the instances you reserved until 30 minutes (for instance types) or 60 minutes (for UltraServer type) before the end time of the Capacity Block. With 30 minutes (for instance types) or 60 minutes (for UltraServer types) left in your Capacity Block reservation, we begin terminating any instances that are running in the Capacity Block. We use this time to clean up your instances before delivering the Capacity Block to the next customer. We emit an event through EventBridge 10 minutes before the termination process begins. For more information, see Monitor Capacity Blocks using EventBridge.

Topics

Supported platforms

Capacity Blocks for ML currently support instances and UltraServers with default tenancy only. When you use the AWS Management Console to purchase a Capacity Block, the default platform option is Linux/UNIX. When you use the AWS Command Line Interface (AWS CLI) or AWS SDK to purchase a Capacity Block, the following platform options are available:

- Linux/Unix
- Red Hat Enterprise Linux
- RHEL with HA
- SUSE Linux
- Ubuntu Pro

Considerations

Before you use Capacity Blocks, consider the following details and limitations.

- If we detect impairment impacting an UltraServer Capacity Block, we will notify you but generally will not take action to terminate your instances on the Capacity Block. This is to minimize unintended disruption to your workloads. You can continue using the UltraServer Capacity Block as is after receiving this notification or request remediation by terminating all instances on the capacity block and submitting an AWS support case. After we receive your support case, we will notify you when we have completed remediation and you can relaunch instances onto your UltraServer Capacity Block.
- For `P6e-GB200` UltraServer Capacity Blocks, you must terminate your instances at least 60 minutes before the Capacity Block end time.
- To use `P6e-GB200` UltraServer Capacity Blocks, you must be opted in to the Dallas Local Zone (N. Virginia) Local Zone.
- Each Capacity Block can have up to 64 instances, and you can have up to 256 instances across Capacity Blocks.
- You can describe Capacity Block offerings that can start in as soon as 30 minutes.
- Capacity Blocks end at 11:30AM Coordinated Universal Time (UTC).
- The termination process for instances running in a Capacity Block begins at 11:00AM Coordinated Universal Time (UTC) on the final day of the reservation.
- Capacity Blocks can be reserved with a start time up to 8 weeks in the future.

- Capacity Block cancellations aren't allowed.
- Capacity Block can't be moved or split.
- Capacity Blocks can't be shared across AWS accounts or within your AWS Organization.
- Only UltraServer Capacity Blocks can be used with resource groups. Instance Capacity Blocks can't be used with resource groups. For more information, see Create a resource group for UltraServer Capacity Blocks.
- The total number of instances that can be reserved in Capacity Blocks across all accounts in your AWS Organization can't exceed 256 instances on a particular date.
- To use a Capacity Block, instances must specifically target the reservation ID.
- Instances in a Capacity Block don't count against your On-Demand Instances limits.
- For P5 instances using a custom AMI, ensure that you have the required software and configuration for EFA.
- For Amazon EKS managed node groups, see Create a managed node group with Amazon EC2 Capacity Blocks for ML. For Amazon EKS self-managed node groups, see Use Capacity Blocks for ML with self-managed nodes.

Related resources

After you create a Capacity Block, you can do the following with the Capacity Block:

- Launch instances into the Capacity Block. For more information, see Launch instances using Capacity Blocks.
- Create an Amazon EC2 Auto Scaling group. For more information, see Use Capacity Blocks for machine learning workloads in the *Amazon EC2 Auto Scaling User Guide*.

> **Note**
>
> If you use Amazon EC2 Auto Scaling or Amazon EKS, you can schedule scaling to run at the start of the Capacity Block reservation. With scheduled scaling, AWS automatically handles retries for you, so you don't need to worry about implementing retry logic to handle transient failures.

- Enhance ML workflows with AWS ParallelCluster. For more information, see Enhancing ML workflows with AWS ParallelCluster and Amazon EC2 Capacity Blocks for ML.

For more information about AWS ParallelCluster, see What is AWS ParallelCluster.

Capacity Blocks pricing and billing

With Amazon EC2 Capacity Blocks for ML, you pay only for what you reserve. The price of a Capacity Block depends on available supply and demand for Capacity Blocks at the time of purchase. You can view the price of a Capacity Block offering before you reserve it. The price of the Capacity Block is charged up front at the time the reservation is made. When you search for a Capacity Block across a range of dates, we return the lowest-priced Capacity Block offering available. After you've reserved a Capacity Block, the price doesn't change.

When you use a Capacity Block, you pay for the operating system you use when your instances are running. For more information about operating system prices, see Amazon EC2 Capacity Blocks for ML Pricing.

Billing

The price of a Capacity Block offering is charged up front. Payment is billed to your AWS account within 5 minutes to 12 hours after you purchase a Capacity Block. While your payment is processing, your Capacity Block reservation resource remains in a state of `payment-pending`. If your payment can't be processed at least 5 minutes before your block start time, or within 12 hours (whichever comes first), your Capacity Block is released and the reservation state changes to `payment-failed`.

After your payment is processed successfully, the Capacity Block resource state changes from `payment-pending` to `scheduled`. You receive an invoice that reflects the one-time upfront payment. In the invoice, you can associate the paid amount with the Capacity Block reservation ID.

When your Capacity Block reservation begins, you are billed based only on the operating system you use while your instances are running in the reservation. You can view your usage and associated charges in your anniversary bill for the month of usage in your AWS Cost and Usage Report.

> **Note**
>
> Savings Plans and Reserved Instance discounts don't apply to Capacity Blocks.

Viewing your bill

You can view your bill in the AWS Billing and Cost Management console. The upfront payment for your Capacity Block appears in the month that you purchased the reservation.

After your reservation begins, your bill shows separate lines for the block reservation used and unused time. You can use these line items to see how much time was used in your reservation. You will see only a usage charge in the line for used time if you use a premium operating system. For more information, see [Capacity Blocks pricing and billing](). There is no additional charge for unused time.

For more information, see [Viewing your bill]() in the *AWS Billing and Cost Management User Guide*.

If your Capacity Block starts in a different month then the month you purchased your reservation, the upfront price and reservation usage show up under separate billing months. In your AWS Cost and Usage Report, the Capacity Block reservation ID is listed in the **reservation/ReservationARN** line item of your upfront fee and the **lineitem/ResourceID** in your anniversary bill so that you can associate the usage to the corresponding upfront price.

Prerequisites for Capacity Blocks

You must launch instances into an AWS Region that supports the instance type that you want to use.

Capacity Blocks with **`p6-b200.48xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-2	US West (Oregon)

Capacity Blocks with **`p5.4xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)

Region code	Region name
us-west-2	US West (Oregon)
eu-west-2	Europe (London)
ap-south-1	Asia Pacific (Mumbai)
ap-northeast-1	Asia Pacific (Tokyo)
ap-southeast-2	Asia Pacific (Sydney)
sa-east-1	South America (São Paulo)

Capacity Blocks with **`p5.48xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
eu-north-1	Europe (Stockholm)
eu-west-2	Europe (London)
sa-east-1	South America (São Paulo)
ap-northeast-1	Asia Pacific (Tokyo)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-2	Asia Pacific (Sydney)
ap-southeast-3	Asia Pacific (Jakarta)

Capacity Blocks with **`p5e.48xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
eu-north-1	Europe (Stockholm)
eu-west-2	Europe (London)
sa-east-1	South America (São Paulo)
ap-northeast-1	Asia Pacific (Tokyo)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-2	Asia Pacific (Sydney)
ap-southeast-3	Asia Pacific (Jakarta)

Capacity Blocks with **`p5en.48xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
eu-north-1	Europe (Stockholm)

Region code	Region name
eu-west-2	Europe (London)
eu-south-2	Europe (Spain)
sa-east-1	South America (São Paulo)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-3	Asia Pacific (Jakarta)

Capacity Blocks with **`p4d.24xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-2	US West (Oregon)

Capacity Blocks with **`p4de.24xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-west-2	US West (Oregon)

Capacity Blocks with **`trn1.32xlarge`** instances are available in the following AWS Regions.

Region code	Region name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
eu-north-1	Europe (Stockholm)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-2	Asia Pacific (Sydney)
ap-southeast-4	Asia Pacific (Melbourne)

Capacity Blocks with **`trn2.48xlarge`** instances are available in the following AWS Region.

Region code	Region name
us-east-2	US East (Ohio)

Amazon EC2 UltraServers Capacity Blocks

Capacity Blocks with **`Trn2`** UltraServers are available in preview the following AWS Region.

Region code	Region name
us-east-2	US East (Ohio)

Capacity Blocks with **`P6e-GB200`** UltraServers are available in the following AWS local zone.

Region code	Region name
us-east-1-dfw-2a	Dallas Local Zone (N. Virginia)

> **Note**
>
> Capacity Block sizes of 64 instances are not supported for all instance types in all AWS Regions.

Find and purchase Capacity Blocks

To reserve a Capacity Block, you first need to find a block of time when capacity is available that matches your needs. To find a Capacity Block that is available to reserve, you specify the following.

- The number of instances that you need
- The duration of time you that you need the instances
- The date range that you need your reservation

To search for an available Capacity Block offering, you specify a reservation duration and instance count. You must specify reservation durations in **1-day increments up to 14 days, and in 7-day increments up to 182 days**. Each Capacity Block can have up to 64 instances, and you can have up to 256 instances across Capacity Blocks.

When you request a Capacity Block that matches your specifications, we provide the details of up to 3 available blocks. All Capacity Blocks end at 11:30AM UTC, so the blocks starting on the same day will have durations that are the closest match to your desired duration. One block will have a duration that is slightly less than your desired duration, while the other will have a duration slightly greater than your desired duration.

The offering details include the start time of the reservation, the Availability Zone for the reservation, and the price of the reservation. For more information, see Capacity Blocks pricing and billing.

You can purchase the Capacity Block offering you are shown, or you can modify your search criteria to see the other options that are available. There is no predefined expiration time for the offering, but offerings are only available on a first-come, first-served basis.

When you purchase a Capacity Block offering, you get an immediate response confirming that your Capacity Block was reserved. After confirmation, you will see a new Capacity Reservation in your account with a reservation type of `capacity-block` and a `start-date` set to the start time of the offering that you purchased. Your Capacity Block reservation is created with a state of payment-pending. After the upfront payment is successfully processed, the reservation state changes to scheduled. For more information, see Billing.

> **Note**
>
> To use P6e-GB200 UltraServer Capacity Blocks, you must be opted in to the Dallas Local Zone (N. Virginia) Local Zone.

Console

To find and purchase a Capacity Block

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 an AWS Region. This choice is important because Capacity Block sizes of 64 instances are not supported for all instance types in all Regions.
3. In the navigation pane, choose **Capacity Reservations**, **Create Capacity Block**.
4. Under **Capacity Block types**, choose either **Instances** or **UltraServers**.
5. Under **Capacity attributes**, you can define your Capacity Block search parameters. By default, the platform is Linux. If you want to select a different operating system, use the AWS CLI. For more information, see Supported platforms.
6. Under **Total capacity** (for Instances) or **UltraServer count** (for UltraServers), specify the number of instances or UltraServers you want to reserve.
7. Under **Duration**, enter the number of days or weeks you need the reservation for.
8. Under **Date range to search for Capacity Blocks**, enter the earliest date that you want your reservation to start.
9. Choose **Find Capacity Blocks**.
10. If a Capacity Block is available that meets your specifications, you see an offering under **Recommended Capacity Blocks**. If there are multiple offerings that meet your specifications, the earliest available Capacity Block offering is shown. To view other Capacity Block offerings, adjust your search inputs and choose **Find Capacity Blocks** again.
11. When you find a Capacity Block offering that you want to purchase, choose **Next**.
12. (Optional) On the **Add tags** page, choose **Add new tag**.

13. The **Review and purchase** page lists the start and end date, duration, total number of instances, and price.

> **Note**
>
> Capacity Blocks can't be canceled after you reserve them.

14. In the popup window **Purchase a Capacity Block**, type confirm, then choose **Purchase**.

AWS CLI

To find an instance Capacity Block

Use the describe-capacity-block-offerings command.

The following example finds instance Capacity Blocks.

```
aws ec2 describe-capacity-block-offerings \
--instance-type p5.48xlarge \
--instance-count 16 \
--start-date-range 2023-08-14T00:00:00Z \
--end-date-range 2023-10-22-T00:00:00Z \
--capacity-duration-hours 48
```

The following example finds UltraServer Capacity Blocks.

```
aws ec2 describe-capacity-block-offerings \
--ultraserver-type u-p6e-gb200x72 \
--ultraserver-count 1 \
--start-date-range 2023-08-14T00:00:00Z \
--end-date-range 2023-10-22-T00:00:00Z \
--capacity-duration-hours 48
```

To purchase a Capacity Block

Use the purchase-capacity-block command with the offering ID of the Capacity Block from the output of the previous example.

```
aws ec2 purchase-capacity-block \
--capacity-block-offering-id cb-0123456789abcdefg \
```

```
--instance-platform Linux/UNIX
```

PowerShell

To find Capacity Blocks

Use the Get-EC2CapacityBlockOffering cmdlet.

The following example finds instance Capacity Blocks.

```
Get-EC2CapacityBlockOffering `
-InstanceType p5.48xlarge `
-InstanceCount 16 `
-CapacityDurationHour 48 `
-StartDateRange 2023-08-14T00:00:00Z `
-EndDateRange 2023-10-22-T00:00:00Z
```

The following example finds UltraServer Capacity Blocks.

```
Get-EC2CapacityBlockOffering `
-UltraserverType u-p6e-gb200x72 `
-UltraserverCount  1 `
-CapacityDurationHour 48 `
-StartDateRange 2023-08-14T00:00:00Z `
-EndDateRange 2023-10-22-T00:00:00Z
```

To purchase a Capacity Block

Use the New-EC2EC2CapacityBlock cmdlet with the offering ID of the Capacity Block from the output of the previous example.

```
New-EC2EC2CapacityBlock `
-CapacityBlockOfferingId cb-0123456789abcdefg `
-InstancePlatform Linux/UNIX
```

Launch instances using Capacity Blocks

To use your Capacity Block, you must specify the Capacity Block reservation ID when launching instances. Launching an instance into a Capacity Block reduces the available capacity by the number of instances launched. For example, if your purchased instance capacity is eight instances and you launch four instances, the available capacity is reduced by four.

If you terminate an instance running in the Capacity Block before the reservation ends, you can launch a new instance in its place. When you stop or terminate an instance in a Capacity Block, it takes several minutes to clean up your instance before you can launch another instance to replace it. During this time, your instance will be in a stopping or `shutting-down` state. After this process is complete, your instance state will change to `stopped` or `terminated`. Then, the available capacity in your Capacity Block will update to show another instance available to use.

Requirements

- Your instance can't launch in a subnet in a different Availability Zone from the Availability Zone where your Capacity Block is located.
- Your instance can't launch using an AMI with a different platform than the platform for your Capacity Block.
- To use `P6e-GB200` UltraServer Capacity Blocks, you must be opted in to the Dallas Local Zone (N. Virginia) Local Zone.

Console

To launch instances into a Capacity Block

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, select the Region for your Capacity Block reservation.
3. From the Amazon EC2 console dashboard, choose **Launch instance**.
4. Follow the procedure to launch an instance.
5. Expand **Advanced details**, and for **Purchasing option**, choose **Capacity Blocks**. Then do one of the following:
   - To launch the instances into a specific Capacity Block, for **Capacity Reservation** choose **Specify Capacity Reservation**, and then select the Capacity Block.
   - (*UltraServers only*) To launch the instances into an UltraServer Capacity Block resource group, for **Capacity Reservation** choose **Specify Capacity Reservation resource group**, and then select the resource group.
6. Choose **Launch instance**.

AWS CLI

To launch instances using into a Capacity Block

Use the [run-instances](https://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html) command with the `instance-market-options MarketType` option.

The following example launches an instance into a specific Capacity Block.

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--count 1 \
--instance-type p5.48xlarge \
--key-name my-key-pair \
--subnet-id subnet-0abcdef1234567890 \
--instance-market-options MarketType='capacity-block' \
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_block_id}
```

The following example launches an instance into an UltraServer Capacity Block resource group.

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--count 1 \
--instance-type p6e-gb200.36xlarge \
--key-name my-key-pair \
--subnet-id subnet-0abcdef1234567890 \
--instance-market-options MarketType='capacity-block' \
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationResourceGroupArn=resource_group_arn}
```

PowerShell

To launch instances into a Capacity Block

Use the [New-EC2Instance](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2Instance.html) cmdlet with the `-InstanceMarketOption` option defined as follows.

```
$marketoption = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketoption.MarketType = "capacity-block"
```

The following example launches an instance into a specific Capacity Block.

```
New-EC2Instance `
```

```
-ImageId ami-0abcdef1234567890 `
-InstanceType p5.48xlarge `
-KeyName "my-key-pair" `
-SubnetId subnet-0abcdef1234567890 `
-InstanceMarketOptions $marketoption `
-CapacityReservationTarget_CapacityReservationId capacity_block_id
```

The following example launches an instance into an UltraServer Capacity Block resource group.

```
New-EC2Instance `
-ImageId ami-0abcdef1234567890 `
-InstanceType p6e-gb200.36xlarge `
-KeyName "my-key-pair" `
-SubnetId subnet-0abcdef1234567890 `
-InstanceMarketOptions $marketoption `
-CapacityReservationTarget_CapacityReservationResourceGroupArn "resource_group_arn"
```

Related resources

- To create a launch template targeting a Capacity Block, see the section called “Launch templates”.
- To launch instances into a Capacity Block using EC2 Fleet, see Tutorial: Configure your EC2 Fleet to launch instances into Capacity Blocks.
- To set up an EKS managed node group with a Capacity Block, see Create a managed node group with Capacity Blocks for ML in the Amazon EKS User Guide.
- To set up AWS ParallelCluster using a Capacity Block, see ML on AWS ParallelCluster.

View Capacity Blocks

After you reserve a Capacity Block, you can view the Capacity Block reservation in your AWS account. You can view the `start-date` and `end-date` to see when your reservation will begin and end. Before a Capacity Block reservation begins, the available capacity appears as zero. You can see how many instances will be available in your Capacity Block by the tag value for the tag key `aws:ec2capacityreservation:incrementalRequestedQuantity`.

When a Capacity Block reservation begins, the reservation state changes from `scheduled` to `active`. We emit an event through Amazon EventBridge to notify you that the Capacity Block is available to use. For more information, see Monitor Capacity Blocks using EventBridge.

Capacity Blocks have the following states:

- `payment-pending` – The upfront payment hasn't been processed yet.
- `payment-failed`—The payment couldn't be processed in the 12 hour time frame. Your Capacity Block was released.
- `scheduled` – The payment was processed and the Capacity Block reservation hasn't started yet.
- `active` – The reserved capacity is available for your use.
- `expired` – The Capacity Block reservation expired automatically at the date and time specified in your reservation request. The reserved capacity is no longer available for your use.

Console

To view Capacity Blocks

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**.
3. On the **Capacity Reservations overview** page, you see a resource table with details about all of your Capacity Reservation resources. To find your Capacity Blocks reservations, select **Capacity Blocks** from the dropdown list above **Capacity Reservation ID**. In the table, you can see information about your Capacity Blocks such as start and end dates, duration, and status.
4. For more details about a Capacity Block, select the reservation ID for the Capacity Block that you want to view. The **Capacity Reservation details** page displays all the properties of the reservation and the number of instances in use and available in the Capacity Block.

> **Note**
>
> Before a Capacity Block reservation begins, the available capacity appears as zero. You can see how many instances will be available when the Capacity Block reservation starts by using the following tag value for the tag key: `aws:ec2capacityreservation:incrementalRequestedQuantity`.

AWS CLI

To view Capacity Blocks

By default, when you use the [describe-capacity-reservations] command both On-Demand Capacity Reservations and Capacity Block reservations are listed. To view only your Capacity Block reservations, filter for reservations of type `capacity-block`.

```
aws ec2 describe-capacity-reservations \
    --filters Name=reservation-type,Values=capacity-block
```

PowerShell

To view Capacity Blocks

Use the [Get-EC2CapacityReservation] cmdlet. By default, both On-Demand Capacity Reservations and Capacity Block reservations are listed. To view only your Capacity Block reservations, filter for reservations of type `capacity-block`.

```
Get-EC2CapacityReservation `
    -Filter @{Name="reservation-type"; Values="capacity-block"}
```

Extend Capacity Blocks

With Capacity Blocks, you reserve compute capacity for your workloads, ensuring availability and consistency. To accommodate your changing needs, you can extend the duration of your existing Capacity Blocks as required.

To extend a Capacity Block, it must have a status of `active` or `scheduled`, and have no extensions that are `payment-pending`. You can request to extend the duration of your Capacity Block up to a minimum of 1 hour or a maximum of 56 days before it expires. You can extend your Capacity Block by 1-day increments up to 14 days, and 7-day increments up to 182 days (26 weeks) total. When you extend your Capacity Block, its end date will be updated so that your instances can continue to run without disruption.

- There is no limit to the number of extensions you can apply to a Capacity Block
- Your Capacity Reservation ID will remain the same after extending the block
- Capacity Blocks can only be extended if there is sufficient capacity available to support them, which is not guaranteed.

Billing

The price of a Capacity Block offering is charged up front. The extension will remain in `payment-pending` until the bill is paid. If your payment can't be processed within 12 hours, or up to 35 minutes before the Capacity Block is scheduled to end (whichever comes first), your extension is not successful and the status changes to `payment-failed`. Your Capacity Block reservation will remain `active` and will be terminated on the original end date.

After your payment is processed successfully, the Capacity Block extension status changes to `payment-succeeded` and the end date of the Capacity Block reservation will be updated to the new end date. The details of your extension can be viewed in the **Capacity Block Extension details** section of the console, or by using the describe-capacity-block-extension-history command.

Extend your Capacity Block

Console

To extend a Capacity Block

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**.
3. On the **Capacity Reservations overview** page, you see a resource table with details about all of your Capacity Reservations resources. Select the reservation ID for the Capacity Block that you want to extend.
4. From the **Actions** drop down menu, choose **Extend Capacity Block**.
5. Under **Duration**, enter the number of days or weeks you need to extend the reservation for.
6. Choose **Find Capacity Block**.
7. If a Capacity Block is available that meets your specifications, an offering appears under **Recommended Capacity Blocks**. To view other Capacity Block offerings, adjust your search inputs and choose **Find Capacity Blocks** again.
8. When you find a Capacity Block offering that you want to purchase, choose **Extend**.
9. In the pop-up window **Extend Capacity Block**, enter *confirm*, then choose **Extend**.

AWS CLI

To find a Capacity Block extension

Use the [describe-capacity-block-extension-offerings] command. The following example searches for a 48 hour Capacity Block extension for the specified Reservation.

```
aws ec2 describe-capacity-block-extension-offerings \
    --capacity-reservation-id cr-1234567890abcdefg \
    --capacity-block-extension-duration-hours 48
```

To extend a Capacity Block

Use the [purchase-capacity-block-extension] command. Specify the extension offering ID from the output of the previous example.

```
aws ec2 purchase-capacity-block-extension \
    --capacity-block-extension-offering-id cbe-0123456789abcdefg \
    --capacity-reservation-id cr-1234567890abcdefg
```

PowerShell

To find a Capacity Block extension

Use the [Get-EC2CapacityBlockExtensionOffering] cmdlet. The following example searches for a 48 hour Capacity Block extension for the specified Reservation.

```
Get-EC2CapacityBlockExtensionOffering `
    -CapacityReservationId cr-1234567890abcdefg `
    -CapacityBlockExtensionDurationHour 48
```

To extend a Capacity Block

Use the [Invoke-EC2CapacityBlockExtension] cmdlet. Specify the extension offering ID from the output of the previous example.

```
Invoke-EC2CapacityBlockExtension `
    -CapacityBlockExtensionOfferingId cbe-0123456789abcdefg `
    -CapacityReservationId cr-1234567890abcdefg
```

Create a resource group for UltraServer Capacity Blocks

You can use AWS Resource Groups to create logical collections of UltraServer Capacity Blocks. After you create the resource group, you can add UltraServer Capacity Blocks that you own in your account. After you add the UltraServer Capacity Blocks, you can target instances launches to the resource group instead of the individual Capacity Blocks. Instances that target a resource group match with any UltraServer Capacity Blocks in the group that has matching attributes and available capacity. If the resource group does not have an UltraServer Capacity Block with matching attributes and available capacity, the instance launch fails.

If an UltraServer Capacity Block is removed from a resource group while it has running instances, those instances continue to run in the Capacity Block. If an UltraServer Capacity Block in a group ends while it has running instances, the instances are terminated.

You can't add instance Capacity Blocks to a resource group.

To create a resource group for UltraServer Capacity Blocks, use one of the following methods.

AWS CLI

To create a resource group for UltraServer Capacity Blocks

Use the create-group AWS CLI command, and for `--configuration`, specify the following:

```
{
  "Configuration": [
    {
      "Type": "AWS::EC2::CapacityReservationPool",
      "Parameters": [
        {
          "Name": "instance-type",
          "Values": [
            "instance_type"
          ]
        },
        {
          "Name": "reservation-type",
          "Values": [
            "capacity-block"
          ]
        }
      ]
    },
    {
      "Type": "AWS::ResourceGroups::Generic",
      "Parameters": [
        {
          "Name": "allowed-resource-types",
```

```
            "Values": [
              "AWS::EC2::CapacityReservation"
            ]
          }
        ]
      }
    ]
}
```

PowerShell

To create a resource group for UltraServer Capacity Blocks

Use the New-RGGroup cmdlet. For `-Configuration`, specify the following:

```
{
  "Configuration": [
    {
      "Type": "AWS::EC2::CapacityReservationPool",
      "Parameters": [
        {
          "Name": "instance-type",
          "Values": [
            "instance_type"
          ]
        },
        {
          "Name": "reservation-type",
          "Values": [
            "capacity-block"
          ]
        }
      ]
    },
    {
      "Type": "AWS::ResourceGroups::Generic",
      "Parameters": [
        {
          "Name": "allowed-resource-types",
          "Values": [
            "AWS::EC2::CapacityReservation"
          ]
        }
      ]
```

```
        }
    ]
}
```

After you create a resource group for UltraServer Capacity Block, use one of the following methods to add existing UltraServer Capacity Blocks to it.

AWS CLI

To add an UltraServer Capacity Block to a resource group

Use the group-resources command. For `--group` specify the name of the resource group you created. For `--resource-arns`, specify the ARNs of the UltraServer Capacity Blocks to add.

```
aws resource-groups group-resources \
--group MyCRGroup \
--resource-arns capacity_block_arn
```

PowerShell

To add an UltraServer Capacity Block to a resource group

Use the Add-RGResource cmdlet. For `-Group` specify the name of the resource group you created. For `-ResourceArn` , specify the ARNs of the UltraServer Capacity Blocks to add.

The following example adds two Capacity Reservations to the specified group.

```
Add-RGResource `
-Group MyCRGroup `
-ResourceArn capacity_block_arn
```

Monitor Capacity Blocks using EventBridge

When your Capacity Block reservation starts, Amazon EC2 will emit an event through EventBridge that indicates your capacity is ready to use. Forty minutes before your Capacity Block reservation ends, you receive another EventBridge event that tells you that any instances running in the reservation will begin to terminate in 10 minutes. For more information about EventBridge events, see Amazon EventBridge Events.

The following event structures for events emitted for Capacity Blocks:

Capacity Block Delivered

The following example shows an event for Capacity Block Delivered.

```
{
  "customer_event_id": "[Capacity Reservation Id]-delivered",
  "detail_type": "Capacity Block Reservation Delivered",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

Capacity Block Expiration Warning

The following example shows an event for Capacity Block Expiration Warning.

```
{
  "customer_event_id": "[Capacity Reservation Id]-approaching-expiry",
  "detail_type": "Capacity Block Reservation Expiration Warning",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

Logging Capacity Blocks API calls with AWS CloudTrail

Capacity Blocks is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Capacity Blocks. CloudTrail captures API calls for Capacity Blocks as events. The calls captured include calls from the Capacity Blocks console and code calls to the Capacity Blocks API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Capacity Blocks. 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 Capacity Blocks, 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.

Capacity Blocks information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Capacity Blocks, 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 Capacity Blocks, 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 AWS 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 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

All Capacity Blocks actions are logged by CloudTrail and are documented in the Amazon EC2 API Reference. For example, calls to the `CapacityBlockScheduled`, and `CapacityBlockActive` 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 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 userIdentity element.

Understanding Capacity Blocks 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 aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following examples show CloudTrail log entries for:

- TerminateCapacityBlocksInstances
- CapacityBlockPaymentFailed
- CapacityBlockScheduled
- CapacityBlockActive
- CapacityBlockFailed
- CapacityBlockExpired

> **Note**
>
> Some fields have been redacted from the examples for data privacy.

TerminateCapacityBlocksInstances


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "TerminateCapacityBlockInstances",
```

```
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
    }
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      }
}
```

CapacityBlockPaymentFailed

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockPaymentFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
```

```
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "payment-failed"
      }
}
```

CapacityBlockScheduled

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockScheduled",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
```

```
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "scheduled"
      }
}
```

CapacityBlockActive

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockActive",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "active"
      }
 }
```

CapacityBlockFailed

```
{
  "eventVersion": "1.05",
```

```
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "failed"
      }
 }
```

CapacityBlockExpired

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockExpired",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
```

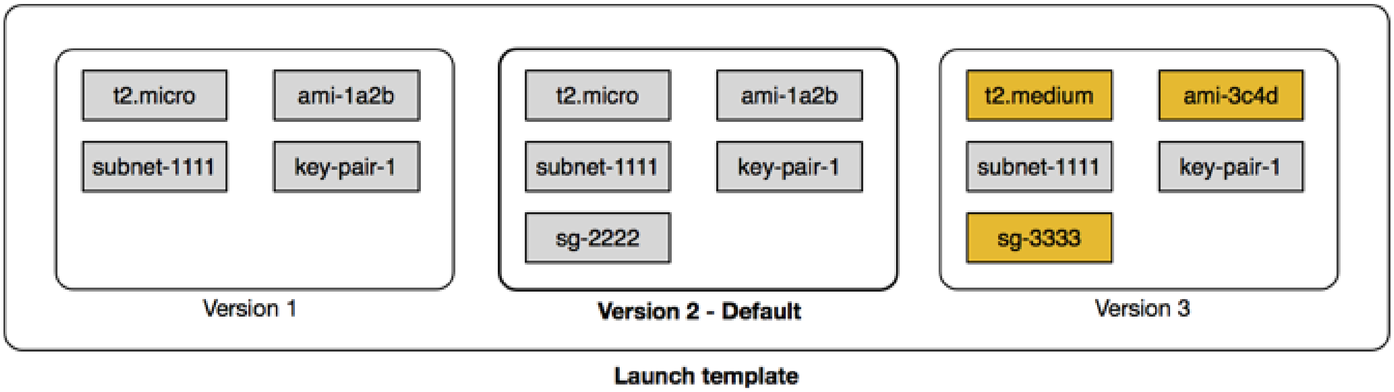
```
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "expired"
      }
}
```


Store instance launch parameters in Amazon EC2 launch templates

You can use an Amazon EC2 *launch template* to store instance launch parameters so that you don't have to specify them every time you launch an Amazon EC2 instance. For example, you can create a launch template that stores 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 instead of entering the parameters again.

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 don't 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

Restrictions for Amazon EC2 launch templates

The following restrictions apply to launch templates and launch template versions:

- **Quotas** – To view the quotas for your launch templates and launch template versions, open the Service Quotas console or use the list-service-quotas AWS CLI command. Each AWS account can have up to a maximum of 5,000 launch templates per Region and up to 10,000 versions per launch template. Your accounts might have different quotas based on their age and usage history.
- **Parameters are optional** – Launch template parameters are optional. However, you must ensure that your instance launch request includes all required parameters. For example, if your launch template does not include an AMI ID, you must specify an AMI ID when launching an instance with this launch template.
- **Parameters not validated** – Launch template parameters are not fully validated when you create the launch template. If you specify incorrect values or use unsupported parameter combinations, instances will fail to launch using this launch template. To avoid issues, make sure to specify

correct values and use supported parameter combinations. For example, to launch an instance in a placement group, you must specify a supported instance type.

- **Tags** – You can tag a launch template, but you can't tag a launch template version.
- **Immutable** – Launch templates are immutable. To modify a launch template, you must create a new version of the launch template.
- **Version numbers** – Launch template versions are numbered in the order in which they are created. When you create a launch template version, you can't specify the version number yourself.

IAM permissions required for Amazon EC2 launch templates

You can use IAM permissions to control whether users can list, view, create, or delete launch templates or launch template versions.

> **Important**
>
> You can't use resource-level permissions to restrict the resources that users can specify in a launch template when they create a launch template or launch template version. Therefore, make sure that only trusted administrators are granted permission to create launch templates and launch template versions.

You must grant anyone that will use a launch template the permissions required to create and access the resources that are specified in the launch template. For example:

- To launch an instance from a shared private Amazon Machine Image (AMI), the user must have launch permission for the AMI.
- To create EBS volumes with tags from existing snapshots, the user must have read access to the snapshots, and permissions to create and tag volumes.

Contents

- [Control versioning permissions]
- [Control access to tags on launch templates]

ec2:CreateLaunchTemplate

To create a launch template in the console or by using the APIs, the principal must have the `ec2:CreateLaunchTemplate` permission in an IAM policy. Whenever possible, use tags to help you control access to the launch templates in your account.

For example, the following IAM policy statement gives the principal permission to create launch templates only if the template uses the specified tag (*purpose*=*testing*).

```
{
    "Sid": "IAMPolicyForCreatingTaggedLaunchTemplates",
    "Action": "ec2:CreateLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

Principals who create launch templates might need some related permissions, such as:

- **ec2:CreateTags** – To add tags to the launch template during the `CreateLaunchTemplate` operation, the `CreateLaunchTemplate` caller must have the `ec2:CreateTags` permission in an IAM policy.
- **ec2:RunInstances** – To launch EC2 instances from the launch template that they created, the principal must also have the `ec2:RunInstances` permission in an IAM policy.

For resource-creating actions that apply tags, users must have the `ec2:CreateTags` permission. The following IAM policy statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `CreateLaunchTemplate`. Users cannot tag existing launch templates or any other resources. For more information, see [Grant permission to tag Amazon EC2 resources during creation].

```
{
```

```
    "Sid": "IAMPolicyForTaggingLaunchTemplatesOnCreation",
    "Action": "ec2:CreateTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateLaunchTemplate"
        }
    }
}
```

The IAM user who creates a launch template doesn't automatically have permission to use the launch template that they created. Like any other principal, the launch template creator needs to get permission through an IAM policy. If an IAM user wants to launch an EC2 instance from a launch template, they must have the `ec2:RunInstances` permission. When granting these permissions, you can specify that users can only use launch templates with specific tags or specific IDs. You can also control the AMI and other resources that anyone using launch templates can reference and use when launching instances by specifying resource-level permissions for the `RunInstances` call. For example policies, see Launch templates.

ec2:DescribeLaunchTemplates

To list and view launch templates in the account, the principal must have the `ec2:DescribeLaunchTemplates` permission in an IAM policy. Because `Describe` actions do not support resource-level permissions, you must specify them without conditions and the value of the resource element in the policy must be `"*"`.

For example, the following IAM policy statement gives the principal permission to list and view all launch templates in the account.

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplates",
    "Action": "ec2:DescribeLaunchTemplates",
    "Effect": "Allow",
    "Resource": "*"
}
```

ec2:DescribeLaunchTemplateVersions

Principals who list and view launch templates should also have the `ec2:DescribeLaunchTemplateVersions` permission to retrieve the entire set of attributes that make up the launch templates.

To list and view launch template versions in the account, the principal must have the `ec2:DescribeLaunchTemplateVersions` permission in an IAM policy. Because `Describe` actions do not support resource-level permissions, you must specify them without conditions and the value of the resource element in the policy must be `"*"`.

For example, the following IAM policy statement gives the principal permission to list and view all launch template versions in the account.

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplateVersions",
    "Effect": "Allow",
    "Action": "ec2:DescribeLaunchTemplateVersions",
    "Resource": "*"
}
```

ec2:DeleteLaunchTemplate

> **Important**
>
> Use caution when giving principals permission to delete a resource. Deleting a launch template might cause a failure in an AWS resource that relies on the launch template.

To delete a launch template, the principal must have the `ec2:DeleteLaunchTemplate` permission in an IAM policy. Whenever possible, use tag-based condition keys to limit the permissions.

For example, the following IAM policy statement gives the principal permission to delete launch templates only if the template has the specified tag (*`purpose`*=*`testing`*).

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplates",
    "Action": "ec2:DeleteLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
```

```
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

Alternatively, you can use ARNs to identify the launch template that the IAM policy applies to.

A launch template has the following ARN.

```
"Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/lt-09477bcd97b0d310e"
```

You can specify multiple ARNs by enclosing them in a list, or you can specify a `Resource` value of `"*"` without the `Condition` element to allow the principal to delete any launch template in the account.

Control versioning permissions

For trusted administrators, you can grant access for creating and deleting versions of a launch template, and for changing the default version of a launch template, by using IAM policies similar to the following examples.

> **Important**
>
> Be cautious when giving principals permission to create launch template versions or modify launch templates.
>
> - When you create a launch template version, you affect any AWS resources that allow Amazon EC2 to launch instances on your behalf with the `Latest` version.
> - When you modify a launch template, you can change which version is the `Default` and therefore affect any AWS resources that allow Amazon EC2 to launch instances on your behalf with this modified version.
>
> You also need to be cautious in how you handle AWS resources that interact with the `Latest` or `Default` launch template version, such as EC2 Fleet and Spot Fleet. When a different launch template version is used for `Latest` or `Default`, Amazon EC2 does not re-check user permissions for actions to be completed when launching new instances to fulfil the fleet's target capacity because there is no user interaction with the AWS

resource. By granting a user permission to call the `CreateLaunchTemplateVersion` and `ModifyLaunchTemplate` APIs, the user is effectively also granted the `iam:PassRole` permission if they point the fleet to a different launch template version that contains an instance profile (a container for an IAM role). It means that a user can potentially update a launch template to pass an IAM role to an instance even if they don't have the `iam:PassRole` permission. You can manage this risk by being careful when granting permissions to who can create and manage launch template versions.

ec2:CreateLaunchTemplateVersion

To create a new version of a launch template, the principal must have the `ec2:CreateLaunchTemplateVersion` permission for the launch template in an IAM policy.

For example, the following IAM policy statement gives the principal permission to create launch template versions only if the version uses the specified tag (*`environment`*=*`production`*). Alternatively, you can specify one or multiple launch template ARNs, or you can specify a `Resource` value of `"*"` without the `Condition` element to allow the principal to create versions of any launch template in the account.

```
{
    "Sid": "IAMPolicyForCreatingLaunchTemplateVersions",
    "Action": "ec2:CreateLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:DeleteLaunchTemplateVersion

Important

As always, you should exercise caution when giving principals permission to delete a resource. Deleting a launch template version might cause a failure in an AWS resource that relies on the launch template version.

To delete a launch template version, the principal must have the `ec2:DeleteLaunchTemplateVersion` permission for the launch template in an IAM policy.

For example, the following IAM policy statement gives the principal permission to delete launch template versions only if the version uses the specified tag (*environment*=*production*). Alternatively, you can specify one or multiple launch template ARNs, or you can specify a `Resource` value of `"*"` without the `Condition` element to allow the principal to delete versions of any launch template in the account.

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplateVersions",
    "Action": "ec2:DeleteLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:ModifyLaunchTemplate

To change the `Default` version that is associated with a launch template, the principal must have the `ec2:ModifyLaunchTemplate` permission for the launch template in an IAM policy.

For example, the following IAM policy statement gives the principal permission to modify launch templates only if the launch template uses the specified tag (*environment*=*production*). Alternatively, you can specify one or multiple launch template ARNs, or you can specify a `Resource` value of `"*"` without the `Condition` element to allow the principal to modify any launch template in the account.

```
{
    "Sid": "IAMPolicyForModifyingLaunchTemplates",
    "Action": "ec2:ModifyLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
```

```
    }
}
```

Control access to tags on launch templates

You can use condition keys to limit tagging permissions when the resource is a launch template. For example, the following IAM policy allows removing only the tag with the *temporary* key from launch templates in the specified account and Region.

```
{
    "Sid": "IAMPolicyForDeletingTagsOnLaunchTemplates",
    "Action": "ec2:DeleteTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
    "Condition": {
        "ForAllValues:StringEquals": {
            "aws:TagKeys": ["temporary"]
        }
    }
}
```

For more information about conditions keys that you can use to control the tag keys and values that can be applied to Amazon EC2 resources, see Control access to specific tags.

Use Amazon EC2 launch templates to control launching Amazon EC2 instances

You can control the configuration of your Amazon EC2 instances by specifying that users can only launch instances if they use a launch template, and that they can only use a specific launch template. You can also control who can create, modify, describe, and delete launch templates and launch template versions.

Use launch templates to control launch parameters

A launch template can contain all or some of the parameters to configure an instance at launch. However, 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 can't remove launch template parameters during launch (for example, you can't 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, users must have permission to use the `ec2:RunInstances` action. 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.

Control the use of launch templates

By default, users do not have permissions to work with launch templates. You can create a 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.

Take care when granting users permissions to use the `ec2:CreateLaunchTemplate` and `ec2:CreateLaunchTemplateVersion` actions. You can't 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.

Important security concerns when using launch templates with EC2 Fleet or Spot Fleet

To use launch templates, you must grant your users permissions to create, modify, describe, and delete launch templates and launch template versions. You can control who can create launch templates and launch template versions by controlling access to the `ec2:CreateLaunchTemplate` and `ec2:CreateLaunchTemplateVersion` actions. You can also control who can modify launch templates by controlling access to the `ec2:ModifyLaunchTemplate` action.

> **Important**
>
> If an EC2 Fleet or Spot Fleet is configured to use the Latest or Default launch template version, the fleet is not aware if Latest or Default are later changed to point to a different launch template version. When a different launch template version is used for Latest or Default, Amazon EC2 does not re-check permissions for actions to be completed when launching new instances to fulfil the fleet's target capacity. This is an important consideration when granting permissions to who can create and manage launch template versions, particularly the `ec2:ModifyLaunchTemplate` action that allows a user to change the Default launch template version.

By granting a user permission to use the EC2 actions for the launch template APIs, the user is effectively also granted the `iam:PassRole` permission if they create or update an EC2 Fleet or Spot Fleet to point to a different launch template version that contains an instance profile (a container for an IAM role). It means that a user can potentially update a launch template to pass an IAM role to an instance even if they don't have the `iam:PassRole` permission. For more information and an example IAM policy, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

For more information, see Control the use of launch templates and Example: Work with launch templates.

Create an Amazon EC2 launch template

You can create an Amazon EC2 launch template by specifying your own values for the instance configuration parameters, or by getting the values from an existing launch template or Amazon EC2 instance.

You don't need to specify a value for every parameter in the launch template; you need only specify one instance configuration parameter to create a launch template. To indicate parameters that you choose not to specify, select **Don't include in launch template** when using the console. When using a command line tool, don't include the parameters to indicate that you're choosing not to specify them in the launch template.

If you want to specify an AMI in the launch template, you can either select an AMI, or specify a Systems Manager parameter that will point to an AMI on instance launch.

When an instance is launched with a launch template, the values that are specified in the launch template are used to configure the corresponding instance parameters. If a value isn't specified in the launch template, then the default value for the corresponding instance parameter is used.

Tasks

- Create a launch template by specifying parameters
- Create a launch template from an existing launch template
- Create a launch template from an instance
- Use a Systems Manager parameter instead of an AMI ID

Create a launch template by specifying parameters

To create a launch template, you must specify the launch template name and at least one instance configuration parameter.

For a description of each parameter, see Reference for Amazon EC2 instance configuration parameters.

Console

To create a launch template

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. Under **Launch template name and description**, do the following:
   a. For **Launch template name**, enter a descriptive name for the launch template.
   b. For **Template version description**, provide a brief description of this version of the launch template.
   c. To tag the launch template on creation, expand **Template tags**, choose **Add new tag**, and then enter a tag key and value pair. Choose **Add new tag** again for each additional tag to add.

> **Note**
>
> To tag the resources that are created when an instance is launched, you must specify the tags under **Resource tags**. For more information, see Step 9 in this procedure.

4. Under **Application and OS Images (Amazon Machine Image)**, you can either keep **Don't include in launch template** selected, or choose the operating system (OS) for the instance, and then choose an AMI. Alternatively, you can specify a Systems Manager parameter instead of specifying an AMI. For more information, see [Use a Systems Manager parameter instead of an AMI ID](#).

   An AMI is a template that contains the operating system and software required to launch an instance.

5. Under **Instance type**, you can either keep **Don't include in launch template** selected, or select an instance type, or specify instance attributes and let Amazon EC2 identify the instance types with those attributes.

   > **Note**
   >
   > Specifying instance attributes is supported only when the launch template is used by Auto Scaling groups, EC2 Fleet, and Spot Fleet to launch instances. For more information, see [Create mixed instances group using attribute-based instance type selection](#) and [Specify attributes for instance type selection for EC2 Fleet or Spot Fleet](#).
   > If you plan to use the launch template in the [launch instance wizard](#) or with the [RunInstances API](#), you can't specify instance type attributes.

   The instance type determines the hardware configuration (CPU, memory, storage, and networking capacity) and size of the host computer used for an instance.

   If you're not sure which instance type to choose, you can do the following:

   - Choose **Compare instance types** to compare different instance types by the following attributes: number of vCPUs, architecture, amount of memory (GiB), amount of storage (GB), storage type, and network performance.

- Choose **Get advice** to get guidance and suggestions for instance types from the EC2 instance type finder. For more information, see Get recommendations from EC2 instance type finder.

> **Note**
>
> Depending on when you created your account, you might be eligible to use Amazon EC2 under the Free Tier.
> If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type, or the **t3.micro** instance type in Regions where **t2.micro** is unavailable. Be aware that when you launch a **t3.micro** instance, it defaults to **Unlimited** mode, which might incur additional charges based on CPU usage. If an instance type can be used under the Free Tier, it is labeled **Free tier eligible**.
> If you created your AWS account on or after July 15, 2025, you can use **t3.micro**, **t3.small**, **t4g.micro**, **t4g.small**, **c7i-flex.large**, and **m7i-flex.large** instance types for 6 months or until your credits are used up.
> For more information, see Free Tier benefits before and after July 15, 2025.

6. Under **Key pair (login)**, for **Key pair name**, either keep **Don't include in launch template** selected, or choose an existing key pair, or create a new one.
7. Under **Network settings**, you can either keep **Dont include in launch template** selected, or you can specify values for the various network settings.
8. Under **Configure storage**, if you specified an AMI in the launch template, the AMI includes one or more volumes of storage, including the root volume (**Volume 1 (AMI Root)**. You can optionally specify additional volumes to attach to the instance. To add a new volume, choose **Add new volume**.
9. Under **Resource tags**, to tag the resources that are created when an instance is launched, choose **Add tag**, and then enter a tag key and value pair. For **Resource types**, specify the resources to tag on creation. You can specify the same tag for all the resources, or specify different tags for different resources. Choose **Add tag** again for each additional tag to add.

   You can specify tags for the following resources that are created when a launch template is used:

   - Instances

- Volumes
- Elastic graphics
- Spot Instance requests
- Network interfaces

> **Note**
>
> To tag the launch template itself, you must specify the tags under **Template tags**. For more information, see Step 3 in this procedure.

10. For **Advanced details**, expand the section to view the fields and optionally specify any additional parameters for your instance.
11. Use the **Summary** panel to review your launch template configuration. You can navigate to any section by choosing its link and then make any necessary changes.
12. When you're ready to create your launch template, choose **Create launch template**.

AWS CLI

To create a launch template

Use the create-launch-template command.

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --version-description WebVersion1 \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

The following is example JSON that specifies the launch template data for the instance configuration. Save the JSON to a file and include it in the `--launch-template-data` parameter as shown in the example command.

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
```

```
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":2
    }
}
```

The following is example output.

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

PowerShell

To create a launch template

Use the New-EC2LaunchTemplate cmdlet.

```
$launchTemplateData = [Amazon.EC2.Model.RequestLaunchTemplateData]@{
    ImageId = 'ami-0abcdef1234567890'
    InstanceType = 'r5.4xlarge'
    NetworkInterfaces = @(

 [Amazon.EC2.Model.LaunchTemplateInstanceNetworkInterfaceSpecificationRequest]@{
```

```
            AssociatePublicIpAddress = $true
            DeviceIndex = 0
            Ipv6AddressCount = 1
            SubnetId = 'subnet-0abcdef1234567890'
        }
    )
    TagSpecifications = @(
        [Amazon.EC2.Model.LaunchTemplateTagSpecificationRequest]@{
            ResourceType = 'instance'
            Tags = [Amazon.EC2.Model.Tag]@{
                Key = 'Name'
                Value = 'webserver'
            }
        }
    )
    CpuOptions = [Amazon.EC2.Model.LaunchTemplateCpuOptionsRequest]@{
        CoreCount = 4
        ThreadsPerCore = 2
    }
}
$tagSpecificationData = [Amazon.EC2.Model.TagSpecification]@{
    ResourceType = 'launch-template'
    Tags = [Amazon.EC2.Model.Tag]@{
        Key = 'purpose'
        Value = 'production'
    }
}
New-EC2LaunchTemplate -LaunchTemplateName 'TemplateForWebServer' `
    -VersionDescription 'WebVersion1' `
    -LaunchTemplateData $launchTemplateData `
    -TagSpecification $tagSpecificationData
```

The following is example output.

```
CreatedBy            : arn:aws:iam::123456789012:root
CreateTime           : 9/19/2023 16:57:55
DefaultVersionNumber : 1
LatestVersionNumber  : 1
LaunchTemplateId     : lt-01238c059eEXAMPLE
LaunchTemplateName   : TemplateForWebServer
Tags                 : {purpose}
```

Create a launch template from an existing launch template

You can clone an existing launch template and then adjust the parameters to create a new launch template. However, you can only do this when using the Amazon EC2 console. The AWS CLI does not support cloning a template. For a description of each parameter, see Reference for Amazon EC2 instance configuration parameters.

Console

To create a launch template from an existing launch template

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 this version of the launch template.
5. To tag the launch template on creation, expand **Template tags**, choose **Add new 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

You can clone the parameters of an existing Amazon EC2 instance and then adjust the parameters to create a launch template. For a description of each parameter, see Reference for Amazon EC2 instance configuration parameters.

Console

To create a launch template 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, and choose **Actions**, **Image and templates**, **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

- Use the get-launch-template-data command and specify the instance ID. You can use the output as a base to create a new launch template or launch template version. By default, the output includes a top-level `LaunchTemplateData` object, which cannot be specified in your launch template data. Use the `--query` option to exclude this object.

```
aws ec2 get-launch-template-data \
    --instance-id i-0123d646e8048babc \
    --query "LaunchTemplateData"
```

The following is example output.

```
    {
        "Monitoring": {},
        "ImageId": "ami-8c1be5f6",
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "DeleteOnTermination": true
                }
            }
```

```
        ],
        "EbsOptimized": false,
        "Placement": {
            "Tenancy": "default",
            "GroupName": "",
            "AvailabilityZone": "us-east-1a"
        },
        "InstanceType": "t2.micro",
        "NetworkInterfaces": [
            {
                "Description": "",
                "NetworkInterfaceId": "eni-35306abc",
                "PrivateIpAddresses": [
                    {
                        "Primary": true,
                        "PrivateIpAddress": "10.0.0.72"
                    }
                ],
                "SubnetId": "subnet-7b16de0c",
                "Groups": [
                    "sg-7c227019"
                ],
                "Ipv6Addresses": [
                    {
                        "Ipv6Address": "2001:db8:1234:1a00::123"
                    }
                ],
                "PrivateIpAddress": "10.0.0.72"
            }
        ]
    }
```

You can write the output directly to a file, for example:

```
aws ec2 get-launch-template-data \
    --instance-id i-0123d646e8048babc \
    --query "LaunchTemplateData" >> instance-data.json
```

To create a launch template using launch template data

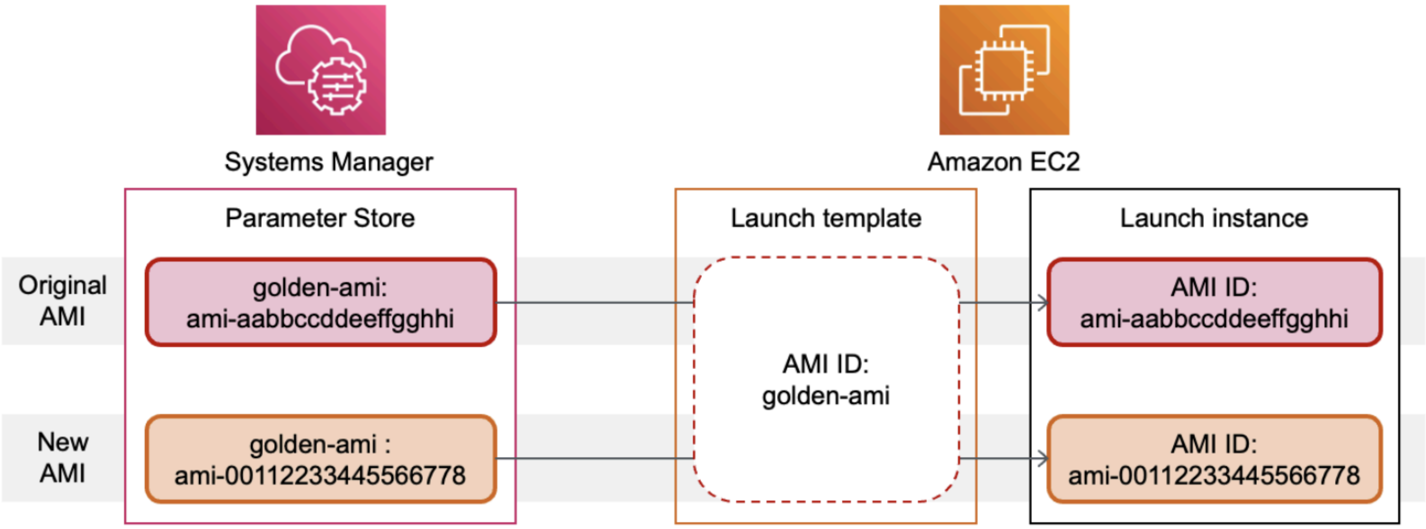
- Use the create-launch-template command to create a launch template using the output from the previous procedure. For more information about creating a launch template using the AWS CLI, see Create a launch template by specifying parameters.

Use a Systems Manager parameter instead of an AMI ID

Instead of specifying an AMI ID in your launch templates, you can specify an AWS Systems Manager parameter. If the AMI ID changes, you can update the AMI ID in one place by updating the Systems Manager parameter in the Systems Manager Parameter Store. Parameters can also be shared with other AWS accounts. You can centrally store and manage AMI parameters in one account and share them with every other account that needs to reference them. By using a Systems Manager parameter, all your launch templates can be updated in a single action.

A Systems Manager parameter is a user-defined key-value pair that you create in the AWS Systems Manager Parameter Store. The Parameter Store provides a central place to store your application configuration values.

In the following diagram, the `golden-ami` parameter is first mapped to the original AMI `ami-aabbccddeeffgghhi` in the Parameter Store. In the launch template, the value for the AMI ID is `golden-ami`. When an instance is launched using this launch template, the AMI ID resolves to `ami-aabbccddeeffgghhi`. Later, the AMI is updated resulting in a new AMI ID. In the Parameter Store, the `golden-ami` parameter is mapped to the new `ami-00112233445566778`. *The launch template remains unchanged.* When an instance is launched using this launch template, the AMI ID resolves to the new `ami-00112233445566778`.



Systems Manager parameter format for AMI IDs

Launch templates require that user-defined Systems Manager parameters adhere to the following format when used in place of an AMI ID:

- Parameter type: `String`
- Parameter data type: `aws:ec2:image` – This ensures that Parameter Store validates that the value you enter is in the proper format for an AMI ID.

For more information about creating a valid parameter for an AMI ID, see [Creating Systems Manager parameters] in the *AWS Systems Manager User Guide*.

Systems Manager parameter format in launch templates

To use a Systems Manager parameter in place of an AMI ID in a launch template, you must use one of the following formats when specifying the parameter in the launch template:

To reference a public parameter:

- `resolve:ssm:public-parameter`

To reference a parameter stored in the same account:

- `resolve:ssm:parameter-name`
- `resolve:ssm:parameter-name:version-number` – The version number itself is a default label
- `resolve:ssm:parameter-name:label`

To reference a parameter shared from another AWS account:

- `resolve:ssm:parameter-ARN`
- `resolve:ssm:parameter-ARN:version-number`
- `resolve:ssm:parameter-ARN:label`

Parameter versions

Systems Manager parameters are versioned resources. When you update a parameter, you create new, successive versions of the parameter. Systems Manager supports [parameter labels] that you can map to specific versions of a parameter.

For example, the `golden-ami` parameter can have three versions: `1`, `2`, and `3`. You can create a parameter label `beta` that maps to version `2`, and a parameter label `prod` that maps to version `3`.

In a launch template, you can specify version 3 of the `golden-ami` parameter by using either of the following formats:

- `resolve:ssm:golden-ami:3`
- `resolve:ssm:golden-ami:prod`

Specifying the version or label is optional. If a version or label is not specified, the latest version of the parameter is used.

Specify a Systems Manager parameter in a launch template

You can specify a Systems Manager parameter in a launch template instead of an AMI ID when you create a launch template or a new version of a launch template.

Console

To specify a Systems Manager parameter in a launch template

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. Under **Application and OS Images (Amazon Machine Image)**, choose **Browse more AMIs**.
5. Choose the arrow button to the right of the search bar, and then choose **Specify custom value/Systems Manager parameter**.
6. In the **Specify custom value or Systems Manager parameter** dialog box, do the following:

   a. For **AMI ID or Systems Manager parameter string**, enter the Systems Manager parameter name using one of the following formats:

      To reference a public parameter:

- **`resolve:ssm:`*`public-parameter`***

To reference a parameter stored in the same account:

- **`resolve:ssm:`*`parameter-name`***
- **`resolve:ssm:`*`parameter-name`*`:`*`version-number`***
- **`resolve:ssm:`*`parameter-name`*`:`*`label`***

To reference a parameter shared from another AWS account:

- **`resolve:ssm:`*`parameter-ARN`***
- **`resolve:ssm:`*`parameter-ARN`*`:`*`version-number`***
- **`resolve:ssm:`*`parameter-ARN`*`:`*`label`***

b. Choose **Save**.

7. Specify any other launch template parameters as needed, and then choose **Create launch template**.

For more information, see [Create a launch template by specifying parameters](#).

AWS CLI

To specify a Systems Manager parameter in a launch template

- Use the [create-launch-template](#) command to create the launch template. To specify the AMI to use, enter the Systems Manager parameter name using one of the following formats:

  To reference a public parameter:

  - **`resolve:ssm:`*`public-parameter`***

  To reference a parameter stored in the same account:

  - **`resolve:ssm:`*`parameter-name`***
  - **`resolve:ssm:`*`parameter-name`*`:`*`version-number`***
  - **`resolve:ssm:`*`parameter-name`*`:`*`label`***

To reference a parameter shared from another AWS account:

- **resolve:ssm:*parameter-ARN***
- **resolve:ssm:*parameter-ARN*:*version-number***
- **resolve:ssm:*parameter-ARN*:*label***

The following example creates a launch template that specifies the following:

- A name for the launch template (*TemplateForWebServer*)
- A tag for the launch template (*purpose*=*production*)
- The data for the instance configuration, specified in a JSON file:
  - The AMI to use (resolve:ssm:*golden-ami*)
  - The instance type to launch (*m5.4xlarge*)
  - A tag for the instance (*Name*=*webserver*)

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

The following is an example JSON file that contains the launch template data for the instance configuration. The value for ImageId is the Systems Manager parameter name, entered in the required format resolve:ssm:*golden-ami*.

```
{"LaunchTemplateData": {
    "ImageId": "resolve:ssm:golden-ami",
    "InstanceType": "m5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }]
  }
```

```
}
```

Verify that a launch template gets the correct AMI ID

To resolve the Systems Manager parameter to the actual AMI ID

Use the describe-launch-template-versions command and include the `--resolve-alias` parameter.

```
aws ec2 describe-launch-template-versions \
    --launch-template-name my-launch-template \
    --versions $Default \
    --resolve-alias
```

The response includes the AMI ID for `ImageId`. In this example, when an instance is launched using this launch template, the AMI ID resolves to `ami-0ac394d6a3example`.

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-089c023a30example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2022-12-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0ac394d6a3example",
                "InstanceType": "t3.micro",
            }
        }
    ]
}
```

Related resources

For more information about working with Systems Manager parameters, see the following reference materials in the Systems Manager documentation.

- For information about how to look up the AMI public parameters supported by Amazon EC2, see Calling AMI public parameters.

- For information about sharing parameters with other AWS accounts or through AWS Organizations, see Working with shared parameters.
- For information about monitoring whether your parameters are created successfully, see Native parameter support for Amazon Machine Image IDs.

Limitations

- Only EC2 Fleets of type `instant` support using a launch template that has a Systems Manager parameter specified in place of an AMI ID.
- EC2 Fleets of type `maintain` and `request`, and Spot Fleets do not support using a launch template that has a Systems Manager parameter specified in place of an AMI ID. For EC2 Fleets of type `maintain` and `request`, and for Spot Fleets, if you specify an AMI in the launch template, you must specify the AMI ID.
- If you use attribute-based instance selection in your EC2 Fleet, you can't specify a Systems Manager parameter in place of an AMI ID. When using attribute-based instance selection, you must specify the AMI ID.
- Amazon EC2 Auto Scaling provides other restrictions. For more information, see Use AWS Systems Manager parameters instead of AMI IDs in launch templates in the *Amazon EC2 Auto Scaling User Guide*.

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 different versions of a launch template, set the default version, describe a launch template version, and delete versions you no longer need.

Tasks

- Create a launch template version
- Set the default launch template version
- Describe a launch template version

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 a description of each parameter, see [Reference for Amazon EC2 instance configuration parameters](#).

Console

To create a launch template version

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 this version of the launch template.
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.
7. Choose **Create launch template**.

AWS CLI

To create a launch template version

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-0abcdef1234567890"
```

PowerShell

To create a launch template version

Use the New-EC2LaunchTemplateVersion Cmdlet. 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 `LaunchTemplateData`. The following example creates a new version based on version 1 of the launch template and specifies a different AMI ID.

```
New-EC2LaunchTemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -VersionDescription WebVersion2 `
    -SourceVersion 1 `
    -LaunchTemplateData (
        New-Object `
            -TypeName Amazon.EC2.Model.RequestLaunchTemplateData `
            -Property @{ImageId = 'ami-0abcdef1234567890'}
    )
```

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

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

Use the modify-launch-template command.

```
aws ec2 modify-launch-template \
```

```
    --launch-template-id lt-0abcd290751193123 \
    --default-version 2
```

PowerShell

To set the default launch template version

Use the Edit-EC2LaunchTemplate cmdlet.

```
Edit-EC2LaunchTemplate `
    -LaunchTemplateId lt-0abcd290751193123 `
    -DefaultVersion 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

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

Use the [describe-launch-template-versions](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-launch-template-versions.html) command and specify the version numbers. In the following example, versions *1* and *3* are specified.

```
aws ec2 describe-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1 3
```

To describe the latest and default launch template versions in your account

Use the [describe-launch-template-versions](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-launch-template-versions.html) command and specify `$Latest`, `$Default`, or both. You must omit the launch template ID and name in the call. You can't specify version numbers.

```
aws ec2 describe-launch-template-versions \
    --versions "$Latest,$Default"
```

PowerShell

To describe a launch template version

Use the [Get-EC2TemplateVersion](https://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2TemplateVersion.html) Cmdlet and specify the version numbers. In the following example, versions *1* and *3* are specified.

```
Get-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1,3
```

To describe the latest and default launch template versions in your account

Use the [Get-EC2TemplateVersion](https://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2TemplateVersion.html) Cmdlet and specify `$Latest`, `$Default`, or both. You must omit the launch template ID and name in the call. You can't specify version numbers.

```
Get-EC2TemplateVersion `
    -Version '$Latest','$Default'
```

Delete a launch template or a launch template version

If you no longer require a launch template, you can delete it. Deleting a launch template deletes all of its versions. If you only want to delete a specific version of a launch template, you can do so while retaining the other versions of the launch template.

Deleting a launch template or launch template version doesn't affect any instances that you've launched from the launch template.

Delete a launch template and all of its versions

If you no longer require a launch template, including all of its versions, you can delete the launch template. Deleting a launch template deletes all of its versions.

Console

To delete a launch template and all its versions

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 and all its versions

Use the delete-launch-template command and specify the launch template.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

PowerShell

To delete a launch template and all its versions

Use the Remove-EC2LaunchTemplate (AWS Tools for PowerShell) command and specify the launch template. If `-Force` is omitted, PowerShell prompts for a confirmation.

```
Remove-EC2LaunchTemplate -LaunchTemplateId lt-0123456789example -Force
```

Delete a launch template version

If you no longer require a launch template version, you can delete it.

Considerations

- You can't replace the version number after you delete it.
- You can't delete the default version of the launch template; you must first assign a different version as the default. If the default version is the only version for the launch template, you must delete the entire launch template.
- When using the console, you can delete one launch template version at a time. When using the AWS CLI, you can delete up to 200 launch template versions in a single request. To delete more than 200 versions in a single request, you can delete the launch template, which also deletes all of its versions.

Console

To delete a launch template version

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions**, **Delete template version**.
4. Select the version to delete and choose **Delete**.

AWS CLI

To delete a launch template version

Use the delete-launch-template-versions command and specify the version numbers to delete. You can specify up to 200 launch template versions to delete in a single request.

```
aws ec2 delete-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1
```

PowerShell

To delete a launch template version

Use the Remove-EC2TemplateVersion cmdlet and specify the version numbers to delete. You can specify up to 200 launch template versions to delete in a single request.

```
Remove-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1
```

Launch an Amazon EC2 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 create your AWS account, you can get started with Amazon EC2 for free using the AWS Free Tier. Your Free Tier benefits depend on when you created your AWS account. If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use the Free Tier to launch and use a `t2.micro` instance for free (in Regions where `t2.micro` is unavailable, you can use a `t3.micro` instance under the Free Tier). You incur charges for your instance or usage that counts against your Free Tier limits while the instance is running, even if it remains idle. For more information, see Amazon EC2 pricing. If you created your AWS account on or after July 15, 2025, you can use `t3.micro`, `t3.small`, `t4g.micro`, `t4g.small`, `c7i-flex.large`, and `m7i-flex.large` instance types for 6 months or until your credits are used up. For more information, see Free Tier benefits before and after July 15, 2025.

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 Get started with 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.
- 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.

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.

Depending on how you plan to connect to your instance, you might want to make certain configurations while launching your instance. These configurations could include specifying inbound security group rules for certain traffic or associating an instance profile role. For more information on the connection methods you can use to connect and their requirements, see Connect to your EC2 instance.

The instance receives a public DNS name that you can use to contact the instance from the internet. The instance also receives a private DNS name that other instances within the same VPC can use to contact the instance.

When you're finished with an instance, to avoid incurring unnecessary costs, be sure to terminate it. For more information, see Terminate Amazon EC2 instances.

The following methods are some of the ways that you can launch an instance.

Method	Tool	Documentation
Use the launch instance wizard to specify the launch parameters.	Amazon EC2 console	Launch an EC2 instance using the launch instance wizard in the console
Create a launch template and launch the instance from the launch template.	Amazon EC2 console	Launch EC2 instances using a launch template
Use an existing instance as the base.	Amazon EC2 console	Launch an EC2 instance using details from an existing instance
Use an AMI that you purchased from the AWS Marketplace.	Amazon EC2 console	Launch an Amazon EC2 instance from an AWS Marketplace AMI

Method	Tool	Documentation
Use an AMI that you specify.	AWS CLI	Launching, listing, and deleting Amazon EC2 instances in the AWS CLI
Use an AMI that you specify.	AWS Tools for Windows PowerShell	Launch an Amazon EC2 Instance Using Windows PowerShell
Use EC2 Fleet to provision capacity across different EC2 instance types and Availability Zones, and across On-Demand Instance, Reserved Instance, and Spot Instance purchasing options.	AWS CLI	EC2 Fleet and Spot Fleet
Use a AWS CloudFormation template to specify an instance.	AWS CloudFormation	AWS::EC2::Instance in the *AWS CloudFormation User Guide*
Use a language-specific AWS SDK to launch an instance.	AWS SDK	AWS SDK for .NET AWS SDK for C++ AWS SDK for Go AWS SDK for Java AWS SDK for JavaScript AWS SDK for PHP V3 AWS SDK for Python AWS SDK for Ruby V3

Tutorials for launching EC2 instances

There are different ways to launch and configure an Amazon EC2 instance. The method and configuration you use depends on your specific use case.

The following tutorials can help you learn how to launch EC2 instances. If you're new to Amazon EC2, we recommend that you start with the first tutorial. The tutorials start by introducing you to the basics, and help you build on the basics by introducing more configuration options.

Objective	Link to tutorial
Launch my very first EC2 instance Learn how to quickly launch an Amazon EC2 instance by using the default settings in the Amazon EC2 launch instance wizard. Also learn how to review the instance configuration fields and terminate the instance. Duration: 10 minutes	Tutorial 1: Launch my very first Amazon EC2 instance
Launch a test EC2 instance and connect to it Learn how to launch an Amazon EC2 instance that you can use for testing purposes. This instance will have no advanced configuration and won't store sensitive information. You will also learn about the essential instance configuration settings, how to connect to the instance, and how to stop it. Duration: 30 minutes	Tutorial 2: Launch a test EC2 instance and connect to it

Looking for other tutorials?

- Tutorial: Install a LAMP server on AL2023
- Tutorial: Configure SSL/TLS on AL2023
- Tutorial: Host a WordPress blog on AL2023

- Tutorial: Complete the configuration required to connect to your instance using EC2 Instance Connect
- Tutorial: Connect an Amazon EC2 instance to an Amazon RDS database

Tutorial 1: Launch my very first Amazon EC2 instance

Tutorial objective	Learn how to quickly launch an Amazon EC2 instance by using the default settings in the Amazon EC2 launch instance wizard. Also learn how to review the instance configuration fields and terminate the instance.
EC2 experience	Beginner
Duration	10 minutes
Cost	Free Tier eligible When you create your AWS account, you can get started with Amazon EC2 for free using the AWS Free Tier. If you created your AWS account before July 15, 2025, it's less than 12 months old, and you haven't already exceeded the Free Tier benefits for Amazon EC2, it won't cost you anything to complete this tutorial, because we help you select options that are within the Free Tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance (even if it remains idle) until you terminate it. If you created your AWS account on or after July 15, 2025, it's less than 6 months old, and you haven't used up all your credits, it won't cost you anything to complete this tutorial,

	because we help you select options that are within the Free Tier benefits. For information on how to determine whether you're eligible for the Free Tier, see the section called "Track your Free Tier usage".
Prerequisites	• You must have an AWS account, configure a user with administrator access, and use the administrator user to sign into the AWS Management Console. Not sure how to do this? Try this tutorial: Setting Up Your AWS Environment • You must have general familiarity with the AWS console. Not sure where to start? Try this getting started guide: Getting Started with the AWS Management Console

Tutorial overview

This tutorial is designed for beginners with no prior experience using Amazon EC2. We'll guide you through the steps for creating—we call it *launching*—your very first EC2 instance using the EC2 console. An *instance* is essentially a web server in the AWS Cloud. After launching your instance, we'll show you how to find it in the console. Finally, to help you manage costs, we'll show you how to delete—we call it *terminate*—your instance.

This tutorial is divided into the following short tasks. You must complete each task before moving to the next one.

- Task 1: Launch your instance
- Task 2: Find your instance
- Task 3: View your instance configuration
- Task 4: Terminate your instance

Task 1: Launch your instance

In this task, you'll take the quickest path to launching your instance by doing only the essentials. We'll use the EC2 launch instance wizard, a web-based form that provides all the fields for configuring and launching your instance. It simplifies the process by providing default values for the instance configuration fields.

Before you start

Make sure you've completed the prerequisites listed in the preceding table, including signing into the AWS Management Console with your administrator user.

Follow these steps to quickly launch your instance

1. **Open the Amazon EC2 console:**

   Go to https://console.aws.amazon.com/ec2/.
2. **Open the EC2 launch instance wizard:**

   From the EC2 dashboard, choose **Launch instance**.

   The **Launch an instance** web-based form opens. This is the EC2 launch instance wizard.
3. **Name your instance:**

   Under **Name and tags**, for **Name**, enter a descriptive name like **`My first EC2 instance`**.

   While naming your instance isn't required, it helps identify your instance later.
4. **Proceed without a key pair:**

   Under **Key pair (login)**, for **Key pair name**, choose **Proceed without a key pair (Not recommended)**.

   A key pair can be used for secure login. However, because we won't be logging into the instance in this tutorial, you don't need a key pair for now.
5. **Launch your instance:**

   In the **Summary** panel on the right, choose **Launch instance**.

   Amazon EC2 quickly launches your instance using the default settings. A **Success** banner confirms the launch.

Congratulations! You've successfully launched your very first EC2 instance!

Task 2: Find your instance

In this task, you'll locate the instance that you just launched in the EC2 console.

Follow these steps to find your instance in the EC2 console

1. **Open the Instances page:**

   If you're still on the success page, choose **Instances** in the breadcrumb at the top of the screen. You might need to choose the three ellipses first to access it.

   If you've navigated away, choose **Instances** from the navigation pane.
2. **Locate your instance:**

   In the **Name** column, find your instance by the name you gave it.

Task 3: View your instance configuration

In this task, you'll become familiar with viewing your instance's configuration details.

Follow these steps to view your instance's configuration

1. **Locate the instance ID:**

   In the **Instance ID** column, take note of your instance's unique ID. It begins with **i–** followed by 17 alphanumeric characters, for example, **i-01aeed690c9fb5322**.

   The instance ID is automatically assigned to your instance when it's launched.
2. **Open the instance details page:**

   In the **Instance ID** column, choose the ID link to open the instance details page where you can review its configuration.
3. **Explore instance configuration details:**

   Take a few minutes to explore the configuration details of your instance. In the next tutorial, we'll dive deeper into the configuration. For now, use this time to familiarize yourself with the instance details page.

   Tip: To quickly find a field, press Ctrl+F or command+F on your keyboard.

a. **Instance type:** Can you find the instance type? It might be **t3.micro**, for example.

b. **Public IPv4 address:** Can you find the public IPv4 address that was allocated to your instance? It's in a format similar to the following example: **34.242.148.128**.

c. **Instance owner:** Can you identify the owner of this instance? It's you! Your AWS account number is listed under the **Owner** field.

d. **Instance tags:** The name you gave your instance is actually a tag. Can you find your instance tags? Choose the **Tags** tab. The key is **Name**, and the value is the name you provided.

e. **Launch time:** Can you find when you launched your instance? Choose the **Details** tab and find the **Launch time** field.

f. **Instance state:** Can you verify the state of your instance? It should be **Running**.

Take a few more minutes to explore the other instance configuration fields. When you're ready, proceed to the next task.

Task 4: Terminate your instance

In this task, you'll delete your instance to preserve your Free Tier benefits. In EC2, *terminate* is the term used for deleting an instance.

Follow these steps to terminate your instance

1. **Initiate termination:**

   If you're still on the instance details page, choose the **Instance state** menu (top right), and then choose **Terminate (delete) instance**.

   If you've navigated away, choose **Instances** from the navigation pane. Then, on the **Instances** page, select the checkbox next to the name of your instance, and then choose the **Instance state** menu (top right), and choose **Terminate (delete) instance**.

2. **Confirm termination:**

   In the **Terminate (delete) instance** window that opens, choose the **Terminate (delete)** button to confirm that you want to terminate your instance.

3. **Monitor instance state:**

On the **Instances** page, check the **Instance state** column. The state of your instance changes to **Shutting-down**. If you don't see the full text, try widening the column.

Once the instance has shut down, Amazon EC2 deletes the instance, and it disappears from the **Instances** page.

Key takeaways

In this tutorial, you covered the following key concepts:

- *Instance* refers to an Amazon EC2 web server in the AWS Cloud.
- *Launch* refers to creating an EC2 instance.
- *Terminate* refers to deleting an EC2 instance.
- The EC2 launch instance wizard contains default values for instance configuration, allowing for a quick and easy instance launch.
- The instance ID is a unique identifier automatically assigned to your instance, while the instance name is an optional tag that you can assign for easier identification.

Next steps

To build confidence in launching and terminating instances, consider repeating the steps in this tutorial. Be sure to terminate any instances that you launch to preserve your Free Tier benefits.

Once you're comfortable with these basics, move onto the next tutorial, which provides a deeper dive into key instance configuration fields.

Tutorial 2: Launch a test EC2 instance and connect to it

Tutorial objective	Learn how to launch an Amazon EC2 instance that you can use for testing purposes. This instance will have no advanced configuration and won't store sensitive information. You will also learn about the essential instance configuration settings, how to connect to the instance, and how to stop it.
EC2 experience	Beginner

Duration	30 minutes
Cost	Free Tier eligible When you create your AWS account, you can get started with Amazon EC2 for free using the AWS Free Tier. If you created your AWS account before July 15, 2025, it's less than 12 months old, and you haven't already exceeded the Free Tier benefits for Amazon EC2, it won't cost you anything to complete this tutorial, because we help you select options that are within the Free Tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance (even if it remains idle) until you terminate it. If you created your AWS account on or after July 15, 2025, it's less than 6 months old, and you haven't used up all your credits, it won't cost you anything to complete this tutorial, because we help you select options that are within the Free Tier benefits. For information on how to determine whether you're eligible for the Free Tier, see the section called “Track your Free Tier usage”.
Prerequisites	Complete Tutorial 1: Launch my very first Amazon EC2 instance.

Tutorial overview

This tutorial is designed for beginners who want to launch an EC2 instance that they can use for testing purposes.

We'll explain the key instance configuration fields, and then guide you through the steps for launching a test instance using the default values in the EC2 console. After launching your instance, we'll show you how to log into—we call it *connect to*—your instance. We'll also show you how to create a key pair, which is required for connecting to your instance in this tutorial. Finally, to help manage costs, we'll show you to stop your instance to avoid usage charges.

You'll launch a Linux instance in this tutorial. While the steps in this tutorial can be used for launching instances with other operating systems, the instructions for *connecting* to an instance are specific to Linux instances.

This tutorial is divided into the following short tasks. You must complete each task before moving to the next one.

- Task 1: Familiarize yourself with key components for launching an instance
- Task 2: Review a technical diagram
- Task 3: Create a key pair
- Task 4: Launch your test instance
- Task 5: Find your instance
- Task 6: View your instance configuration
- Task 7: Familiarize yourself with key components for connecting to an instance
- Task 8: Connect to your instance
- Task 9: Stop your instance

Task 1: Familiarize yourself with key components for launching an instance

In this task, you'll explore the key components required to launch an EC2 instance. These are the AMI, instance type, key pair, security group, network (VPC and subnet), and Amazon EBS volume. You'll also explore an optional component, the **Name** tag.

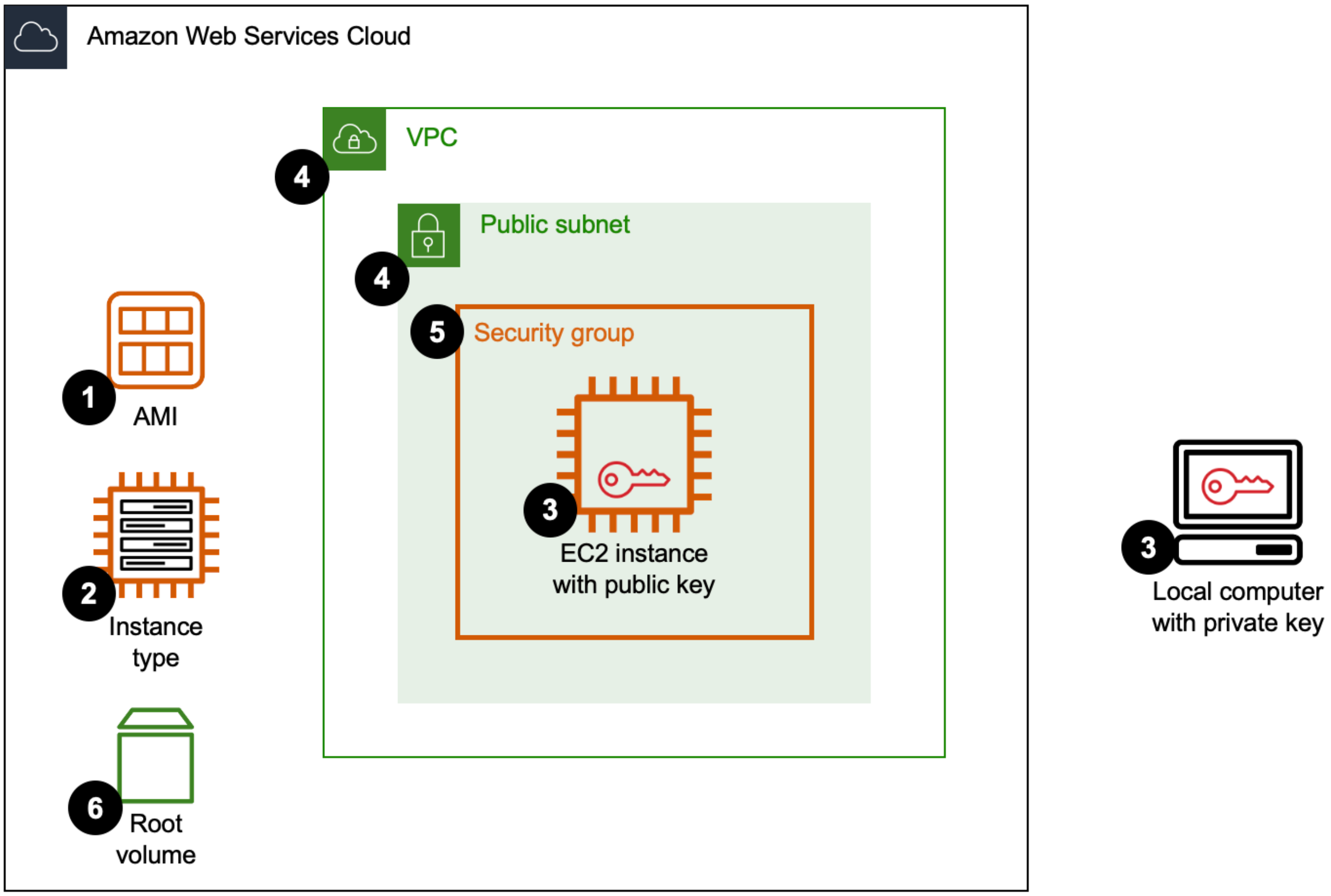
To help visualize these components, think of an instance like a rental house. Just as renting a house gives you a place to live without your needing to own and maintain the property, EC2 instances provide computing power without your needing to own and maintain the underlying infrastructure.

When deciding on the kind of instance to launch, you'll consider the configuration criteria for the instance, just as you would consider the criteria you want from a house. While this analogy simplifies things, it offers a helpful way to visualize the components until you're more familiar with them.

- **AMI – House building materials and amenities:** The Amazon Machine Image (AMI) determines the operating system and applications your instance starts with. This is like choosing the building materials (like brick, steel, or wood) and amenities (like appliances and furnishings) of your house. A base AMI is like an unfurnished house with basic appliances, while a custom AMI with pre-installed software is like a fully furnished house.
- **Instance type – House size and power:** The instance type defines the size and capabilities of your EC2 instance, much like choosing the size of a house, number of rooms, and energy capacity. Each instance type determines the amount of CPU, memory, storage, and networking capacity of your instance. The selected AMI might limit what instance types you can choose.
- **Key pair – Front door key**: A key pair is like the lock and key to the front door of your house. The public key acts as the lock on your instance, while the private key is the key you must keep securely on your local computer. If someone else gets hold of your private key, they can access your instance, much like how someone with your front door key can enter your house.
- **Network (VPC and subnet) – Property boundary, sectioned areas, and house number**: Your virtual private cloud (VPC) is like the entire property where your house is located, and the subnet is the sectioned-off area around the house. If you have multiple houses (instances) on your property, you might want to section them off into distinct areas (different subnets) depending on their purpose. Some houses allow visitors to roam freely through the gardens (public subnets with internet access), while others have fenced-off gardens to restrict entry (private subnets without internet access). Each subnet contains a range of IP addresses, much like house numbers, which can be assigned to instances within the subnet.
- **Security group – The gatekeeper**: The security group acts like a gatekeeper, controlling who is allowed to visit your house. It enforces a set of rules that controls what traffic is allowed to reach your instance. For example, a rule that allows SSH traffic from a specific IP address is like the gatekeeper letting in only a specific person to deliver groceries. Similarly, allowing HTTPS traffic from anywhere is like letting the public come and take a look at the exterior of your house.
- **Amazon EBS volume – Storage units**: EBS volumes are like storage units where you can store your belongings. Each instance has a root volume (where the AMI is stored), and you can add more volumes (storage) at any time as needed.
- **Name tag – The house name:** The **Name** tag functions like a sign on a house, helping you easily identify who lives there. While the **Name** tag makes it easier to distinguish between instances, it's not required when launching an instance.

Task 2: Review a technical diagram

In this task, you'll become familiar with a typical technical diagram that we use in the AWS documentation. The following diagram represents the configuration for the test instance you'll launch in this tutorial. In the previous task, we introduced these components using the analogy of a rental house. Now, we'll focus on the actual EC2 components themselves. The numbered labels correspond to the descriptions that follow.



1. **AMI** – The AMI is the image you choose when launching an instance. It's a template that contains the operating system and software to run on your instance. For example, if you want to launch a Linux instance, you can choose the Amazon Linux 2023 AMI. Or, if you want to launch a Windows instance, you can choose the Microsoft Windows Server 2022 Base AMI. The AMI catalog in the Amazon EC2 console contains 1000s of images to choose from.
2. **Instance type** – The instance type is the hardware that determines the CPU, memory, storage, and networking capacity of the host computer used for your instance. Amazon EC2 offers over

600 instance types to choose from, each varying in hardware configuration and size, allowing you to choose the best fit for your application's needs.

3. **Key pair –** A key pair is set of security credentials that you use to prove your identity when connecting to your instance. The public key is on your instance and the private key is on your local computer.

   In EC2, *connecting to your instance* refers to logging into your instance from your local computer. While there are other ways to securely connect to your instance, in this tutorial we use a key pair.
4. **Network –** The network is made up of a VPC and one or more subnets. A VPC is a virtual network within the AWS Cloud. Every AWS customer has their own VPC dedicated to their AWS account. You'll launch your instance into a subnet in your VPC. A subnet is a range of IP addresses within a VPC. Your default subnet is a public subnet, which means it will assign a public IP address and provide internet access to your instance from outside the Amazon network.
5. **Security group** – A security group acts as a firewall to control the traffic to your instance. A security group contains rules that allow certain types of traffic to enter your instance. To connect through SSH from your local computer to your instance (using your key pair), you need a rule that allows SSH traffic from your local computer.
6. **EBS volume –** An Amazon EBS volume is a storage device that functions like a physical hard drive. Your instance comes with a root volume, which is a special EBS volume that stores the AMI with the operating system and software needed to boot your instance. You can optionally add data volumes. However, since your test instance won't store any sensitive data, you don't need additional encrypted data volumes.

Congratulations! You've completed the conceptual tasks in this tutorial. In the following tasks, you'll use the Amazon EC2 console to create the components you've learned about.

Task 3: Create a key pair

In this task, you'll create a key pair. A key pair consists of two parts: a public key, which you'll add to your instance, and a matching private key, which you'll use to securely connect to your instance. In the next task, you'll select this key pair when launching your instance, which automatically adds the public key to the instance. It's crucial to store the private key securely on your local computer, because anyone with access to it can connect to your instance.

If you prefer to use an existing key pair when you launch your test instance, feel free to skip this task. Otherwise, proceed to create a new key pair.

Before you start

Make sure you've completed the prerequisites listed in the preceding table, including signing into the AWS Management Console with your administrator user.

Follow these steps to create a key pair

1. **Open the Amazon EC2 console:**

   Go to https://console.aws.amazon.com/ec2/.
2. **Navigate to the Key pairs console page:**

   In the navigation pane, under **Network & Security**, choose **Key Pairs**.

   - If you previously created key pairs, they appear in the table.
   - If no key pairs exist, the table is empty.
3. **Create a new key pair:**

   Choose the **Create key pair** button (top right) to open the **Create key pair** web-based form, and enter your key pair details, as follows:

   a. **Name your key pair:** For **Name**, enter a name that will help you recognize the key pair, like **`test-instance-key-pair`**.

      The name can be up to 255 ASCII characters long. It can’t include leading or trailing spaces.
   b. **Choose the key pair type:** For **Key pair type**, choose **ED25519**.

      Linux instances support both RSA and ED25519 key types, while Windows instances support only RSA. Since you'll be launching a Linux instance in this tutorial, you can use an ED25519 key.
   c. **Choose the private key file format:** For **Private key file format**, choose **.pem**.

      This is the format in which your private key file will be saved.
4. **Save the public key to Amazon EC2 and download the private key:**

   Choose the **Create key pair** button (bottom right).

   Amazon EC2 saves the public key, while your browser downloads the private key file automatically to your local computer. The file is named according to the name that you specified for the key pair, and the extension is the file format that you chose. Move the private key file to a secure location on your computer.

> **Important**
>
> This is the only chance you'll have to save the private key file.

5. **Set the permissions on the key (for macOS and Linux users):**

   If you plan to connect to your instance using SSH on a macOS or Linux computer, you must set the correct permissions for your private key file. Open a terminal window and run the following command, replacing *test-instance-key-pair* with the name of your key pair:

   ```
   chmod 400 test-instance-key-pair.pem
   ```

   This command ensures that only you can read the private key file, which is necessary for establishing a secure connection to your instance. Without these permissions, you won't be able to connect using this key pair.

Congratulations! You've successfully created a key pair!

Task 4: Launch your test instance

In this task, you'll quickly launch a test instance using the EC2 launch instance wizard. You'll configure the main instance configuration settings for a Linux instance and use the default values for the other fields.

To help you manage costs, we recommend choosing **Free tier eligible** components.

Follow these steps to launch a test instance

1. **Open the Amazon EC2 console:**

   Go to https://console.aws.amazon.com/ec2/.

2. **Open the EC2 launch instance wizard:**

   From the EC2 dashboard, choose **Launch instance**.

   The **Launch an instance** web-based form opens. This is the EC2 launch instance wizard.

3. **Name your instance:**

Under **Name and tags**, for **Name**, enter a descriptive name like `Test instance`.

The instance name is a tag, where the key is **Name**, and the value is the name that you specify.

Tip: For test instances, a name tag is sufficient. However, for production instances, it’s best practice to establish a tagging policy to standardize tagging across all your resources.

4. **Choose your operating system and software—the Amazon Machine Image (AMI):**

   Under **Application and OS Images (Amazon Machine Image)**, for **Amazon Machine Image (AMI)**, the default selection is **Amazon Linux 2023 AMI**. This AMI is marked **Free tier eligible**. In this tutorial, you'll be launching a Linux instance, so leave the default setting to keep under the Free Tier limits.

5. **Choose your hardware—the instance type:**

   Under **Instance type**, for **Instance type**, keep the default selection for this tutorial. The default instance type can be used under the Free Tier and its hardware is suitable for your test instance.

6. **Prepare for secure login with a key pair:**

   Under **Key pair (login)**, for **Key pair name**, choose the key pair you created in the previous task. If you don't see your key pair in the list, choose the refresh icon (to the right of the list).

   When your instance launches, it will place the public key on the instance. To connect to your instance after it has launched, you'll use the corresponding private key that you downloaded in the previous task.

7. **Configure the network settings to enable internet access:**

   Under **Network settings**, the **Network** (your VPC) and **Subnet** fields are configured by default. Keep the default settings for this tutorial to help you get started quickly. If you haven’t modified your default subnet, your instance will have internet access.

   Tip: Your default subnet is a public subnet, which means it will assign a public IP address and provide internet access to your instance from outside the Amazon network. For test instances, it’s okay to use the default subnet settings that provide internet access. However, for production instances, it’s best practice to only assign a public IP address and use a subnet with internet access when absolutely necessary.

8. **Set up the instance firewall (security group):**

Under **Network settings**, under **Firewall (security groups)**, keep the checkbox **Allow SSH traffic from Anywhere (0.0.0.0)** selected. This will create a new security group for your test instance that allows SSH traffic from any IP address.

A security group acts as a firewall to control the traffic to your instance. To connect through SSH from your local computer to your instance, you need a rule that allows SSH traffic from your local computer.

Tip: The IP address of your local computer might change over time if your internet service provider uses dynamic IP assignment. We're assuming that when you use an instance for testing purposes, you won't use the instance to store sensitive information, and therefore security measures can be less restrictive. For test instances, it's generally acceptable to allow traffic from any IP address (`0.0.0.0/0`) so that you can always connect even if your IP address changes. However, for production instances, especially those with sensitive data, it's best practice to allow traffic only from known IP addresses.

9. **Configure the instance storage:**

   Under **Configure storage**, the **Root volume (Encrypted)** fields are configured by default. Leave the settings as they are to keep under the Free Tier limits.

   Since our test instance won't store any sensitive data, we don't need additional encrypted data volumes.

10. **Review the instance configuration:**

    In the **Summary** panel on the right, you can review your high-level settings before launching your instance.

11. **Launch your instance:**

    When you're ready to launch your instance, in the **Summary** panel, choose **Launch instance**.

    Amazon EC2 quickly launches your instance using the settings that you specified. If you didn't specify a setting, the default is used. A **Success** banner confirms the launch.

Congratulations! You've successfully launched your test instance!

Task 5: Find your instance

In this task, you'll locate the instance that you just launched in the EC2 console.

Follow these steps to find your instance in the EC2 console

1. **Open the Instances page:**

   If you're still on the success page, choose the instance ID in the **Success** banner.

   If you've navigated away, choose **Instances** from the navigation pane.
2. **Locate your instance:**

   In the **Name** column, find your instance by the name you gave it.

Task 6: View your instance configuration

In this task, you'll become familiar with viewing your instance's configuration details.

Follow these steps to view your instance's configuration

1. **Locate your instance:**

   In the **Name** column, find your instance by the name you gave it.
2. **Open the instance details page:**

   Select the checkbox next to the name of your instance, and then choose the **Actions** menu (top right), and choose **View details** to open the instance details page where you can review its configuration.

   In the previous tutorial, you chose the instance's ID link to open the instance details page. You'll discover that there's more than one way to accomplish a task in the EC2 console.
3. **Explore instance configuration details:**

   Take a few minutes to explore the configuration details of your instance.

   Tip: To quickly find a field, press Ctrl+F or command+F on your keyboard.

   a. **AMI:** Can you find the AMI that you used to launch your instance? You can find the information in **AMI ID** and **AMI name** on the **Details** tab.
   b. **Instance type:** Can you find the instance type? It's might be **t3.micro**, for example.
   c. **Key pair:** Can you find the key pair that you selected when you launched your instance? It's specified for **Key pair assigned at launch**. Note that if you change the key pair in the future, the value here won't change.

d. **VPC:** Can you find the ID of your VPC? You'll find all networking-related configuration settings on the **Networking** tab. The VPC ID is in a format similar to the following example: **vpc-1a2b3c4d**

e. **Subnet:** Can you find the ID of the subnet in which you launched your instance? It's in a format similar to the following example: **subnet-1a2b3c4d**

f. **Public IPv4 address:** Can you find the public IPv4 address that was allocated to your instance? It's in a format similar to the following example: **34.242.148.128**.

g. **Security group:** Can you find the inbound rule that was created to allow SSH traffic from anywhere (0.0.0.0./0)? You'll find all security-related configuration settings on the **Security** tab.

h. **Storage:** Can you find the volume that was created for this instance? You'll find all storage-related configuration settings on the **Storage** tab.

i. **Instance tags:** The name you gave your instance is actually a tag. Can you find your instance tags? Choose the **Tags** tab. The key is **Name**, and the value is the name you provided.

j. **Instance state:** Can you verify the state of your instance? It should be **Running**.

Take a few more minutes to explore the other instance configuration fields. When you're ready, proceed to the next task.

Task 7: Familiarize yourself with key components for connecting to an instance

In this task, you'll explore the key components required to connect to an EC2 instance. These are the connection protocol, public DNS, security group, key pair, and instance username.

To help visualize these components, think of connecting to an instance like going to your house:

- **Connection protocol – Your mode of transport:** Just like choosing how to get home, you choose the connection protocol that will take you to your instance. In this tutorial, we'll use SSH (Secure Shell), which creates a secure tunnel for connecting your computer to your instance over the internet.
- **Public DNS – The house address:** Just like your house has a unique address, your EC2 instance has its own public DNS name (for example, `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com`). This public DNS name enables SSH to connect directly to your instance.

- **Security group – The gatekeeper:** Imagine your house has a gatekeeper who controls who may enter or leave. Similarly, the EC2 instance has a security group that acts like a gatekeeper, controlling which types of network traffic are allowed in or out of your instance. Only the traffic you explicitly permit (for example, SSH traffic from your computer's IP address) is allowed in.
- **Private key – Your front door key:** When you launched the instance, you specified a key pair. The public key was placed on the instance, and you kept the private key on your computer. The private key acts as your front door key—without it, you can't get into your instance.
- **Instance username – The resident:** When you arrive at your house, you need to identify yourself to prove you're a resident. Similarly, when connecting to an instance, you provide a username. Different instances have different default usernames, depending on their operating system. For example, Amazon Linux instances use `ec2-user` as the default username.

The connection command

To connect to your EC2 instance, use the following command in a terminal window:

```
ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
west-1.compute.amazonaws.com
```

Here's a breakdown of what the command does:

- `ssh` – This command specifies the connection protocol, initiating an SSH (Secure Shell) connection to your instance.
- `-i "test-instance-key-pair.pem"` – The `-i` flag indicates the private key file needed to authenticate the connection. This private key file must match the key pair you specified when launching the instance. If your private key file is saved in a specific folder, specify the full path to the file.
- `ec2-user` – This is the username for logging into the instance. For Amazon Linux instances, the default username is `ec2-user`. Other AMIs might use different default usernames, such as `ubuntu` for Ubuntu instances.
- `@` – This symbol separates the username from the instance's address.
- `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com` – This is the public address of your instance (the public DNS), which includes the public IPv4 address and the AWS Region. It uniquely identifies the instance.

What happens when you run the command

After you run the command, SSH establishes a secure tunnel and authenticates with your private key. If the instance's security group permits the traffic, you gain access to your EC2 instance. You can now control the instance from your computer as if you were sitting right in front of it. You can run commands, install software, and manage files—just like you would on your local machine.

Task 8: Connect to your instance

In this task, you'll connect to your instance using an SSH client on your computer. In the previous task, we introduced the components for connecting to an instance using the analogy of going to your house. Now, we'll focus on connecting to the actual EC2 instance.

There are different ways to connect to an instance. The method you use to connect depends on the instance's operating system. Since you've launched a Linux instance, you'll use an SSH client on your local computer.

First, check if your computer has an SSH client installed

Most computers come with an SSH client pre-installed. To check, open a terminal window on your computer and run the following command:

```
ssh
```

If the command is recognized, you're ready to connect.

If the command isn't recognized, you must install an SSH client. Instructions for installing an SSH client are beyond the scope of this tutorial. If you need help, see SSH connection prerequisites in this user guide or search online for instructions on how to install an SSH client on your operating system.

Follow these steps to connect to your instance

1. **Initiate connecting:**

   If you're on the instance details page in the Amazon EC2 console, choose the **Connect** button (top right).

   If you've navigated away, choose **Instances** from the navigation pane. Then, on the **Instances** page, select the checkbox next to the name of your instance and choose the **Connect** button (top right).

   This opens the **Connect to instance** page.

2. **Choose the connection method:**

   On the **Connect to instance** page, choose the **SSH client** tab.

   Take a moment to review the text on this page, as these are the steps that you'll follow next.

3. **Review the SSH command:**

   Under **Example**, you'll see a command that is automatically generated and customized with your instance's details. The private key name is derived from the name of the public key specified at launch.

   The command looks something like this:

   ```
   ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
   west-1.compute.amazonaws.com
   ```

4. **Copy the SSH command:**

   Choose the copy icon next to the example SSH command.

5. **Open a terminal window:**

   On your local computer, open a terminal window.

6. **Paste and run the SSH command:**

   Paste the SSH command into the terminal window. If you saved your private key file in a specific folder, edit the command to include the full file path.

   Press Enter on your keyboard.

   You'll see a response similar to the following:

   ```
   The authenticity of host 'ec2-18-201-118-201.eu-west-1.compute.amazonaws.com
    (18-201-118-201)' can't be established.
   ED25519 key fingerprint is SHA256:examplehxj9aOr1MogvKOoMNskVVIRBQBoq0example.This
    key is not known by any other names.
   Are you sure you want to continue connecting (yes/no/[fingerprint])?
   ```

7. **Complete the connection:**

   Enter **yes** and press Return on your keyboard.

Verifying the fingerprint is beyond the scope of this tutorial. To learn more, see (Optional) Get the instance fingerprint.

Upon a successful connection, the terminal prompt changes to display your instance's public DNS.

Congratulations! You've successfully connected to your instance!

Task 9: Stop your instance

In this task, you'll stop your instance to preserve your Free Tier benefits. When your instance is stopped, you stop incurring costs for it. If you created your AWS account before July 15, 2025 and your qualify for the Free Tier, you will continue to incur costs for the EBS storage.

Follow these steps to stop your instance

1. **Initiate stopping:**

   If you're still on the **Connect to instance** page, choose **Instances** from the breadcrumb. If you've navigated away, choose **Instances** from the navigation pane.

   Then, on the **Instances** page, select the checkbox next to the name of your instance, and then choose the **Instance state** menu (top right), and choose **Stop instance**. When prompted, choose **Stop**.

2. **Monitor instance state:**

   On the **Instances** page, check the **Instance state** column. The state of your instance changes to **Stopping** and then **Stopped**. If you don't see the full text, try widening the column.

   If you think the instance state has changed from **Stopping** to **Stopped**, but you don't see it yet, choose the refresh icon (above the table) to refresh the **Instances** table.

Key takeaways

In this tutorial, you covered the following key concepts:

- *AMI* refers to an Amazon Machine Image, which is a template that contains the operating system and software required to launch an instance.

- *Instance type* refers to the hardware of the host computer used for your instance. It determines the CPU, memory, storage, and networking capacity of your instance.
- *Key pair* refers to the set of public and private keys that you can use for securely connecting to your instance.
- *Network* refers to a *VPC* (a virtual private cloud dedicated to your account within the AWS Cloud) and a *subnet* (a range of IP addresses within your VPC).
- *Security group* refers to a set of rules that controls what traffic can reach your instance.
- *EBS volume* refers to the data storage for your instance. Every instance has a root volume for storing the AMI and one or more optional data volumes.
- *Tags* are metadata that you can optionally assign to your instance. The instance name is a tag, whose **Key** is **Name**, and the **Value** is your choice.
- *Connecting* refers to accessing your instance over the internet.
- *SSH* refers to the Secure Shell connection protocol that you can use to connect to your instance.
- *Public DNS* is your instance's unique public address.
- *Instance username* is determined by the operating system of your instance and required for connecting.
- *Stopping* your instance stops the charges for the instance, but EBS storage charges continue.

Next steps

To build confidence in launching, connecting to, and stopping instances, consider repeating the steps in this tutorial. Be sure to terminate any instances that you launch to preserve your Free Tier benefits.

Once you're comfortable with these basics, you can explore more advanced tutorials. For more tutorials, see Looking for other tutorials?

If you created your AWS account before July 15, 2025, consider watching the following 6-minute video: How can I avoid charges on my account when using AWS Free Tier services

If you created your AWS account on or after July 15, 2025, consider reviewing the following information: Explore AWS services with AWS Free Tier in the *AWS Billing User Guide*

Reference for Amazon EC2 instance configuration parameters

The launch instance wizard and launch template in the Amazon EC2 console provide all the parameters for configuring an Amazon EC2 instance.

Except for the key pair, the launch instance wizard provides a default value for each parameter. You can accept any or all of the defaults, or configure an instance with your own values. When creating a launch template, the parameters are optional. If you use a launch template to launch an instance, the parameters specified in the launch template override the default values in the launch instance wizard. Any parameter not specified in the launch template will default to the value provided by the launch instance wizard.

The parameters are grouped in the launch instance wizard and launch template. The following descriptions are presented according to the parameter groupings in the console.

Parameters for instance configuration

- Name and tags
- Application and OS Images (Amazon Machine Image)
- Instance type
- Key pair (login)
- Network settings
- Configure storage
- Advanced details
- Summary

Name and tags

The instance name is a tag, where the key is **Name**, and the value is the name that you specify. You can tag the instance, volumes, and network interfaces. For Spot Instances, you can tag the Spot Instance request only. For information about tags, see Tag your Amazon EC2 resources.

Specifying an instance name and additional tags is optional.

- For **Name**, enter a descriptive name for the instance. If you don't specify a name, the instance can be identified by its ID, which is automatically generated when you launch the instance.
- To add additional tags, choose **Add additional tags**. Choose **Add tag**, and then enter a key and value, and select the resource type to tag. Choose **Add tag** again for each additional tag to add.

You can only specify the instance name when launch an instance. You can't name the instance when you create a launch template, but you can add tags for the resources that are created when the instance is launched.

Application and OS Images (Amazon Machine Image)

An Amazon Machine Image (AMI) contains the information required to create an instance. For example, an AMI might contain the software that's required to act as a web server, such as Linux, Apache, and your website.

You can find a suitable AMI as follows. With each option for finding an AMI, you can choose **Cancel** (at top right) to return to the launch instance wizard without choosing an AMI.

Search bar

To search through all available AMIs, enter a keyword in the AMI search bar and then press **Enter**. To select an AMI, choose **Select**.

Recents

The AMIs that you've recently used.

Choose **Recently launched** or **Currently in use**, and then, from **Amazon Machine Image (AMI)**, select an AMI.

My AMIs

The private AMIs that you own, or private AMIs that have been shared with you.

Choose **Owned by me** or **Shared with me**, and then, from **Amazon Machine Image (AMI)**, select an AMI.

Quick Start

AMIs are grouped by operating system (OS) to help you get started quickly.

First select the OS that you need, and then, from **Amazon Machine Image (AMI)**, select an AMI. To select an AMI that can be used under the AWS Free Tier, make sure that the AMI is marked **Free tier eligible**.

Browse more AMIs

Choose **Browse more AMIs** to browse the full AMI catalog.

- To search through all available AMIs, enter a keyword in the search bar and then press **Enter**.
- To find an AMI by using a Systems Manager parameter, choose the arrow button to the right of the search bar, and then choose **Search by Systems Manager parameter**. For more information, see Reference AMIs using Systems Manager parameters.

- To search by category, choose **Quickstart AMIs**, **My AMIs**, **AWS Marketplace AMIs**, or **Community AMIs**.

  The AWS Marketplace is 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 Amazon EC2 instance from an AWS Marketplace AMI. In **Community AMIs**, you can find AMIs that AWS community members have made available for others to use. AMIs from Amazon or a verified partner are marked **Verified provider**.
- To filter the list of AMIs, select one or more check boxes under **Refine results** on the left of the screen. The filter options are different depending on the selected search category.
- Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need: either **ebs** (backed by Amazon EBS) or **instance-store** (backed by instance store). For more information, see Root device type.
- Check the **Virtualization** type listed for each AMI. Notice which AMIs are the type that you need: either **hvm** or **paravirtual**. For example, some instance types require HVM. For more information about Linux virtualization types, see Virtualization types.
- Check the **Boot mode** listed for each AMI. Notice which AMIs use the boot mode that you need: either **legacy-bios**, **uefi**, or **uefi-preferred**. For more information, see Instance launch behavior with Amazon EC2 boot modes.
- Choose an AMI that meets your needs, and then choose **Select**.

Warning when changing the AMI

When you launch an instance, if you modify the configuration of any volumes or security groups associated with the selected AMI, and then you choose a different AMI, a window opens to warn you that some of your current settings will be changed or removed. You can review the changes to the security groups and volumes. Furthermore, you can either view which volumes will be added and deleted, or view only the volumes that will be added. This warning does not appear when creating a launch template.

Instance type

The instance type defines the hardware configuration and size of the instance. Larger instance types have more CPU and memory. For more information, see Amazon EC2 instance types.

- **Instance type**: Ensure that the instance type is compatible with the AMI that you've specified. For more information, see Amazon EC2 instance types.

Free Tier – You can use instance types that are labeled **Free tier eligible** for free under the Free Tier. The specific instance types depend on when you created your AWS account.

If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type, or the **t3.micro** instance type in Regions where **t2.micro** is unavailable. Be aware that when you launch a **t3.micro** instance, it defaults to **Unlimited** mode, which might incur additional charges based on CPU usage.

If you created your AWS account on or after July 15, 2025, you can use **t3.micro**, **t3.small**, **t4g.micro**, **t4g.small**, **c7i-flex.large**, and **m7i-flex.large** instance types for 6 months or until your credits are used up.

For more information, see Free Tier benefits before and after July 15, 2025.

- **Compare instance types**: You can compare different instance types by the following attributes: number of vCPUs, architecture, amount of memory (GiB), amount of storage (GB), storage type, and network performance.
- **Get advice**: You can get guidance and suggestions for instance types from the EC2 instance type finder. For more information, see Get recommendations from EC2 instance type finder.
- (Launch templates only) **Advanced**: To specify instance attributes and let Amazon EC2 identify the instance types with those attributes, choose **Advanced**, and then choose **Specify instance type attributes**.
  - **Number of vCPUs**: Enter the minimum and maximum number of vCPUs for your compute requirements. To indicate no limits, enter a minimum of `0`, and leave the maximum blank.
  - **Amount of memory (MiB)**: Enter the minimum and maximum amount of memory, in MiB, for your compute requirements. To indicate no limits, enter a minimum of `0`, and leave the maximum blank.
  - Expand **Optional instance type attributes** and choose **Add attribute** to express your compute requirements in more detail. For information about each attribute, see InstanceRequirementsRequest in the *Amazon EC2 API Reference*.
  - **Resulting instance types**: You can preview the instance types that match the specified attributes. To exclude instance types, choose **Add attribute**, and from the **Attribute** list, choose **Excluded instance types**. From the **Attribute value** list, select the instance types to exclude.

Key pair (login)

For **Key pair name**, choose an existing key pair, or choose **Create new key pair** to create a new one. For more information, see Amazon EC2 key pairs and Amazon EC2 instances.

> **Important**
>
> If you choose the **Proceed without key pair (Not recommended)** 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.

Network settings

The network settings define the IP addresses, security groups, and network interfaces for your instances. You can use the default network settings or configure them as needed.

- (Launch instance wizard only) **VPC**: Choose an existing VPC for your instance. The default VPC for the Region is selected by default. Alternatively, you can choose a VPC that you created or that was shared with you. For more information, see Virtual private clouds for your EC2 instances.
- **Subnet**: Choose a subnet for your instance or choose **Create new subnet** to create a new subnet using the Amazon VPC console.
  - You can create a subnet in any Availability Zone, Local Zone, Wavelength Zone, or Outpost Zone for the selected VPC.
  - To launch the instance in an IPv6-only subnet, the instance must be a Nitro-based instance.
- (Launch instance wizard only) **Auto-assign Public IP**: Enable or disable auto-assignment of public IPv4 addresses. When launching instances into a default subnet, the default value is **Enable**. When launching instances into a nondefault subnet the default value is **Disable**. For more information, see Public IPv4 addresses.

  You can't enable this option for nondefault subnets if you add a secondary network interface. For more information, see the section called “Assign a public IPv4 address at launch”.
- (Launch instance wizard only) **Auto-assign IPv6 IP**: Enable or disable auto-assignment of IPv6 addresses. For more information, see IPv6 addresses.
- **Firewall (security groups)**: Choose an existing security group or create a new one. Ensure that your security group has rules that allow traffic to and from your instances. All other traffic is ignored.

If you create a new security group, we automatically create an inbound rule that allows you to connect to your instance from all IP addresses over SSH (Linux instances) or RDP (Windows instances. You can remove or modify this rule as needed. You can add rules as needed. For more information, see Configure security group rules.

> **Warning**
>
> Rules that enable all IP addresses to access your instance over SSH or RDP are acceptable if you are briefly launching a test instance and will stop or terminate it after a short time. They are unsafe for production environments. You should authorize only a specific IP address range to access your instances.

This security group is added to the primary network interface and any secondary network interfaces. You can select additional security groups for your network interfaces, but you can't remove the one that you select here.

- **Advanced network configuration** – You can configure the primary network interface as needed. To add a secondary network interface, choose **Add network interface**. The number of network interfaces that you can add depends on the instance type that you selected. Note that this section is available only if you choose a subnet.
  - **Device index**: The device index. The primary network interface must be assigned to index 0.
  - **Network interface**: The network interface. Select **New interface** to let Amazon EC2 create a new interface, or select an existing, available network interface. If you select an existing network interface as the primary network interface, you can't enable **Auto-assign Public IP** for nondefault subnets.
  - **Description**: A description for the new network interface.
  - **Subnet**: The subnet in which to create the new network interface. The instance is launched in the same subnet as the primary network interface.

    You must choose a subnet for a secondary network interface from the same Availability Zone as the subnet for the primary network interface. If you select a subnet from another VPC, the label **Multi-VPC** appears next to the network interface. This enables you to create multi-homed instances across VPCs with different networking and security configurations.

    To launch an EC2 instance into an IPv6-only subnet, you must use a Nitro-based instance. When launching an IPv6-only instance, it is possible that DHCPv6 might not immediately provide the instance with the IPv6 DNS name server. During this initial delay, the instance might not resolve public domains. You can change the configuration file and re-image your AMI so that the file has the IPv6 DNS name server address immediately on booting.

- **Security groups**: The security groups to associate with the network interface. You must choose a security group from the same VPC as the subnet for the network interface.
- (Launch templates only) **Auto-assign public IP**: Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address and instances in a nondefault subnet do not. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see Public IPv4 addresses.
- **Primary IP**: A private IPv4 address from the range of your subnet. Leave blank to let Amazon EC2 choose a private IPv4 address for you.
- **Secondary IP**: Additional private IPv4 addresses from the range of your subnet. Choose **Manually assign** and enter an IPv4 address. Choose **Add IP** to add another IPv4 address. Alternatively, choose **Automatically assign** and enter a value to indicate the number of IPv4 addresses that Amazon EC2 chooses for you.
- (IPv6-only) **IPv6 IPs**: IPv6 addresses from the range of the subnet. Choose **Manually assign** and enter an IPv6 address. Choose **Add IP** to add another IPv6 address. Alternatively, choose **Automatically assign** and enter a value to indicate the number of IPv6 addresses that Amazon EC2 chooses for you.
- **IPv4 Prefixes**: The IPv4 prefixes for the network interface. Choose **Manually assign** and enter an IPv4 prefix. Alternatively, choose **Automatically assign** and enter a value to indicate the number of IPv4 prefixes that Amazon EC2 chooses for you.
- **IPv6 Prefixes**: The IPv6 prefixes for the network interface. Choose **Manually assign** and enter an IPv6 prefix. Alternatively, choose **Automatically assign** and enter a value to indicate the number of IPv6 prefixes that Amazon EC2 chooses for you.
- (Dual-stack and IPv6-only) **Assign Primary IPv6 IP**: If you select a dual-stack or IPv6-only subnet, assign a primary IPv6 address. This helps prevent disruptions to traffic to the instance or network interface. Enable this option if you rely on the IPv6 address not changing. You can't remove the primary IPv6 address later on. When you enable an IPv6 GUA address to be a primary IPv6, the first IPv6 GUA becomes the primary IPv6 address until the instance is terminated or the network interface is detached. If you have multiple IPv6 addresses associated with a network interface and you let Amazon EC2 assign a primary IPv6 address, the first IPv6 GUA address associated with the network interface is the primary IPv6 address.
- **Delete on termination**: Indicates whether the network interface is deleted when the instance is deleted.

- **Interface type**: The network interface type:
  - **ENA**: A high-performance network interface designed to handle high throughput and packet-per-second rates for TCP/IP protocols while minimizing CPU usage. This is the default value. For more information about ENA, see Elastic Network Adapter.
  - **EFA with ENA**: A network interface that supports both ENA and EFA devices for traditional TCP/IP based transport along with SRD based transport. For more information about EFA, see Elastic Fabric Adapter.
  - **EFA-only**: A high-performance network interface designed to handle high throughput, low latency inter-node communication for SRD based transport while bypassing the operating system stack. EFA-only network interfaces do not support IP addresses. For more information about EFA, see Elastic Fabric Adapter.
- **Elastic Fabric Adapter**: Indicates whether the network interface is an Elastic Fabric Adapter. For more information, see Elastic Fabric Adapter for AI/ML and HPC workloads on Amazon EC2.
- **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.
- **ENA Express**: ENA Express is powered by AWS Scalable Reliable Datagram (SRD) technology. SRD technology uses a packet spraying mechanism to distribute load and avoid network congestion. Enabling ENA Express allows supported instances to communicate using SRD on top of regular TCP traffic when possible. The launch instance wizard or launch template does not include ENA Express configuration for the instance unless you select **Enable** or **Disable** from the list.
- **ENA Express UDP**: If you've enabled ENA Express, you can optionally use it for UDP traffic. The launch instance wizard or launch template does not include ENA Express configuration for the instance unless you select **Enable** or **Disable**.

Configure storage

The AMI you selected includes one or more volumes of storage, including the root volume. You can specify additional volumes to attach to the instance.

(Launch instance wizard only) You can use the **Simple** or **Advanced** view. With the **Simple** view, you specify the size and type of the volume. To specify all volume parameters, choose the **Advanced** view (at top right of the card).

By using the **Advanced** view, you can configure each volume as follows:

- **Storage type**: Select Amazon EBS or instance store volumes to associate with your instance. The volume types available in the list depend on the instance type that you've chosen. For more information, see Instance store temporary block storage for EC2 instances and Amazon EBS volumes.
- **Device name**: Select from the list of available device names for the volume.
- **Snapshot**: Select the snapshot from which to restore the volume. You can search for available shared and public snapshots by entering text into the **Snapshot** field.
- **Size (GiB)**: For EBS volumes, you can specify a storage size.
- **Volume type**: For EBS volumes, select a volume type. For more information, see Amazon EBS volume types in the *Amazon EBS User Guide*.
- **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, choose **Yes** to delete the volume when the instance is terminated, or choose **No** to keep the volume. For more information, see Preserve data when an instance is terminated.
- **Encrypted**: If the instance type supports EBS encryption, you can choose **Yes** to 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 in the *Amazon EBS User Guide*.
- **KMS key**: If you selected **Yes** for **Encrypted**, then you must select a customer managed key to use to encrypt 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 specify the ARN of any customer managed key that you created.
- **Throughput**: If you have selected the `gp3` volume type, then you can enter the throughput, in MiB/s, that the volume can support.
- **Volume initialization rate**: If you have selected a snapshot, you can optionally specify the volume initialization rate, in MiB/s, at which the snapshot blocks are to be downloaded from Amazon S3 to the volume. For more information, see Use an Amazon EBS Provisioned Rate for Volume Initialization. To use the default initialization rate or fast snapshot restore (if it is enabled for the selected snapshot), don't specify a rate.
- **File systems**: Mount an Amazon EFS or Amazon FSx file system to the instance. For more information about mounting an Amazon EFS file system, see Use Amazon EFS with Amazon EC2 Linux instances. For more information about mounting an Amazon FSx file system, see Use Amazon FSx with Amazon EC2 instances

Advanced details

For **Advanced details**, expand the section to view the fields and specify any additional parameters for the instance.

- (Launch instance wizard only) **Domain join directory**: Select the AWS Directory Service directory (domain) to which your instance is joined to after launch. If you select a domain, you must select an IAM role with the required permissions. For more information about domain joining, see Seamlessly join an Amazon EC2 Linux instance to your AWS Managed Microsoft AD directory (Linux instances) and Seamlessly join an Amazon EC2 Windows instance to your AWS Managed Microsoft AD directory (Windows instances).
- **IAM instance profile**: Select an IAM instance profile to associate with the instance. This is a container for an IAM role. For more information, see IAM roles for Amazon EC2.
- **Hostname type**: Select whether the guest OS hostname of the instance will include the resource name or the IP name. For more information, see EC2 instance hostnames and domains.
- **DNS Hostname**: Determines if the DNS queries to the resource name or the IP name (depending on what you selected for **Hostname type**) will respond with the IPv4 address (A record), IPv6 address (AAAA record), or both. For more information, see EC2 instance hostnames and domains.
- **Instance auto-recovery**: When enabled, recovers your instance if system status checks fail. This setting is enabled by default at launch for supported instance types. For more information, see Configure simplified automatic recovery on an Amazon EC2 instance.
- **Shutdown behavior**: Select whether the instance should stop or terminate when shut down. For more information, see Change instance initiated shutdown behavior.
- **Stop - Hibernate behavior**: To enable hibernation, choose **Enable**. This field is available only if your instance meets the hibernation prerequisites. For more information, see Hibernate your Amazon EC2 instance.
- **Termination protection**: To prevent accidental termination, choose **Enable**. For more information, see Change instance termination protection.
- **Stop protection**: To prevent accidental stopping, choose **Enable**. For more information, see Enable stop protection.
- **Detailed CloudWatch monitoring**: Choose **Enable** to turn on detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see Monitor your instances using CloudWatch.

- **Credit specification**: Choose **Unlimited** to enable applications to burst beyond the baseline for as long as needed. This field is only valid for **T** instances. Additional charges may apply. For more information, see Burstable performance instances.
- **Placement group**: Specify a placement group in which to launch the instance. You can select an existing placement group, or create a new one. Not all instance types support launching an instance in a placement group. For more information, see Placement groups for your Amazon EC2 instances.
- **EBS-optimized instance**: An instance that's optimized for Amazon EBS uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, choose **Enable** to enable it. Additional charges apply. For more information, see the section called "EBS optimization".
- **Instance bandwidth configuration**: You can boost either your networking bandwidth or your EBS bandwidth. For supported instance types only. For more information, see EC2 instance bandwidth weighting configuration.
- **Purchasing option**: Choose **Spot Instances** to request Spot Instances at the Spot price, capped at the On-Demand price, and choose **Customize Spot Instance options** to change the default Spot Instance settings. You can set your maximum price (not recommended), and change the request type, request duration, and interruption behavior. If you do not request a Spot Instance, Amazon EC2 launches an On-Demand Instance by default. For more information, see Manage your Spot Instances.
- **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.
- **Tenancy**: Choose whether to run your instance on shared hardware (**Shared**), isolated, dedicated hardware (**Dedicated**), or on a Dedicated Host (**Dedicated host**). If you choose to launch the instance onto a Dedicated Host, you can specify whether to launch the instance into a host resource group or you can target a specific Dedicated Host. Additional charges may apply. For more information, see Amazon EC2 Dedicated Instances and Amazon EC2 Dedicated Hosts.
- **RAM disk ID**: (Only valid for paravirtual (PV) AMIs) Select a RAM disk for the instance. If you have selected a kernel, you might need to select a specific RAM disk with the drivers to support it.
- **Kernel ID**: (Only valid for paravirtual (PV) AMIs) Select a kernel for the instance.
- **Nitro Enclave**: Allows you to create isolated execution environments, called enclaves, from Amazon EC2 instances. 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*.

- **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*.
- **Specify CPU options**: In the launch instance wizard, this field is only visible if the selected instance type supports specifying 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 CPU options for Amazon EC2 instances.
- **Metadata accessible**: You can enable or disable access to the Instance Metadata Service (IMDS). For more information, see Configure instance metadata options for new instances.
- **Metadata IPv6 endpoint**: You can enable the instance to use the IMDS IPv6 address `[fd00:ec2::254]` to retrieve instance metadata. This option is only available if you are launching Nitro-based instances into an IPv6-supported subnet (dual stack or IPv6 only). For more information about retrieving instance metadata, see Access instance metadata for an EC2 instance.
- **Metadata version**: If you enable access to the IMDS, 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.
- **Metadata response hop limit**: If you enable the IMDS, you can set the allowable number of network hops for the metadata token. For more information, see Configure instance metadata options for new instances.
- **Allow tags in metadata**: If you select **Enable**, the instance will allow access to all of its tags from its metadata. If no value is specified, then by default, access to the tags in instance metadata is not allowed. For more information, see Enable access to tags in instance metadata.
- **User data**: You can specify user data to configure an instance during launch, or to run a configuration script. For more information about user data for Linux instances, see Run commands when you launch an EC2 instance with user data input. For more information about user data for Windows instances, see How Amazon EC2 handles user data for Windows instances.

Summary

Use the **Summary** panel to specify the number of instances to launch, to review your instance configuration, and to launch your instances.

- **Number of instances**: Enter the number of instances to launch. All of the instances will launch with the same configuration.

> **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) If you specify more than one instance, to help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **consider EC2 Auto Scaling** to create a launch template 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 instances in an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.

- Review the details of your instance, and make any necessary changes. You can navigate directly to a section by choosing its link in the **Summary** panel.
- When you're ready to launch your instance, choose **Launch instance**.

Launch an EC2 instance using the launch instance wizard in the console

You can launch an Amazon EC2 instance using the launch instance wizard in the Amazon EC2 console. The wizard provides default values for the launch parameters, which you can either accept or modify to suit your requirements. The only parameter that is not specified is the key pair. If you choose to accept the default values, you can quickly launch an instance by selecting only a key pair.

> **Important**
>
> You incur charges for the instance while the instance is in the `running` state, even if it remains idle. However, if you qualify for the Free Tier, you might not incur charges. For more information, see Track your Free Tier usage for Amazon EC2.

For a description of each parameter in the launch instance wizard, see Reference for Amazon EC2 instance configuration parameters.

Topics

- Quickly launch an instance
- Launch an instance using defined parameters

Quickly launch an instance

To set up an instance quickly for testing purposes, follow these steps. You'll select the operating system and your key pair, and accept the default values. Except for the key pair, the launch instance wizard provides default values for all of the parameters. You can accept any or all of the defaults, or configure an instance by specifying your own values for each parameter.

For a description of each parameter in the launch instance wizard, see Reference for Amazon EC2 instance configuration parameters.

To quickly launch an instance using the launch instance wizard

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 AWS Region is displayed (for example, US East (Ohio)). If needed, select a different Region in which to launch the instance.
3. From the Amazon EC2 console dashboard, choose **Launch instance**.
4. (Optional) Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
5. Under **Application and OS Images (Amazon Machine Image)**, choose **Quick Start**, and then choose the operating system (OS) for your instance.
6. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or create a new one.
7. In the **Summary** panel, choose **Launch instance**.

Launch an instance using defined parameters

If you're launching an instance that you'll use in production, you'll need to configure the instance to suit your requirements. For a description of each parameter in the launch instance wizard, see Reference for Amazon EC2 instance configuration parameters.

To launch an instance by defining all the launch parameters using the launch instance wizard

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 AWS Region is displayed (for example, US East (Ohio)). If needed, select a different Region in which to launch the instance.
3. From the Amazon EC2 console dashboard, choose **Launch instance**.
4. (Optional) Under **Name and tags**, for **Name**, enter a descriptive name for your instance so that you can easily keep track of it.

   The instance name is a tag, where the key is **Name**, and the value is the name that you specify.
5. Under **Application and OS Images (Amazon Machine Image)**, choose the operating system (OS) for your instance, and then choose an AMI.

   An AMI is a template that contains the operating system and software required to launch your instance.
6. Under **Instance type**, choose an instance type.

   The instance type determines the hardware configuration (CPU, memory, storage, and networking capacity) and size of the host computer used for your instance.

   If you're not sure which instance type to choose, you can do the following:

   - Choose **Compare instance types** to compare different instance types by the following attributes: number of vCPUs, architecture, amount of memory (GiB), amount of storage (GB), storage type, and network performance.
   - Choose **Get advice** to get guidance and suggestions for instance types from the EC2 instance type finder. For more information, see Get recommendations from EC2 instance type finder.

   > **Note**
   >
   > Depending on when you created your account, you might be able to use instance types for free under the Free Tier. These instance types are labeled **Free tier eligible**.

> If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type, or the **t3.micro** instance type in Regions where **t2.micro** is unavailable. Be aware that when you launch a **t3.micro** instance, it defaults to **Unlimited** mode, which might incur additional charges based on CPU usage.
> If you created your AWS account on or after July 15, 2025, you can use **t3.micro**, **t3.small**, **t4g.micro**, **t4g.small**, **c7i-flex.large**, and **m7i-flex.large** instance types for 6 months or until your credits are used up.
> For more information, see Free Tier benefits before and after July 15, 2025.

7. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or create a new one. If you do not require a key pair to connect to your instance, you can choose **Proceed without a key pair (not recommended)**.
8. Under **Network settings**, you can keep the defaults if you're launching a test instance. If you're launching a production instance, it's best practice to control traffic into and out of your instance using network settings and security groups that you define.
9. Under **Configure storage**, you can keep the defaults or specify additional storage. The AMI you selected includes one or more volumes of storage, including the root volume. You can specify additional volumes to attach to the instance.

   You can use the **Simple** or **Advanced** view. With the **Simple** view, you specify the size and type of the volume. To specify all volume parameters, choose the **Advanced** view (at top right of the card).
10. For **Advanced details**, expand the section to view the fields and specify any additional parameters for your instance.
11. In the **Summary** panel, you can do the following:

    a. Specify the number of instances to launch.

    b. Review your instance configuration, and navigate directly to a section by choosing its link.

    c. When you're ready to launch your instance, choose **Launch instance**.

    If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.
12. (Optional) You can create a billing alert for the instance. On the confirmation screen, under **Next Steps**, choose **Create billing alerts** and follow the directions. Billing alerts can also be

created after you launch the instance. For more information, see Creating a billing alarm to monitor your estimated AWS charges in the *Amazon CloudWatch User Guide*.

Launch EC2 instances using a launch template

An Amazon EC2 launch template stores instance launch parameters so that you don't have to specify them every time you launch an instance.

Several instance launch services can optionally use launch templates when launching instances, while for other services, like EC2 Fleet, instances can't be launched unless a launch template is used. This topic describes how to use a launch template when launching an instance using the EC2 launch instance wizard, Amazon EC2 Auto Scaling, EC2 Fleet, and Spot Fleet.

For more information about launch templates, including how to create a launch template, see Store instance launch parameters in Amazon EC2 launch templates.

Topics

Launch an Amazon EC2 instance using a launch template

You can use the parameters contained in a launch template to launch an Amazon EC2 instance. After selecting the launch template, but before launching the instance, you can modify the launch parameters.

Instances that are launched using a launch template are automatically assigned two tags with the keys `aws:ec2launchtemplate:id` and `aws:ec2launchtemplate:version`. You can't remove or edit these tags.

Console

To launch an instance using a launch template

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

2. Use one of the following options to select the launch template:

   - From the Amazon EC2 console dashboard, choose the down arrow next to **Launch instance**, choose **Launch instance from template**, and then for **Source template**, select a launch template.
   - In the navigation pane, choose **Launch Templates**, select the launch template, and choose **Actions**, **Launch instance from template**.

3. For **Source template version**, select the launch template version to use.
4. (Optional) You can modify the values for any of the launch parameters. If you don't modify a value, the value defined by the launch template is used. If no value was specified in the launch template, the default value for the parameter is used.
5. In the **Summary** panel, for **Number of instances**, specify the number of instances to launch.
6. Choose **Launch instance**.

   If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

AWS CLI

To launch an instance from a launch template

- Use the run-instances command and specify the `--launch-template` parameter. Optionally specify the launch template version to use. If you don't specify the version, the default version is used.

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- To override a launch template parameter, specify the parameter in the run-instances command. The following example overrides the instance type that's specified in the launch template (if any).

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --instance-type t2.small
```

- If you specify a nested parameter that's part of a complex structure, the instance is launched using the complex structure as specified in the launch template plus any additional nested parameters that you specify.

  In the following example, the instance is launched with the tag *Owner*=*TeamA* as well as any other tags that are specified in the launch template. If the launch template has an existing tag with a key of *Owner*, the value is replaced with *TeamA*.

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]"
```

  In the following example, the instance is launched with a volume with the device name */dev/xvdb* as well as any other block device mappings that are specified in the launch template. If the launch template has an existing volume defined for */dev/xvdb*, its values are replaced with the specified values.

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --block-device-mappings "DeviceName=/dev/
xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

PowerShell

To launch an instance from a launch template using the AWS Tools for PowerShell

- Use the New-EC2Instance command and specify the `-LaunchTemplate` parameter. Optionally specify the launch template version to use. If you don't specify the version, the default version is used.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
```

```
    }
)
```

- To override a launch template parameter, specify the parameter in the [New-EC2Instance](#) command. The following example overrides the instance type that's specified in the launch template (if any).

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
)
```

- If you specify a nested parameter that's part of a complex structure, the instance is launched using the complex structure as specified in the launch template plus any additional nested parameters that you specify.

  In the following example, the instance is launched with the tag *Owner*=*TeamA* as well as any other tags that are specified in the launch template. If the launch template has an existing tag with a key of *Owner*, the value is replaced with *TeamA*.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
        }
) `
    -TagSpecification (
        New-Object -TypeName Amazon.EC2.Model.TagSpecification -Property @{
            ResourceType = 'instance';
            Tags         = @(
                @{key = "Owner"; value = "TeamA" },
                @{key = "Department"; value = "Operations" }
```

```
            )
        }
)
```

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.

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
) `
    -BlockDeviceMapping  (
        New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping -Property @{
            DeviceName = '/dev/xvdb';
            EBS        = (
                New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property @{
                    VolumeSize = 25;
                    VolumeType = 'gp3'
                }
            )
        }
)
```

If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

Launch instances in an Amazon EC2 Auto Scaling group using a launch template

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.

Before you can create an Auto Scaling group using a launch template, you must first create a launch template that includes the parameters required to launch an instance in an Auto Scaling group. Some parameters are required, such as the ID of the AMI, and some parameters are not available to use with an Auto Scaling group. The console provides guidance to help you create a template that you can use with Amazon EC2 Auto Scaling.

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

- For the instructions, see Create an Auto Scaling group using a launch template in the *Amazon EC2 Auto Scaling User Guide*.

To create or update an 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.

For more information, see the following topics in the *Amazon EC2 Auto Scaling User Guide*:

- Create a launch template for an Auto Scaling group
- Create a launch template using advanced settings
- Examples for creating and managing launch templates with the AWS Command Line Interface (AWS CLI) – Provides examples that show how to create launch templates with various parameter combinations.
- Create Auto Scaling groups using launch templates
- Update an Auto Scaling group

Launch an EC2 Fleet using a launch template

A launch template is a requirement when creating an EC2 Fleet request. 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.

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.

Launch a Spot Fleet using a launch template

A launch template is optional when creating a Spot Fleet request. If you don't use a launch template, you can manually specify the launch parameters. If you use a launch template, 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 [Create a Spot Fleet](#).

To create a Spot Fleet request using a launch template

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Choose **Request Spot Instances**.
4. Under **Launch parameters**, choose **Use a launch template**.
5. For **Launch template**, choose a launch template, and then, from the field to the right, choose the launch template version.
6. Configure your Spot Fleet by selecting different options on this screen. For more information about the options, see [Create a Spot Fleet request using defined parameters](#).
7. When you're ready to create your Spot Fleet, choose **Launch**.

To create a Spot Fleet request using a launch template

- Use the [request-spot-fleet](#) command. Use the `LaunchTemplateConfigs` parameter to specify the launch template and any overrides for the launch template.

Launch an EC2 instance using details from an existing instance

The Amazon EC2 console provides a **Launch more like this** option that enables you to use a current instance as a base for launching other instances. This option automatically populates the Amazon EC2 launch instance wizard with certain configuration details from the selected instance.

Considerations

- We do not clone your instances; we only replicate some of the configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI. Create a [launch template](#) to ensure that you launch your instances using the same launch details.
- The current instance must be in the `running` state.

Copied details

The following configuration details are copied from the selected instance into the launch instance 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.
- 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
- [Windows instances] 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 to your AWS Managed Microsoft AD Active Directory in the *AWS Directory Service Administration Guide*.

Details not copied

The following configuration details are not copied from your selected instance. Instead, the wizard applies their default settings or behavior:

- Number of network interfaces – The default is one network interface, which is the primary network interface (eth0).
- Storage – The default storage configuration is determined by the AMI and the instance type.

To launch more instances like 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 an instance, and then choose **Actions**, **Images and templates**, **Launch more like this**.
4. The launch instance wizard opens. You can make any necessary changes to the instance configuration by selecting different options on this screen.

   When you are ready to launch your instance, choose **Launch instance**.
5. If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

Launch an Amazon EC2 instance from an AWS Marketplace AMI

You can subscribe to an AWS Marketplace AMI and launch an instance from it using the Amazon EC2 console or a command line tool. For more information about AWS Marketplace AMIs, see Paid AMIs in the AWS Marketplace for Amazon EC2 instances.

To cancel your subscription to the AMI after launch, you must first terminate all instances that were launched from the AMI. For more information, see Manage your AWS Marketplace subscriptions.

To launch an instance from an AWS Marketplace AMI using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 console dashboard, choose **Launch instance**.
3. (Optional) Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
4. Under **Application and OS Images (Amazon Machine Image)**, choose **Browse more AMIs**, and then choose the **AWS Marketplace AMIs** tab. Find a suitable AMI by browsing the categories or using the search functionality. To choose a product, choose **Select**.
5. A window opens with 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 one of the following buttons:

- **Subscribe on instance launch** – Your subscription starts when you choose **Launch instance** (at Step 10).
- **Subscribe now** – Your subscription starts immediately. While the subscription is underway, you can configure the instance by continuing with the steps in this procedure. If there are any problems with your credit card details, you will be asked to update your account details.

> **Note**
>
> You're 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 when you select an instance type. Additional taxes might also apply to the product.

6. For **Instance type**, select an instance type for your instance. The instance type defines the hardware configuration and size of the instance to launch.
7. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or create a new one.
8. Under **Network settings**, for **Firewall (security groups)**, take note of the new security group that was created according to the vendor's specifications for the product. The security group might 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.
9. You can use the other fields on the screen to configure your instance, add storage, and add tags. For information about the different options that you can configure, see [Reference for Amazon EC2 instance configuration parameters](https://docs.aws.amazon.com).
10. In the **Summary** panel, under **Software Image (AMI)**, check the details of the AMI from which you're about to launch the instance. Also check the other configuration details that you specified. When you're ready to launch your instance, choose **Launch instance**.
11. Depending on the product you've subscribed to, the instance might take a few minutes or more to launch. If you chose **Subscribe on instance launch** at Step 5, 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 all instances** to go to the **Instances** page.

> **Note**
>
> You are charged the subscription price as long as your instance is in the `running` state, even if it is idle. If your instance is stopped, you might still be charged for storage.

12. When your instance is in the `running` state, you can connect to it. To do this, select your instance in the list, choose **Connect**, and choose a connection option. For more information about connecting to your instance, see Connect to your EC2 instance.

> **Important**
>
> Check the vendor's usage instructions carefully, as you might need to use a specific username to connect to your instance. For information about accessing your subscription details, see Manage your AWS Marketplace subscriptions.

13. If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

To launch an instance from an AWS Marketplace AMI using a command line tool

To launch instances from AWS Marketplace products using a command line tool, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launch your instance in the *AWS Command Line Interface User Guide*.
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
Query API	Use the RunInstances request.

Connect to your EC2 instance

Your Amazon EC2 instance is a virtual server in the AWS Cloud. To log on to your instance, you must establish a connection to the instance. How you connect to your instance depends on the operating system of the instance and the operating system on the computer that you use to connect to the instance. The following table details the requirements for each connection method.

Connection option	Instance operating system	Inbound traffic rule	IAM permissions	Instance profile role	Software on instance	Software on connecting system	Key pair
SSH client	Linux	Yes	No	No	No	Yes	Yes
EC2 Instance Connect	Linux	Yes	Yes	No	Yes [1]	No	No
PuTTY	Linux	Yes	No	No	No	Yes	Yes
RDP client	Windows	Yes	No	No	No	Yes	Yes [2]
Fleet Manager	Windows	No	Yes	Yes	Yes [1]	No	Yes
Session Manager	Linux, Windows	No	Yes	Yes	Yes [1]	No	No
EC2 Instance Connect Endpoint	Linux, Windows	Yes	Yes	No	No	No	Yes

[1] The required software is only pre-installed on certain AMIs. You can manually install the required software as needed on supported operating systems.

[2] The key pair is only required if you are using the randomly generated password for the local Administrator user account.

For more information, see the documentation for the connection option that you intend to use.

Connection options

- Connect to your Linux instance using an SSH client
- Connect to your Linux instance using PuTTY
- Connect to your Windows instance using an RDP client
- Connect to your Windows instance using Fleet Manager
- Connect using Session Manager
- Connect using a public IP and EC2 Instance Connect
- Connect using a private IP and EC2 Instance Connect Endpoint

General connection prerequisites

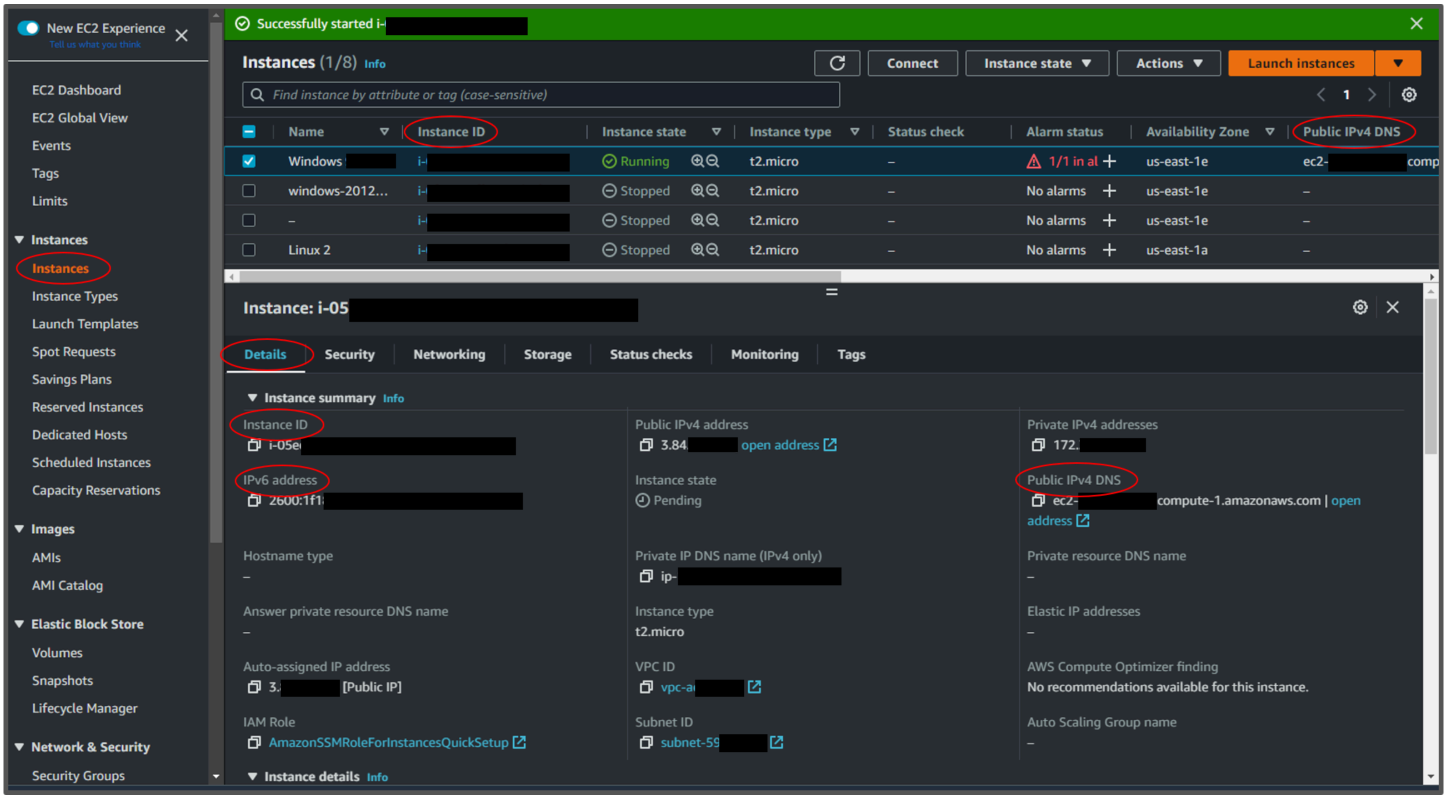
The following are general prerequisites to connect to an instance. Note that there might be additional prerequisites that are specific to the connection option that you choose.

General prerequisites

- Check that your instance has passed its status checks. It can take a few minutes for an instance to be ready to accept connection requests. For more information, see View status checks.
- Get the required instance details.
- Locate the private key and set permissions.
- (Optional) Get the instance fingerprint.

Get the required instance details

To prepare to connect to your instance, get the following information from the Amazon EC2 console or by using the command line.



- **Get the public DNS name of the instance.**

  You can get the public DNS for your instance from the Amazon EC2 console. Check the **Public IPv4 DNS** column of the **Instances** pane. If this column is hidden, choose the settings icon (

) in the top-right corner of the screen, and select **Public IPv4 DNS**. You can also find the public DNS in the instance information section of the **Instances** pane. When you select the instance in the **Instances** pane of the Amazon EC2 console, information about that instance will appear on the lower half of the page. Under the **Details** tab, look for **Public IPv4 DNS**.

  If you prefer, you can use the describe-instances (AWS CLI) or Get-EC2Instance (AWS Tools for Windows PowerShell) commands.

  If no **Public IPv4 DNS** is displayed, verify that the **Instance state** is **Running**, and that you have not launched the instance in a private subnet. If you launched your instance using the launch instance wizard, you may have edited the **Auto-assign public IP** field under **Network settings** and changed the value to **Disable**. If you disable the **Auto-assign public IP** option, the instance is not assigned a public IP address when it is launched.

- **(IPv6 only instances) Get the IPv6 address of the instance.**

If you assigned an IPv6 address to your instance, you can optionally connect to the instance using its IPv6 address instead of a public IPv4 address or public IPv4 DNS hostname. Your local computer must have an IPv6 address and must be configured to use IPv6. You can get the IPv6 address of your instance from the Amazon EC2 console. Check the **IPv6 IPs** column of the **Instances** pane. Or, you can find the IPv6 address in the instance information section. When you select the instance in the **Instances** pane of the Amazon EC2 console, information about that instance will appear on the lower half of the page. Under the **Details** tab, look for **IPv6 address**.

If you prefer, you can use the describe-instances (AWS CLI) or Get-EC2Instance (AWS Tools for Windows PowerShell) commands. For more information about IPv6, see IPv6 addresses.

- **(Linux instances) Get the username for your instance.**

  You can connect to your instance using the username for your user account or the default username for the AMI that you used to launch your instance.

  - **Get the username for your user account.**

    For more information about how to create a user account, see Manage system users on your Amazon EC2 Linux instance.

  - **Get the default username for the AMI that you used to launch your instance.**
    - **Amazon Linux** – `ec2-user`
    - **CentOS** – `centos` or `ec2-user`
    - **Debian** – `admin`
    - **Fedora** – `fedora` or `ec2-user`
    - **RHEL** – `ec2-user` or `root`
    - **SUSE** – `ec2-user` or `root`
    - **Ubuntu** – `ubuntu`
    - **Oracle** – `ec2-user`
    - **Bitnami** – `bitnami`
    - **Rocky Linux** – `rocky`
    - **Other** – Check with the AMI provider

Locate the private key and set permissions

You must know the location of your private key file to make the initial connection to a Linux instance using SSH or a Windows instances using RDP. For SSH connections, you must set file permissions so that only you can read the private key.

For information about how key pairs work when using Amazon EC2, see Amazon EC2 key pairs and Amazon EC2 instances.

- **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 the section called "Identify the public key specified at launch".

  If you can't find your private key file, see I've lost my private key. How can I connect to my instance?

  (Linux instances) If you are connecting to your instance using PuTTY and need to convert the `.pem` file to `.ppk`, see Convert your private key using PuTTYgen.

- **(Linux instances) Set the permissions of your private key so that only you can read it.**
  - **Connect from macOS or Linux**

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

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

    If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see Error: Unprotected private key file.
  - **Connect from Windows**

    Open File Explorer and right-click on the `.pem` file. Select **Properties** > **Security tab** and choose **Advanced**. Choose **Disable inheritance**. Remove access to all users except for the current user.

(Optional) Get the instance fingerprint

To protect yourself from man-in-the-middle attacks, you can verify the authenticity of the instance you're about to connect to by verifying the fingerprint that is displayed. Verifying the fingerprint is useful if you launched your instance from a public AMI provided by a third party.

Task overview

First, get the instance fingerprint from the instance. Then, when you connect to the instance and are prompted to verify the fingerprint, compare the fingerprint you obtained in this procedure with the fingerprint that is displayed. If the fingerprints don't match, someone might be attempting a man-in-the-middle attack. If they match, you can confidently connect to your instance.

Prerequisites to get the instance fingerprint

- The instance must not be in the `pending` state. The fingerprint is available only after the first boot of the instance is complete.
- You must be the instance owner to get the console output.
- There are various ways to get the instance fingerprint. If you want to use the AWS CLI, it must be installed on your local computer. For information about installing the AWS CLI, see Getting started with the AWS CLI in the *AWS Command Line Interface User Guide*.

To get the instance fingerprint

In Step 1, you get the console output, which includes the instance fingerprint. In Step 2, you find the instance fingerprint in the console output.

1. Get the console output using one of the following methods.

   Console

   1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   2. From the left navigator, choose **Instances**.
   3. Select your instance, and then choose **Actions**, **Monitor and troubleshoot**, **Get system log**.

AWS CLI

On your local computer (not on the instance you're connecting to), use the [get-console-output](#) command. If the output is large, [you can pipe it to a text file](#), where it might be easier to read.

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0 \
    --query Output \
    --output text > temp.txt
```

PowerShell

On your local computer, use the following [Get-EC2ConsoleOutput](#) cmdlet.

```
$encodedOutput = (Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0).Output
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($encodedOutput
```

2. In the console output, find the instance (host) fingerprint, which is located under `BEGIN SSH HOST KEY FINGERPRINTS`. There might be several instance fingerprints. When you connect to your instance, it will display only one of the fingerprints.

   The exact output can vary by operating system, AMI version, and whether AWS created the key pairs. The following is example output.

```
ec2:#############################################################
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 256 SHA256:l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY no comment (ECDSA)
ec2: 256 SHA256:kpEa+rw/Uq3zxaYZN8KT501iBtJOIdHG52dFi66EEfQ no comment (ED25519)
ec2: 2048 SHA256:L8l6pepcA7iqW/jBecQjVZClUrKY+o2cHLI0iHerbVc no comment (RSA)
ec2: -----END SSH HOST KEY FINGERPRINTS-----
ec2: #############################################################
```

> **Note**
>
> You'll reference this fingerprint when you connect to the instance.

Connect to your Linux instance using SSH

There are multiple ways to connect to your Linux instance using SSH. Some ways depend on the operating system of the local computer that you connect from. Other methods are browser-based, such as EC2 Instance Connect or AWS Systems Manager Session Manager, and can be used from any computer. You can use SSH to connect to your Linux instance and run commands, or use SSH to transfer files between your local computer and your instance.

Before you connect to your Linux instance using SSH, complete the following prerequisites:

- Check that your instance has passed its status checks. It can take a few minutes for an instance to be ready to accept connection requests. For more information, see View status checks.
- Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Rules to connect to instances from your computer.
- Get the required instance details.
- Locate the private key and set permissions.
- (Optional) Get the instance fingerprint.

Then, choose from one of the following options to connect to your Linux instance using SSH.

- Connect using an SSH client
- Connect using PuTTY
- Transfer files using SCP

If you can't connect to your instance and need help troubleshooting, see Troubleshoot issues connecting to your Amazon EC2 Linux instance.

Connect to your Linux instance using an SSH client

You can use Secure Shell (SSH) to connect to your Linux instance from your local computer. For more information about other options, see Connect to your EC2 instance.

> **Note**
>
> If you receive an error while attempting to connect to your instance, make sure that your instance meets all of the SSH connection prerequisites. If it meets all of the prerequisites,

and you're still not able to connect to your Linux instance, see Troubleshoot issues connecting to your Amazon EC2 Linux instance.

Contents

SSH connection prerequisites

Before you can connect to your Linux instance using SSH, complete the following tasks.

Complete the general prerequisites.

- Check that your instance has passed its status checks. It can take a few minutes for an instance to be ready to accept connection requests. For more information, see View status checks.
- Get the required instance details.
- Locate the private key and set permissions.
- (Optional) Get the instance fingerprint.

Allow inbound SSH traffic from your IP address.

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Rules to connect to instances from your computer.

Install an SSH client on your local computer (if needed).

Your local computer might have an SSH client installed by default. You can verify this by entering the following command in a terminal window. If your computer doesn't recognize the command, you must install an SSH client.

```
ssh
```

The following are some of the possible options for Windows. If your computer runs a different operating system, see the documentation for that operating system for SSH client options.

Install OpenSSH on Windows

After you install OpenSSH on Windows, you can connect to your Linux instance from your Windows computer using SSH. Before you begin, ensure that you meet the following requirements.

Windows version

The version of Windows on your computer must be Windows Server 2019 or later.

For earlier versions of Windows, download and install Win32-OpenSSH instead.

PowerShell requirements

To install OpenSSH on your Windows OS using PowerShell, you must be running PowerShell version 5.1 or later, and your account must be a member of the built-in Administrators group. Run `$PSVersionTable.PSVersion` from PowerShell to check your PowerShell version.

To check whether you are a member of the built-in Administrators group, run the following PowerShell command:

```
(New-Object
 Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurrent())).Is
```

If you are a member of the built-in Administrators group, the output is `True`.

To install OpenSSH for Windows using PowerShell, run the following PowerShell command.

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

The following is example output.

```
Path          :
Online        : True
RestartNeeded : False
```

To uninstall OpenSSH from Windows using PowerShell, run the following PowerShell command.

```
Remove-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

The following is example output.

```
Path          :
Online        : True
RestartNeeded : True
```

Install Windows Subsystem for Linux (WSL)

After you install WSL on Windows, you can connect to your Linux instance from your Windows computer using Linux command line tools, such as an SSH client.

Follow the instructions in Install Windows Subsystem for Linux on your EC2 Windows instance. If you follow the instructions in Microsoft's installation guide, they install the Ubuntu distribution of Linux. You can install a different Linux distribution if you prefer.

In a WSL terminal window, copy the `.pem` file (for the key pair that you specified for your instance at launch) from Windows to WSL. Note the fully-qualified path to the `.pem` file on WSL to use when connecting to your instance. For information about how to specify the path to your Windows hard drive, see How do I access my C drive?.

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

For information about uninstalling Windows Subsystem for Linux, see How do I uninstall a WSL Distribution?.

Connect to your Linux instance using an SSH client

Use the following procedure to connect to your Linux instance using an SSH client.

To connect to your instance using an SSH client

1. Open a terminal window on your computer.
2. Use the **ssh** command to connect to the instance. You need the details about your instance that you gathered as part of the prerequisites. For example, you need the location of the private key (`.pem` file), the username, and the public DNS name or IPv6 address. The following are example commands.
   - (Public DNS) To use the public DNS name, enter the following command.

     ```
     ssh -i /path/key-pair-name.pem instance-user-name@instance-public-dns-name
     ```

   - (IPv6) Alternatively, if your instance has an IPv6 address, enter the following command to use the IPv6 address.

```
ssh -i /path/key-pair-name.pem instance-user-name@2001:db8::1234:5678:1.2.3.4
```

The following is an example response.

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'
 can't be established.
ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no)?
```

3. (Optional) Verify that the fingerprint in the security alert matches the fingerprint. If these fingerprints don't match, someone might be attempting a man-in-the-middle attack. If they match, continue to the next step. For more information, see Get the instance fingerprint.
4. Enter **yes**.

   You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to
 the list of known hosts.
```

Connect to your Linux instance using PuTTY

You can connect to your Linux instance using PuTTY, a free SSH client for Windows.

If you're running Windows Server 2019 or later, we recommend that you use OpenSSH, an open source connectivity tool for remote login using the SSH protocol.

> **Note**
>
> If you receive an error while attempting to connect to your instance, make sure that your instance meets all of the SSH connection prerequisites. If it meets all of the prerequisites, and you're still not able to connect to your Linux instance, see Troubleshoot issues connecting to your Amazon EC2 Linux instance.

Contents

- Connect to your Linux instance

Prerequisites

Before you connect to your Linux instance using PuTTY, complete the following tasks.

Complete the general prerequisites.

- Check that your instance has passed its status checks. It can take a few minutes for an instance to be ready to accept connection requests. For more information, see View status checks.
- Get the required instance details.
- Locate the private key and set permissions.
- (Optional) Get the instance fingerprint.

Allow inbound SSH traffic from your IP address.

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Rules to connect to instances from your computer.

Install PuTTY on your local computer (if needed).

Download and install PuTTY from the PuTTY download page. If you already have an earlier version of PuTTY installed, we recommend that you download the latest version. Be sure to install the entire suite.

Convert your private key to PPK format using PuTTYgen.

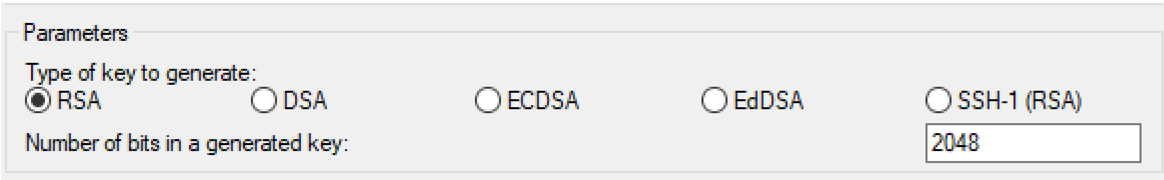
You must specify the private key for the key pair that you specified when you launched the instance. If you created the private key in .pem format, you must convert it to a PPK file for use with PuTTY. Locate the private key (.pem file), and then follow the steps in Convert your private key using PuTTYgen.

(Optional) Convert your private key using PuTTYgen

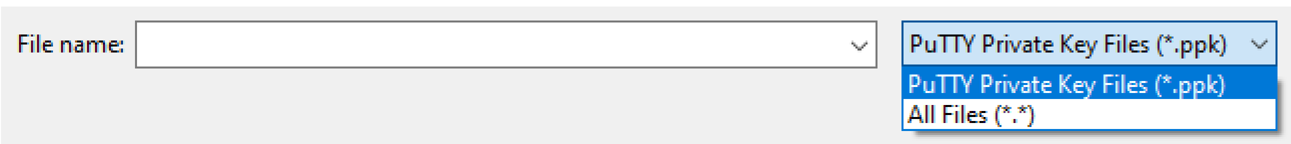
PuTTY does not natively support the PEM format for SSH keys. PuTTY provides a tool named PuTTYgen, which converts PEM keys to the required PPK format for PuTTY. If you created the key using PEM format instead of PPK format, you must convert your private key (.pem file) into this format (.ppk file) for use with PuTTY.

To convert your private key from PEM to PPK format

1. From the **Start** menu, choose **All Programs**, **PuTTY**, **PuTTYgen**.
2. Under **Type of key to generate**, choose **RSA**. If your version of PuTTYgen does not include this option, choose **SSH-2 RSA**.

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

4. Select your `.pem` file for the key pair that you specified when you launched your instance and choose **Open**. PuTTYgen displays a notice that the `.pem` file was successfully imported. Choose **OK**.
5. To save the key in the format that PuTTY can use, choose **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.

   Note

   A passphrase on a private key is an extra layer of protection. Even if your private key is discovered, it can't be used without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or to copy files to an instance.

6. Specify the same name for the key that you used for the key pair (for example, `key-pair-name`) and choose **Save**. PuTTY automatically adds the `.ppk` file extension.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

Connect to your Linux instance

Use the following procedure to connect to your Linux instance using PuTTY. You need the `.ppk` file that you created for your private key. For more information, see (Optional) Convert your private key using PuTTYgen in the preceding section. If you receive an error while attempting to connect to your instance, see Troubleshoot issues connecting to your Amazon EC2 Linux instance.

Last tested version – PuTTY .78

To connect to your instance using PuTTY

1. Start PuTTY (from the **Start** menu, search for **PuTTY** and then choose **Open**).
2. In the **Category** pane, choose **Session** and complete the following fields:

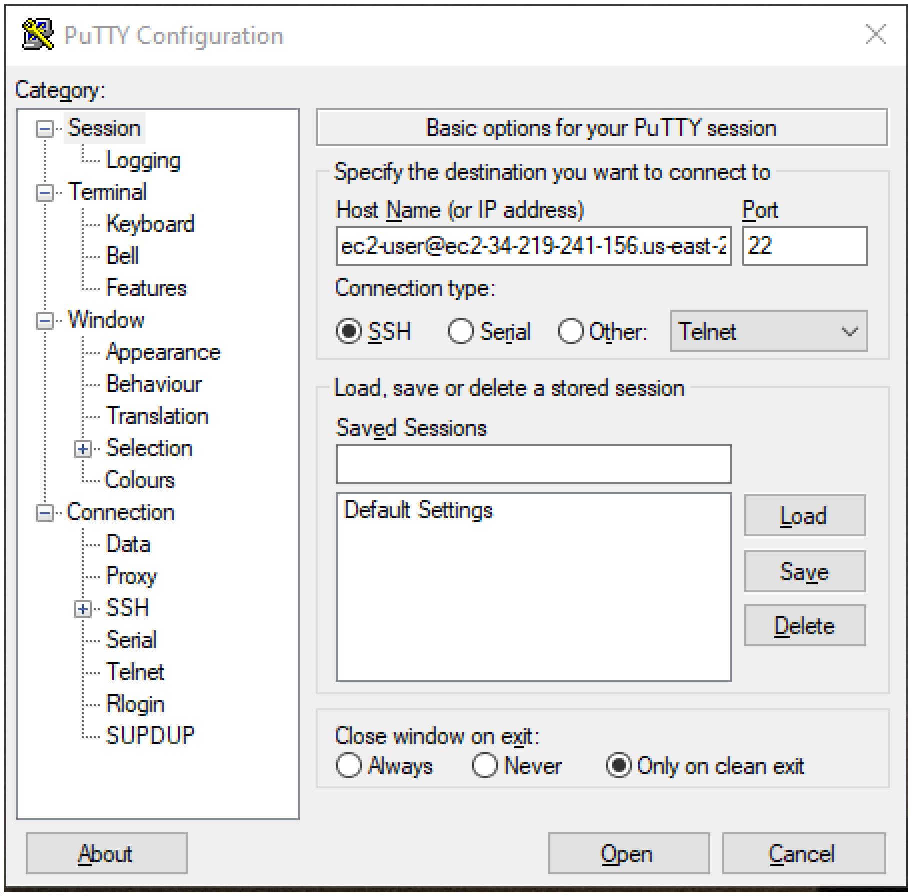
   a. In the **Host Name** box, do one of the following:

      - (Public DNS) To connect using your instance's public DNS name, enter *`instance-user-name`*@*`instance-public-dns-name`*.
      - (IPv6) Alternatively, if your instance has an IPv6 address, to connect using your instance's IPv6 address, enter *`instance-user-name`*@*`2001:db8::1234:5678:1.2.3.4`*.

      For information about how to get the username for your instance, and the public DNS name or IPv6 address of your instance, see Get the required instance details.

   b. Ensure that the **Port** value is 22.

   c. Under **Connection type**, select **SSH**.



3. (Optional) You can configure PuTTY to automatically send 'keepalive' data at regular intervals to keep the session active. This is useful to avoid disconnecting from your instance due to session inactivity. In the **Category** pane, choose **Connection**, and then enter the required interval in **Seconds between keepalives**. For example, if your session disconnects after 10 minutes of inactivity, enter 180 to configure PuTTY to send keepalive data every 3 minutes.
4. In the **Category** pane, expand **Connection**, **SSH**, and **Auth**. Choose **Credentials**.
5. Next to **Private key file for authentication**, choose **Browse**. In the **Select private key file** dialog box, select the `.ppk` file that you generated for your key pair. You can either double-click the file or choose **Open** in the **Select private key file** dialog box.
6. (Optional) If you plan to connect to this instance again after this session, you can save the session information for future use. In the **Category** pane, choose **Session**. Enter a name for the session in **Saved Sessions**, and then choose **Save**.
7. To connect to the instance, choose **Open**.
8. If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host to which you are connecting.

a. (Optional) Verify that the fingerprint in the security alert dialog box matches the fingerprint that you previously obtained in (Optional) Get the instance fingerprint. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.

b. Choose **Accept**. A window opens and you are connected to your instance.

> **Note**
>
> If you specified a passphrase when you converted your private key to the PuTTY format, you must provide that passphrase when you log in to the instance.

If you receive an error while attempting to connect to your instance, see Troubleshoot issues connecting to your Amazon EC2 Linux instance.

Transfer files to a Linux instance using SCP

One way to transfer files between your local computer and a Linux instance is to use the secure copy protocol (SCP). SCP is a good option for simple operations, such as as one-time file copies. SCP secures files transfers using the same .pem file that you use to connect to an instance using SSH. If you need to keep files synchronized, or if the files are large, **rsync** is faster and more efficient than SCP. For security, use **rsync** over SSH, as **rsync** transfers data using plain text by default.

Before you connect to your Linux instance using SCP, complete the following tasks:

- **Complete the general prerequisites.**
  - Check that your instance has passed its status checks. It can take a few minutes for an instance to be ready to accept connection requests. For more information, see View status checks.
  - Get the required instance details.
  - Locate the private key and set permissions.
  - (Optional) Get the instance fingerprint.
- **Allow inbound SSH traffic from your IP address.**

  Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Rules to connect to instances from your computer.
- **Install an SCP client.**

Most Linux, Unix, and Apple computers include an SCP client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools, including an SCP client. For more information, see https://www.openssh.com.

The following procedure steps you through using SCP to transfer a file using the instance's public DNS name, or the IPv6 address if your instance has one.

To use SCP to transfer files between your computer and your instance

1. Determine the location of the source file on your computer and the destination path on the instance. In the following examples, the name of the private key file is `key-pair-name.pem`, the file to transfer is `my-file.txt`, the username for the instance is ec2-user, the public DNS name of the instance is `instance-public-dns-name`, and the IPv6 address of the instance is `2001:db8::1234:5678:1.2.3.4`.

   - (Public DNS) To transfer a file to the destination on the instance, enter the following command from your computer.

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@instance-public-dns-name:path/
     ```

   - (IPv6) To transfer a file to the destination on the instance if the instance has an IPv6 address, enter the following command from your computer. The IPv6 address must be enclosed in square brackets (`[ ]`), which must be escaped (`\`).

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@
     \[2001:db8::1234:5678:1.2.3.4\]:path/
     ```

2. If you haven't already connected to the instance using SSH, you see a response like the following:

   ```
   The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'
   can't be established.
   RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
   Are you sure you want to continue connecting (yes/no)?
   ```

   (Optional) You can optionally verify that the fingerprint in the security alert matches the instance fingerprint. For more information, see (Optional) Get the instance fingerprint.

Enter **yes**.

3. If the transfer is successful, the response is similar to the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
my-file.txt                                100%   480     24.4KB/s   00:00
```

4. To transfer a file in the other direction (from your Amazon EC2 instance to your computer), reverse the order of the host parameters. For example, you can transfer `my-file.txt` from your EC2 instance to the a destination on your local computer as `my-file2.txt`, as shown in the following examples.

   - (Public DNS) To transfer a file to a destination on your computer, enter the following command from your computer.

     ```
     scp -i /path/key-pair-name.pem ec2-user@instance-public-dns-name:path/my-
     file.txt path/my-file2.txt
     ```

   - (IPv6) To transfer a file to a destination on your computer if the instance has an IPv6 address, enter the following command from your computer. The IPv6 address must be enclosed in square brackets (`[ ]`), which must be escaped (`\`).

     ```
     scp -i /path/key-pair-name.pem ec2-user@\[2001:db8::1234:5678:1.2.3.4\]:path/
     my-file.txt path/my-file2.txt
     ```

Manage system users on your Amazon EC2 Linux instance

Each Linux instance launches with a default Linux system user. You can add users to your instance and delete users.

For the default user, the default username is determined by the AMI that was specified when you launched the instance.

> **Note**
>
> By default, password authentication and root login are disabled, and sudo is enabled. To log in to your instance, you must use a key pair. For more information about logging in, see Connect to your Linux instance using SSH.

You can allow password authentication and root login for your instance. For more information, see the documentation for your operating system.

Note

Linux system users should not be confused with IAM users. For more information, see IAM users in the *IAM User Guide*.

Contents

Default usernames

The default username for your EC2 instance is determined by the AMI that was specified when you launched the instance.

The default usernames are:

- For an Amazon Linux AMI, the username is `ec2-user`.
- For a CentOS AMI, the username is `centos` or `ec2-user`.
- For a Debian AMI, the username is `admin`.
- For a Fedora AMI, the username is `fedora` or `ec2-user`.
- For a RHEL AMI, the username is `ec2-user` or `root`.
- For a SUSE AMI, the username is `ec2-user` or `root`.
- For an Ubuntu AMI, the username is `ubuntu`.
- For an Oracle AMI, the username is `ec2-user`.
- For a Bitnami AMI, the username is `bitnami`.

> **Note**
>
> To find the default username for other Linux distributions, check with the AMI provider.

Considerations

Using the default user is adequate for many applications. However, you may choose to add users so that individuals can have their own files and workspaces. Furthermore, creating users for new users is much more secure than granting multiple (possibly inexperienced) users access to the default user, because the default user can cause a lot of damage to a system when used improperly. For more information, see Tips for Securing Your EC2 Instance.

To enable users SSH access to your EC2 instance using a Linux system user, you must share the SSH key with the user. Alternatively, you can use EC2 Instance Connect to provide access to users without the need to share and manage SSH keys. For more information, see Connect to your Linux instance using a public IP address and EC2 Instance Connect.

Create a user

First create the user, and then add the SSH public key that allows the user to connect to and log into the instance.

> **Important**
>
> In Step 1 of this procedure, you create a new key pair. Because a key pair functions like a password, it's crucial to handle it securely. If you create a key pair for a user, you must ensure that the private key is sent to them securely. Alternatively, the user can complete Steps 1 and 2 by creating their own key pair, keeping the private key secure on their machine, and then sending you the public key to complete the procedure from Step 3.

To create a user

1. Create a new key pair. You must provide the `.pem` file to the user for whom you are creating the user. They must use this file to connect to the instance.
2. Retrieve the public key from the key pair that you created in the previous step.

   ```
   $ ssh-keygen -y -f /path_to_key_pair/key-pair-name.pem
   ```

The command returns the public key, as shown in the following example.

```
ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6Vhz2ItxCih
+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
d6RJhJOI0iBXrlsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/
i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco
+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3RbBQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

3. Connect to the instance.

4. Use the **adduser** command to create the user and add it to the system (with an entry in the `/etc/passwd` file). The command also creates a group and a home directory for the user. In this example, the user is named *newuser*.

   - AL2023 and Amazon Linux 2

     With AL2023 and Amazon Linux 2, the user is created with password authentication disabled by default.

     ```
     [ec2-user ~]$ sudo adduser newuser
     ```

   - Ubuntu

     Include the `--disabled-password` parameter to create the user with password authentication disabled.

     ```
     [ubuntu ~]$ sudo adduser newuser --disabled-password
     ```

5. Switch to the new user so that the directory and file that you create will have the proper ownership.

   ```
   [ec2-user ~]$ sudo su - newuser
   ```

   The prompt changes from `ec2-user` to *newuser* to indicate that you have switched the shell session to the new user.

6. Add the SSH public key to the user. First create a directory in the user's home directory for the SSH key file, then create the key file, and finally paste the public key into the key file, as described in the following sub-steps.

a. Create a `.ssh` directory in the *newuser* home directory and change its file permissions to `700` (only the owner can read, write, or open the directory).

```
[newuser ~]$ mkdir .ssh
```

```
[newuser ~]$ chmod 700 .ssh
```

> **Important**
>
> Without these exact file permissions, the user will not be able to log in.

b. Create a file named `authorized_keys` in the `.ssh` directory and change its file permissions to `600` (only the owner can read or write to the file).

```
[newuser ~]$ touch .ssh/authorized_keys
```

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

> **Important**
>
> Without these exact file permissions, the user will not be able to log in.

c. Open the `authorized_keys` file using your favorite text editor (such as **vim** or **nano**).

```
[newuser ~]$ nano .ssh/authorized_keys
```

Paste the public key that you retrieved in **Step 2** into the file and save the changes.

> **Important**
>
> Ensure that you paste the public key in one continuous line. The public key must not be split over multiple lines.

The user should now be able to log into the *newuser* user on your instance, using the private key that corresponds to the public key that you added to the `authorized_keys` file. For more information about the different methods of connecting to a Linux instance, see [Connect to your Linux instance using SSH](#).

Remove a user

If a user is no longer needed, you can remove that user so that it can no longer be used.

Use the **userdel** command to remove the user from the system. When you specify the `-r` parameter, the user's home directory and mail spool are deleted. To keep the user's home directory and mail spool, omit the `-r` parameter.

```
[ec2-user ~]$ sudo userdel -r olduser
```

Connect to your Windows instance using RDP

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.

> **Tip**
>
> If you need to connect to your instance in order to troubleshoot boot, network configuration, and other issues for instances built on the AWS Nitro System, you can use the EC2 Serial Console for instances.

Contents

Connect to your Windows instance using an RDP client

You can connect to your Windows instance using an RDP client as follows.

> **Tip**
>
> Alternatively, you can connect to your Windows instance using Systems Manager Fleet Manager or EC2 Instance Connect Endpoint.

Prerequisites

You must meet the following prerequisites to connect to your Windows instance using an RDP client.

- **Complete the general prerequisites.**
  - Check that your instance has passed its status checks. It can take a few minutes for an instance to be ready to accept connection requests. For more information, see View status checks.
  - Get the required instance details.
  - Locate the private key and set permissions.
  - (Optional) Get the instance fingerprint.
- **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, download the Microsoft Remote Desktop app from the Microsoft Store.
  - (macOS X) Download the Windows App for Mac (previously named Microsoft Remote Desktop from the Mac App Store.
  - (Linux) Use Remmina.
- **Allow inbound RDP traffic from your IP address.**

  Ensure that the security group associated with your instance allows incoming RDP traffic from your IP address. For more information, see Rules to connect to instances from your computer.

Retrieve the administrator password

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

To connect to a Windows instance using RDP, you must retrieve the initial administrator password and then enter this password when you connect to your instance. It takes a few minutes after instance launch before this password is available. Your account must have permission to call the GetPasswordData action. For more information, see Example policies to control access the Amazon EC2 API.

The default username for the Administrator account depends on the language of the operating system (OS) contained in the AMI. To determine the correct username, identify the language of the OS, and then choose the corresponding username. For example, for an English OS, the username is `Administrator`, for a French OS it's `Administrateur`, and for a Portuguese OS it's `Administrador`. If a language version of the OS does not have a username in the same language, choose the username `Administrator (Other)`. For more information, see Localized Names for Administrator Account in Windows in the Microsoft website.

To retrieve the initial administrator password

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 then choose **Connect**.
4. On the **Connect to instance** page, choose the **RDP client** tab.
5. For **Username**, choose the default username for the Administrator account. The username you choose must match the language of the operating system (OS) contained in the AMI that you used to launch your instance. If there is no username in the same language as your OS, choose **Administrator (Other)**.
6. Choose **Get password**.
7. On the **Get Windows password** page, do the following:

    a. Choose **Upload private key file** and navigate to the private key (`.pem`) file that you specified when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file to this window.

    b. Choose **Decrypt password**. The **Get Windows password** page closes, and the default administrator password for the instance appears under **Password**, replacing the **Get password** link shown previously.

c. Copy the password and save it in a safe place. This password is required to connect to the instance.

Connect to your Windows instance

The following procedure uses the Remote Desktop Connection client for Windows (MSTSC). If you're using a different RDP client, download the RDP file and then see the documentation for the RDP client for the steps to establish the RDP connection.

To connect to a Windows instance using an RDP client

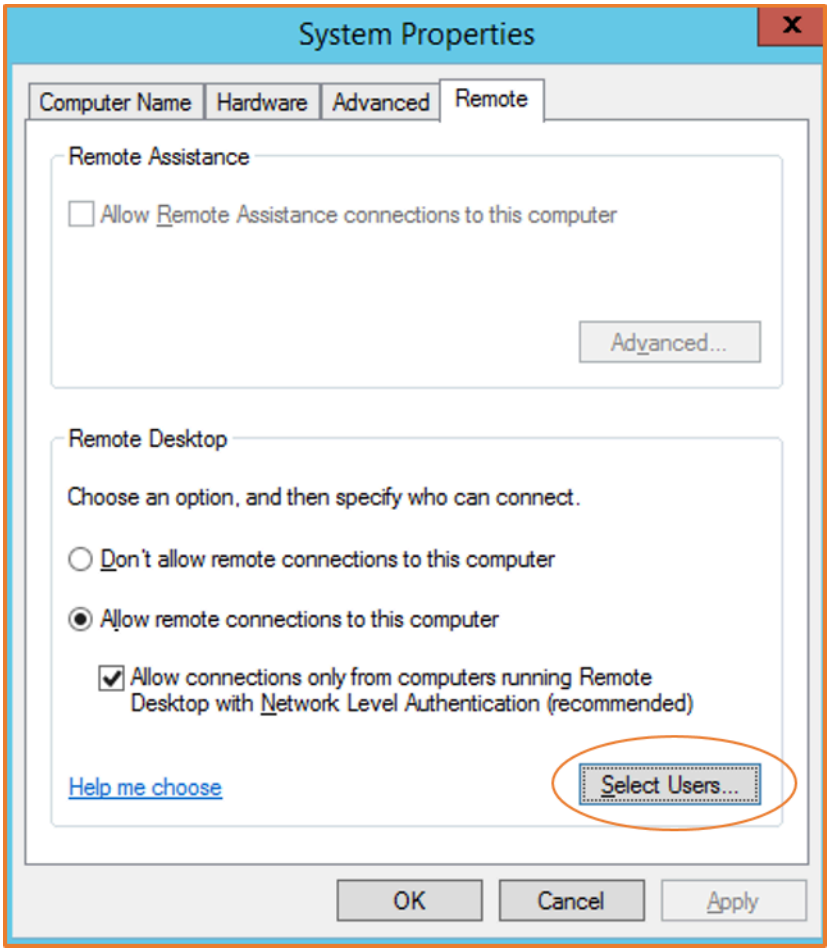
1. On the **Connect to instance** page, choose **Download remote desktop file**. When the file download is finished, choose **Cancel** to return to the **Instances** page. The RDP file is downloaded to your `Downloads` folder.
2. Run `mstsc.exe` to open the RDP client.
3. Expand **Show options**, choose **Open**, and select the .rdp file from your `Downloads` folder.
4. By default, **Computer** is the public IPv4 DNS name of the instance and **User name** is the administrator account. To connect to the instance using IPv6 instead, replace the public IPv4 DNS name of the instance with its IPv6 address. Review the default settings and change them as needed.
5. Choose **Connect**. If you receive a warning that the publisher of the remote connection is unknown, choose **Connect** to continue.
6. Enter the password that you saved previously, and then choose **OK**.
7. Due to the nature of self-signed certificates, you might get a warning that the security certificate could not be authenticated. Do one of the following:
   - If you trust the certificate, choose **Yes** to connect to your instance.
   - [Windows] Before you proceed, compare the thumbprint of the certificate with the value in the system log to confirm the identity of the remote computer. Choose **View certificate** and then choose **Thumbprint** from the **Details** tab. Compare this value to the value of `RDPCERTIFICATE-THUMBPRINT` in **Actions**, **Monitor and troubleshoot**, **Get system log**.
   - [Mac OS X] Before you proceed, compare the fingerprint of the certificate with the value in the system log to confirm the identity of the remote computer. Choose **Show Certificate**, expand **Details**, and choose **SHA1 Fingerprints**. Compare this value to the value of `RDPCERTIFICATE-THUMBPRINT` in **Actions**, **Monitor and troubleshoot**, **Get system log**.

8. If the RDP connection is successful, the RDP client displays the Windows login screen and then the Windows desktop. If you receive an error message instead, see the section called “Remote Desktop can't connect to the remote computer”. When you are finished with the RDP connection, you can close the RDP client.

Configure user accounts

After you connect to your instance over RDP, we recommend that you perform the following tasks:

- 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 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 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.



Connect to your Windows instance using Fleet Manager

You can use Fleet Manager, a capability of AWS Systems Manager, to connect to Windows instances using the Remote Desktop Protocol (RDP) and display up to four Windows instances on the same page in the AWS Management Console. You can connect to the first instance in the Fleet Manager Remote Desktop directly from the **Instances** page in the Amazon EC2 console. For more information about Fleet Manager, see Connect to a managed instance using Remote Desktop in the *AWS Systems Manager User Guide*.

You do not need to specifically allow incoming RDP traffic from your IP address if you use Fleet Manager to connect. Fleet Manager handles that for you.

Prerequisites

Before attempting to connect to an instance using Fleet Manager, you must set up your environment. For more information, see Setting up your environment in the *AWS Systems Manager User Guide*.

To connect to a Windows instance using Fleet Manager

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation pane, choose **Instances**.
3. Select the instance and then choose **Connect**.
4. On the **RDP client** tab, for **Connection Type**, choose **Connect using Fleet Manager**.
5. Choose **Fleet Manager Remote Desktop**. This opens the **Fleet Manager Remote Desktop** page in the AWS Systems Manager console.
6. Enter your credentials and then choose **Connect**.
7. If the RDP connection is successful, Fleet Manager displays the Windows desktop. When you are finished with the session, choose **Actions**, **End session**.

For more information, see Connecting to a Windows Server managed instance using Remote Desktop in the *AWS Systems Manager User Guide*.

Transfer files to a Windows instance using RDP

You can work with your Windows instance in 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 (RDP) software. You can access local files on hard disk drives, DVD drives, portable media drives, and mapped network drives.

To access your local files from your Windows instances, you must enable the local file sharing feature by mapping the remote session drive to your local drive. The steps are slightly different depending on whether your local computer operating system is Windows or macOS X.

For more information about the prerequisites to connect using RDP, see Prerequisites.

Windows

To map the remote session drive to your local drive on your local Windows computer

1. Open the Remote Desktop Connection client.
2. Choose **Show Options**.
3. Add the instance host name to the **Computer** field and username to the **User name** field, as follows:

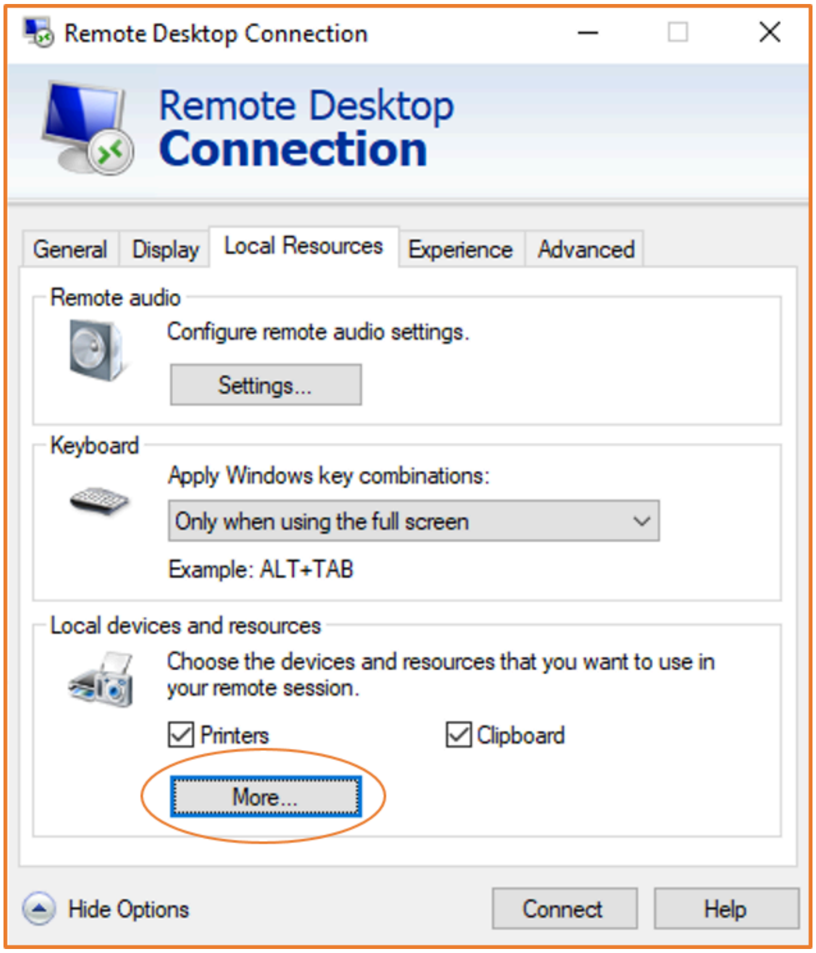
a. Under **Connection settings**, choose **Open...**, and browse to the RDP shortcut file that you downloaded from the Amazon EC2 console. The file contains the Public IPv4 DNS host name, which identifies the instance, and the Administrator user name.

b. Select the file and choose **Open**. The **Computer** and **User name** fields are populated with the values from the RDP shortcut file.

c. Choose **Save**.

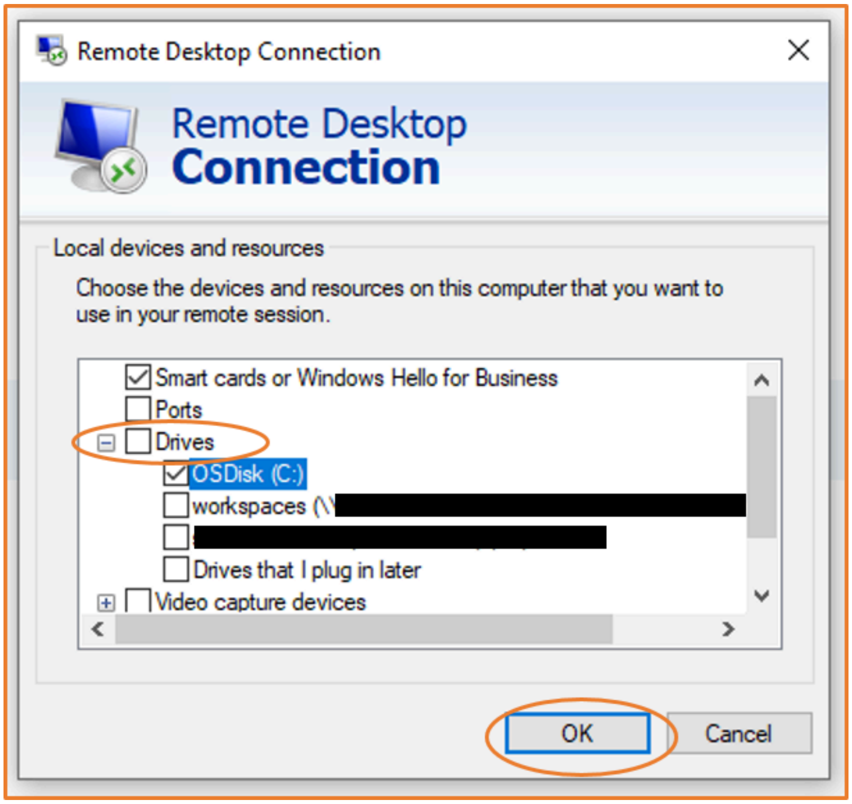
4. Choose the **Local Resources** tab.

5. Under **Local devices and resources**, choose **More...**



6. Open **Drives** and select the local drive to map to your Windows instance.

7. Choose **OK**.

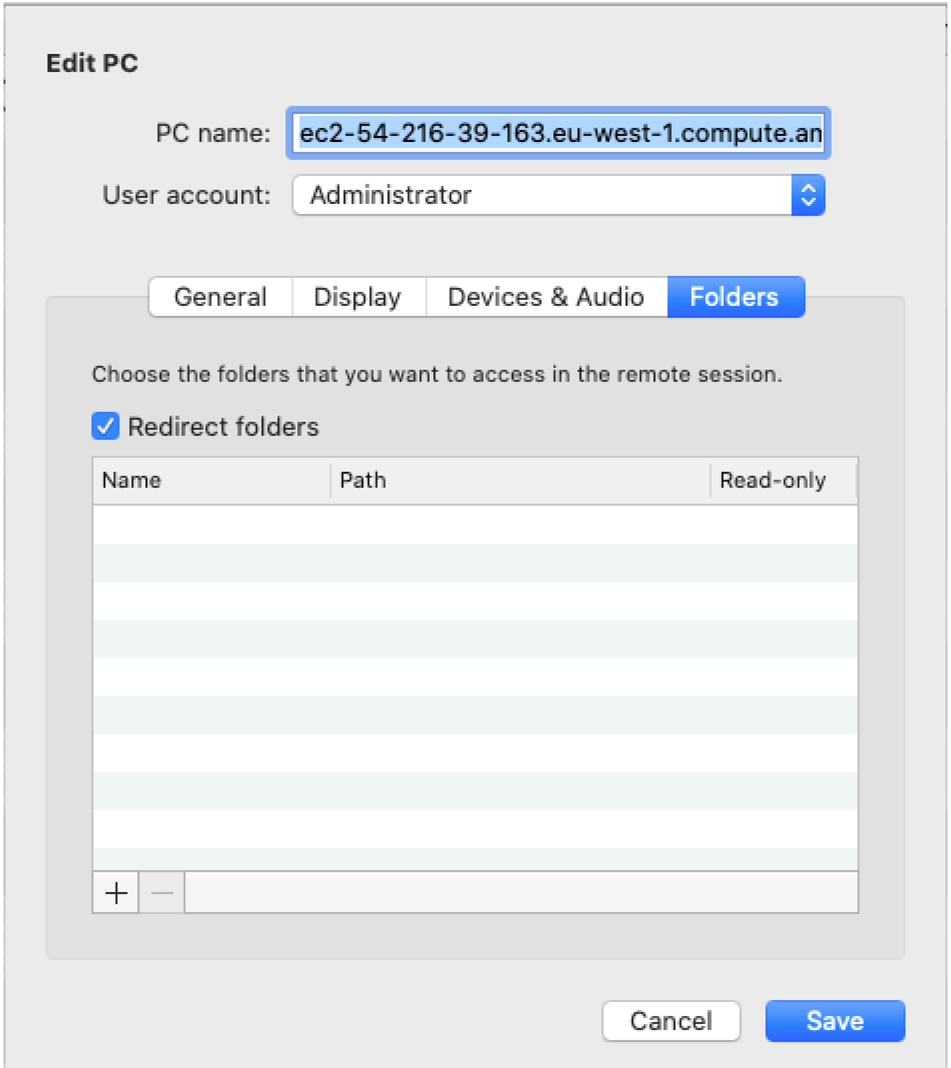


8. Choose **Connect** to connect to your Windows instance.

macOS X

To map the remote session drive to your local folder on your local macOS X computer

1. Open the Remote Desktop Connection client.
2. Browse to the RDP file that you downloaded from the Amazon EC2 console (when you initially connected to the instance), and drag it onto the Remote Desktop Connection client.
3. Right-click the RDP file, and choose **Edit**.
4. Choose the **Folders** tab, and select the **Redirect folders** checkbox.

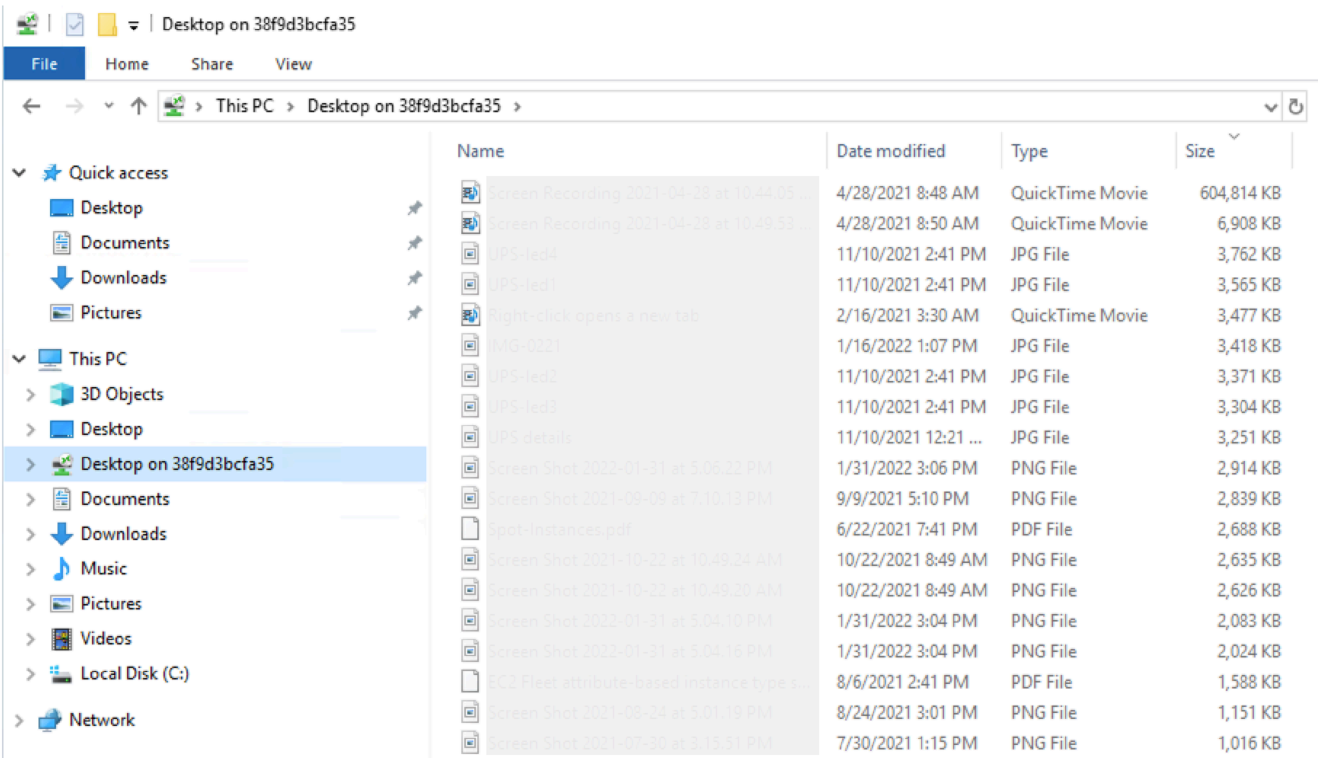


5. Choose the **+** icon at bottom left, browse to the folder to map, and choose **Open**. Repeat this step for every folder to map.

6. Choose **Save**.

7. Choose **Connect** to connect to your Windows instance. You'll be prompted for the password.

8. On the instance, in File Explorer, expand **This PC**, and find the shared folder from which you can access your local files. In the following screenshot, the **Desktop** folder on the local computer was mapped to the remote session drive on the instance.



For more information on making local devices available to a remote session on a Mac computer, see Get started with the macOS client.

Connect to your Amazon EC2 instance using Session Manager

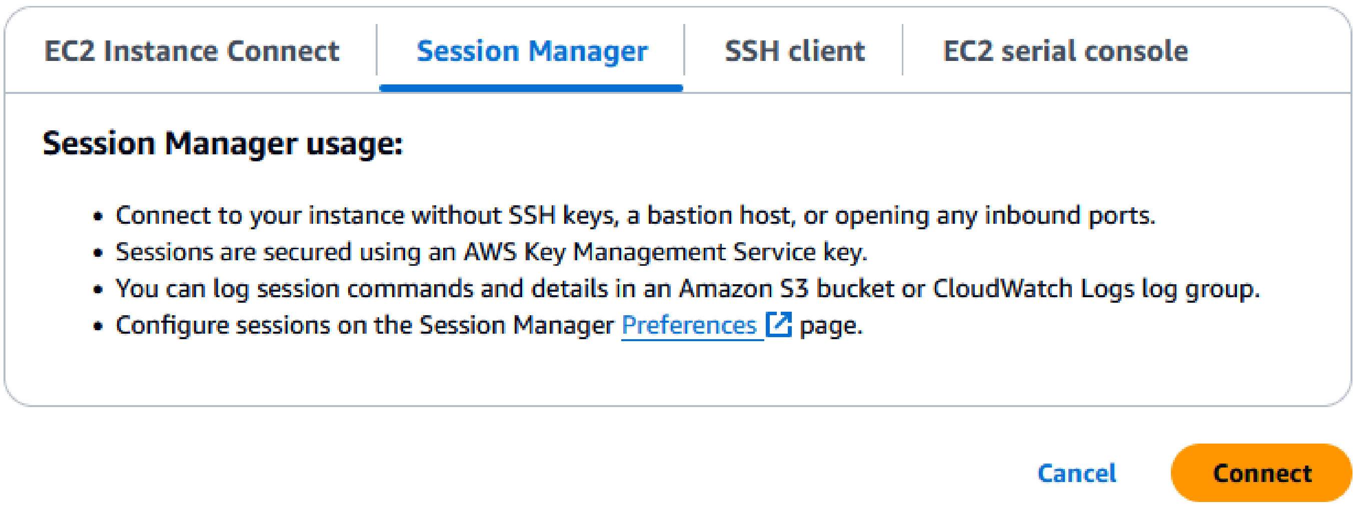
Session Manager is a fully-managed AWS Systems Manager capability for managing your Amazon EC2 instances through an interactive, one-click, browser-based shell, or through the AWS CLI. You can use Session Manager to start a session with an instance in your account. After the session is started, you can run interactive commands on the instance 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*.

Prerequisites

Before you attempt to connect to an instance using Session Manager, you must complete the required setup steps. For example, the instance must be managed by SSM and must have an attached IAM role with the **AmazonSSMManagedInstanceCore** policy. For more information, see Setting up Session Manager.

To connect to an Amazon EC2 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 the connection method, choose **Session Manager**.
5. Choose **Connect** to start the session.



Troubleshooting

If you receive an error that you're not authorized to perform one or more Systems Manager actions (`ssm:`*`command-name`*), 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*.

Connect to your Linux instance using a public IP address and EC2 Instance Connect

Amazon EC2 Instance Connect provides a secure way to connect to your Linux instances over Secure Shell (SSH). With EC2 Instance Connect, you use AWS Identity and Access Management (IAM) [policies]() and [principals]() to control SSH access to your instances, removing the need to share and manage SSH keys. All connection requests using EC2 Instance Connect are [logged to AWS CloudTrail]() so that you can audit connection requests.

You can use EC2 Instance Connect to connect to your instances using the Amazon EC2 console or the SSH client of your choice.

When you connect to an instance using EC2 Instance Connect, the EC2 Instance Connect API pushes an SSH public key to the instance metadata where it remains for 60 seconds. An IAM policy attached to your user authorizes your user to push the public key to the instance metadata. The SSH daemon uses `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, which are configured when EC2 Instance Connect is installed, to look up the public key from the instance metadata for authentication, and connects you to the instance.

> **Tip**
>
> EC2 Instance Connect is one of the options to connect to your Linux instance. For other options, see Connect to your Linux instance using SSH. To connect to a Windows instance, see Connect to your Windows instance using RDP.

Pricing

EC2 Instance Connect is available at no additional cost.

Region availability

EC2 Instance Connect is available in all AWS Regions, except Asia Pacific (Taipei). It is not supported in Local Zones.

Contents

For a blog post that discusses how to improve the security of your bastion hosts using EC2 Instance Connect, see Securing your bastion hosts with Amazon EC2 Instance Connect.

Tutorial: Complete the configuration required to connect to your instance using EC2 Instance Connect

To connect to your instance using EC2 Instance Connect in the Amazon EC2 console, you first need to complete the prerequisite configuration that will allow you to successfully connect to your instance. The purpose of this tutorial is to guide you through the tasks to complete the prerequisite configuration.

Tutorial overview

In this tutorial, you'll complete the following four tasks:

- Task 1: Grant permissions required to use EC2 Instance Connect

  First you'll create an IAM policy that contains the IAM permissions that allow you to push a public key to the instance metadata. You'll attach this policy to your IAM identity (user, user group, or role) so that your IAM identity gets these permissions.
- Task 2: Allow inbound traffic from the EC2 Instance Connect service to your instance

  Then you'll create a security group that allows traffic from the EC2 Instance Connect service to your instance. This is required when you use EC2 Instance Connect in the Amazon EC2 console to connect to your instance.
- Task 3: Launch your instance

  You'll then launch an EC2 instance using an AMI that is pre-installed with EC2 Instance Connect and you'll add the security group that you created in the previous step.
- Task 4: Connect to your instance

  Finally, you'll use EC2 Instance Connect in the Amazon EC2 console to connect to your instance. If you can connect, then you can be sure that the prerequisite configuration you completed in Tasks 1, 2, and 3 were successful.

Task 1: Grant permissions required to use EC2 Instance Connect

When you connect to an instance using EC2 Instance Connect, the EC2 Instance Connect API pushes an SSH public key to the instance metadata where it remains for 60 seconds. You need an IAM policy attached to your IAM identity (user, user group, or role) to grant you the required permission to push the public key to the instance metadata.

Task objective

You'll create the IAM policy that grants the permission to push the public key to the instance. The specific action to allow is `ec2-instance-connect:SendSSHPublicKey`. You must also allow the `ec2:DescribeInstances` action so that you can view and select your instance in the Amazon EC2 console.

After you've created the policy, you'll attach the policy to your IAM identity (user, user group, or role) so that your IAM identity gets the permissions.

You'll create a policy that is configured as follows:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

> **Important**
>
> The IAM policy created in this tutorial is a highly permissive policy; it allows you to connect to any instance using any AMI username. We're using this highly permissive policy to keep the tutorial simple and focused on the specific configurations that this tutorial is teaching. However, in a production environment, we recommend that your IAM policy is configured to provide least-privilege permissions. For example IAM policies, see Grant IAM permissions for EC2 Instance Connect.

To create and attach an IAM policy that allows you to use EC2 Instance Connect to connect to your instances

1. **First create the IAM policy**

   a. Open the IAM console at https://console.aws.amazon.com/iam/.

   b. In the navigation pane, choose **Policies**.

   c. Choose **Create policy**.

   d. On the **Specify permission** page, do the following:

      i. For **Service**, choose **EC2 Instance Connect**.

      ii. Under **Actions allowed**, in the search field start typing **`send`** to show the relevant actions, and then select **SendSSHPublicKey**.

      iii. Under **Resources**, choose **All**. For a production environment, we recommend specifying the instance by its ARN, but for this tutorial, you're allowing all instances.

      iv. Choose **Add more permissions**.

      v. For **Service**, choose **EC2**.

      vi. Under **Actions allowed**, in the search field start typing **`describein`** to show the relevant actions, and then select **DescribeInstances**.

      vii. Choose **Next**.

   e. On the **Review and create** page, do the following:

      i. For **Policy name**, enter a name for the policy.

      ii. Choose **Create policy**.

2. **Then attach the policy to your identity**

   a. In the IAM console, in the navigation pane, choose **Policies**.

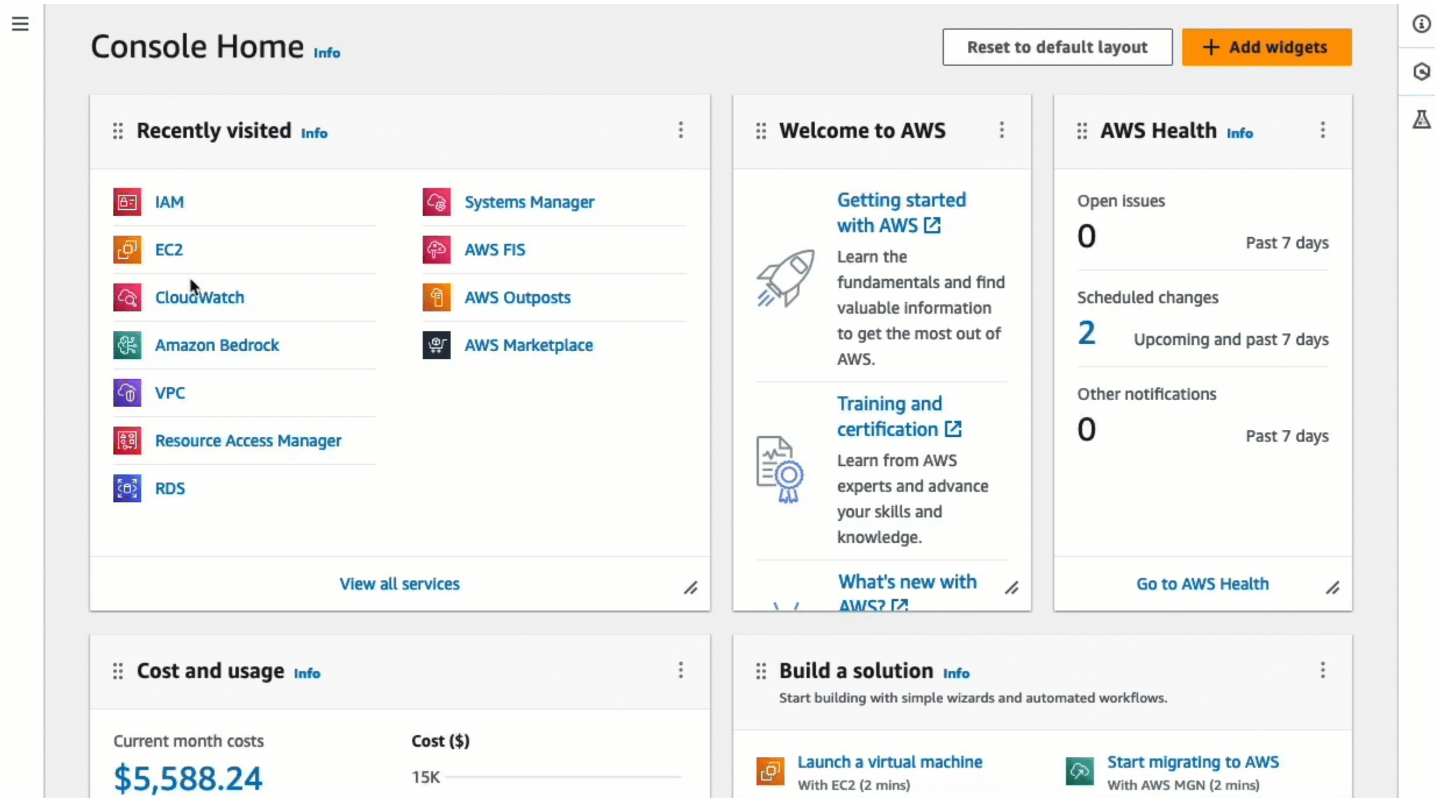
   b. In the list of policies, select the option button next to the name of the policy you created. You can use the search box to filter the list of policies.

   c. Choose **Actions**, **Attach**.

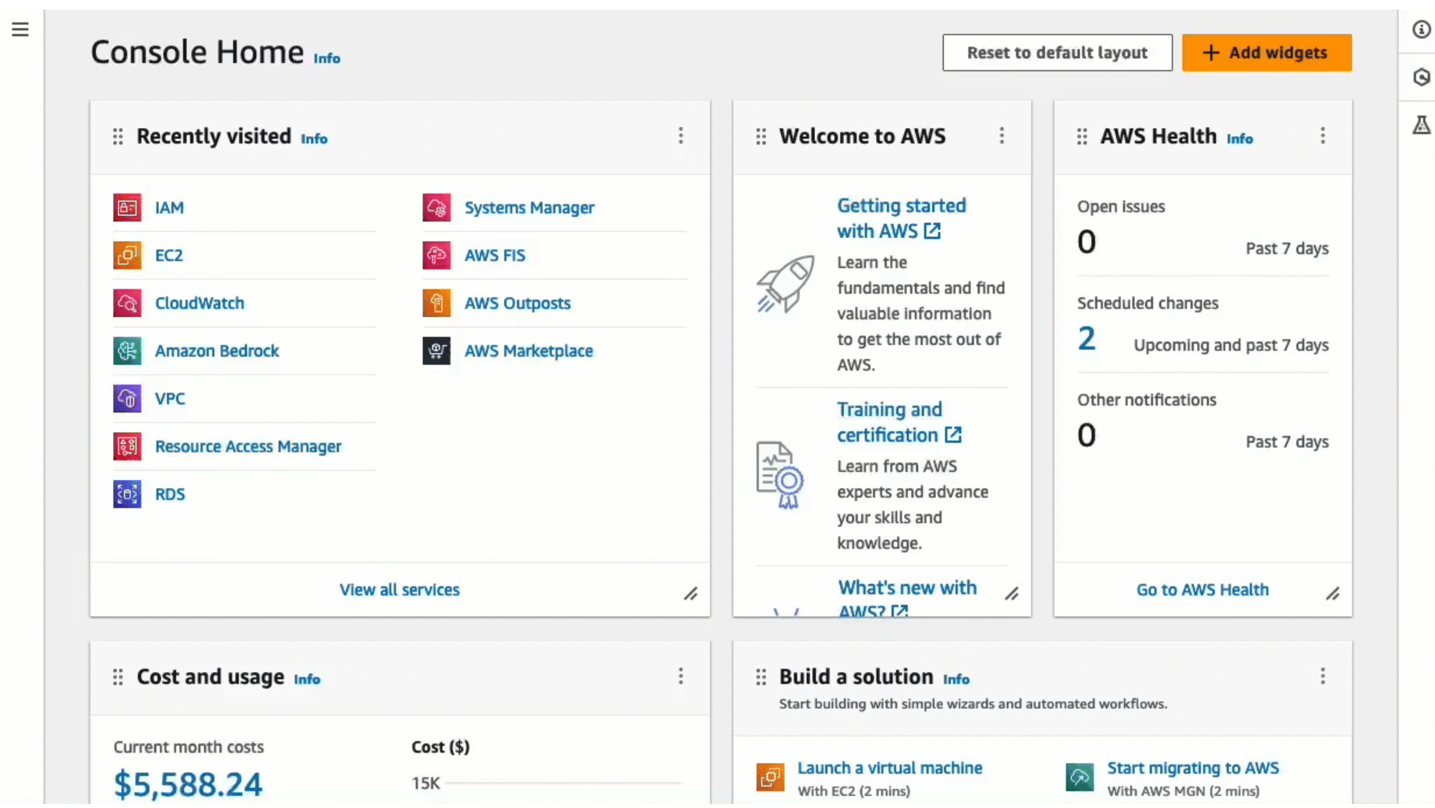
   d. Under **IAM entities**, select the checkbox next to your identity (user, user group, or role). You can use the search box to filter the list of entities.

   e. Choose **Attach policy**.

View an animation: Create an IAM policy



View an animation: Attach an IAM policy



Task 2: Allow inbound traffic from the EC2 Instance Connect service to your instance

When you use EC2 Instance Connect in the Amazon EC2 console to connect to an instance, the traffic that must be allowed to reach the instance is traffic from the EC2 Instance Connect service. This is different to connecting from your local computer to an instance; in that case, you must allow traffic from your local computer to your instance. To allow traffic from the EC2 Instance Connect service, you must create a security group that allows inbound SSH traffic from the IP address range for the EC2 Instance Connect service.

AWS uses prefix lists to manage IP address ranges. The names of the EC2 Instance Connect prefix lists are as follows, with *region* replaced by the Region code:

- IPv4 prefix list name: `com.amazonaws.region.ec2-instance-connect`
- IPv6 prefix list name: `com.amazonaws.region.ipv6.ec2-instance-connect`

Task objective

You'll create a security group that allows inbound SSH traffic on port 22 from the IPv4 prefix list in the Region in which your instance is located.

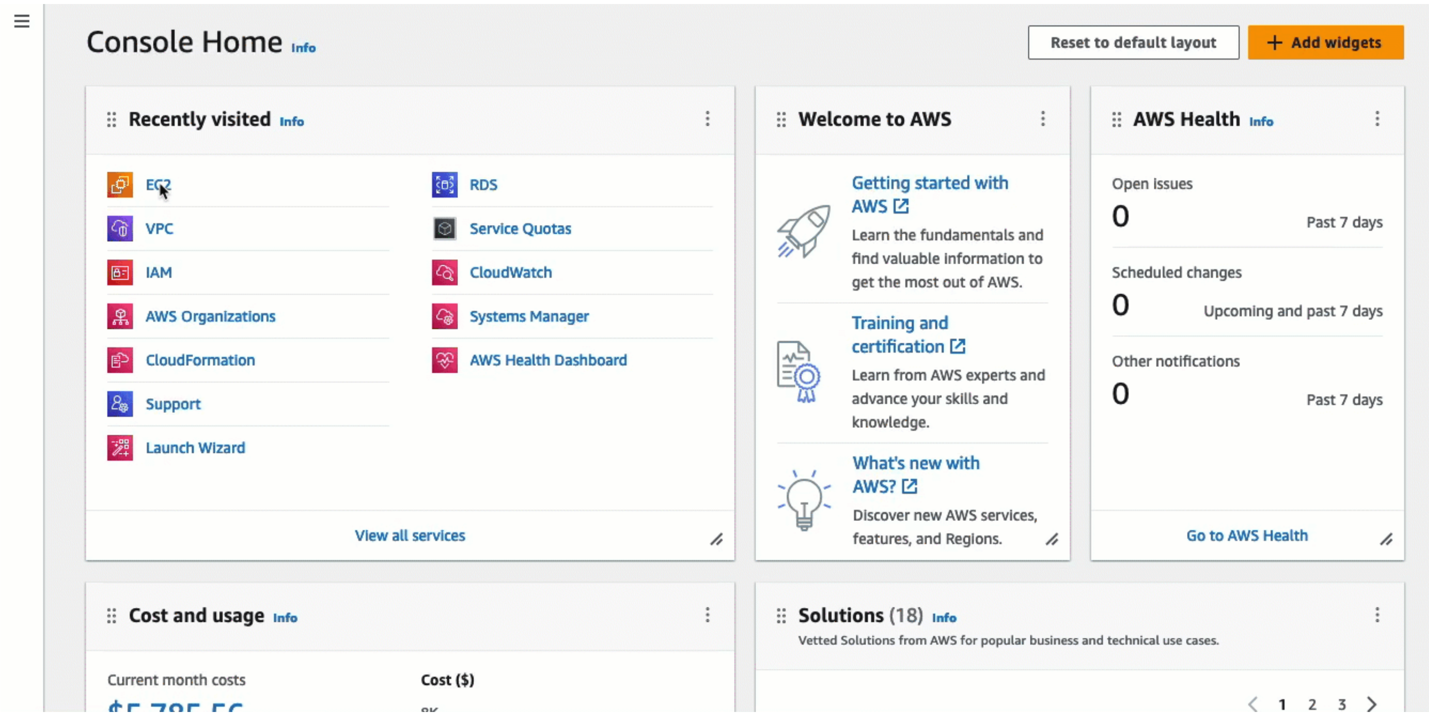
To create a security group that allows inbound traffic from the EC2 Instance Connect service to your instance

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. Under **Basic details**, do the following:
   a. For **Security group name**, enter a meaningful name for your security group.
   b. For **Description**, enter a meaningful description for your security group.
5. Under **Inbound rules**, do the following:
   a. Choose **Add rule**.
   b. For **Type**, choose **SSH**.
   c. For **Source**, leave **Custom**.
   d. In the field next to **Source**, select the prefix list for EC2 Instance Connect.

For example, if your instance is located in the US East (N. Virginia) (`us-east-1`) Region and your users will connect to its public IPv4 address, choose the following prefix list: **com.amazonaws.us-east-1.ec2-instance-connect**

6. Choose **Create security group**.

View an animation: Create the security group



Task 3: Launch your instance

When you launch an instance, you must specify an AMI that contains the information required to launch the instance. You can choose to launch an instance with or without EC2 Instance Connect pre-installed. In this task, we specify an AMI that comes pre-installed with EC2 Instance Connect.

If you launch your instance without EC2 Instance Connect pre-installed, and you want to use EC2 Instance Connect to connect to your instance, you'll need to perform additional configuration steps. These steps are outside the scope of this tutorial.

Task objective

You'll launch an instance with the Amazon Linux 2023 AMI, which comes pre-installed with EC2 Instance Connect. You'll also specify the security group that you created earlier so that you can use EC2 Instance Connect in the Amazon EC2 console to connect to your instance. Because you'll use EC2 Instance Connect to connect to your instance, which pushes a public key to your instance's metadata, you won't need to specify an SSH key when you launch your instance.

To launch an instance that can use EC2 Instance Connect in the Amazon EC2 console for connection

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 AWS Region is displayed (for example, **Ireland**). Select a Region in which to launch your instance. This choice is important because you created a security group that allows traffic for a specific Region, so you must select the same Region in which to launch your instance.
3. From the Amazon EC2 console dashboard, choose **Launch instance**.
4. (Optional) Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
5. Under **Application and OS Images (Amazon Machine Image)**, choose **Quick Start**. **Amazon Linux** is selected by default. Under **Amazon Machine Image (AMI)**, **Amazon Linux 2023 AMI** is selected by default. Keep the default selection for this task.
6. Under **Instance type**, for **Instance type**, keep the default selection, or choose a different instance type.
7. Under **Key pair (login)**, for **Key pair name**, choose **Proceed without a key pair (Not recommended)**. When you use EC2 Instance Connect to connect to an instance, EC2 Instance Connect pushes a key pair to the instance's metadata, and it is this key pair that is used for the connection.
8. Under **Network settings**, do the following:

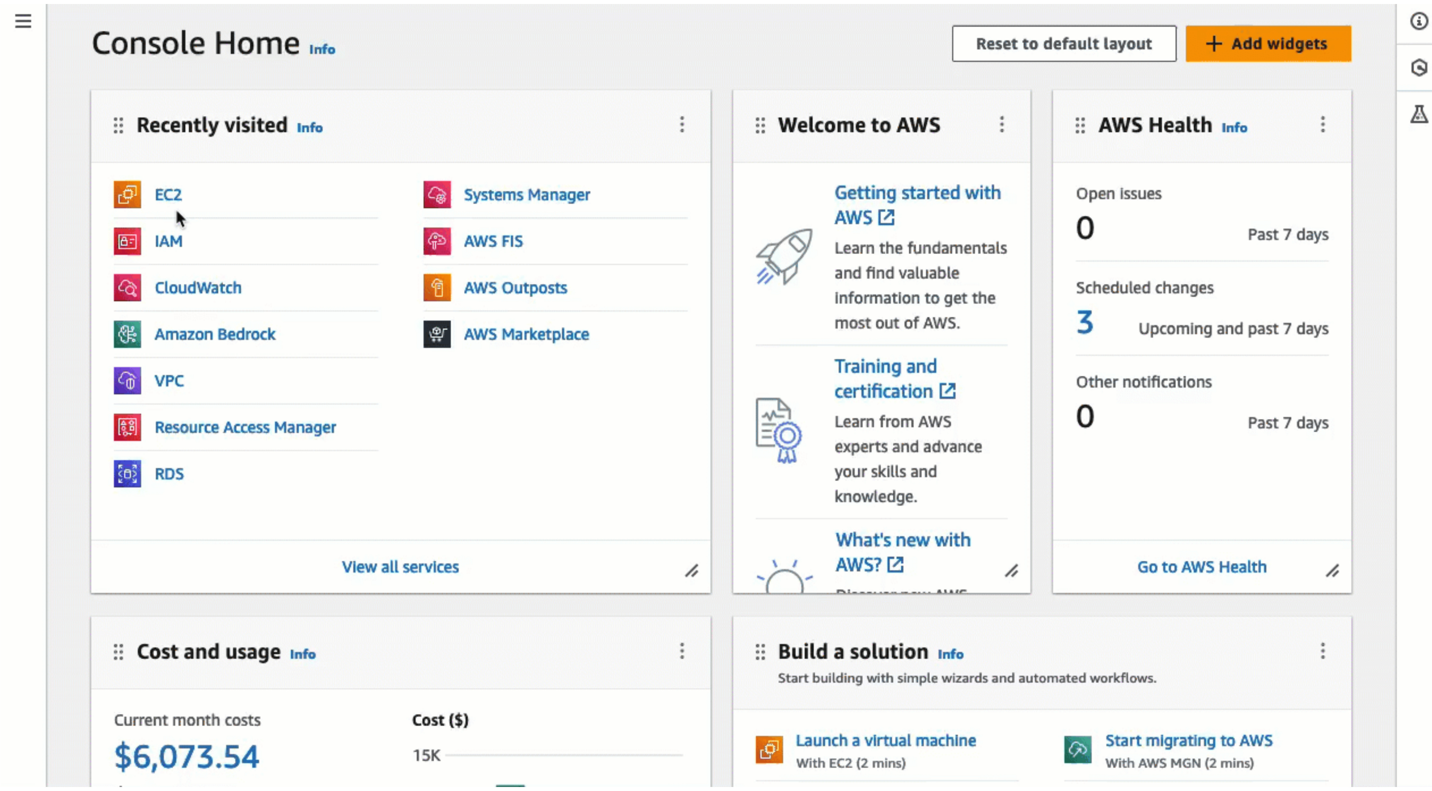
   a. For **Auto-assign public IP**, leave **Enable**.

      > **Note**
      >
      > To use EC2 Instance Connect in the Amazon EC2 console to connect to an instance, the instance must have a public IPv4 or public IPv6 address.

   b. For **Firewall (security groups)**, choose **Select existing security group**.

   c. Under **Common security groups**, choose the security group that you created earlier.
9. In the **Summary** panel, choose **Launch instance**.

View an animation: Launch your instance



Task 4: Connect to your instance

When you connect to an instance using EC2 Instance Connect, the EC2 Instance Connect API pushes an SSH public key to the [instance metadata](#) where it remains for 60 seconds. The SSH daemon uses `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` to look up the public key from the instance metadata for authentication, and connects you to the instance.

Task objective

In this task, you'll connect to your instance using EC2 Instance Connect in the Amazon EC2 console. If you completed the prerequisite Tasks 1, 2, and 3, the connection should be successful.

Steps to connect to your instance

Use the following steps to connect to your instance. To view an animation of the steps, see [View an animation: Connect to your instance](#).

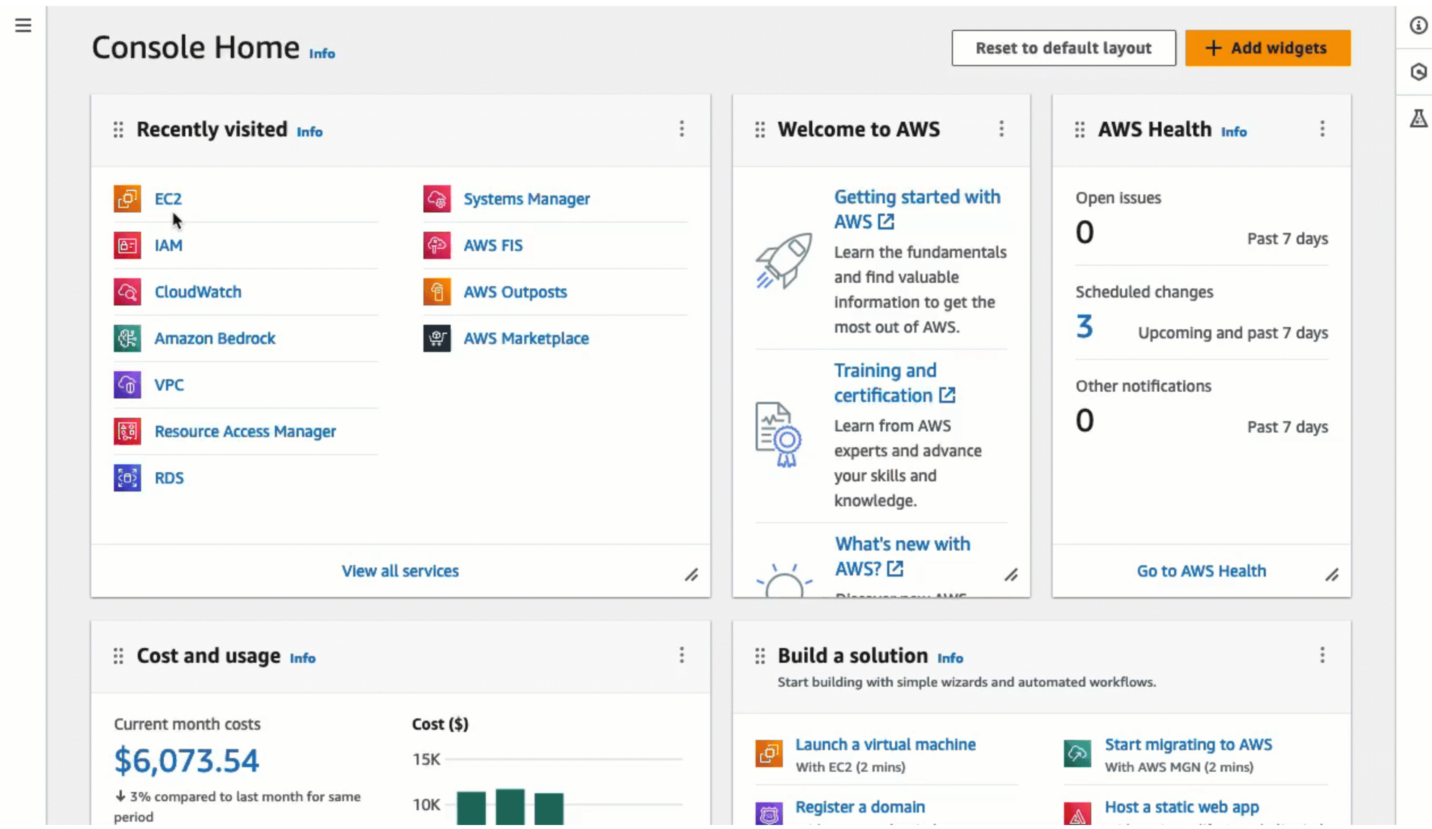
To connect an instance using EC2 Instance Connect in the Amazon EC2 console

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

2. In the navigation bar at the top of the screen, the current AWS Region is displayed (for example, **Ireland**). Select the Region in which your instance is located.
3. In the navigation pane, choose **Instances**.
4. Select your instance and choose **Connect**.
5. Choose the **EC2 Instance Connect** tab.
6. Choose **Connect using a Public IP**.
7. Choose **Connect**.

   A terminal window opens in the browser, and you are connected to your instance.

View an animation: Connect to your instance



Prerequisites for EC2 Instance Connect

The following are the prerequisites for installing and using EC2 Instance Connect:

- Install EC2 Instance Connect
- Ensure network connectivity
- Allow inbound SSH traffic

- Grant permissions
- Install an SSH client on your local computer
- Meet username requirements

Install EC2 Instance Connect

To use EC2 Instance Connect to connect to an instance, the instance must have EC2 Instance Connect installed. You can either launch the instance using an AMI that comes pre-installed with EC2 Instance Connect, or you can install EC2 Instance Connect on instances that are launched with supported AMIs. For more information, see Install EC2 Instance Connect on your EC2 instances.

Ensure network connectivity

Instances can be configured to allow users to connect to your instance over the internet or through the instance's private IP address. Depending on how your users will connect to your instance using EC2 Instance Connect, you must configure the following network access:

- If your users will connect to your instance over the internet, then your instance must have a public IPv4 or public IPv6 address and be in a public subnet with a route to the internet. If you haven't modified your default public subnet, then it contains a route to the internet for IPv4 only, and not for IPv6. For more information, see Enable VPC internet access using internet gateways in the *Amazon VPC User Guide*.
- If your users will connect to your instance through the instance's private IPv4 address, then you must establish private network connectivity to your VPC, such as by using AWS Direct Connect, AWS Site-to-Site VPN, or VPC peering, so that your users can reach the instance's private IP address.

If your instance does not have a public IPv4 or public IPv6 address and you prefer not to configure the network access as described above, you can consider EC2 Instance Connect Endpoint as an alternative to EC2 Instance Connect. With EC2 Instance Connect Endpoint, you can connect to an instance using SSH or RDP even if the instance does not have a public IPv4 or public IPv6 address. For more information, see Connect to your Linux instance using the Amazon EC2 console.

Allow inbound SSH traffic

When using the Amazon EC2 console to connect to an instance

When users connect to an instance using the Amazon EC2 console, the traffic that must be allowed to reach the instance is traffic from the EC2 Instance Connect service. The service is identified by specific IP address ranges, which AWS manages through prefix lists. You must create a security group that allows inbound SSH traffic from the EC2 Instance Connect service. To configure this, for the inbound rule, in the field next to **Source**, select the EC2 Instance Connect prefix list.

AWS provides different managed prefix lists for IPv4 and IPv6 addresses for each Region. The names of the EC2 Instance Connect prefix lists are as follows, with *region* replaced by the Region code:

- IPv4 prefix list name: `com.amazonaws.region.ec2-instance-connect`
- IPv6 prefix list name: `com.amazonaws.region.ipv6.ec2-instance-connect`

For the instructions for creating the security group, see Task 2: Allow inbound traffic from the EC2 Instance Connect service to your instance. For more information, see Available AWS-managed prefix lists in the *Amazon VPC User Guide*.

When using the CLI or SSH to connect to an instance

Ensure that the security group associated with your instance allows inbound SSH traffic on port 22 from your IP address or from your network. The default security group for the VPC does not allow incoming SSH traffic by default. The security group created by the launch instance wizard allows incoming SSH traffic by default. For more information, see Rules to connect to instances from your computer.

Grant permissions

You must grant the required permissions to every IAM user who will use EC2 Instance Connect to connect to an instance. For more information, see Grant IAM permissions for EC2 Instance Connect.

Install an SSH client on your local computer

If your users will connect using SSH, they must ensure that their local computer has an SSH client.

A user's local computer most likely has an SSH client installed by default. They can check for an SSH client by typing **ssh** at the command line. If their local computer doesn't recognize the command, they can install an SSH client. For information about installing an SSH client on Linux or macOS X, see http://www.openssh.com. For information about installing an SSH client on Windows 10, see OpenSSH in Windows.

There is no need to install an SSH client on a local computer if your users use only the Amazon EC2 console to connect to an instance.

Meet username requirements

When using EC2 Instance Connect to connect to an instance, the username must meet the following requirements:

- First character: Must be a letter (`A-Z`, `a-z`), a digit (`0-9`), or an underscore (`_`)
- Subsequent characters: Can be letters (`A-Z`, `a-z`), digits (`0-9`), or the following characters: `@ . _ -`
- Minimum length: 1 character
- Maximum length: 31 characters

Grant IAM permissions for EC2 Instance Connect

To connect to an instance using EC2 Instance Connect, you must create an IAM policy that grants your users permissions for the following actions and condition:

- `ec2-instance-connect:SendSSHPublicKey` action – Grants permission to push the public key to an instance.
- `ec2:osuser` condition – Specifies the name of the OS user that can push the public key to an instance. Use the default username for the AMI that you used to launch the instance. The default username for AL2023 and Amazon Linux 2 is `ec2-user`, and for Ubuntu it's `ubuntu`.
- `ec2:DescribeInstances` action – Required when using the EC2 console because the wrapper calls this action. Users might already have permission to call this action from another policy.
- `ec2:DescribeVpcs` action – Required when connecting to an IPv6 address.

Consider restricting access to specific EC2 instances. Otherwise, all IAM principals with permission for the `ec2-instance-connect:SendSSHPublicKey` action can connect to all EC2 instances. You can restrict access by specifying resource ARNs or by using resource tags as condition keys.

For more information, see Actions, resources, and condition keys for Amazon EC2 Instance Connect.

For information about creating IAM policies, see Creating IAM policies in the *IAM User Guide*.

Allow users to connect to specific instances

The following IAM policy grants permission to connect to specific instances, identified by their resource ARNs.

In the following example IAM policy, the following actions and condition are specified:

- The `ec2-instance-connect:SendSSHPublicKey` action grants users permission to connect to two instances, specified by the resource ARNs. To grant users permission to connect to *all* EC2 instances, replace the resource ARNs with the `*` wildcard.
- The `ec2:osuser` condition grants permission to connect to the instances only if the *`ami-username`* is specified when connecting.
- The `ec2:DescribeInstances` action is specified to grant permission to users who will use the console to connect to your instances. If your users will only use an SSH client to connect to your instances, you can omit `ec2:DescribeInstances`. Note that the `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.
- The `ec2:DescribeVpcs` action is specified to grant permission to users who will use the console to connect to your instances using an IPv6 address. If your users will only use a public IPv4 address, you can omit `ec2:DescribeVpcs`. Note that the `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:111122223333:instance/i-0598c7d356eba48d7"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

Allow users to connect to instances with specific tags

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on tags that can be attached to users and AWS resources. You can use resource tags to control access to an instance. For more information about using tags to control access to your AWS resources, see Controlling access to AWS resources in the *IAM User Guide*.

In the following example IAM policy, the `ec2-instance-connect:SendSSHPublicKey` action grants users permission to connect to any instance (indicated by the `*` wildcard in the resource ARN) on condition that the instance has a resource tag with key=`tag-key` and value=`tag-value`.

The `ec2:DescribeInstances` action is specified to grant permission to users who will use the console to connect to your instances. If your users will use only an SSH client to connect to your instances, you can omit `ec2:DescribeInstances`. Note that the `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.

The `ec2:DescribeVpcs` action is specified to grant permission to users who will use the console to connect to your instances using an IPv6 address. If your users will only use a public IPv4 address, you can omit `ec2:DescribeVpcs`. Note that the `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/tag-key": "tag-value"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

Install EC2 Instance Connect on your EC2 instances

To connect to a Linux instance using EC2 Instance Connect, the instance must have EC2 Instance Connect installed. Installing EC2 Instance Connect configures the SSH daemon on the instance.

For more information about the EC2 Instance Connect package, see aws/aws-ec2-instance-connect-config on the GitHub website.

> **Note**
>
> If you configured the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` settings for SSH authentication, the EC2 Instance Connect installation will not update them. As a result, you can't use EC2 Instance Connect.

Install prerequisites

Before you install EC2 Instance Connect, ensure that you meet the following prerequisites.

- **Verify that the instance uses one of the following:**

- Amazon Linux 2 prior to version 2.0.20190618 *
- AL2023 minimal AMI or Amazon ECS-optimized AMI
- CentOS Stream 8 and 9
- macOS Sonoma prior to 14.2.1, Ventura prior to 13.6.3, and Monterey prior to 12.7.2 *
- Red Hat Enterprise Linux (RHEL) 8 and 9
- Ubuntu 16.04 and 18.04 *

> **Tip**
>
> * For Amazon Linux 2, macOS, and Ubuntu: If you launched your instance using a later version than those listed above, EC2 Instance Connect comes preinstalled and no manual installation is required.

- **Verify the general prerequisites for EC2 Instance Connect.**

  For more information, see Prerequisites for EC2 Instance Connect.
- **Verify the prerequisites for connecting to your instance using an SSH client on your local machine.**

  For more information, see Connect to your Linux instance using SSH.
- **Get the ID of the instance.**

  You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the describe-instances (AWS CLI) or Get-EC2Instance (AWS Tools for Windows PowerShell) command.

Manually install EC2 Instance Connect

> **Note**
>
> If you launched your instance using one of the following AMIs, EC2 Instance Connect is pre-installed and you can skip this procedure:
>
> - AL2023 standard AMI
> - Amazon Linux 2 2.0.20190618 or later
> - macOS Sonoma 14.2.1 or later

- macOS Ventura 13.6.3 or later
- macOS Monterey 12.7.2 or later
- Ubuntu 20.04 or later

Use one of the following procedures for installing EC2 Instance Connect, depending on the operating system of your instance.

Amazon Linux 2

To install EC2 Instance Connect on an instance launched with Amazon Linux 2

1. Connect to your instance using SSH.

   Replace the example values in the following command with your values. Use the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For Amazon Linux 2, the default username is `ec2-user`.

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
   ```

   For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. Install the EC2 Instance Connect package on your instance.

   ```
   [ec2-user ~]$ sudo yum install ec2-instance-connect
   ```

   You should see three new scripts in the `/opt/aws/bin/` folder:

   ```
   eic_curl_authorized_keys
   eic_parse_authorized_keys
   eic_run_authorized_keys
   ```

3. (Optional) Verify that EC2 Instance Connect was successfully installed on your instance.

   ```
   [ec2-user ~]$ sudo less /etc/ssh/sshd_config
   ```

EC2 Instance Connect was successfully installed if the AuthorizedKeysCommand and AuthorizedKeysCommandUser lines contain the following values:

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand sets the eic_run_authorized_keys script to look up the keys from the instance metadata
- AuthorizedKeysCommandUser sets the system user as ec2-instance-connect

Note

If you previously configured AuthorizedKeysCommand and AuthorizedKeysCommandUser, the EC2 Instance Connect installation will not change the values and you will not be able to use EC2 Instance Connect.

CentOS

To install EC2 Instance Connect on an instance launched with CentOS

1. Connect to your instance using SSH.

   Replace the example values in the following command with your values. Use the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For CentOS, the default username is centos or ec2-user.

   ```
   $ ssh -i my_ec2_private_key.pem centos@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
   ```

   For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. If you use an HTTP or HTTPS proxy, you must set the http_proxy or https_proxy environment variables in the current shell session.

   If you're not using a proxy, you can skip this step.

- For an HTTP proxy server, run the following commands:

```
$ export http_proxy=http://hostname:port
$ export https_proxy=http://hostname:port
```

- For an HTTPS proxy server, run the following commands:

```
$ export http_proxy=https://hostname:port
$ export https_proxy=https://hostname:port
```

3. Install the EC2 Instance Connect package on your instance by running the following commands.

   The EC2 Instance Connect configuration files for CentOS are provided in a Red Hat Package Manager (RPM) package, with different RPM packages for CentOS 8 and CentOS 9 and for instance types that run on Intel/AMD (x86_64) or ARM (AArch64).

   Use the command block for your operating system and CPU architecture.

   - CentOS 8

     Intel/AMD (x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-5.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

     ARM (AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
```

```
connect-2.0.0-5.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- CentOS 9

  Intel/AMD (x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-2.0.0-5.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

  ARM (AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-2.0.0-5.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

You should see the following new script in the `/opt/aws/bin/` folder:

```
eic_run_authorized_keys
```

4. (Optional) Verify that EC2 Instance Connect was successfully installed on your instance.

   - For CentOS 8:

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-connect.conf
     ```

   - For CentOS 9:

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

   EC2 Instance Connect was successfully installed if the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` lines contain the following values:

   ```
   AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - `AuthorizedKeysCommand` sets the `eic_run_authorized_keys` script to look up the keys from the instance metadata
   - `AuthorizedKeysCommandUser` sets the system user as `ec2-instance-connect`

   > **Note**
   >
   > If you previously configured `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, the EC2 Instance Connect installation will not change the values and you will not be able to use EC2 Instance Connect.

macOS

To install EC2 Instance Connect on an instance launched with macOS

1. Connect to your instance using SSH.

Replace the example values in the following command with your values. Use the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For macOS instances, the default username is `ec2-user`.

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. Update Homebrew using the following command. The update will list the software that Homebrew knows about. The EC2 Instance Connect package is provided via Homebrew on macOS instances. For more information, see Update the operating system and software on Amazon EC2 Mac instances.

```
[ec2-user ~]$ brew update
```

3. Install the EC2 Instance Connect package on your instance. This will install the software and configure sshd to use it.

```
[ec2-user ~]$ brew install ec2-instance-connect
```

You should see the following new script in the `/opt/aws/bin/` folder:

```
eic_run_authorized_keys
```

4. (Optional) Verify that EC2 Instance Connect was successfully installed on your instance.

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
```

EC2 Instance Connect was successfully installed if the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` lines contain the following values:

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- `AuthorizedKeysCommand` sets the `eic_run_authorized_keys` script to look up the keys from the instance metadata
- `AuthorizedKeysCommandUser` sets the system user as `ec2-instance-connect`

> **Note**
>
> If you previously configured `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, the EC2 Instance Connect installation will not change the values and you will not be able to use EC2 Instance Connect.

RHEL

To install EC2 Instance Connect on an instance launched with Red Hat Enterprise Linux (RHEL)

1. Connect to your instance using SSH.

   Replace the example values in the following command with your values. Use the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For RHEL, the default username is `ec2-user` or `root`.

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
   ```

   For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. If you use an HTTP or HTTPS proxy, you must set the `http_proxy` or `https_proxy` environment variables in the current shell session.

   If you're not using a proxy, you can skip this step.

   - For an HTTP proxy server, run the following commands:

     ```
     $ export http_proxy=http://hostname:port
     $ export https_proxy=http://hostname:port
     ```

- For an HTTPS proxy server, run the following commands:

```
$ export http_proxy=https://hostname:port
$ export https_proxy=https://hostname:port
```

3. Install the EC2 Instance Connect package on your instance by running the following commands.

   The EC2 Instance Connect configuration files for RHEL are provided in a Red Hat Package Manager (RPM) package, with different RPM packages for RHEL 8 and RHEL 9 and for instance types that run on Intel/AMD (x86_64) or ARM (AArch64).

   Use the command block for your operating system and CPU architecture.

   - RHEL 8

     Intel/AMD (x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-5.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

     ARM (AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-5.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
```

```
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- RHEL 9

  Intel/AMD (x86_64)

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-2.0.0-5.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect.rpm
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  [ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  ```

  ARM (AArch64)

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-2.0.0-5.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect.rpm
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  [ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  ```

You should see the following new script in the `/opt/aws/bin/` folder:

```
eic_run_authorized_keys
```

4. (Optional) Verify that EC2 Instance Connect was successfully installed on your instance.

   - For RHEL 8:

```
[ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-connect.conf
```

- For RHEL 9:

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
```

EC2 Instance Connect was successfully installed if the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` lines contain the following values:

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- `AuthorizedKeysCommand` sets the `eic_run_authorized_keys` script to look up the keys from the instance metadata
- `AuthorizedKeysCommandUser` sets the system user as `ec2-instance-connect`

> **Note**
>
> If you previously configured `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, the EC2 Instance Connect installation will not change the values and you will not be able to use EC2 Instance Connect.

Ubuntu

To install EC2 Instance Connect on an instance launched with Ubuntu 16.04 or later

1. Connect to your instance using SSH.

   Replace the example values in the following command with your values. Use the SSH key pair that was assigned to your instance when you launched it and use the default username of the AMI that you used to launch your instance. For an Ubuntu AMI, the username is `ubuntu`.

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. (Optional) Ensure your instance has the latest Ubuntu AMI.

   Run the following commands to update all the packages on your instance.

   ```
   ubuntu:~$ sudo apt-get update
   ```

   ```
   ubuntu:~$ sudo apt-get upgrade
   ```

3. Install the EC2 Instance Connect package on your instance.

   ```
   ubuntu:~$ sudo apt-get install ec2-instance-connect
   ```

   You should see three new scripts in the `/usr/share/ec2-instance-connect/` folder:

   ```
   eic_curl_authorized_keys
   eic_parse_authorized_keys
   eic_run_authorized_keys
   ```

4. (Optional) Verify that EC2 Instance Connect was successfully installed on your instance.

   ```
   ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
   ```

   EC2 Instance Connect was successfully installed if the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` lines contain the following values:

   ```
   AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %%u %%f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - `AuthorizedKeysCommand` sets the `eic_run_authorized_keys` script to look up the keys from the instance metadata
   - `AuthorizedKeysCommandUser` sets the system user as `ec2-instance-connect`

> **Note**
>
> If you previously configured `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, the EC2 Instance Connect installation will not change the values and you will not be able to use EC2 Instance Connect.

Connect to a Linux instance using EC2 Instance Connect

The following instructions explain how to connect to your Linux instance using EC2 Instance Connect through the Amazon EC2 console, the AWS CLI, or an SSH client.

When you connect to an instance using EC2 Instance Connect through the console or AWS CLI, the EC2 Instance Connect API automatically pushes an SSH public key to the instance metadata where it remains for 60 seconds. An IAM policy attached to your user authorizes this action. If you prefer using your own SSH key, you can use an SSH client and explicitly push your SSH key to the instance using EC2 Instance Connect.

Considerations

After connecting to an instance using EC2 Instance Connect, the connection persists until the SSH session is terminated. The duration of the connection is not determined by the duration of your IAM credentials. If your IAM credentials expire, the connection continues to persist. When using the EC2 Instance Connect console experience, if your IAM credentials expire, terminate the connection by closing the browser page. When using your own SSH client and EC2 Instance Connect to push your key, you can set a SSH timeout value to terminate the SSH session automatically.

Requirements

Before you begin, be sure to review the prerequisites.

Connection options

- Connect using the Amazon EC2 console
- Connect using the AWS CLI
- Connect using your own key and SSH client
- Troubleshoot

Connect using the Amazon EC2 console

You can connect to an instance using EC2 Instance Connect through the Amazon EC2 console.

Requirements

To connect using the Amazon EC2 console, the instance must have either a public IPv4 or a public IPv6 address. If the instance only has a private IPv4 address, you can use the ec2-instance-connect AWS CLI to connect.

To connect to your instance using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Connect**.
4. Choose the **EC2 Instance Connect** tab.
5. Choose **Connect using a Public IP**.
6. If there is a choice, select the IP address to connect to. Otherwise, the IP address is selected automatically.
7. For **Username**, verify the username.
8. Choose **Connect** to establish a connection. An in-browser terminal window opens.

Connect using the AWS CLI

You can use the ec2-instance-connect AWS CLI to connect to your instance with an SSH client. EC2 Instance Connect attempts to establish a connection using an available IP address in a predefined order, based on the specified connection type. If an IP address isn't available, it automatically tries the next one in the order.

Connection types

`auto` (default)

EC2 Instance Connect tries to connect using the instance's IP addresses in the following order and with the corresponding connection type:

1. Public IPv4: `direct`
2. Private IPv4: `eice`

3. Public IPv6: `direct`

`direct`

EC2 Instance Connect tries to connect using the instance's IP addresses in the following order:

1. Public IPv4
2. Public IPv6
3. Private IPv4 (it does not connect over an EC2 Instance Connect Endpoint)

`eice`

EC2 Instance Connect tries to connect using the instance's private IPv4 address and an EC2 Instance Connect Endpoint.

> **Note**
>
> In the future, we might change the behavior of the `auto` connection type. To ensure that your desired connection type is used, we recommend that you explicitly set the `--connection-type` to either `direct` or `eice`.
> Connecting to a *private* IPv6 address is not supported when using the ec2-instance-connect AWS CLI.

Requirements

You must use AWS CLI version 2. For more information, see Install or update to the latest version of the AWS CLI.

To connect to an instance using the instance ID

If you only know the instance ID, and want to let EC2 Instance Connect determine the connection type to use when connecting to your instance, use the ec2-instance-connect ssh CLI command with the instance ID.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example
```

To connect to an instance using the instance ID and an EC2 Instance Connect Endpoint

If you want to connect to your instance over an EC2 Instance Connect Endpoint, use the preceding command and also specify the `--connection-type` parameter with the `eice` value.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --connection-type eice
```

To connect to an instance using the instance ID and your own private key file

If you want to connect to your instance over an EC2 Instance Connect Endpoint using your own private key, specify the instance ID and the path to the private key file. Do not include *file://* in the path; the following example will fail: *file:///path/to/key*.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --private-key-file /
path/to/key.pem
```

> **Tip**
>
> If you get an error when using these commands, make sure that you're using AWS CLI version 2, because the `ssh` command is only available in this major version. We also recommend regularly updating to the latest minor version of AWS CLI version 2 to access the latest features. For more information, see About AWS CLI version 2 in the *AWS Command Line Interface User Guide*.

Connect using your own key and SSH client

You can use your own SSH key and connect to your instance from the SSH client of your choice while using the EC2 Instance Connect API. This enables you to benefit from the EC2 Instance Connect capability to push a public key to the instance. This connection method works for instances with public and private IP addresses.

Requirements

- Requirements for key pairs
  - Supported types: RSA (OpenSSH and SSH2) and ED25519
  - Supported lengths: 2048 and 4096
  - For more information, see Create a key pair using a third-party tool and import the public key to Amazon EC2.
- When connecting to an instance that has only private IP addresses, the local computer from which you are initiating the SSH session must have connectivity to the EC2 Instance Connect service endpoint (to push your SSH public key to the instance) as well as network connectivity to

the instance's private IP address to establish the SSH session. The EC2 Instance Connect service endpoint is reachable over the internet or over an AWS Direct Connect public virtual interface. To connect to the instance's private IP address, you can leverage services such as AWS Direct Connect, AWS Site-to-Site VPN, or VPC peering.

To connect to your instance using your own key and any SSH client

1. **(Optional) Generate new SSH private and public keys**

   You can generate new SSH private and public keys, `my_key` and `my_key.pub`, using the following command:

   ```
   ssh-keygen -t rsa -f my_key
   ```

2. **Push your SSH public key to the instance**

   Use the send-ssh-public-key command to push your SSH public key to the instance. If you launched your instance using AL2023 or Amazon Linux 2, the default username for the AMI is `ec2-user`. If you launched your instance using Ubuntu, the default username for the AMI is `ubuntu`.

   The following example pushes the public key to the specified instance in the specified Availability Zone, to authenticate `ec2-user`.

   ```
   aws ec2-instance-connect send-ssh-public-key \
       --region us-west-2 \
       --availability-zone us-west-2b \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --instance-os-user ec2-user \
       --ssh-public-key file://my_key.pub
   ```

3. **Connect to the instance using your private key**

   Use the **ssh** command to connect to the instance using the private key before the public key is removed from the instance metadata (you have 60 seconds before it is removed). Specify the private key that corresponds to the public key, the default username for the AMI that you used to launch your instance, and the instance's public DNS name (if connecting over a private network, specify the private DNS name or IP address). Add the `IdentitiesOnly=yes` option to ensure that only the files in the ssh config and the specified key are used for the connection.

```
ssh -o "IdentitiesOnly=yes" -i my_key ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

The following example uses `timeout 3600` to set your SSH session to terminate after 1 hour. Processes started during the session may continue running on your instance after the session terminates.

```
timeout 3600 ssh -o “IdentitiesOnly=yes” -i my_key ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

Troubleshoot

If you receive an error while attempting to connect to your instance, see the following:

- Troubleshoot issues connecting to your Amazon EC2 Linux instance
- How do I troubleshoot issues connecting to my EC2 instance using EC2 Instance Connect?

Uninstall EC2 Instance Connect

To disable EC2 Instance Connect, connect to your Linux instance and uninstall the `ec2-instance-connect` package that is installed on the OS. If the `sshd` configuration matches what it was set to when you installed EC2 Instance Connect, uninstalling `ec2-instance-connect` also removes the `sshd` configuration. If you modified the `sshd` configuration after installing EC2 Instance Connect, you must update it manually.

Amazon Linux

You can uninstall EC2 Instance Connect on AL2023 and Amazon Linux 2 2.0.20190618 or later, where EC2 Instance Connect is preconfigured.

To uninstall EC2 Instance Connect on an instance launched using Amazon Linux

1. Connect to your instance using SSH. Specify the SSH key pair you used for your instance when you launched it and the default username for the AL2023 or Amazon Linux 2 AMI, which is `ec2-user`.

For example, the following **ssh** command connects to the instance with the public DNS name `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, using the key pair `my_ec2_private_key.pem`.

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

2. Uninstall the `ec2-instance-connect` package using the **yum** command.

```
[ec2-user ~]$ sudo yum remove ec2-instance-connect
```

Ubuntu

To uninstall EC2 Instance Connect on an instance launched using an Ubuntu AMI

1. Connect to your instance using SSH. Specify the SSH key pair you used for your instance when you launched it and the default username for the Ubuntu AMI, which is `ubuntu`.

   For example, the following **ssh** command connects to the instance with the public DNS name `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, using the key pair `my_ec2_private_key.pem`.

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

2. Uninstall the `ec2-instance-connect` package using the **apt-get** command.

```
ubuntu:~$ sudo apt-get remove ec2-instance-connect
```

Connect to your instances using a private IP address and EC2 Instance Connect Endpoint

EC2 Instance Connect Endpoint allows you to connect securely to an instance from the internet, without using a bastion host, or requiring that your virtual private cloud (VPC) has direct internet connectivity.

Benefits

- You can connect to your instances without requiring the instances to have a public IPv4 address. AWS charges for all public IPv4 addresses, including public IPv4 addresses associated with running instances and Elastic IP addresses. For more information, see the **Public IPv4 Address** tab on the Amazon VPC pricing page.
- You can connect to your instances from the internet without requiring that your VPC has direct internet connectivity through an internet gateway.
- You can control access to the creation and use of the EC2 Instance Connect Endpoints to connect to instances using IAM policies and permissions.
- All attempts to connect to your instances, both successful and unsuccessful, are logged to CloudTrail.

Pricing

There is no additional cost for using EC2 Instance Connect Endpoints. If you use an EC2 Instance Connect Endpoint to connect to an instance in a different Availability Zone, there is an additional charge for data transfer across Availability Zones.

Contents

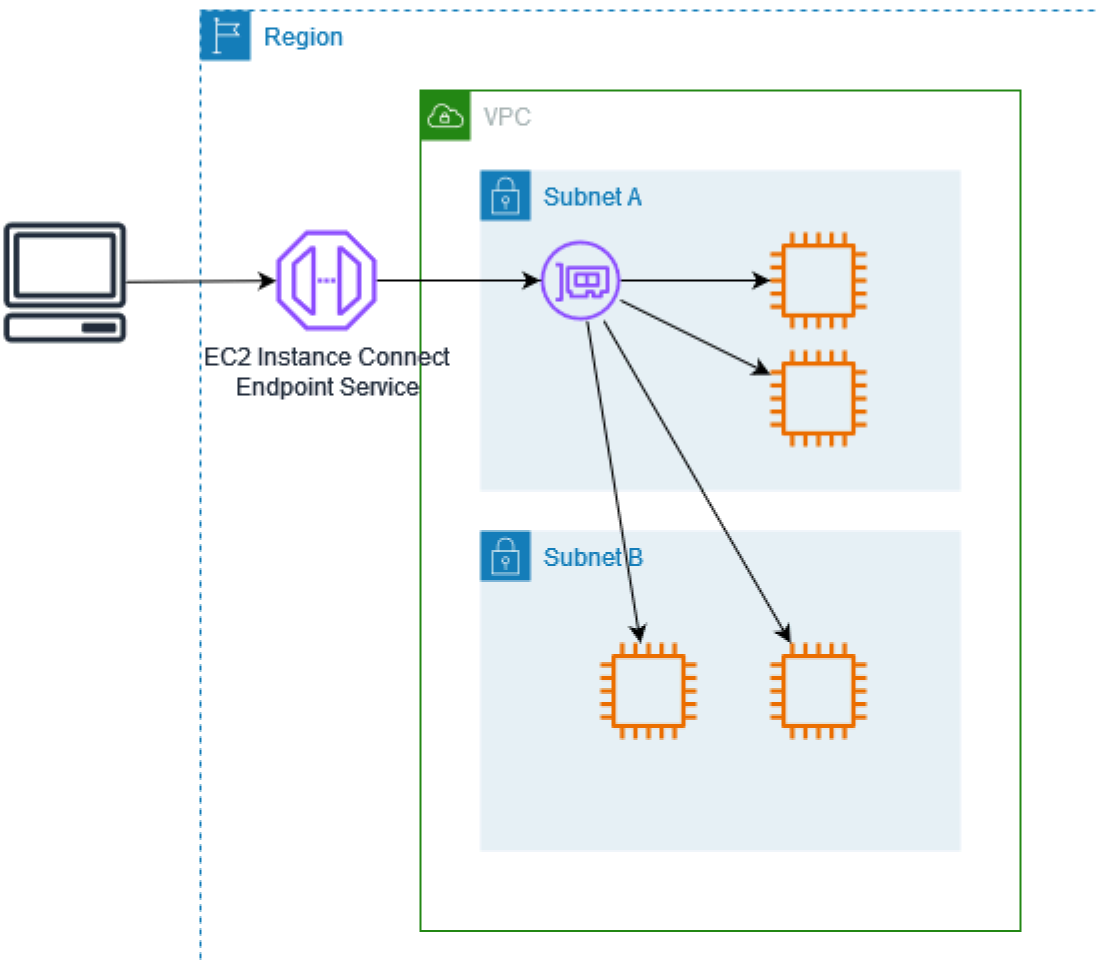
How it works

EC2 Instance Connect Endpoint is an identity-aware TCP proxy. The EC2 Instance Connect Endpoint Service establishes a private tunnel from your computer to the endpoint using the credentials for your IAM entity. Traffic is authenticated and authorized before it reaches your VPC.

You can configure additional security group rules to restrict inbound traffic to your instances. For example, you can use inbound rules to allow traffic on management ports only from the EC2 Instance Connect Endpoint.

You can configure route table rules to allow the endpoint to connect to any instance in any subnet of the VPC.

The following diagram shows how a user can connect to their instances from the internet using an EC2 Instance Connect Endpoint. First, create an **EC2 Instance Connect Endpoint** in subnet A. We create a network interface for the endpoint in the subnet, which serves as the entry point for traffic destined to your instances in the VPC. If the route table for subnet B allows traffic from subnet A, then you can use the endpoint to reach instances in subnet B.



Considerations

Before you begin, consider the following.

- EC2 Instance Connect Endpoint is intended specifically for management traffic use cases, not for high volume data transfers. High volume data transfers are throttled.
- Your instance must have an IPv4 address (either private or public). EC2 Instance Connect Endpoint does not support connecting to instances using IPv6 addresses.
- (Linux instances) If you use your own key pair, you can use any Linux AMI. Otherwise, your instance must have EC2 Instance Connect installed. For information about which AMIs include EC2 Instance Connect and how to install it on other supported AMIs, see Install EC2 Instance Connect.
- You can assign a security group to an EC2 Instance Connect Endpoint when you create it. Otherwise, we use the default security group for the VPC. The security group for an EC2 Instance Connect Endpoint must allow outbound traffic to the destination instances. For more information, see Security groups for EC2 Instance Connect Endpoint.
- You can configure an EC2 Instance Connect Endpoint to preserve the source IP addresses of clients when routing requests to the instances. Otherwise, the IP address of the network interface becomes the client IP address for all incoming traffic.
  - If you turn on client IP preservation, the security groups for the instances must allow traffic from the clients. Also, the instances must be in the same VPC as the EC2 Instance Connect Endpoint.
  - If you turn off client IP preservation, the security groups for the instances must allow traffic from the VPC. This is the default.
  - The following instance types do not support client IP preservation: C1, CG1, CG2, G1, HI1, M1, M2, M3, and T1. If you turn on client IP preservation and attempt to connect to an instance with one of these instance types by using EC2 Instance Connect Endpoint, the connection fails.
  - Client IP preservation is not supported when traffic is routed through a transit gateway.
- When you create an EC2 Instance Connect Endpoint, a service-linked role is automatically created for the Amazon EC2 service in AWS Identity and Access Management (IAM). Amazon EC2 uses the service-linked role to provision network interfaces in your account, which are required when creating EC2 Instance Connect Endpoints. For more information, see Service-linked role for EC2 Instance Connect Endpoint.

- You can create only 1 EC2 Instance Connect Endpoint per VPC and per subnet. For more information, see Quotas for EC2 Instance Connect Endpoint. If you need to create another EC2 Instance Connect Endpoint in a different Availability Zone within the same VPC, you must first delete the existing EC2 Instance Connect Endpoint. Otherwise, you'll receive a quota error.
- Each EC2 Instance Connect Endpoint can support up to 20 concurrent connections.
- The maximum duration for an established TCP connection is 1 hour (3,600 seconds). You can specify the maximum allowed duration in an IAM policy, which can be up to 3,600 seconds. For more information, see Permissions to use EC2 Instance Connect Endpoint to connect to instances.

  The duration of the connection is not determined by the duration of your IAM credentials. If your IAM credentials expire, the connection continues to persist until the specified maximum duration is reached. When you connect to an instance using the EC2 Instance Connect Endpoint console experience, set **Max tunnel duration (seconds)** to a value that is less than the duration of your IAM credentials. If your IAM credentials expire early, terminate the connection to your instance by closing the browser page.

Grant permissions to use EC2 Instance Connect Endpoint

By default, IAM entities don't have permission to create, describe, or modify EC2 Instance Connect Endpoints. An IAM administrator can create IAM policies that grant the permissions required to perform specific actions on the resources that they need.

For information about creating IAM policies, see Creating IAM policies in the *IAM User Guide*.

The following example policies show how you can control the permissions that users have to EC2 Instance Connect Endpoints.

Examples

- Permissions to create, describe, modify, and delete EC2 Instance Connect Endpoints
- Permissions to use EC2 Instance Connect Endpoint to connect to instances
- Permissions to connect only from a specific IP address range

Permissions to create, describe, modify, and delete EC2 Instance Connect Endpoints

To create and modify an EC2 Instance Connect Endpoint, users require permissions for the following actions:

- `ec2:CreateInstanceConnectEndpoint`
- `ec2:CreateNetworkInterface`
- `ec2:CreateTags`
- `ec2:ModifyInstanceConnectEndpoint`
- `iam:CreateServiceLinkedRole`

To describe and delete EC2 Instance Connect Endpoints, users require permissions for the following actions:

- `ec2:DescribeInstanceConnectEndpoints`
- `ec2:DeleteInstanceConnectEndpoint`

You can create a policy that grants permission to create, describe, modify, and delete EC2 Instance Connect Endpoints in all subnets. Alternatively, you can restrict actions for specified subnets only by specifying the subnet ARNs as the allowed `Resource` or by using the `ec2:SubnetID` condition key. You can also use the `aws:ResourceTag` condition key to explicitly allow or deny endpoint creation with certain tags. For more information, see [Policies and permissions in IAM]() in the *IAM User Guide*.

Example IAM policy

In the following example IAM policy, the `Resource` section grants permission to create, modify, and delete endpoints in all subnets, specified by the asterisk (`*`). The `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.

Permissions to use EC2 Instance Connect Endpoint to connect to instances

The `ec2-instance-connect:OpenTunnel` action grants permission to establish a TCP connection to an instance to connect over the EC2 Instance Connect Endpoint. You can specify the EC2 Instance Connect Endpoint to use. Alternatively, a `Resource` with an asterisk (`*`) allows users to use any available EC2 Instance Connect Endpoint. You can also restrict access to instances based on the presence or absence of resource tags as condition keys.

Conditions

- `ec2-instance-connect:remotePort` – The port on the instance that can be used to establish a TCP connection. When this condition key is used, attempting to connect to an instance on any other port other than the port specified in the policy results in a failure.
- `ec2-instance-connect:privateIpAddress` – The destination private IP address associated with the instance that you want to establish a TCP connection with. You can specify a single IP address, such as `10.0.0.1/32`, or a range of IPs through CIDRs, such as `10.0.1.0/28`. When this condition key is used, attempting to connect to an instance with a different private IP address or outside the CIDR range results in a failure.
- `ec2-instance-connect:maxTunnelDuration` – The maximum duration for an established TCP connection. The unit is seconds and the duration ranges from a minimum of 1 second to a maximum of 3,600 seconds (1 hour). If the condition is not specified, the default duration is set to 3,600 seconds (1 hour). Attempting to connect to an instance for longer than the specified duration in the IAM policy or for longer than the default maximum results in a failure. The connection is disconnected after the specified duration.

  If `maxTunnelDuration` is specified in the IAM policy and the value specified is less than 3,600 seconds (the default), then you must specify `--max-tunnel-duration` in the command when connecting to an instance. For information about how to connect to an instance, see Connect to an Amazon EC2 instance using EC2 Instance Connect Endpoint.

You can also grant a user access to establish connections to instances based on the presence of resource tags on the EC2 Instance Connect Endpoint. For more information, see Policies and permissions in IAM in the *IAM User Guide*.

For Linux instances, the `ec2-instance-connect:SendSSHPublicKey` action grants permission to push the public key to an instance. The `ec2:osuser` condition specifies the name of the OS (operating system) user that can push the public key to an instance. Use the default username for the AMI that you used to launch the instance. For more information, see Grant IAM permissions for EC2 Instance Connect.

Example IAM policy

The following example IAM policies allow an IAM principal to connect to an instance using only the specified EC2 Instance Connect Endpoint, identified by the specified endpoint ID `eice-123456789abcdef`. The connection is successfully established only if all the conditions are satisfied.

> **Note**
>
> The `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.

Linux

This example evaluates if the connection to the instance is established on —port 22 (SSH), if the private IP address of the instance lies within the range of `10.0.1.0/31` (between `10.0.1.0` and `10.0.1.1`), and the `maxTunnelDuration` is less than or equal to `3600` seconds. The connection is disconnected after `3600` seconds (1 hour).

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Windows

This example evaluates if the connection to the instance is established on port 3389 (RDP), if the private IP address of the instance lies within the range of `10.0.1.0/31` (between `10.0.1.0` and `10.0.1.1`), and the `maxTunnelDuration` is less than or equal to `3600` seconds. The connection is disconnected after `3600` seconds (1 hour).

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "3389"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
```

```
                    "NumericLessThanEquals": {
                        "ec2-instance-connect:maxTunnelDuration": "3600"
                    }
                }
            },
            {
                "Sid": "Describe",
                "Action": [
                    "ec2:DescribeInstances",
                    "ec2:DescribeInstanceConnectEndpoints"
                ],
                "Effect": "Allow",
                "Resource": "*"
            }
        ]
}
```

Permissions to connect only from a specific IP address range

The following example IAM policy allows an IAM principal to connect to an instance on condition they are connecting from an IP address within the IP address range specified in the policy. If the IAM principal calls `OpenTunnel` from an IP address not within `192.0.2.0/24` (the example IP address range in this policy), the response is `Access Denied`. For more information, see aws:SourceIp in the *IAM User Guide*.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                }
```

```
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Resource": "*"
        }
    ]
}
```

Security groups for EC2 Instance Connect Endpoint

A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For example, we deny traffic to and from an Amazon EC2 instance unless it is specifically allowed by the security groups associated with the instance.

The following examples show you how to configure the security group rules for the EC2 Instance Connect Endpoint and the target instances.

Examples

- EC2 Instance Connect Endpoint security group rules
- Target instance security group rules

EC2 Instance Connect Endpoint security group rules

The security group rules for an EC2 Instance Connect Endpoint must allow outbound traffic destined for the target instances to leave the endpoint. You can specify either the instance security group or the IPv4 address range of the VPC as the destination.

Traffic to the endpoint originates from the EC2 Instance Connect Endpoint Service, and it is allowed regardless of the inbound rules for the endpoint security group. To control who can use EC2 Instance Connect Endpoint to connect to an instance, use an IAM policy. For more information, see Permissions to use EC2 Instance Connect Endpoint to connect to instances.

Example outbound rule: Security group referencing

The following example uses security group referencing, which means that the destination is a security group associated with the target instances. This rule allows outbound traffic from the endpoint to all instances that use this security group.

Protocol	Destination	Port range	Comment
TCP	*ID of instance security group*	22	Allows outbound SSH traffic to all instances associated with the instance security group

Example outbound rule: IPv4 address range

The following example allows outbound traffic to the specified IPv4 address range. The IPv4 addresses of an instance is assigned from its subnet, so you can use the IPv4 address range of the VPC.

Protocol	Destination	Port range	Comment
TCP	*VPC IPv4 CIDR*	22	Allows outbound SSH traffic to the VPC

Target instance security group rules

The security group rules for target instances must allow inbound traffic from the EC2 Instance Connect Endpoint. You can specify either the endpoint security group or an IPv4 address range as the source. If you specify an IPv4 address range, the source depends on whether client IP preservation is off or on. For more information, see Considerations.

Because security groups are stateful, the response traffic is allowed to leave the VPC regardless of the outbound rules for the instance security group.

Example inbound rule: Security group referencing

The following example uses security group referencing, which means that the source is the security group associated with the endpoint. This rule allows inbound SSH traffic from the endpoint to all instances that use this security group, whether client IP preservation is on or off. If there are no other inbound security group rules for SSH, then the instances accept SSH traffic only from the endpoint.

Protocol	Source	Port range	Comment
TCP	*ID of endpoint security group*	22	Allows inbound SSH traffic from the resources associated with the endpoint security group

Example inbound rule: Client IP preservation off

The following example allows inbound SSH traffic from the specified IPv4 address range. Because client IP preservation is off, the source IPv4 address is the address of the endpoint network interface. The address of the endpoint network interface is assigned from its subnet, so you can use the IPv4 address range of the VPC to allow connections to all instances in the VPC.

Protocol	Source	Port range	Comment
TCP	*VPC IPv4 CIDR*	22	Allows inbound SSH traffic from the VPC

Example inbound rule: Client IP preservation on

The following example allows inbound SSH traffic from the specified IPv4 address range. Because client IP preservation is on, the source IPv4 address is the address of the client.

Protocol	Source	Port range	Comment
TCP	*Public IPv4 address range*	22	Allows inbound traffic from the specified client IPv4 address range

Create an EC2 Instance Connect Endpoint

You can create an EC2 Instance Connect Endpoint to allow secure connection to your instances.

Prerequisites

You must have the required IAM permissions to create an EC2 Instance Connect Endpoint. For more information, see [Permissions to create, describe, modify, and delete EC2 Instance Connect Endpoints].

Shared subnets

You can create an EC2 Instance Connect Endpoint in a subnet that is shared with you. You can't use a EC2 Instance Connect Endpoint that the VPC owner created in a subnet that is shared with you.

Console

To create an EC2 Instance Connect Endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the left navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**, and then specify the endpoint settings as follows:

    a. (Optional) For **Name tag**, enter a name for the endpoint.

    b. For **Type**, choose **EC2 Instance Connect Endpoint**.

    c. Under **Network settings**, for **VPC**, select the VPC that has the target instances.

    d. (Optional) To preserve client IP addresses, expand **Additional settings** and select the **Preserve Client IP** check box. Otherwise, the default is to use the endpoint network interface as the client IP address.

e. (Optional) For **Security groups**, select the security group to associate with the endpoint. Otherwise, the default is to use the default security group for the VPC. For more information, see Security groups for EC2 Instance Connect Endpoint.

f. For **Subnet**, select the subnet in which to create the endpoint.

g. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.

4. Review your settings and then choose **Create endpoint**.

   The initial status of the endpoint is **Pending**. Before you can connect to an instance using this endpoint, you must wait until the endpoint status is **Available**. This can take a few minutes.

5. To connect to an instance using your endpoint, see Connect to an instance.

AWS CLI

To create an EC2 Instance Connect Endpoint

Use the create-instance-connect-endpoint command.

```
aws ec2 create-instance-connect-endpoint --subnet-id subnet-0123456789example
```

The following is example output.

```
{
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
        "State": "create-complete",
        "StateMessage": "",
        "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-endpoint.us-east-1.amazonaws.com",
        "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-endpoint.us-east-1.amazonaws.com",
        "NetworkInterfaceIds": [
            "eni-0123abcd"
        ],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1a",
        "CreatedAt": "2023-04-07T15:43:53.000Z",
```

```
        "SubnetId": "subnet-0123abcd",
        "PreserveClientIp": false,
        "SecurityGroupIds": [
            "sg-0123abcd"
        ],
        "Tags": []
}
```

To monitor the creation status

The initial value for the `State` field is `create-in-progress`. Before you can connect to an instance using this endpoint, wait until the state is `create-complete`. Use the describe-instance-connect-endpoints command to monitor the status of the EC2 Instance Connect Endpoint. The `--query` parameter filters the results to the `State` field.

```
aws ec2 describe-instance-connect-endpoints --instance-connect-endpoint-
ids eice-0123456789example --query InstanceConnectEndpoints[*].State --output text
```

The following is example output.

```
create-complete
```

PowerShell

To create the EC2 Instance Connect Endpoint

Use the New-EC2InstanceConnectEndpoint cmdlet.

```
New-EC2InstanceConnectEndpoint -SubnetId subnet-0123456789example
```

The following is example output.

```
OwnerId                     : 111111111111
InstanceConnectEndpointId   : eice-0123456789example
InstanceConnectEndpointArn  : arn:aws:ec2:us-east-1:111111111111:instance-connect-
endpoint/eice-0123456789example
State                       : create-complete
StateMessage                :
DnsName                     : eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com
```

```
FipsDnsName                  : eice-0123456789example.0123abcd.fips.ec2-instance-
connect-endpoint.us-east-1.amazonaws.com
NetworkInterfaceIds          : {eni-0123abcd}
VpcId                        : vpc-0123abcd
AvailabilityZone             : us-east-1a
CreatedAt                    : 4/7/2023 3:43:53 PM
SubnetId                     : subnet-0123abcd
PreserveClientIp             : False
SecurityGroupIds             : {sg-0123abcd}
Tags                         : {}
```

To monitor the creation status

The initial value for the `State` field is `create-in-progress`. Before you can connect to an instance using this endpoint, wait until the state is `create-complete`. Use the Get-EC2InstanceConnectEndpoint cmdlet to monitor the status of the EC2 Instance Connect Endpoint. `.State.Value` filters the results to the `State` field.

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

The following is example output.

```
create-complete
```

Modify an EC2 Instance Connect Endpoint

You can modify existing EC2 Instance Connect Endpoints using the AWS CLI or an SDK. The Amazon EC2 console doesn't support endpoint modification.

Before you begin, you must have the required IAM permissions. For more information, see Permissions to create, describe, modify, and delete EC2 Instance Connect Endpoints.

Parameters you can modify

You can modify the following EC2 Instance Connect Endpoint parameters:

Security groups

You can specify new security groups for the EC2 Instance Connect Endpoint. The new security groups replace the current security groups.

When modifying the security groups, you must specify:

- At least one security group, even if it's just the default security group in the VPC.
- The IDs of the security groups, not the names.

IP address type

You can specify a new IP address type for the EC2 Instance Connect Endpoint.

Valid values: `ipv4` | `dualstack` | `ipv6`

Preserve client IP setting

You can specify whether to preserve the client IP address as the source.

> **Note**
>
> Preserving the client IP is only supported on IPv4 EC2 Instance Connect Endpoints. When enabling `PreserveClientIp`, either the endpoint's existing IP address type must be `ipv4`, or if modifying the IP address type in the same request, the new value must be `ipv4`.

AWS CLI

To modify an EC2 Instance Connect Endpoint

Use the modify-instance-connect-endpoint command and specify the EC2 Instance Connect Endpoint and the parameters to modify. The following example modifies all the parameters in a single request.

```
aws ec2 modify-instance-connect-endpoint \
    --instance-connect-endpoint-id eice-0123456789example \
    --security-group-ids sg-0123456789example \
    --ip-address-type dualstack \
    --no-preserve-client-ip
```

The following is example output.

```
{
    "Return": true
}
```

To monitor the update status

During modification, the EC2 Instance Connect Endpoint status changes to `update-in-progress`. The update process runs asynchronously and completes with either an `update-complete` or `update-failed` status. The endpoint uses its old configuration until the status changes to `update-complete`.

Use the describe-instance-connect-endpoints command to monitor the update status. The `--query` parameter filters the results to the `State` field.

```
aws ec2 describe-instance-connect-endpoints \
    --instance-connect-endpoint-ids eice-0123456789example \
    --query InstanceConnectEndpoints[*].State --output text
```

The following is example output.

```
update-complete
```

PowerShell

To modify an EC2 Instance Connect Endpoint

Use the Edit-EC2InstanceConnectEndpoint cmdlet and specify the EC2 Instance Connect Endpoint and the parameters to modify. The following example modifies all the parameters in a single request.

```
Edit-EC2InstanceConnectEndpoint `
    -InstanceConnectEndpointId eice-0123456789example `
    -SecurityGroupIds sg-0123456789example `
    -IpAddressType dualstack `
    -PreserveClientIp $false
```

The following is example output.

```
True
```

To monitor the update status

During modification, the EC2 Instance Connect Endpoint status changes to `update-in-progress`. The update process runs asynchronously and completes with either an `update-`

complete or update-failed status. The endpoint uses its old configuration until the status changes to update-complete.

Use the Get-EC2InstanceConnectEndpoint command to monitor the update status. .State.Value filters the results to the State field.

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

The following is example output.

```
update-complete
```

Delete an EC2 Instance Connect Endpoint

When you are finished with an EC2 Instance Connect Endpoint, you can delete it.

You must have the required IAM permissions to create an EC2 Instance Connect Endpoint. For more information, see Permissions to create, describe, modify, and delete EC2 Instance Connect Endpoints.

When you delete an EC2 Instance Connect Endpoint using the console, it enters the **Deleting** state. If deletion is successful, the deleted endpoint no longer appears. If deletion fails, the state is delete-failed and **Status message** provides the failure reason.

When you delete an EC2 Instance Connect Endpoint using the AWS CLI, it enters the delete-in-progress state. If deletion is successful, it enters the delete-complete state. If deletion fails, the state is delete-failed and StateMessage provides the failure reason.

Console

To delete an EC2 Instance Connect Endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the left navigation pane, choose **Endpoints**.
3. Select the endpoint.
4. Choose **Actions**, **Delete VPC endpoints**.
5. When prompted for confirmation, enter **delete**.

6. Choose **Delete**.

AWS CLI

To delete an EC2 Instance Connect Endpoint

Use the delete-instance-connect-endpoint command and specify the ID of the EC2 Instance Connect Endpoint to delete.

```
aws ec2 delete-instance-connect-endpoint --instance-connect-endpoint-
id eice-03f5e49b83924bbc7
```

The following is example output.

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

PowerShell

To delete an EC2 Instance Connect Endpoint

Use the Remove-EC2InstanceConnectEndpoint cmdlet and specify the ID of the EC2 Instance Connect Endpoint to delete.

```
Remove-EC2InstanceConnectEndpoint -InstanceConnectEndpointId eice-03f5e49b83924bbc7
```

The following is example output.

```
@{
    InstanceConnectEndpoint = @{
        OwnerId = "111111111111"
        InstanceConnectEndpointId = "eice-0123456789example"
        InstanceConnectEndpointArn = "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/eice-0123456789example"
        State = "delete-in-progress"
        StateMessage = ""
        NetworkInterfaceIds = @()
        VpcId = "vpc-0123abcd"
        AvailabilityZone = "us-east-1d"
        CreatedAt = "2023-02-07T12:05:37+00:00"
        SubnetId = "subnet-0123abcd"
    }
}
```

Connect to an Amazon EC2 instance using EC2 Instance Connect Endpoint

You can use EC2 Instance Connect Endpoint to connect to an Amazon EC2 instance that supports SSH or RDP.

Prerequisites

- You must have the required IAM permission to connect to an EC2 Instance Connect Endpoint. For more information, see Permissions to use EC2 Instance Connect Endpoint to connect to instances.
- The EC2 Instance Connect Endpoint must be in the **Available** (console) or `create-complete` (AWS CLI) state. If you do not have an EC2 Instance Connect Endpoint for your VPC, you can create one. For more information, see Create an EC2 Instance Connect Endpoint.
- Your instance must have an IPv4 address (either private or public). EC2 Instance Connect Endpoint does not support connecting to instances using IPv6 addresses.
- (Linux instances) To use the Amazon EC2 console to connect to your instance, or to use the CLI to connect and have EC2 Instance Connect handle the ephemeral key, your instance must have EC2 Instance Connect installed. For more information, see Install EC2 Instance Connect.
- Ensure that the security group of the instance allows inbound SSH traffic from the EC2 Instance Connect Endpoint. For more information, see Target instance security group rules.

Connection options

Connect to your Linux instance using the Amazon EC2 console

You can connect to an instance using the Amazon EC2 console (a browser-based client) as follows.

To connect to your instance using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and then choose **Connect**.
4. Choose the **EC2 Instance Connect** tab.
5. Choose **Connect using a Private IP**.
6. For **EC2 Instance Connect Endpoint**, choose the ID of the EC2 Instance Connect Endpoint.
7. For **Username**, if the AMI that you used to launch the instance uses a username other than `ec2-user`, enter the correct username.
8. For **Max tunnel duration (seconds)**, enter the maximum allowed duration for the SSH connection.

   The duration must comply with any `maxTunnelDuration` condition specified in the IAM policy. If you don't have access to the IAM policy, contact your administrator.
9. Choose **Connect**. This opens a terminal window for your instance.

Connect to your Linux instance using SSH

You can use SSH to connect to your Linux instance, and use the `open-tunnel` command to establish a private tunnel. You can use `open-tunnel` in single connection or multi-connection mode.

For information about using the AWS CLI to connect to your instance using SSH, see Connect using the AWS CLI.

The following examples use OpenSSH. You can use any other SSH client that supports a proxy mode.

Single connection

To allow only a single connection to an instance using SSH and the open-tunnel command

Use `ssh` and the open-tunnel AWS CLI command as follows. The `-o` proxy command encloses the `open-tunnel` command that creates the private tunnel to the instance.

```
ssh -i my-key-pair.pem ec2-user@i-1234567890abcdef0 \
    -o ProxyCommand='aws ec2-instance-connect open-tunnel --instance-id i-1234567890abcdef0'
```

For:

- `-i` – Specify the key pair that was used to launch the instance.
- *`ec2-user`*`@`*`i-1234567890abcdef0`* – Specify the username of the AMI that was used to launch the instance, and the instance ID.
- `--instance-id` – Specify the ID of the instance to connect to. Alternatively, specify `%h`, which extracts the instance ID from the user.

Multi-connection

To allow multiple connections to an instance, first run the open-tunnel AWS CLI command to start listening for new TCP connections, and then use `ssh` to create a new TCP connection and a private tunnel to your instance.

To allow multiple connections to your instance using SSH and the open-tunnel command

1. Run the following command to start listening for new TCP connections on the specified port on your local machine.

   ```
   aws ec2-instance-connect open-tunnel \
       --instance-id i-1234567890abcdef0 \
       --local-port 8888
   ```

   Expected output

```
Listening for connections on port 8888.
```

2. In a *new terminal window*, run the following `ssh` command to create a new TCP connection and a private tunnel to your instance.

   ```
   ssh -i my-key-pair.pem ec2-user@localhost -p 8888
   ```

   Expected output – In the *first* terminal window, you'll see the following:

   ```
   [1] Accepted new tcp connection, opening websocket tunnel.
   ```

   You might also see the following:

   ```
   [1] Closing tcp connection.
   ```

Connect to your Linux instance using the AWS CLI

If you only know your instance ID, you can use the ec2-instance-connect ssh AWS CLI command to connect to your instance using an SSH client. For more information, see Connect using the AWS CLI.

Prerequisites

Install AWS CLI version 2 and configure it using your credentials. For more information, see Install or update to the latest version of the AWS CLI and Configure the AWS CLI in the *AWS Command Line Interface User Guide*. Alternatively, open AWS CloudShell and run AWS CLI commands in its pre-authenticated shell.

To connect to an instance using the instance ID and an EC2 Instance Connect Endpoint

If you only know the instance ID, use the ec2-instance-connect ssh CLI command, and specify the `ssh` command, the instance ID, and the `--connection-type` parameter with the `eice` value.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --os-user ec2-user --
connection-type eice
```

> **Tip**
>
> If you get an error when using this command, make sure that you're using AWS CLI version 2. The `ssh` parameter is only available in AWS CLI version 2. For more information, see About AWS CLI version 2 in the *AWS Command Line Interface User Guide*.

Connect to your Windows instance using RDP

You can use Remote Desktop Protocol (RDP) over EC2 Instance Connect Endpoint to connect to a Windows instance without a public IPv4 address or public DNS name.

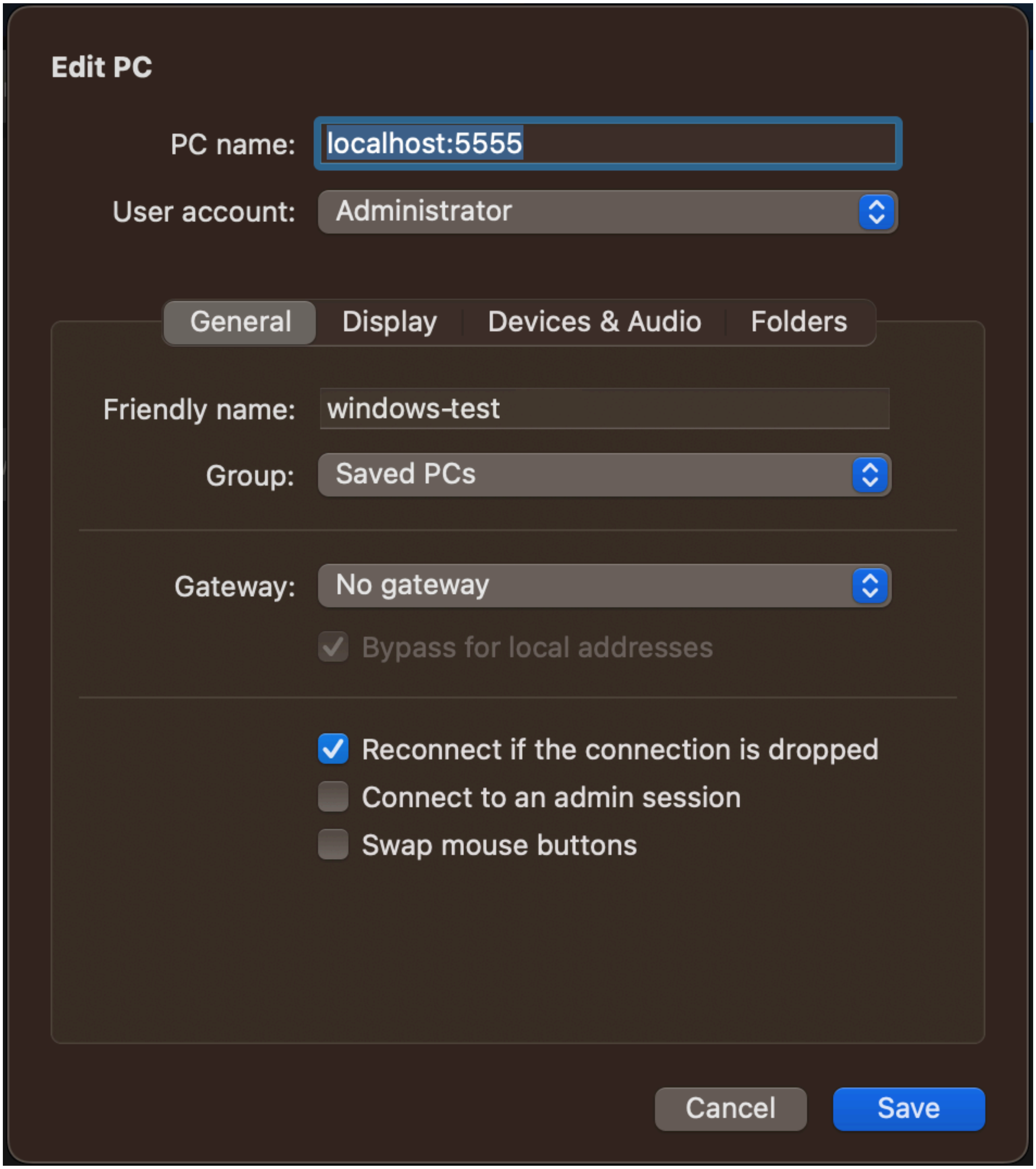
To connect to your Windows instance using an RDP client

1. Complete Steps 1 – 8 in Connect to your Windows instance using RDP. After downloading the RDP desktop file at Step 8, you'll get an **Unable to connect** message, which is to be expected because your instance does not have a public IP address.
2. Run the following command to establish a private tunnel to the VPC in which the instance is located. `--remote-port` must be 3389 because RDP uses port 3389 by default.

```
aws ec2-instance-connect open-tunnel \
    --instance-id i-1234567890abcdef0 \
    --remote-port 3389 \
    --local-port any-port
```

3. In your **Downloads** folder, find the RDP desktop file that you downloaded, and drag it onto the RDP client window.
4. Right-click the RDP desktop file and choose **Edit**.
5. In the **Edit PC** window, for **PC name** (the instance to connect to), enter `localhost:local-port`, where *`local-port`* uses the same value as you specified in Step 2, and then choose **Save**.

   Note that the following screenshot of the **Edit PC** window is from Microsoft Remote Desktop on a Mac. If you are using a Windows client, the window might be different.



6. In the RDP client, right-click the PC (that you just configured) and choose **Connect** to connect to your instance.

7. At the prompt, enter the decrypted password for the administrator account.

Troubleshoot

Use the following information to help diagnose and fix issues that you might encounter when using EC2 Instance Connect Endpoint to connect an instance.

Can't connect to your instance

The following are common reasons why you might not be able to connect to your instance.

- Security groups – Check the security groups assigned to the EC2 Instance Connect Endpoint and your instance. For more information about the required security group rules, see Security groups for EC2 Instance Connect Endpoint.
- Instance state – Verify that your instance is in the `running` state.
- Key pair – If the command you're using to connect requires a private key, verify that your instance has a public key and that you have the corresponding private key.
- IAM permissions – Verify that you have the required IAM permissions. For more information, see Grant permissions to use EC2 Instance Connect Endpoint.

For more troubleshooting tips for Linux instances, see Troubleshoot issues connecting to your Amazon EC2 Linux instance. For troubleshooting tips for Windows instances, see the section called "Windows instance RDP issues".

ErrorCode: AccessDeniedException

If you receive an `AccessDeniedException` error, and the `maxTunnelDuration` condition is specified in the IAM policy, be sure to specify the `--max-tunnel-duration` parameter when connecting to an instance. For more information about this parameter, see open-tunnel in the *AWS CLI Command Reference*.

Log connections established over EC2 Instance Connect Endpoint

You can log resource operations and audit connections established over the EC2 Instance Connect Endpoint with AWS CloudTrail logs.

For more information about using AWS CloudTrail with Amazon EC2, see Log Amazon EC2 API calls using AWS CloudTrail.

Log EC2 Instance Connect Endpoint API calls with AWS CloudTrail

EC2 Instance Connect Endpoint resource operations are logged to CloudTrail as management events. When the following API calls are made, the activity is recorded as a CloudTrail event in **Event history**:

- `CreateInstanceConnectEndpoint`
- `DescribeInstanceConnectEndpoints`
- `DeleteInstanceConnectEndpoint`

You can view, search, and download recent events in your AWS account. For more information, see [Viewing events with CloudTrail Event history] in the *AWS CloudTrail User Guide*.

Use AWS CloudTrail to audit users who connect to an instance using EC2 Instance Connect Endpoint

Connection attempts to instances via EC2 Instance Connect Endpoint are logged in CloudTrail in **Event history**. When a connection to an instance is initiated through an EC2 Instance Connect Endpoint, the connection is logged as a CloudTrail management event with the `eventName` of `OpenTunnel`.

You can create Amazon EventBridge rules that route the CloudTrail event to a target. For more information, see the [Amazon EventBridge User Guide].

The following is an example of an `OpenTunnel` management event that was logged in CloudTrail.

```
{
     "eventVersion": "1.08",
     "userIdentity": {
         "type": "IAMUser",
         "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
         "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
         "accountId": "123456789012",
         "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
         "userName": "IAM-friendly-name"
     },
     "eventTime": "2023-04-11T23:50:40Z",
     "eventSource": "ec2-instance-connect.amazonaws.com",
     "eventName": "OpenTunnel",
     "awsRegion": "us-east-1",
     "sourceIPAddress": "1.2.3.4",
```

```
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceConnectEndpointId": "eici-0123456789EXAMPLE",
        "maxTunnelDuration": "3600",
        "remotePort": "22",
        "privateIpAddress": "10.0.1.1"
    },
    "responseElements": null,
    "requestID": "98deb2c6-3b3a-437c-a680-03c4207b6650",
    "eventID": "bbba272c-8777-43ad-91f6-c4ab1c7f96fd",
    "readOnly": false,
    "resources": [{
        "accountId": "123456789012",
        "type": "AWS::EC2::InstanceConnectEndpoint",
        "ARN": "arn:aws:ec2:us-east-1:123456789012:instance-connect-endpoint/
eici-0123456789EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

Service-linked role for EC2 Instance Connect Endpoint

Amazon EC2 uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon EC2. Service-linked roles are predefined by Amazon EC2 and include all the permissions that Amazon EC2 requires to call other AWS services on your behalf. For more information, see Service-linked roles in the *IAM User Guide*.

Service-linked role permissions for EC2 Instance Connect Endpoint

Amazon EC2 uses **AWSServiceRoleForEC2InstanceConnect** to create and manage network interfaces in your account that are required by EC2 Instance Connect Endpoint.

The **AWSServiceRoleForEC2InstanceConnect** service-linked role trusts the following service to assume the role:

- `ec2-instance-connect.amazonaws.com`

The **AWSServiceRoleForEC2InstanceConnect** service-linked role uses the following managed policy:

- **Ec2InstanceConnectEndpoint**

To view the permissions for the managed policy, see Ec2InstanceConnectEndpoint in the *AWS Managed Policy Reference*.

Create a service-linked role for EC2 Instance Connect Endpoint

You don't need to manually create this service-linked role. When you create an EC2 Instance Connect Endpoint, Amazon EC2 creates the service-linked role for you.

Edit a service-linked role for EC2 Instance Connect Endpoint

EC2 Instance Connect Endpoint doesn't allow you to edit the **AWSServiceRoleForEC2InstanceConnect** service-linked role.

Delete a service-linked role for EC2 Instance Connect Endpoint

If you no longer need to use EC2 Instance Connect Endpoint, we recommend that you delete the **AWSServiceRoleForEC2InstanceConnect** service-linked role.

You must delete all EC2 Instance Connect Endpoint resources before you can delete the service-linked role.

To delete the service-linked role, see Delete a service-linked role in the *IAM User Guide*.

You must configure permissions to allow an IAM entity (a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Quotas for EC2 Instance Connect Endpoint

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific.

Your AWS account has the following quotas related to EC2 Instance Connect Endpoint.

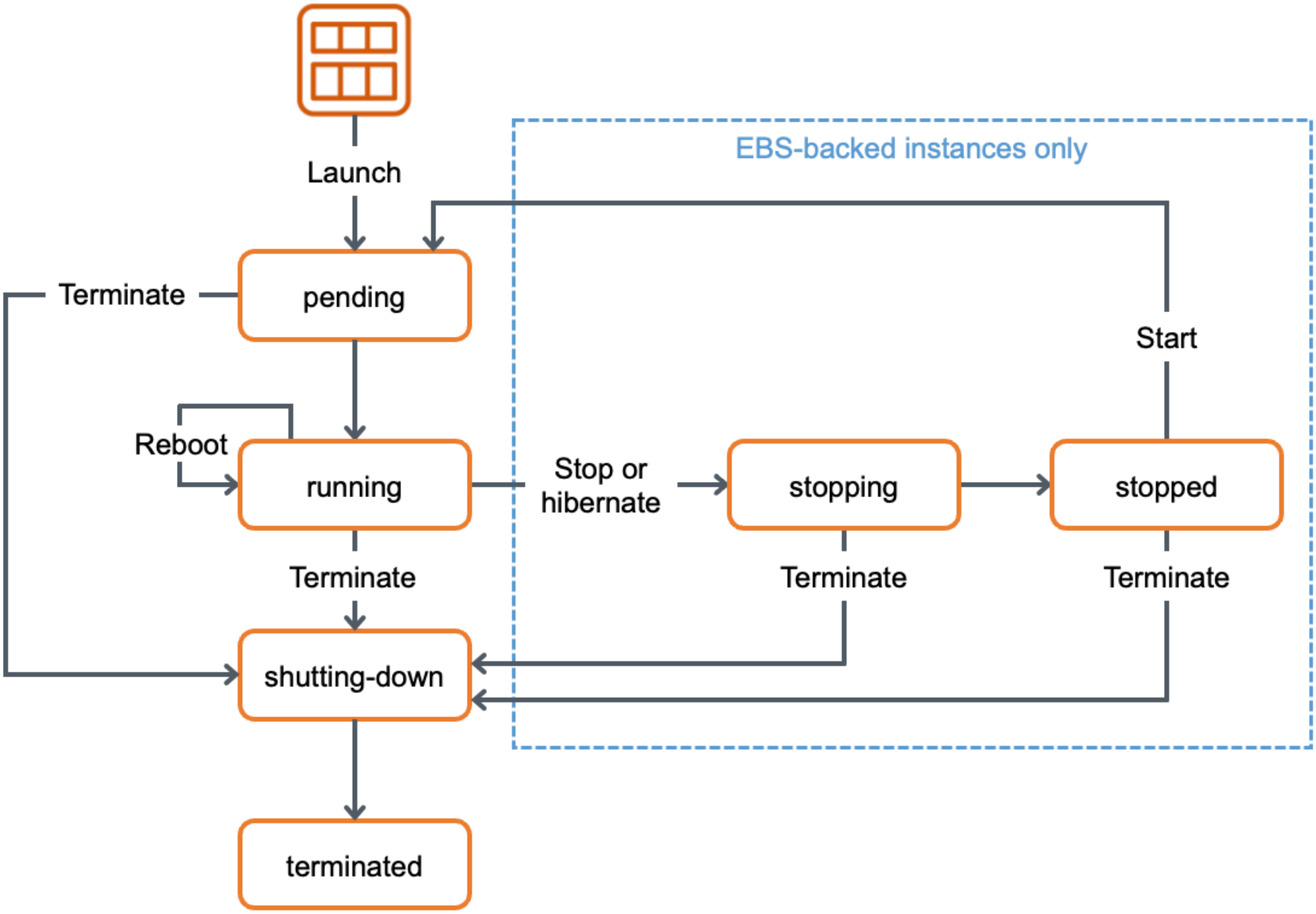
Name	Default	Adjustable
Maximum number of EC2 Instance Connect Endpoints per AWS account per AWS Region	5	No

Name	Default	Adjustable
Maximum number of EC2 Instance Connect Endpoints per VPC	1	No
Maximum number of EC2 Instance Connect Endpoints per subnet	1	No
Maximum number of concurrent connections per EC2 Instance Connect Endpoint	20	No

Amazon EC2 instance state changes

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.



You can receive notifications when your instances change states. For more information, see [the section called “State change events”](#).

Billing by instance state

The following table provides a brief description of each instance state and indicates whether instance usage is billed. 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 User Guide*.

Instance state	Description	Instance usage billing
`pending`	The instance is preparing to enter the `running`	Not billed

Instance state	Description	Instance usage billing
	state. An instance enters the `pending` state when it is launched or when it is started after being in the `stopped` state.	
`running`	The instance is running and ready for use.	Billed
`stopping`	The instance is preparing to be stopped.	Not billed **Note** If you hibernate an instance, you're billed while the instance is in the `stopping` state.
`stopped`	The instance is shut down and cannot be used. The instance can be started at any time.	Not billed
`shutting-down`	The instance is preparing to be terminated.	Not billed
`terminated`	The instance has been permanently deleted and cannot be started.	Not billed **Note** Reserved Instances that applied to terminated instances are billed until the end of their term according to their payment option. For more information, see [Reserved Instances for Amazon EC2 overview](#)

Pending instances

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.

Stopped instances

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. You are not charged for usage or data transfer fees for your instance when it is `stopped`. Charges are incurred for the storage of 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 the instance is moved 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 attached to 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 remains associated with your instance. If your instance has an IPv6 address, it retains the IPv6 address.

Each time you transition an instance from `stopped` to `running`, you are charged per second when the instance is running, with a minimum of one minute per instance start.

For more details about stopping and starting an instance, see Stop and start Amazon EC2 instances.

Hibernated instances

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 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 Amazon EC2 instance.

Rebooting instances

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 Amazon EC2 instance.

Terminated instances

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

If you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You can also describe a terminated instance using the CLI and API. Resources (such as tags) are gradually disassociated from the terminated instance, therefore may no longer be visible on the terminated instance after a short while. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate shutdown from within the instance itself (for example, by using the **shutdown** command on Linux). The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see Terminate Amazon EC2 instances.

Differences between instance states

The following table summarizes the key differences between rebooting, stopping, hibernating, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer.	We move the instance to a new host computer (though in some cases, it remains on the current host).	We move the instance to a new host computer (though in some cases, it remains on the current host).	None

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
Private IPv4 address	The instance keeps its private IPv4 address.	The instance keeps its private IPv4 address.	The instance keeps its private IPv4 address.	None
Public IPv4 address	The instance keeps its public IPv4 address.	The instance gets a new public IPv4 address, unless it has a secondary network interface or a secondary private IPv4 address that is associated with an Elastic IP address.	The instance gets a new public IPv4 address, unless it has a secondary network interface or a secondary private IPv4 address that is associated with an Elastic IP address.	None
Elastic IP address (IPv4)	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address is disassociated from the instance
IPv6 address	The instance keeps its IPv6 address	The instance keeps its IPv6 address	The instance keeps its IPv6 address	None
Instance store volumes	The data is preserved	The data is erased	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is preserved	The volume is deleted by default

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
RAM (contents of memory)	The RAM is erased	The RAM is erased	The RAM is saved to a file on the root volume	The RAM is erased
Billing	The instance billing hour doesn't change	You stop incurring charges for an instance as soon as its state changes to `stopping`. Each time an instance transitions from `stopped` to `running`, we start a new instance billing period, billing a minimum of one minute every time you start your instance.	You incur charges while the instance is in the `stopping` state, but stop incurring charges when the instance is in the `stopped` state. Each time an instance transitions from `stopped` to `running`, we start a new instance billing period, billing a minimum of one minute every time you start your instance.	You stop incurring charges for an instance as soon as its state changes to `shutting-down`

Operating system shutdown commands always terminate an instance store-backed instance. You can control whether operating system shutdown commands stop or terminate an Amazon EBS-backed instance. For more information, see Change instance initiated shutdown behavior.

Stop and start Amazon EC2 instances

You can stop and start your instance if it has an Amazon EBS volume as its root device. When you stop an instance, it shuts down. When you start an instance, it is typically migrated to a new underlying host computer and assigned a new public IPv4 address.

An instance stop can be user-initiated (where you manually stop the instance) or initiated by AWS (in response to a scheduled stop event when AWS detects irreparable failure of the underlying host for your instance).

For user-initiated stops, we recommend using the Amazon EC2 console, CLI, or API instead of running the operating system stop command from your instance. When using Amazon EC2, if the instance does not cleanly shut down within a few minutes, Amazon EC2 performs a hard shut down. Furthermore, AWS CloudTrail creates an API record of when your instance was stopped.

This topic describes how to perform a user-initiated stop. For information about a stop performed by AWS, see Manage Amazon EC2 instances scheduled to stop or retire.

When you stop an instance, it is not deleted. If you decide that you no longer need an instance, you can terminate it. For more information, see Terminate Amazon EC2 instances. If you want to hibernate an instance to save the contents from the instance memory (RAM), see Hibernate your Amazon EC2 instance. For distinctions between instance lifecycle actions, see Differences between instance states.

Contents

How EC2 instance stop and start works

When you stop an Amazon EC2 instance, changes are registered at the operating system (OS) level of the instance, some resources are lost, and some resources persist. When you start an instance, changes are registered at the instance level.

Topics

- Test application response to stop and start
- Costs related to instance stop and start

What happens when you stop an instance

The following describes what typically happens when you stop an instance using the default stop method. Note that some aspects might vary depending on which stop method you use.

Changes 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 OS shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
- ACPI shutdown is initiated.
- The instance shuts down when the graceful OS shutdown process exits. There is no configurable OS shutdown time.
- If the instance OS does not cleanly shut down within a few minutes, a hard shutdown is performed.
- The instance stops running.
- The instance state changes to `stopping` and then `stopped`.
- [Auto Scaling] If your instance is in an Auto Scaling group, when the instance is in any Amazon EC2 state other than `running`, or if its status for the status checks becomes `impaired`, Amazon EC2 Auto Scaling considers the instance to be unhealthy and replaces it. For more information, see Health checks for instances in an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.
- [Windows instances] When you stop and start a Windows instance, the launch agent 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 the section called "EC2Launch v2".

Resources lost

- Data stored on the RAM.
- Data stored on the instance store volumes.

- The public IPv4 address that Amazon EC2 automatically assigned to the instance upon launch or start. To retain a public IPv4 address that never changes, you can associate an Elastic IP address with your instance.

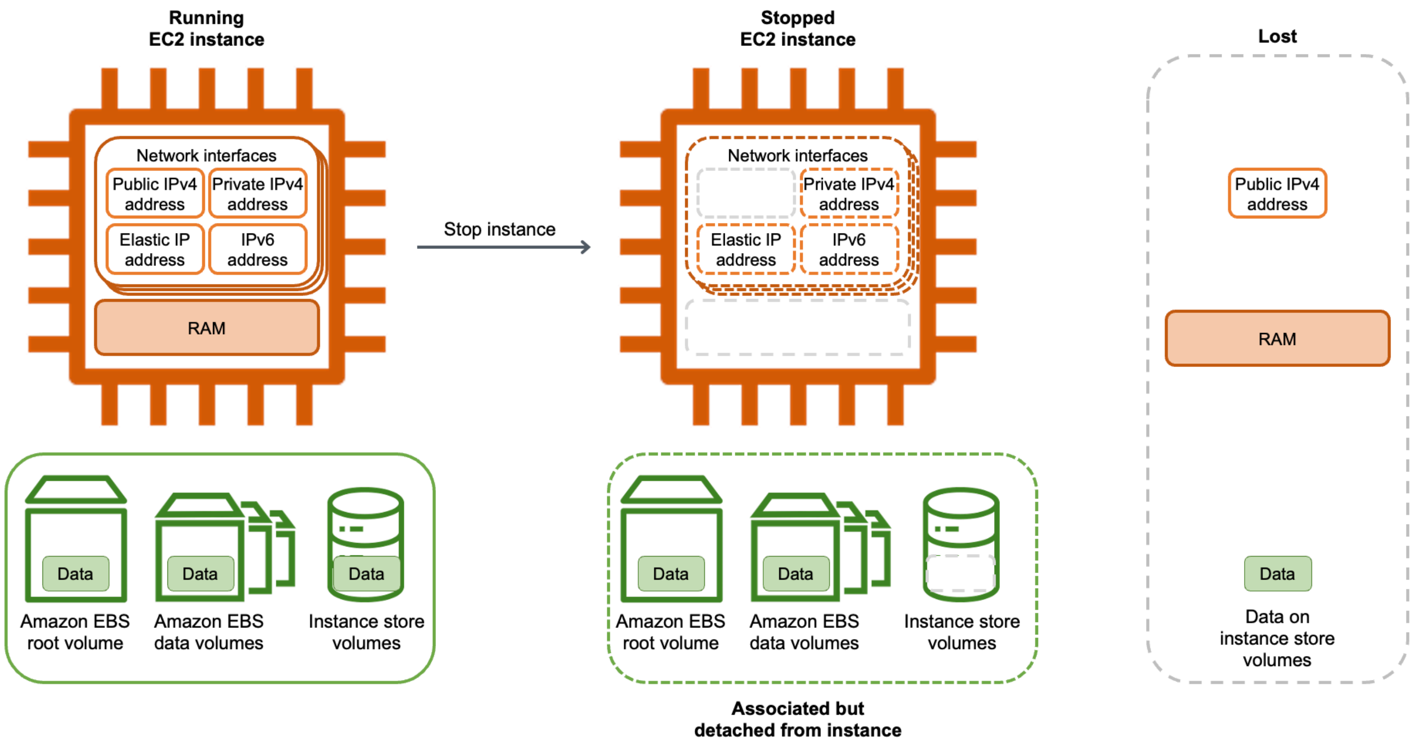
Resources that persist

- Any attached Amazon EBS root and data volumes.
- Data stored on the Amazon EBS volumes.
- Any attached network interfaces.

  A network interface includes the following resources, which also persist:

  - Private IPv4 addresses.
  - IPv6 addresses.
  - Elastic IP addresses associated with the instance. Note that when the instance is stopped, you are charged for the associated Elastic IP addresses.

The following diagram illustrates what persists and what is lost when an EC2 instance is stopped. The diagram is divided into three parts: the first part, labeled **Running EC2 instance**, shows the instance in the `running` state with its resources. The second part, labeled **Stopped EC2 instance**, shows the instance in the `stopped` state with the resources that persist. The third part, labeled **Lost**, shows the resources that are lost when the instance is stopped.



For information about what happens when you stop a Mac instance, see [Stop or terminate your Amazon EC2 Mac instance](#).

What happens when you start an instance

- In most cases, the instance is migrated to a new underlying host computer (though in some cases, such as when an instance is allocated to a host in a [Dedicated Host](#) configuration, it remains on the current host).
- The associated EBS volumes and network interfaces are reattached to the instance.
- Amazon EC2 assigns a new public IPv4 address to the instance if the instance is configured to receive a public IPv4 address, unless it has a secondary network interface or a secondary private IPv4 address that is associated with an Elastic IP address.

Test application response to stop and start

You can use AWS Fault Injection Service to test how your application responds when your instance is stopped and started. For more information, see the [AWS Fault Injection Service User Guide](#).

Costs related to instance stop and start

The following costs are associated with stopping and starting an instance.

Stopping – As soon as the state of an instance changes to `shutting-down` or `terminated`, charges are no longer incurred for the instance. You are not charged for usage or data transfer fees for a stopped instance. Charges are incurred to store Amazon EBS storage volumes.

Starting – Each time you start a stopped instance, you are charged for a minimum of one minute of usage. After one minute, you are charged for only the seconds you use. For example, if you run an instance for 20 seconds and then stop it, you are charged for a minute of usage. If you run an instance for 3 minutes and 40 seconds, you are charged for 3 minutes and 40 seconds of usage.

Methods for stopping an instance

There are four ways to perform a user-initiated stop: default stop, stop with skip OS shutdown, force stop, and force stop with skip OS shutdown. The following table compares the key differences between the stop methods:

Stop method	Key purpose	Use case	CLI command
Default stop	Normal instance shutdown with attempted graceful OS shutdown.	Typical instance stop.	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0`
Stop with skip OS shutdow	Bypasses the graceful OS shutdown when stopping an instance.	When bypassing graceful OS shutdown is required.	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --skip-os-shutdown`
Force stop	Handles stuck instances. Attempts a default stop first; if instance fails to stop, then forcibly stops the instance.	When instance is stuck in `stopping` state.	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \`

Stop method	Key purpose	Use case	CLI command
			`--force`
Force stop with skip OS shutdown	Force stops and bypasses the graceful OS shutdown when stopping an instance.	When force stop and bypassing graceful OS shutdown is required.	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --force \ --skip-os-shutdown`

For instructions on how to use each method, see the following:

- Stop an instance with a graceful OS shutdown
- Stop an instance and bypass the graceful OS shutdown
- Force stop an instance

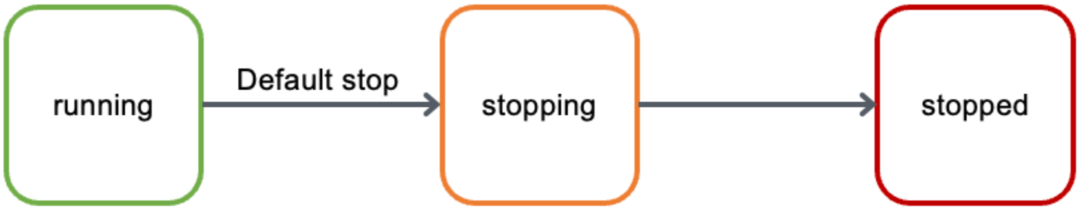
Stop methods:

- Default stop
- Stop with skip OS shutdown
- Force stop
- Force stop with skip OS shutdown

The following sections provide more detailed information about the four different user-initiated stop methods.

Default stop

The default stop method is the standard way to stop an instance. When you issue the StopInstances command, the instance transitions from the `running` state, to `stopping`, and finally to `stopped`, as illustrated by the following diagram:



Purpose: Normal instance shutdown with attempted graceful OS shutdown.

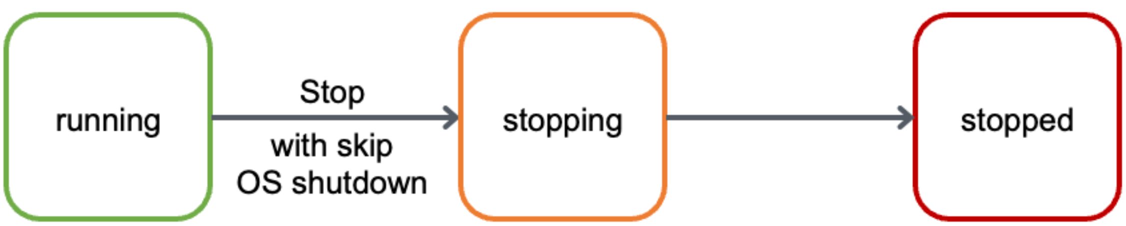
Data impact: Preserves data on the EBS root volume and data volumes. Loses data on the instance store volume.

When to use: First stop attempt for typical stops.

> **Note**
>
> If you've already attempted a stop with skip OS shutdown, a subsequent default stop attempt during the same state transition session will not perform a graceful OS shutdown. Bypassing the graceful OS shutdown is irreversible for the instance's current session.

Stop with skip OS shutdown

When bypassing the graceful OS shutdown is required, the stop with skip OS shutdown method can be used to stop an instance and bypass the graceful OS shutdown, as illustrated by the following diagram:



> **Warning**
>
> Bypassing the graceful OS shutdown might result in data loss or corruption (for example, memory contents not flushed to disk or loss of in-flight IOs) or skipped shutdown scripts.

Purpose: Bypass the graceful OS shutdown when stopping an instance.

Data impact: Might result in data loss or corruption. Contents of memory might not be flushed to disk and in-flight IOs might be lost. Might skip shutdown scripts.

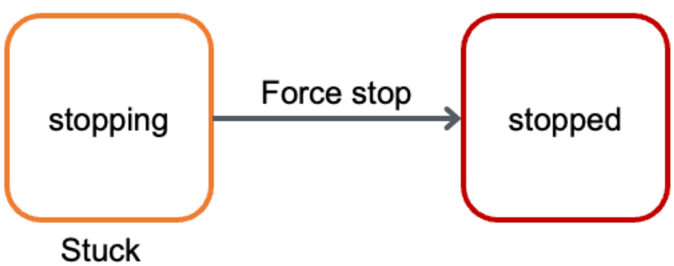
When to use: When bypassing the graceful OS shutdown is required. If used while a default stop with graceful OS shutdown is in progress, the graceful OS shutdown will be bypassed.

> **Note**
>
> Bypassing the graceful OS shutdown is irreversible for the instance's current state transition session. A subsequent default stop attempt during this session will not attempt a graceful OS shutdown.

Force stop

The force stop method is used to handle instances that are stuck in the `stopping` state. An instance typically becomes stuck due to an underlying hardware issue (indicated by a failed [system status check](#)).

The force stop method first attempts a default stop. If the instance remains stuck in the `stopping` state, the `force` parameter forcibly shuts down the instance and transitions the instance to the `stopped` state, as indicated by the following diagram:



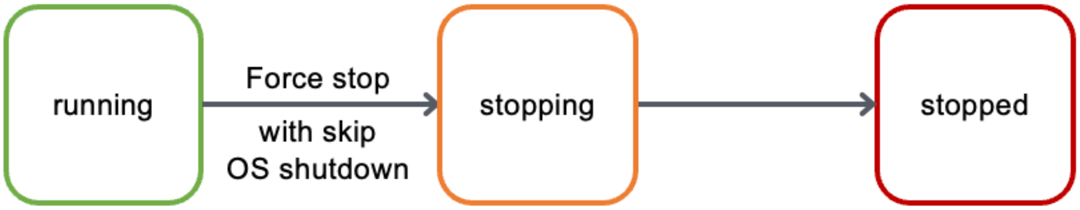
Purpose: Handles instances stuck in the `stopping` state. Attempts a default stop first. If the instance fails to stop, then forcibly shuts down the instance.

Data impact: Attempts a default stop first, but if force stop goes ahead, then might cause data loss or corruption. In rare cases, results in post-stop writes to EBS volumes or other shared resources.

When to use: Second stop attempt when an instance remains stuck after a default stop. For more information, see [Troubleshoot Amazon EC2 instance stop issues](#).

Force stop with skip OS shutdown

When force stopping and bypassing the graceful OS shutdown is required, the force stop with skip OS shutdown method can be used to bring an instance to the `stopped` state, as illustrated in the following diagram:



Purpose: Combines force stop with bypassing a graceful OS shutdown when stopping an instance.

Data impact: Skip OS shutdown might result in data loss or corruption. Contents of memory might not be flushed to disk and in-flight IOs might be lost. Might skip shutdown scripts. If force stop goes ahead, then might cause additional data loss or corruption. In rare cases, results in post-stop writes to the EBS volumes or other shared resources.

When to use: When you want to be sure that your instance will stop and you want to bypass the graceful OS shutdown. If used while a default stop with graceful OS shutdown is in progress, the graceful OS shutdown will be bypassed.

Manually stop and start your instances

You can stop and start your Amazon EBS-backed instances (instances with EBS root volumes). You can't stop and start instances with an instance store root volume.

When using the default method to stop an instance, a graceful operating system (OS) shutdown is attempted. You can bypass the graceful OS shutdown; however, this might risk data integrity.

> **Warning**
>
> 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 the instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

[Linux instances] Using the OS **halt** command from an instance does not initiate a shutdown. If you use the **halt** command, the instance does not terminate; instead, it places the CPU into HLT, which suspends CPU operation. The instance remains running.

You can initiate a shutdown using the OS **shutdown** or **poweroff** commands. When you use an OS command, the instance stops by default. You can change this behavior. For more information, see Change instance initiated shutdown behavior.

> **Note**
>
> If you stopped an Amazon EBS-backed instance and it appears "stuck" in the `stopping` state, you can forcibly stop it. For more information, see Troubleshoot Amazon EC2 instance stop issues.

Stop and start methods

- Stop an instance with a graceful OS shutdown
- Stop an instance and bypass the graceful OS shutdown
- Start an instance

Stop an instance with a graceful OS shutdown

You can stop an instance using the default stop method, which includes an attempt at a graceful OS shutdown. For more information, see Default stop.

Console

To stop an instance using the default stop method

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**, and then 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.

AWS CLI

To stop an instance using the default stop method

Use the stop-instances command.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To stop an instance using the default stop method

Use the Stop-EC2Instance cmdlet

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

Stop an instance and bypass the graceful OS shutdown

You can bypass the graceful OS shutdown when stopping an instance. For more information, see Stop with skip OS shutdown.

Warning

Bypassing the graceful OS shutdown might result in data loss or corruption (for example, memory contents not flushed to disk or loss of in-flight IOs) or skipped shutdown scripts.

Console

To stop an instance and bypass the graceful OS shutdown

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 **Instance state**, **Stop instance**.
4. Under **Skip OS shutdown**, select the **Skip OS shutdown** checkbox. If you don't see this option in the console, it's not yet available in the console in the current Region. You can, however, access this feature using the AWS CLI or SDK, or try another Region in the console.
5. Choose **Stop**.

AWS CLI

To stop an instance and bypass the graceful OS shutdown

Use the stop-instances command with `--skip-os-shutdown`.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --skip-os-shutdown
```

PowerShell

To stop an instance and bypass the graceful OS shutdown

Use the [Stop-EC2Instance](#) cmdlet with `-SkipOsShutdown $true`.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -SkipOsShutdown $true
```

Start an instance

You can start a stopped instance.

Console

To start an 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, and choose **Instance state**, **Start instance**.

   It can take a few minutes for the instance to enter the `running` state.

AWS CLI

To start an instance

Use the start-instances command.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To start an instance

Use the Start-EC2Instance cmdlet.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

Automatically stop and start your instances

You can automate stopping and starting instances with the following services:

Instance Scheduler on AWS

You can use Instance Scheduler on AWS to automate the starting and stopping of EC2 instances. For more information, see How do I use Instance Scheduler with CloudFormation to schedule EC2 instances? Note that additional charges apply.

AWS Lambda and an Amazon EventBridge rule

You can use Lambda and an EventBridge rule to stop and start your instances on a schedule. For more information, see How do I use Lambda to stop and start Amazon EC2 instances at regular intervals?

Amazon EC2 Auto Scaling

To ensure you have the correct number of Amazon EC2 instances available to handle the load for an application, create Auto Scaling groups. Amazon EC2 Auto Scaling ensures that your application always has the right capacity to handle the traffic demand, and saves costs by launching instances only when they are needed. Note that Amazon EC2 Auto Scaling terminates, rather than stops, unneeded instances. To set up Auto Scaling groups, see Get started with Amazon EC2 Auto Scaling.

Find all running and stopped instances

You can find all of your running and stopped instances across all AWS Regions on a single page using Amazon EC2 Global View. This capability is especially useful for taking inventory and finding forgotten instances. For information about how to use Global View, see View resources across Regions using AWS Global View.

Alternatively, you can run a command or cmdlet in each Region where you have instances.

AWS CLI

To get the number of EC2 instances in a Region

Use the following describe-instances command to count the instances in the current Region. You must run this command in each Region where you have instances.

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "length(Reservations[].Instances[])"
```

The following is example output.

```
27
```

To get summary info about your EC2 instances in a Region

Use the following describe-instances command. You must run this command in each Region where you have instances.

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "Reservations[].Instances[].[InstanceId,InstanceType,PrivateIpAddress]"
 \
    --output table
```

The following is example output.

```
---------------------------------------------------------
|                    DescribeInstances                  |
+----------------------+----------------+-----------------+
|  i-0e3e777f4362f1bf7|  t2.micro     |  10.0.12.9      |
|  i-09453945dcf1529e9|  t2.micro     |  10.0.143.213   |
|  i-08fd74f3f1595fdbd|  m7i.4xlarge  |  10.0.1.103     |
+----------------------+----------------+-----------------+
```

PowerShell

To get the number of EC2 instances in a Region

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -Region us-east-2).Instances.Length
```

The following is example output.

```
27
```

To get summary info about your EC2 instances in a Region

Use the Get-EC2Instance cmdlet. You must run this command in each Region where you have instances.

```
(Get-EC2Instance).Instances | Select InstanceId, InstanceType, PrivateIpAddress
```

The following is example output.

```
InstanceId          InstanceType PrivateIpAddress
----------          ------------ ----------------
i-0e3e777f4362f1bf7 t2.micro     10.0.12.9
i-09453945dcf1529e9 t2.micro     10.0.143.213
i-08fd74f3f1595fdbd m7i.4xlarge  10.0.1.103
```

Find the initial and most recent launch times

When you describe an instance, the launch time for the instance is its most recent launch time. After you stop and start an instance, the launch time reflects the new instance start time. To find the initial launch time for an instance, even after stopping and starting it, view the time at which the primary network interface was attached to the instance.

Console

To find the most recent launch time

Select the instance and find **Launch time** under **Instance details** on the **Details** tab.

To find the initial launch time

Select the instance and find the primary network interface (device index is 0) under **Network interfaces** on the **Networking** tab.

AWS CLI

To find the initial and most recent launch times

Use the following [describe-instances](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instances.html) command to display both the initial launch time and the most recent launch time for the specified instance.

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query 'Reservations[].Instances[].
{InstanceID:InstanceId,InitialLaunch:NetworkInterfaces[0].Attachment.AttachTime,LastLaunch:L
```

The following is example output.

```
[
    {
        "InstanceID": "i-1234567890abcdef0",
        "InitialLaunch": "2024-04-19T00:47:08+00:00",
        "LastLaunch": "2024-05-27T06:24:06+00:00"
    }
]
```

PowerShell

To find the most recent launch time

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.LaunchTime
```

The following is example output.

```
Monday, May 27, 2024 6:24:06 AM
```

To find the initial launch time

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -
InstanceId i-1234567890abcdef0).Instances.NetworkInterfaces.Attachment.AttachTime
```

The following is example output.

```
Friday, April 19, 2024 12:47:08 AM
```

Enable stop protection for your EC2 instances

To prevent an instance from being accidentally stopped, you can enable stop protection for the instance. Stop protection also protects your instance from accidental termination.

The `DisableApiStop` attribute of the Amazon EC2 [`ModifyInstanceAttribute`] API controls whether the instance can be stopped by using the Amazon EC2 console, the AWS CLI, or the Amazon EC2 API. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped.

Considerations

- Enabling stop protection does not prevent you from accidentally stopping an instance by initiating a shutdown from the instance using an operating system command such as **shutdown** or **poweroff**.
- Enabling stop protection does not prevent AWS from stopping the instance when there is a scheduled event to stop the instance.
- Enabling stop protection does not prevent Amazon EC2 Auto Scaling from terminating an instance when the instance is unhealthy or during scale-in events. You can control whether an Auto Scaling group can terminate a particular instance when scaling in by using instance scale-in protection.
- Stop protection not only prevents your instance from being accidentally stopped, but also from accidental termination when using the console, AWS CLI, or API. However, it does not automatically set the `DisableApiTermination` attribute. Note that when the `DisableApiStop` attribute is set to `false`, the `DisableApiTermination` attribute setting determines whether the instance can be terminated using the console, AWS CLI, or API. For more information see Terminate Amazon EC2 instances.
- You can't enable stop protection for instance store-backed instances.
- You can't enable stop protection for Spot Instances.
- The Amazon EC2 API follows an eventual consistency model when you enable or disable stop protection. This means that the result of running commands to set the stop protection attribute might not be immediately visible to all subsequent commands you run. For more information, see Eventual consistency in the *Amazon EC2 Developer Guide*.

Stop protection tasks

- Enable stop protection for an instance at launch

- Enable stop protection for a running or stopped instance
- Disable stop protection for a running or stopped instance

Enable stop protection for an instance at launch

You can enable stop protection for an instance when launching the instance.

Console

To enable stop protection for an instance at launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the dashboard, choose **Launch instance**.
3. Configure your instance in the new launch instance wizard.
4. In the wizard, enable stop protection by choosing **Enable** for **Stop protection** under **Advanced details**.

AWS CLI

To enable stop protection for an instance at launch

Use the run-instances command to launch the instance. Add the following parameter.

```
--disable-api-stop
```

PowerShell

To enable stop protection for an instance at launch

Use the New-EC2Instance cmdlet to launch the instance. Add the following parameter.

```
-DisableApiStop $true
```

Enable stop protection for a running or stopped instance

You can enable stop protection for an instance while the instance is running or stopped.

Console

To enable stop protection for an 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, and then choose **Actions**>**Instance settings**>**Change stop protection**.
4. Select the **Enable** checkbox, and then choose **Save**.

AWS CLI

To enable stop protection for an instance

Use the modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-stop
```

PowerShell

To enable stop protection for an instance

Use the Edit-EC2InstanceAttribute cmdlet.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $true
```

Disable stop protection for a running or stopped instance

You can disable stop protection for a running or stopped instance using one of the following methods.

Console

To disable stop protection for a running or stopped 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, and then choose **Actions**, **Instance settings**, **Change stop protection**.
4. Clear the **Enable** checkbox, and then choose **Save**.

AWS CLI

To disable stop protection for a running or stopped instance

Use the modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-stop
```

PowerShell

To disable stop protection for an instance

Use the Edit-EC2InstanceAttribute cmdlet.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $false
```

Hibernate your Amazon EC2 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 your instance is started:

- 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 and it meets the hibernation prerequisites.

If an instance or application takes a long time to bootstrap and build a memory footprint in order 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 so that it's ready to be resumed to the desired state whenever needed.

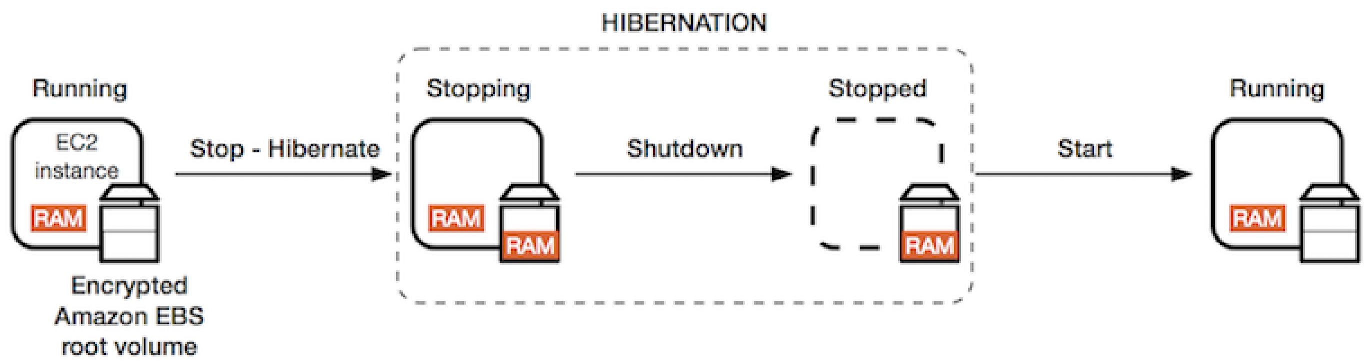
You're not charged for instance usage for a hibernated instance when it is in the `stopped` state or for data transfer when the contents of the RAM are transferred to the EBS root volume. You are charged for storage of any EBS volumes, including storage for the RAM contents.

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 Amazon EC2 instances.

Contents

How Amazon EC2 instance hibernation works

The following diagram shows a basic overview of the hibernation process for EC2 instances.



What happens when you hibernate an instance

When you hibernate an instance, the following happens:

- 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 the instance is started, 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 the instance is started, the instance continues to retain its private IPv4 addresses and any IPv6 addresses.
- Amazon EC2 releases the public IPv4 address. When the instance is started, 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 that are associated with a hibernated instance.

For information about how hibernation differs from reboot, stop, and terminate, see [Differences between instance states](#).

Limitations

- When you hibernate an instance, the data on any instance store volumes is lost.
- (Linux instances) You can't hibernate a Linux instance that has more than 150 GiB of RAM.
- (Windows instances) You can't hibernate a Windows instance that has more than 16 GiB of RAM.
- While your instance is hibernated, you can't modify it. This is different to a stopped instance that isn't hibernated, where you can modify certain attributes, such as the instance type or size.
- 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 a new instance that is launched from the AMI or from an AMI that was created from the snapshot.
- (Spot Instances only) If Amazon EC2 hibernates your Spot Instance, only Amazon EC2 can resume your instance. If you hibernate your Spot Instance (user-initiated hibernation), you can resume your instance. A hibernated Spot Instance can only be resumed if capacity is available and the Spot price is less than or equal to your specified maximum price.
- 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 instances in an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.
- You can't hibernate an instance that is configured to boot in UEFI mode with UEFI Secure Boot enabled.
- 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.
- You can’t hibernate an instance that uses a kernel below 5.10 if Federal Information Processing Standard (FIPS) mode is enabled.
- 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.

Considerations for hibernating a Spot Instance

- If *you* hibernate your Spot Instance, you can restart it provided capacity is available and the Spot price is less than or equal to your specified maximum price.
- If *Amazon EC2* hibernates your Spot Instance:
  - Only Amazon EC2 can resume your instance.
  - Amazon EC2 resumes the hibernated Spot Instance when capacity becomes available with a Spot price that is less than or equal to your specified maximum price.
  - Before Amazon EC2 hibernates your Spot Instance, you'll receive an interruption notice two minutes before hibernation starts.

  For more information, see Spot Instance interruptions.

Prerequisites for EC2 instance hibernation

You can enable hibernation support for an On-Demand Instance or a Spot Instance when you launch it. You can't enable hibernation on an existing instance, whether it is running or stopped. For more information, see Enable instance hibernation.

Requirements to hibernate an instance

- AWS Regions
- AMIs
- Instance families
- Instance RAM size
- Root volume type
- Root volume size
- Root volume encryption
- EBS volume type
- Spot Instance requests

AWS Regions

You can use hibernation with instances in all AWS Regions.

AMIs

You must use an HVM AMI that supports hibernation. The following AMIs support hibernation:

Linux AMIs

AMIs for Intel and AMD instance types

- AL2023 AMI released 2023.09.20 or later [1]
- Amazon Linux 2 AMI released 2019.08.29 or later
- Amazon Linux AMI 2018.03 released 2018.11.16 or later
- CentOS version 8 AMI [2] (Additional configuration is required)
- Fedora version 34 or later AMI [2] (Additional configuration is required)
- Red Hat Enterprise Linux (RHEL) 9 AMI [2] (Additional configuration is required)
- Red Hat Enterprise Linux (RHEL) 8 AMI [2] (Additional configuration is required)
- Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI released with serial number 20230303 or later [3]
- Ubuntu 20.04 LTS (Focal Fossa) AMI released with serial number 20210820 or later [3]
- Ubuntu 18.04 LTS (Bionic Beaver) AMI released with serial number 20190722.1 or later [3] [5]
- Ubuntu 16.04 LTS (Xenial Xerus) AMI [3] [4] [5] (Additional configuration is required)

AMIs for Graviton instance types

- AL2023 AMI (64-bit Arm) released 2024.07.01 or later [1]
- Amazon Linux 2 AMI (64-bit Arm) released 2024.06.20 or later
- Ubuntu 22.04.2 LTS (64-bit Arm) (Jammy Jellyfish) AMI released with serial number 20240701 or later [3]
- Ubuntu 20.04 LTS (64-bit Arm) (Focal Fossa) AMI released with serial number 20240701 or later [3]

[1] For AL2023 minimal AMI, additional configuration is required.

[2] For CentOS, Fedora, and Red Hat Enterprise Linux, hibernation is supported on Nitro-based instances only.

[3] We recommend disabling KASLR on instances with Ubuntu 22.04.2 LTS (Jammy Jellyfish), Ubuntu 20.04 LTS (Focal Fossa), Ubuntu 18.04 LTS (Bionic Beaver), and Ubuntu 16.04 LTS (Xenial Xerus). For more information, see Disable KASLR on an instance (Ubuntu only).

[4] For the Ubuntu 16.04 LTS (Xenial Xerus) AMI, hibernation is not supported on `t3.nano` instance types. No patch will be made available because Ubuntu (Xenial Xerus) ended support in April 2021. If you want to use `t3.nano` instance types, we recommend that you upgrade to the Ubuntu 22.04.2 LTS (Jammy Jellyfish), Ubuntu 20.04 LTS (Focal Fossa) AMI, or the Ubuntu 18.04 LTS (Bionic Beaver) AMI.

[5] Support for Ubuntu 18.04 LTS (Bionic Beaver) and Ubuntu 16.04 LTS (Xenial Xerus) has reached end of life.

To configure your own AMI to support hibernation, see Configure a Linux AMI to support hibernation.

Support for other versions of Ubuntu and other operating systems is coming soon.

Windows AMIs

- Windows Server 2022 AMI released 2023.09.13 or later
- Windows Server 2019 AMI released 2019.09.11 or later
- Windows Server 2016 AMI released 2019.09.11 or later
- Windows Server 2012 R2 AMI released 2019.09.11 or later
- Windows Server 2012 AMI released 2019.09.11 or later

Instance families

You must use an instance family that supports hibernation. However, bare metal instances are not supported.

- General purpose: M3, M4, M5, M5a, M5ad, M5d, M6a, M6g, M6gd, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd, M8i, M8i-flex, T2, T3, T3a, T4g
- Compute optimized: C3, C4, C5, C5d, C6a, C6g, C6gd, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd, C8gn
- Memory optimized: R3, R4, R5, R5a, R5ad, R5d, R6a, R6g, R6gd, R6idn, R6in, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd, R8i, R8i-flex, X2gd
- Storage optimized: I3, I3en, I4g, I7i, I7ie, I8g, I8ge, Im4gn, Is4gen

Console

To get the instance types that support hibernation

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instance Types**.
3. Add the filter **On-Demand Hibernation support = true**.
4. (Optional) Add filters to further scope to specific instance types of interest.

AWS CLI

To get the instance types that support hibernation

Use the describe-instance-types command. Note that the available instance types vary by Region.

```
aws ec2 describe-instance-types \
    --filters Name=hibernation-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

To get the instance types that support hibernation

Use the Get-EC2InstanceType cmdlet. Note that the available instance types vary by Region.

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-
Object
```

Instance RAM size

Linux instances – Must be less than 150 GiB.

Windows instances – Must be less than or equal to 16 GiB. For hibernating a T3 or T3a Windows instance, we recommend at least 1 GiB of RAM.

Root volume type

The root volume must be an EBS volume, not an instance store volume.

Root volume size

The root volume 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.

Root volume encryption

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 encryption by default** – You can enable EBS encryption by default to ensure that 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 Enable encryption by default.
- **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.
- **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 and Copy an AMI.

EBS volume type

The EBS volumes must use one of the following EBS volume types:

- General Purpose SSD (`gp2` and `gp3`)
- Provisioned IOPS SSD (`io1` and `io2`)

If you choose a Provisioned IOPS SSD volume type, you must provision the EBS volume with the appropriate IOPS to achieve optimum performance for hibernation. For more information, see Amazon EBS volume types in the *Amazon EBS User Guide*.

Spot Instance requests

For Spot Instances, the following requirements apply:

- The Spot Instance request type must be `persistent`.
- You can't specify a launch group in the Spot Instance request.

Configure a Linux AMI to support hibernation

The following Linux AMIs can support hibernating an Amazon EC2 instance, provided you complete the additional configuration steps described in this section.

Additional configuration required for:

- AL2023 minimal AMI released 2023.09.20 or later
- Amazon Linux 2 minimal AMI released 2019.08.29 or later
- Amazon Linux 2 released before 2019.08.29
- Amazon Linux released before 2018.11.16
- CentOS version 8 or later
- Fedora version 34 or later
- Red Hat Enterprise Linux version 8 or 9
- Ubuntu 20.04 LTS (Focal Fossa) released before serial number 20210820
- Ubuntu 18.04 (Bionic Beaver) released before serial number 20190722.1
- Ubuntu 16.04 (Xenial Xerus)

For the Linux and Windows AMIs that support hibernation and for which *no additional* configuration is required, see AMIs.

For more information, see Update instance software on your Amazon Linux 2 instance.

AL2023 minimal AMI released 2023.09.20 or later

To configure an AL2023 minimal AMI released 2023.09.20 or later to support hibernation

1. Install the `ec2-hibinit-agent` package from the repositories.

   ```
   [ec2-user ~]$ sudo dnf install ec2-hibinit-agent
   ```

2. Restart the service.

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

Amazon Linux 2 minimal AMI released 2019.08.29 or later

To configure an Amazon Linux 2 minimal AMI released 2019.08.29 or later to support hibernation

1. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

2. Restart the service.

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

Amazon Linux 2 released before 2019.08.29

To configure an Amazon Linux 2 AMI released before 2019.08.29 to support hibernation

1. Update the kernel to `4.14.138-114.102` or later.

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

2. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. Reboot the instance.

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

4. Confirm that the kernel version is updated to `4.14.138-114.102` or later.

```
[ec2-user ~]$ uname -a
```

5. Stop the instance and create an AMI. For more information, see Create an Amazon EBS-backed AMI.

Amazon Linux released before 2018.11.16

To configure an Amazon Linux AMI released before 2018.11.16 to support hibernation

1. Update the kernel to `4.14.77-70.59` or later.

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

2. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. Reboot the instance.

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

4. Confirm that the kernel version is updated to `4.14.77-70.59` or greater.

```
[ec2-user ~]$ uname -a
```

5. Stop the instance and create an AMI. For more information, see Create an Amazon EBS-backed AMI.

CentOS version 8 or later

To configure a CentOS version 8 or later AMI to support hibernation

1. Update the kernel to `4.18.0-305.7.1.el8_4.x86_64` or later.

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

2. Install the Fedora Extra Packages for Enterprise Linux (EPEL) repository.

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

3. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. Enable the hibernate agent to start on boot.

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. Reboot the instance.

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

6. Confirm that the kernel version is updated to `4.18.0-305.7.1.el8_4.x86_64` or later.

```
[ec2-user ~]$ uname -a
```

Fedora version 34 or later

To configure a Fedora version 34 or later AMI to support hibernation

1. Update the kernel to `5.12.10-300.fc34.x86_64` or later.

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

2. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

3. Enable the hibernate agent to start on boot.

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

4. Reboot the instance.

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

5. Confirm that the kernel version is updated to `5.12.10-300.fc34.x86_64` or later.

```
[ec2-user ~]$ uname -a
```

Red Hat Enterprise Linux version 8 or 9

To configure a Red Hat Enterprise Linux 8 or 9 AMI to support hibernation

1. Update the kernel to `4.18.0-305.7.1.el8_4.x86_64` or later.

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

2. Install the Fedora Extra Packages for Enterprise Linux (EPEL) repository.

   RHEL version 8:

   ```
   [ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
   ```

   RHEL version 9:

   ```
   [ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
   ```

3. Install the `ec2-hibinit-agent` package from the repositories.

   ```
   [ec2-user ~]$ sudo yum install ec2-hibinit-agent
   ```

4. Enable the hibernate agent to start on boot.

   ```
   [ec2-user ~]$ sudo systemctl enable hibinit-agent.service
   ```

5. Reboot the instance.

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

6. Confirm that the kernel version is updated to `4.18.0-305.7.1.el8_4.x86_64` or later.

   ```
   [ec2-user ~]$ uname -a
   ```

Ubuntu 20.04 LTS (Focal Fossa) released before serial number 20210820

To configure an Ubuntu 20.04 LTS (Focal Fossa) AMI released before serial number 20210820 to support hibernation

1. Update the linux-aws-kernel to `5.8.0-1038.40` or later, and grub2 to `2.04-1ubuntu26.13` or later.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. Reboot the instance.

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

3. Confirm that the kernel version is updated to `5.8.0-1038.40` or later.

```
[ec2-user ~]$ uname -a
```

4. Confirm that the grub2 version is updated to `2.04-1ubuntu26.13` or later.

```
[ec2-user ~]$ dpkg --list | grep grub2-common
```

Ubuntu 18.04 (Bionic Beaver) released before serial number 20190722.1

To configure an Ubuntu 18.04 LTS AMI released before serial number 20190722.1 to support hibernation

1. Update the kernel to `4.15.0-1044` or later.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. Reboot the instance.

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

4. Confirm that the kernel version is updated to `4.15.0-1044` or later.

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04 (Xenial Xerus)

To configure Ubuntu 16.04 LTS to support hibernation, you need to install the linux-aws-hwe kernel package version 4.15.0-1058-aws or later and the ec2-hibinit-agent.

> **Important**
>
> The `linux-aws-hwe` kernel package is supported by Canonical. The standard support for Ubuntu 16.04 LTS ended in April 2021, and the package no longer receives regular updates. However, it will receive additional security updates until the Extended Security Maintenance support ends in 2024. For more information, see [Amazon EC2 Hibernation for Ubuntu 16.04 LTS now available](#) on the Canonical Ubuntu Blog.
> We recommend that you upgrade to the Ubuntu 20.04 LTS (Focal Fossa) AMI or the Ubuntu 18.04 LTS (Bionic Beaver) AMI.

To configure an Ubuntu 16.04 LTS AMI to support hibernation

1. Update the kernel to `4.15.0-1058-aws` or later.

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt install linux-aws-hwe
```

2. Install the `ec2-hibinit-agent` package from the repositories.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. Reboot the instance.

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

4. Confirm that the kernel version is updated to `4.15.0-1058-aws` or later.

```
[ec2-user ~]$ uname -a
```

Enable hibernation for an Amazon EC2 instance

To hibernate an instance, you must first enable it for hibernation while launching the instance.

> **Important**
>
> You can't enable or disable hibernation for an instance after you launch it.

Tasks

Enable hibernation for On-Demand Instances

You can enable hibernation for your On-Demand Instances.

Console

To enable hibernation for an On-Demand Instance

1. Follow the procedure to launch an instance, but don't launch the instance until you've completed the following steps to enable hibernation.
2. To enable hibernation, configure the following fields in the launch instance wizard:

   a. Under **Application and OS Images (Amazon Machine Image)**, select an AMI that supports hibernation. For more information, see AMIs.

   b. Under **Instance type**, select a supported instance type. For more information, see Instance families.

   c. Under **Configure storage**, choose **Advanced** (at the right), and specify the following information for the root volume:

      - 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 **Encrypted**, choose **Yes**. If you enabled encryption by default in this AWS Region, **Yes** is selected.
- For **KMS key**, 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 [Prerequisites for EC2 instance hibernation](#).

d. Expand **Advanced details**, and for **Stop - Hibernate behavior**, choose **Enable**.

3. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see [Launch an EC2 instance using the launch instance wizard in the console](#).

AWS CLI

To enable hibernation for an On-Demand Instance

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
        }
    }
```

```
]
```

The value for `DeviceName` must match the root device name that's associated with the AMI. To find the root device name, use the describe-images command.

```
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 for an On-Demand Instance

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.

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair
```

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.

```
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.

Enable hibernation for Spot Instances

You can enable hibernation for your Spot Instances. For more information about hibernating a Spot Instance on interruption, see Spot Instance interruptions.

Console

To enable hibernation for a Spot Instance

1. Follow the procedure to request a Spot Instance using the launch instance wizard, but don't launch the instance until you've completed the following steps to enable hibernation.
2. To enable hibernation, configure the following fields in the launch instance wizard:
   a. Under **Application and OS Images (Amazon Machine Image)**, select an AMI that supports hibernation. For more information, see AMIs.
   b. Under **Instance type**, select a supported instance type. For more information, see Instance families.
   c. Under **Configure storage**, choose **Advanced** (at the right), and specify the following information for the root volume:
      - 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 **Encrypted**, choose **Yes**. If you enabled encryption by default in this AWS Region, **Yes** is selected.
      - For **KMS key**, 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 Prerequisites for EC2 instance hibernation.
   d. Expand **Advanced details**, and, in addition to the fields for configuring a Spot Instance, do the following:
      i. For **Request type**, choose **Persistent**.

ii. For **Interruption behavior**, choose **Hibernate**. Alternatively, for **Stop - Hibernate behavior**, choose **Enable**. Both fields enable hibernation on your Spot Instance. You need only configure one of them.

3. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

AWS CLI

To enable hibernation for a Spot Instance

Use the run-instances command to request a Spot 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. The Spot request type (`SpotInstanceType`) must be `persistent`.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c4.xlarge \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
    --instance-market-options
        {
           "MarketType":"spot",
           "SpotOptions":{
              "MaxPrice":"1",
              "SpotInstanceType":"persistent"
            }
        }
```

Specify the EBS root volume parameters in `mapping.json` as follows.

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
            "VolumeType": "gp2",
```

```
                "Encrypted": true
            }
        }
]
```

The value for DeviceName must match the root device name that's associated with the AMI. To find the root device name, use the describe-images command.

```
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 for a Spot Instance

Use the New-EC2Instance command to request a Spot 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.

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair `
    -InstanceMarketOption @(
        MarketType = spot;
        SpotOptions @{
        MaxPrice = 1;
        SpotInstanceType = persistent}
    )
```

The value for `DeviceName` must match the root device name associated with the AMI. To find the root device name, use the [Get-EC2Image](https://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2Image.html) command.

```
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.

View if an instance is enabled for hibernation

You can check whether an instance is enabled for hibernation.

Console

To view if an instance is enabled for hibernation

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.

AWS CLI

To view if an instance is enabled for hibernation

Use the describe-instances command and specify the `--filters "Name=hibernation-options.configured,Values=true"` parameter to filter instances that are enabled for hibernation.

```
aws ec2 describe-instances \
    --filters "Name=hibernation-options.configured,Values=true"
```

The following field in the output indicates that the instance is enabled for hibernation.

```
"HibernationOptions": {
    "Configured": true
}
```

PowerShell

To view if an instance is enabled for hibernation

Use the Get-EC2Instance cmdlet and filter instances that are enabled for hibernation.

```
(Get-EC2Instance `
    -Filter @{Name="hibernation-options.configured"; Values="true"}).Instances
```

Disable KASLR on an instance (Ubuntu only)

To run hibernation on a newly launched instance with Ubuntu 16.04 LTS (Xenial Xerus), Ubuntu 18.04 LTS (Bionic Beaver) released with serial number 20190722.1 or later, or Ubuntu 20.04 LTS (Focal Fossa) released with serial number 20210820 or later, we recommend disabling KASLR (Kernel Address Space Layout Randomization). On Ubuntu 16.04 LTS, Ubuntu 18.04 LTS, or Ubuntu 20.04 LTS, KASLR is enabled by default.

KASLR is a standard Linux kernel security feature that helps to mitigate exposure to and ramifications of yet-undiscovered memory access vulnerabilities by randomizing the base address value of the kernel. With KASLR enabled, there is a possibility that the instance might not resume after it has been hibernated.

To learn more about KASLR, see Ubuntu Features.

To disable KASLR on an instance launched with Ubuntu

1. Connect to your instance using SSH. For more information, see Connect to your Linux instance using SSH.
2. Open the `/etc/default/grub.d/50-cloudimg-settings.cfg` file in your editor of choice. Edit the `GRUB_CMDLINE_LINUX_DEFAULT` line to append the `nokaslr` option to its end, as shown in the following example.

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
    nvme_core.io_timeout=4294967295 nokaslr"
   ```
3. Save the file and exit your editor.
4. Run the following command to rebuild the grub configuration.

   ```
   sudo update-grub
   ```

5. Reboot the instance.

```
sudo reboot
```

6. Run the following command to confirm that `nokaslr` has been added.

```
cat /proc/cmdline
```

The output of the command should include the `nokaslr` option.

Hibernate an Amazon EC2 instance

You can initiate hibernation on an On-Demand Instance or Spot Instance if the instance is an EBS-backed instance, is enabled for hibernation, and meets the hibernation prerequisites. If an instance cannot hibernate successfully, a normal shutdown occurs.

Console

To hibernate 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, 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 Prerequisites for EC2 instance hibernation.
4. When prompted for confirmation, choose **Hibernate**. It can take a few minutes for the instance to hibernate. The instance state first changes to **Stopping**, and then changes to **Stopped** when the instance has hibernated.

AWS CLI

To hibernate an instance

Use the stop-instances command and specify the `--hibernate` parameter.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

PowerShell

To hibernate an instance

Use the Stop-EC2Instance cmdlet.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Hibernate $true
```

You can check whether hibernation was initiated on an instance.

Console

To view if hibernation was initiated on 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 and, on the **Details** tab, in the **Instance details** section, check the value for **State transition message**.

   Client.UserInitiatedHibernate: User initiated hibernate indicates that you initiated hibernation on the On-Demand Instance or Spot Instance.

AWS CLI

To view if hibernation was initiated on an instance

Use the describe-instances command and specify the `state-reason-code` filter to see the instances on which hibernation was initiated.

```
aws ec2 describe-instances \
    --filters "Name=state-reason-code,Values=Client.UserInitiatedHibernate"
```

The following field in the output indicates that hibernation was initiated on the On-Demand Instance or Spot Instance.

```
"StateReason": {
    "Code": "Client.UserInitiatedHibernate"
```

```
}
```

PowerShell

To view if hibernation was initiated on an instance

Use the Get-EC2Instance cmdlet and specify the `state-reason-code` filter to see the instances on which hibernation was initiated.

```
Get-EC2Instance `
    -Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

Start a hibernated Amazon EC2 instance

Start a hibernated instance by starting it in the same way that you would start a stopped instance.

For Spot Instances, if Amazon EC2 hibernated the instance, then only Amazon EC2 can resume it. You can only resume a hibernated Spot Instance if *you* hibernated it. Spot Instances can only be resumed if capacity is available and the Spot price is less than or equal to your specified maximum price.

Console

To start a hibernated instance

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 show the instance in a failed state until the instance has started.

AWS CLI

To start a hibernated instance

Use the start-instances command.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To start a hibernated instance

Use the Start-EC2Instance cmdlet.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

Troubleshoot Amazon EC2 instance hibernation

Use this information to help diagnose and fix issues that you might encounter when hibernating an instance.

Hibernation issues

- Can't hibernate immediately after launch
- Takes too long to transition from stopping to stopped, and memory state not restored after start
- Instance "stuck" in the stopping state
- Can’t start Spot Instance immediately after hibernate
- Resume Spot Instances failed

Can't hibernate immediately after launch

If you try to hibernate an instance too quickly after you've launched it, you get an error.

You must wait for about two minutes for Linux instances and about five minutes for Windows instances 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.

Linux instances

Check the instance system log and look for messages that are related to hibernation. To access the system log, connect to the instance or use the get-console-output command. Find the log lines from the `hibinit-agent`. If the log lines indicate a failure or the log lines are missing, there was most likely a failure configuring hibernation at launch.

For example, the following message indicates that the instance root volume is not large enough: `hibinit-agent: Insufficient disk space. Cannot create setup for hibernation. Please allocate a larger root device.`

If the last log line from the `hibinit-agent` is `hibinit-agent: Running: swapoff /swap`, hibernation was successfully configured.

If you do not see any logs from these processes, your AMI might not support hibernation. For information about supported AMIs, see Prerequisites for EC2 instance hibernation. If you used your own Linux AMI, make sure that you followed the instructions for Configure a Linux AMI to support hibernation.

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 to the instance and open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\Ec2Launch.log` file in a text editor. If you're using EC2Launch v2, open `C:\ProgramData\Amazon\EC2Launch\log\agent.log`.

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 error message includes decimal ASCII values, you can convert the ASCII values to plain text in order to read the full error message.

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 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.

Windows instance size

If you're using a T3 or T3a Windows instance with less than 1 GiB of RAM, try increasing the size of the instance to one that has at least 1 GiB 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 Amazon EC2 instance stop issues.

Can't start Spot Instance immediately after hibernate

If you try to start a Spot Instance within two minutes of hibernating it, you might get the following error:

`You failed to start the Spot Instance because the associated Spot Instance request is not in an appropriate state to support start.`

Wait for about two minutes for Linux instances and about five minutes for Windows instances and then retry starting the instance.

Resume Spot Instances failed

If your Spot Instance was hibernated successfully but failed to resume, and instead rebooted (a fresh restart where the hibernated state is not retained), it might be because the user data contained the following script:

```
/usr/bin/enable-ec2-spot-hibernation
```

Remove this script from the **User data** field in the launch template, and then request a new Spot Instance.

Note that even if the instance failed to resume, without the hibernated state preserved, the instance can still be started in the same way as starting from the `stopped` state.

Reboot your Amazon EC2 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 the following:

- Public DNS name (IPv4)
- Private IPv4 address
- Public IPv4 address
- IPv6 address (if applicable)
- Any data on its instance store volumes

Rebooting an instance doesn't start a new instance billing period, unlike stopping and starting an instance (which starts a new billing period with a one-minute minimum charge).

An instance reboot can be user-initiated (where you manually reboot the instance) or initiated by AWS (for automatic instance recovery, or in response to a scheduled reboot event for necessary maintenance, such as to apply updates that require a reboot).

For user-initiated reboots, we recommend using the Amazon EC2 console, CLI, or API instead of running the operating system reboot command from your instance. When using Amazon EC2, if the instance does not cleanly shut down within a few minutes, Amazon EC2 performs a hard reboot. Furthermore, AWS CloudTrail creates an API record of when your instance was rebooted.

This topic describes how to perform a user-initiated reboot. For information about reboots performed by AWS, see Automatic instance recovery and Manage Amazon EC2 instances scheduled for reboot.

> **Important**
>
> If updates are installing 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.

Console

To reboot 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 and choose **Instance state**, **Reboot instance**.
4. Choose **Reboot** when prompted for confirmation.

   The instance remains in the `running` state.

AWS CLI

To reboot an instance

Use the reboot-instances command.

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To reboot an instance

Use the Restart-EC2Instance cmdlet.

```
Restart-EC2Instance -InstanceId i-1234567890abcdef0
```

To run a controlled fault injection experiment

You can use AWS Fault Injection Service to test how your application responds when your instance is rebooted. For more information, see the AWS Fault Injection Service User Guide.

Terminate Amazon EC2 instances

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 new instances using the same AMI.

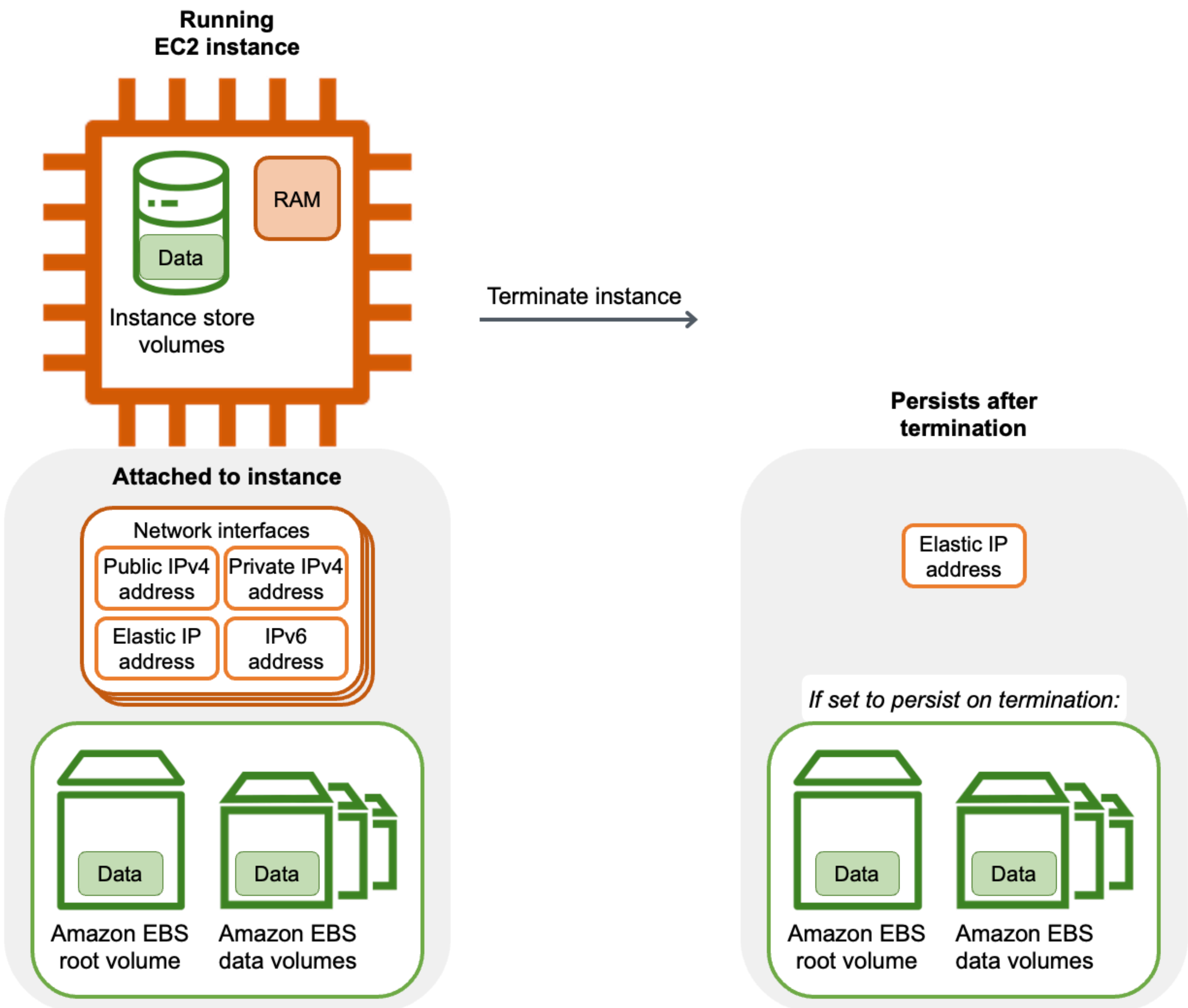
If you'd rather stop or hibernate your instance, see Stop and start Amazon EC2 instances or Hibernate your Amazon EC2 instance. For more information, see Differences between instance states.

Contents

How instance termination works

When you terminate an instance, changes are registered at the operating system (OS) level of the instance, some resources are lost, and some resources persist.

The following diagram shows what is lost and what persists when an Amazon EC2 instance is terminated. When an instance terminates, the data on any instance store volumes and the data stored in the instance RAM is erased. Any Elastic IP addresses associated with the instance are detached. For Amazon EBS root volumes and data volumes, the outcome depends on the **Delete on termination** setting of each volume.



Considerations

- **Data persistence**
  - Instance store volumes: All data is permanently deleted when the instance terminates.
  - EBS root volume:
    - When attached at launch, deleted by default when the instance terminates.
    - When attached after launch, persists by default when the instance terminates.
  - EBS data volumes:

  - When attached at launch using the console: Persists by default when the instance terminates.
  - When attached at launch using the CLI: Deleted by default when the instance terminates.
  - When attached after launch using the console or CLI: Persists by default when the instance terminates.

> **Note**
>
> Any volumes that are not deleted on instance termination continue to incur charges. You can change the setting so that a volume is deleted or persists on instance termination. For more information, see Preserve data when an instance is terminated.

- **Protection against accidental termination**
  - To prevent an instance from being accidentally terminated by someone, enable termination protection.
  - To control whether an instance stops or terminates when shutdown is initiated from the instance, change the instance initiated shutdown behavior.
- **Shutdown scripts** – If you run a script on instance termination, your instance might have an abnormal termination because we have no way of ensuring that shutdown scripts run. Amazon EC2 attempts to cleanly shut down an instance and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.
- **Bare metal instances** – x86 bare metal instances don't support cooperative shutdown.

What happens when you terminate an instance

Changes 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 of the system is provided by **systemd** (Linux) or the System process (Windows). Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
- ACPI shutdown is initiated.

- The instance shuts down after the graceful shutdown process exits. There is no configurable OS shutdown time. The instance remains visible in the console for a short time, then the entry is automatically deleted.

Resources lost

- Data stored on the instance store volumes.
- EBS root volume if the `DeleteOnTermination` attribute is set to `true`.
- EBS data volumes (attached at launch or after) if the `DeleteOnTermination` attribute is set to `true`.

Resources that persist

- EBS root volume if the `DeleteOnTermination` attribute is set to `false`.
- EBS data volumes (attached at launch or after) if the `DeleteOnTermination` attribute is set to `false`.

Test application response to instance termination

You can use AWS Fault Injection Service to test how your application responds when your instance is terminated. For more information, see the AWS Fault Injection Service User Guide.

Methods for terminating an instance

There are four ways to perform a user-initiated instance termination: default terminate, terminate with skip OS shutdown, force terminate, and force terminate with skip OS shutdown. The following table compares the key differences between the termination methods:

Termination method	Key purpose	Use case	CLI command
Default terminat	Normal instance shutdown with attempted graceful OS shutdown.	Typical instance termination.	`aws ec2 terminate-instances \`

Termination method	Key purpose	Use case	CLI command
			`--instance-id i-1234567890abcdef0`
Terminate with skip OS shutdown	Bypasses the graceful OS shutdown when terminating an instance.	When bypassing graceful OS shutdown is required.	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --skip-os-shutdown`
Force terminate	Handles stuck instances. Attempts a default termination first; if instance fails to terminate, then forcibly terminates the instance.	When instance is stuck in `shutting-down` state.	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --force`
Force terminate with skip OS shutdown	Force terminates and bypasses the graceful OS shutdown when terminating an instance.	When force termination and bypassing graceful OS shutdown is required.	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --force \ --skip-os-shutdown`

For instructions on how to use each method, see the following:

- Terminate an instance with a graceful OS shutdown
- Terminate an instance and bypass the graceful OS shutdown
- Force terminate an instance

Terminate an instance with a graceful OS shutdown

You can terminate an instance using the default terminate method, which includes an attempt at a graceful OS shutdown. For more information, see Methods for terminating an instance.

Console

To terminate an instance using the default terminate method

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 **Instance state**, **Terminate (delete) instance**.
4. Choose **Terminate (delete)** when prompted for confirmation.
5. After you terminate an instance, it remains visible for a short while, with a state of `terminated`.

   If termination fails or if a terminated instance is visible for more than a few hours, see Terminated instance still displayed.

AWS CLI

To terminate an instance using the default terminate method

Use the terminate-instances command.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To terminate an instance using the default terminate method

Use the Remove-EC2Instance cmdlet.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

Terminate an instance and bypass the graceful OS shutdown

You can bypass the graceful OS shutdown when terminating an instance. For more information, see Methods for terminating an instance.

Console

To terminate an instance and bypass the graceful OS shutdown

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 **Instance state**, **Terminate (delete) instance**.
4. Under **Skip OS shutdown**, select the **Skip OS shutdown** checkbox. If you don't see this option in the console, it's not yet available in the console in the current Region. You can, however, access this feature using the AWS CLI or SDK, or try another Region in the console.
5. Choose **Terminate (delete)**.
6. After you terminate an instance, it remains visible for a short while, with a state of `terminated`.

   If termination fails or if a terminated instance is visible for more than a few hours, see Terminated instance still displayed.

AWS CLI

To terminate an instance and bypass the graceful OS shutdown

Use the terminate-instances command with `--skip-os-shutdown`..

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --skip-os-shutdown
```

PowerShell

To terminate an instance and bypass the graceful OS shutdown

Use the Remove-EC2Instance cmdlet with `-SkipOsShutdown $true`..

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
```

```
    -SkipOsShutdown $true
```

Troubleshoot instance termination

The requester must have permission to call `ec2:TerminateInstances`. For more information, see [Example policies to work with instances].

If you terminate your instance and another instance starts, most likely you have configured automatic scaling through a feature like EC2 Fleet or Amazon EC2 Auto Scaling. For more information, see [Instances automatically launched or terminated].

You can't terminate an instance if termination protection is turned on. For more information, see [termination protection].

If your instance is in the `shutting-down` state for longer than usual, you can attempt to force terminate it. If it remains in the `shutting-down` state, it should be cleaned up (terminated) by automated processes within the Amazon EC2 service. For more information, see [Delayed instance termination].

Change instance termination protection

To prevent your instance from being accidentally terminated using the Amazon EC2 API, whether you call `TerminateInstances` directly or using another interface such as the Amazon EC2 console, enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch an instance, or while the instance is running or stopped.

The `DisableApiTermination` attribute doesn't prevent you from terminating an instance by initiating shutdown from the instance (for example, by using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set to `terminate`. For more information, see [Change instance initiated shutdown behavior].

Considerations

- Enabling termination protection does not prevent AWS from terminating the instance when there is a [scheduled event] to terminate the instance.
- Enabling termination protection does not prevent Amazon EC2 Auto Scaling from terminating an instance when the instance is unhealthy or during scale-in events. You can control whether

an Auto Scaling group can terminate a particular instance when scaling using instance scale-in protection. You can control whether an Auto Scaling group can terminate unhealthy instances by suspending the ReplaceUnhealthy scaling process.

- You can't enable termination protection for Spot Instances.

Console

To enable termination protection for an instance at launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the dashboard, choose **Launch instance**.
3. Expand **Advanced details**. For **Termination protection**, select **Enable**.
4. When you are finishing specifying the details for your instance, choose **Launch instance**.

To change termination protection for an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, select **Instances**.
3. Select the instance.
4. Choose **Actions**, **Instance settings**, **Change termination protection**.
5. For **Termination protection** select or clear **Enable**.
6. Choose **Save**.

AWS CLI

To enable termination protection for an instance

Use the modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-termination
```

To disable termination protection for an instance

Use the modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-termination
```

PowerShell

To enable termination protection for an instance

Use the Edit-EC2InstanceAttribute cmdlet.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $true
```

To disable termination protection for an instance

Use the Edit-EC2InstanceAttribute cmdlet.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $false
```

Terminate multiple instances with termination protection

If you terminate multiple instances across multiple Availability Zones in the same request, 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.

Example

Suppose that you have the following four instances across two Availability Zones.

Instance	Availability Zone	Terminate protection
Instance 1	AZ A	`Disabled`
Instance 2		`Disabled`
Instance 3	AZ B	`Enabled`
Instance 4		`Disabled`

If you attempt to terminate all of these instances in the same request, the request reports failure with the following results:

- **Instance 1** and **Instance 2** are successfully terminated because neither instance is enabled for termination protection.
- **Instance 3** and **Instance 4** fail to terminate because **Instance 3** is enabled for termination protection.

Change 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. You can change this behavior so that the instance terminates instead by changing the `InstanceInitiatedShutdownBehavior` attribute for the instance. You can change this attribute while the instance is running or stopped.

The **halt** command doesn't initiate a shutdown. If used, the instance doesn't terminate; instead, it places the CPU into HLT and the instance continues to run.

> **Note**
>
> The `InstanceInitiatedShutdownBehavior` attribute only applies when you perform a shutdown from the operating system of the instance itself. It doesn't apply when you stop an instance using the `StopInstances` API or the Amazon EC2 console.

Console

To change the instance initiated shutdown behavior

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**.

   Shutdown behavior displays the current behavior.
5. To change the behavior, for **Shutdown behavior**, choose **Stop** or **Terminate**.
6. Choose **Save**.

AWS CLI

To change the instance initiated shutdown behavior

Use the modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --instance-initiated-shutdown-behavior terminate
```

PowerShell

To change the instance initiated shutdown behavior

Use the Edit-EC2InstanceAttribute cmdlet.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -InstanceInitiatedShutdownBehavior terminate
```

Preserve data when an instance is terminated

When an Amazon EC2 instance is terminated, you can preserve the data on your instance store volumes or Amazon EBS volumes. This topic explains how to ensure your data persists beyond instance termination.

How instance termination affects root and data volumes

Instance store volumes

When an instance is terminated, the instance store volumes are automatically deleted and the data is lost. To preserve this data beyond the lifetime of the instance, before terminating the instance, manually copy the data to persistent storage, such as an Amazon EBS volume, an Amazon S3 bucket, or an Amazon EFS file system. For more information, see Storage options for your Amazon EC2 instances.

Amazon EBS volumes

When an instance is terminated, the EBS volumes are either deleted or preserved, depending on the value of the `DeleteOnTermination` attribute for each volume:

- **Yes** (console) / `true` (CLI) – The volume is deleted when the instance is terminated.
- **No** (console) / `false` (CLI) – The volume is preserved when the instance is terminated. Preserved volumes continue to incur charges.

> **Note**
>
> After an instance terminates, you can take a snapshot of the preserved volume or attach it to another instance. To avoid incurring charges, you must delete the volume.

Default deletion behavior for EBS volumes

The default `DeleteOnTermination` value differs depending on the volume type, whether the volume was attached at launch or after, and the method (console or CLI) used to attach the volume:

Volume type	Attached when	Method for attaching	Default behavior on instance termination
Root volume	At launch	Console or CLI	Delete
Root volume	After launch	Console or CLI	Preserve
Data volume	At launch	Console	Preserve

Volume type	Attached when	Method for attaching	Default behavior on instance termination
Data volume	At launch	CLI	Delete
Data volume	After launch	Console and CLI	Preserve

Check volume persistence settings

The default value at launch for an EBS volume is determined by the `DeleteOnTermination` attribute set on the AMI. You can change the value at instance launch, overriding the AMI setting. We recommend that you verify the default setting for the `DeleteOnTermination` attribute after you launch an instance.

To check if an Amazon EBS volume will be deleted on instance termination

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 the **Storage** tab.
5. Under **Block devices**, scroll right to check the **Delete on termination** column.
   - If **Yes**, the volume is deleted when the instance is terminated.
   - If **No**, the volume is not be deleted when the instance is terminated. Any volumes not deleted continue to incur charges.

Change the root volume to persist at launch

You can change the `DeleteOnTermination` attribute of an EBS root volume when you launch an instance. You can also use the following procedure for a data volume.

Console

To change the root volume of an instance to persist at launch

1. Follow the procedure to launch an instance, but don't launch the instance until you've completed the following steps to change the root volume to persist.

2. On the **Configure storage** pane, choose **Advanced**.
3. Under **EBS volumes**, expand the root volume information.
4. For **Delete on termination**, choose **No**.
5. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

AWS CLI

To change the root volume of an instance to persist at launch

Use the run-instances command to change the value of `DeleteOnTermination` in the block device mapping.

Add the `--block-device-mappings` option:

```
--block-device-mappings file://mapping.json
```

In `mapping.json`, specify the device name, for example `/dev/sda1` or `/dev/xvda`, and for `DeleteOnTermination`, specify `false`.

```
[
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

To change the root volume of an instance to persist at launch

Use the New-EC2Instance cmdlet to change the value of `DeleteOnTermination` in the block device mapping.

Add the `-BlockDeviceMapping` option:

```
-BlockDeviceMapping $bdm
```

In bdm, specify the device name, for example /dev/sda1 or /dev/xvda, and for DeleteOnTermination, specify false.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

Change the root volume of a running instance to persist

You can change the EBS root volume of a running instance to persist. You can also use the following procedure for a data volume.

AWS CLI

To change the root volume to persist

Use the modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0  \
    --block-device-mappings file://mapping.json
```

In mapping.json, specify the device name, for example /dev/sda1 or /dev/xvda, and for --DeleteOnTermination, specify false.

```
[
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

To change the root volume to persist

Use the Edit-EC2InstanceAttribute cmdlet.

Add the `-BlockDeviceMapping` option:

```
-BlockDeviceMapping $bdm
```

In bdm, specify the device name, for example `/dev/sda1` or `/dev/xvda`, and for `DeleteOnTermination`, specify `false`.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

Instance retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware that hosts the instance. The instance root device determines the behavior of instance retirement:

- If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware.
- If your instance root device is an instance store volume, the instance is terminated, and can't be used again.

For more information about the types of instance events, see [Scheduled events for Amazon EC2 instances](#).

Contents

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.

> **Important**
>
> If an instance is scheduled for retirement, we recommend that you take action as soon as possible, because the instance might already be unreachable. For more information, see Check if your instance is reachable.

Options to identify instances scheduled for retirement

- Monitor the email for the account contacts
- Check your instances

Monitor the email for the account contacts

If an instance is scheduled for retirement, the primary contact for the account and the operations contact receive an email prior to the event. This email includes the instance ID and scheduled retirement date. For more information, see Update the primary contact for your AWS account and Update the alternate contacts for your AWS account in the *AWS Account Management Reference Guide*.

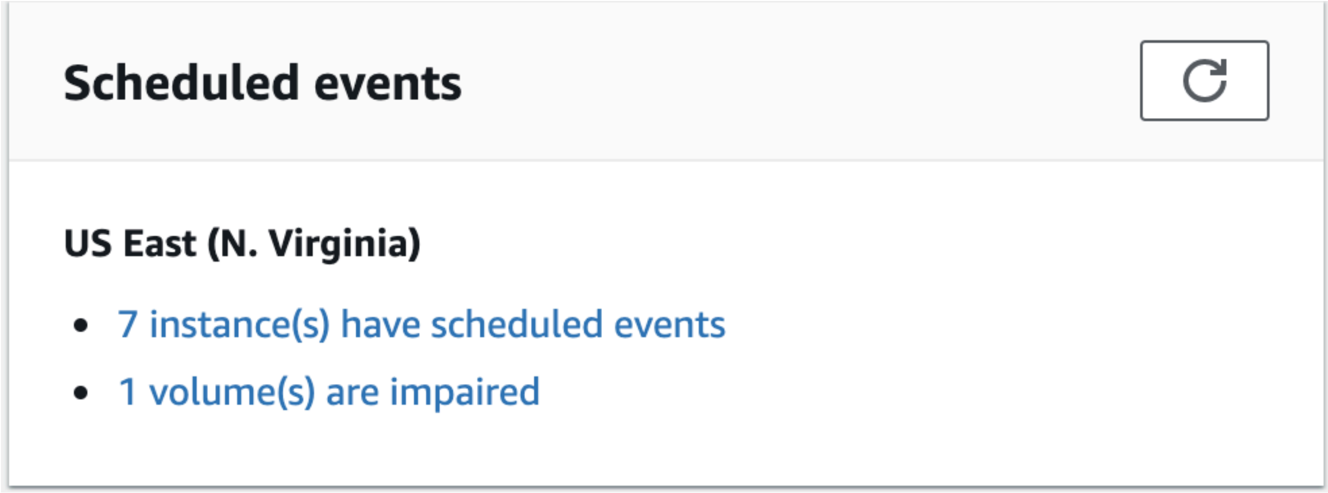
Check your instances

If you use an email account that you do not check regularly, you might miss an instance retirement notification. You can check whether any of your instances are scheduled for retirement at any time.

Console

To identify instances scheduled for retirement

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **EC2 Dashboard**. Under **Scheduled events**, you can see the events that are associated with your Amazon EC2 instances and volumes, organized by Region.



3. If you have an instance with a scheduled event listed, select its link below the Region name to go to the **Events** page.
4. The **Events** page lists all resources that have events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance stop or retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

AWS CLI

To find instances scheduled for retirement

Use the following describe-instance-status command. Repeat in each Region where you have running instances.

```
aws ec2 describe-instance-status --filters Name=event.code,Values=instance-
retirement
```

PowerShell

To find instances scheduled for retirement

Use the Get-EC2InstanceStatus cmdlet. Repeat in each Region where you have running instances.

```
Get-EC2InstanceStatus -Filter @{Name="event.code"; Values="instance-retirement"}
```

Actions to take for EBS-backed instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

For Linux instances, if you are not sure whether your instance is backed by EBS or instance store, see Root volumes for your Amazon EC2 instances.

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 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 Amazon EC2 instances. 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 to recover your instance.
- Alternatively, if you want to terminate 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 an Amazon EBS-backed AMI.

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 **Launch instance from AMI**. Configure the parameters for your instance and then choose **Launch instance**. For more information about each field, see Launch an EC2 instance using the launch instance wizard in the console.

Actions to take for instance-store backed instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

> **Warning**
>
> If your instance store-backed instance passes its retirement date, it is terminated and you cannot recover the instance or any data that was stored on it. Regardless of the root device of your instance, the data on instance store volumes is lost when the instance is retired, even if the volumes are attached to an EBS-backed instance.

Check if your instance is reachable

When you are notified that your instance is scheduled for retirement, we recommend that you take the following action as soon as possible:

- Check if your instance is reachable by either connecting to or pinging your instance.
- If your instance is unreachable, there is likely very little that can be done to recover your instance. For more information, see Troubleshoot an unreachable Amazon EC2 instance. AWS will terminate your instance on the scheduled retirement date, so, for an unreachable instance, you can immediately terminate the instance yourself.

Launch a replacement instance

Create an instance store-backed AMI from your instance using the AMI tools, as described in Create an instance store-backed AMI. From the Amazon EC2 console, select your new AMI and then choose **Launch instance from AMI**. Configure the parameters for your instance and then choose **Launch instance**. For more information about each field, see Launch an EC2 instance using the launch instance wizard in the console.

Convert your instance to an EBS-backed instance

Transfer your data to an EBS volume, take a snapshot of the volume, and then create AMI from the snapshot. You can launch a replacement instance from your new AMI. For more information, see Convert your instance store-backed AMI to an EBS-backed AMI.

Automatic instance recovery

> **Important**
>
> This section describes how to proactively configure recovery mechanisms on an EC2 instance. These recovery mechanisms are designed to restore instance availability when AWS detects an underlying hardware or software issue that causes a system status check to fail. If you are currently experiencing problems accessing your instance, see Troubleshoot EC2 instances.

If AWS detects that an instance is unavailable due to an underlying hardware or software issue, there are two mechanisms that can automatically restore instance availability—simplified automatic recovery and Amazon CloudWatch action based recovery. Restoring instance availability is also known as *instance recovery*.

During the instance recovery process, AWS will attempt to move your instance from the host with the underlying hardware or software issue to a different host. If successful, the instance recovery process will appear to the instance as an unplanned reboot. You can verify if instance recovery occurred.

If the recovery process is unsuccessful, the instance might continue running on the host with the underlying hardware or software issue. In this case, manual intervention is required. If the instance becomes unreachable or the system status check continues to fail, we recommend that you manually stop and start the instance. When you start an instance, it is typically migrated to a new underlying host computer. However, unlike automatic instance recovery, where the instance retains its public IPv4 address, a restarted instance receives a new public IPv4 address unless it has an Elastic IP address.

To benefit from the automatic recovery mechanisms, they must be configured in advance on an instance before a system status check fails. By default, simplified automatic recovery is enabled during instance launch. You can optionally configure Amazon CloudWatch action based recovery after launch. Having one of these mechanisms configured makes your instance more resilient.

Simplified automatic recovery and Amazon CloudWatch action based recovery are only available on supported instances. For more information, see Requirements for enabling simplified automatic recovery and Requirements for enabling CloudWatch action based recovery.

> **Warning**
>
> When AWS recovers your instance due to an underlying hardware or software issue, be aware of the following consequences: data stored in volatile memory (RAM) will be lost and the operating system's uptime will start over from zero. Furthermore, with CloudWatch action based recovery, data on instance store volumes will also be lost. To help protect against data loss, we recommend that you regularly create backups of valuable data. For more information about backup and recovery best practices for EC2 instances, see Best practices for Amazon EC2.
> Automatic instance recovery mechanisms are designed for *individual instances*. For guidance on building a resilient *system*, see Build a resilient system.

Topics

Key concepts of automatic instance recovery

Automatic instance recovery is an Amazon EC2 feature that automatically restores instance availability when underlying hardware or software failures occur, enhancing the resilience and reliability of your EC2 instances.

The following are key concepts of automatic instance recovery:

Configuration options

Two mechanisms can be configured to support automatic instance recovery:

- Simplified automatic recovery: Enabled by default on supported instances.
- CloudWatch action based recovery: Requires manual configuration on supported instances.

System status checks

System status checks automatically monitor the AWS infrastructure on which your EC2 instance runs.

- If a system status check fails, AWS initiates automatic instance recovery, which attempts to migrate the affected instance to different hardware.
- A failed system status check indicates a problem with the host's hardware or software, and not a problem with the instance itself. Automatic instance recovery can recover an instance that fails a system status check. However, automatic instance recovery does not operate if only the instance status check fails.
- For the differences between instance and system status checks, see Types of status checks.

Examples of underlying hardware or software problems

Hardware or software issues that can cause a system status check to fail include loss of network connectivity, loss of system power, software issues on the physical host, and hardware issues on the physical host that impact network reachability.

Characteristics of recovered instances

A recovered instance is identical to the original instance, except for the elements that are lost.

Preserved elements:

- Instance ID
- Public, private, and Elastic IP addresses
- Instance metadata
- Placement group
- Attached EBS volumes
- Availability Zone

Lost elements:

- Data stored in volatile memory (RAM)
- Data stored on instance store volumes (applicable to CloudWatch action based recovery only)
- Operating system uptime resets to zero

Monitoring system status checks with CloudWatch

The StatusCheckFailed_System metric in CloudWatch indicates whether a system status check passed or failed.

Metric values:

- **0 –** The system status check passed.
- **1 –** The system status check failed.

Events in AWS Health Dashboard

During automatic instance recovery attempts, AWS sends events to your AWS Health Dashboard based on the configured recovery mechanism and its outcome:

- Simplified automatic recovery
  - Success event: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
  - Failure event: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`
- CloudWatch action based recovery
  - Success event: `AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS`
  - Failure event: `AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE`

Differences between simplified automatic recovery and CloudWatch action based recovery

The following table compares the key differences between simplified automatic recovery and CloudWatch action based recovery.

Comparison point	Simplified automatic recovery	CloudWatch action based recovery
Configuration	Enabled by default on supported instances	Requires manual configura tion of CloudWatch alarms and actions
Flexibility	Fixed recovery behavior managed by AWS	Customizable actions and conditions

Comparison point	Simplified automatic recovery	CloudWatch action based recovery
Notification	Basic notifications through AWS Health Dashboard	Customizable notifications through SNS
Metal instance size	Excluded	Included
Instance store volumes attached at launch	Not supported for instances that attach instance store volumes at launch	Supported on selected instance types. Note that data on instance store volumes is lost during instance recovery.
Recovery time	Standard recovery attempt	Faster recovery attempts than simplified automatic recovery
Host problem resolves during migration	Migration might be canceled and the instance stays on the original host	Migration continues to a new host
Cost	No additional cost	Might incur CloudWatch charges

Build a resilient system

While simplified automatic recovery and CloudWatch action based recovery are effective for maintaining individual instance availability, AWS recommends implementing a high-availability architecture that allows failover of traffic to healthy instances.

To achieve this, consider using AWS services such as Elastic Load Balancing (which distributes incoming traffic across multiple EC2 instances) and Amazon EC2 Auto Scaling (which automatically adjusts the number of instances based on demand and health).

For more information about building a resilient, fault-tolerant system with EC2 instances, see the following resources:

- Back to Basics: Designing for Failure with EC2 on the *AWS YouTube channel*

- Disaster Recovery (DR) Architecture on AWS, Part I: Strategies for Recovery in the Cloud on the *AWS Architecture Blog* site
- Application Load Balancers User Guide
- Amazon EC2 Auto Scaling User Guide
- REL11-BP02 Fail over to healthy resources in the *Reliability Pillar AWS Well-Architected Framework*

Verify if automatic instance recovery occurred

If your instance appears to have been offline and then unexpectedly rebooted, it might have undergone automatic instance recovery in response to an underlying hardware or software issue. You can verify this by checking for automatic instance recovery events in your AWS Health Dashboard. You can also check whether an underlying hardware or software issue was detected for your instance by checking the **StatusCheckFailed_System** Amazon CloudWatch metric.

Check for events in AWS Health Dashboard

When an automatic instance recovery attempt occurs, AWS sends events to your AWS Health Dashboard. The specific event depends on the configured recovery mechanism and whether the attempt succeeded or failed.

To check for automatic instance recovery events in the AWS Health Dashboard

1. Open the AWS Health Dashboard at https://phd.aws.amazon.com/phd/home#/.
2. Look for the events associated with automatic instance recovery. The presence of these events can confirm whether an attempt at automatic instance recovery occurred and its outcome.
   - Simplified automatic recovery
     - Success event: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
     - Failure event: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`
   - CloudWatch action based recovery
     - Success event: `AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS`
     - Failure event: `AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE`

Monitor system status checks with CloudWatch

You can verify if an underlying hardware or software issue was detected for your instance by checking the StatusCheckFailed_System metric in CloudWatch. The metric value indicates whether a system status check passed (no hardware or software issue) or failed (hardware or software issue).

To verify if an underlying hardware or software issue was detected

1. Open the CloudWatch console **Metrics** page at https://console.aws.amazon.com/cloudwatch/home?#metricsV2.
2. Verify that you're in the same Region as your EC2 instance.
3. Paste the following metric in the **Metrics** search field, and press Enter.

```
StatusCheckFailed_System
```

4. Choose **EC2 > Per-Instance Metrics**.
5. In the table, select the check box next to the instance that you want to check.
6. Change the query period to the time that you suspect the recovery event occurred.
7. Choose the **Graphed metrics** tab, and for **StatusCheckFailed_System**, do the following:
   a. For **Statistic**, choose either **Average**, **Maximum**, or **Minimum**.
   b. For **Period**, choose **1 minute**.
8. Check the value for **StatusCheckFailed_System**.
   - Value of **0**: The system status check passed, indicating no underlying hardware or software issue.
   - Value of **1**: The system status check failed, indicating an underlying hardware or software issue.

For more information, see Automatic instance recovery.

Configure simplified automatic recovery on an Amazon EC2 instance

> **Important**
>
> This section describes how to proactively configure recovery mechanisms on an EC2 instance. These recovery mechanisms are designed to restore instance availability when AWS detects an underlying hardware or software issue that causes a system status check to fail. If you are currently experiencing problems accessing your instance, see Troubleshoot EC2 instances.

If AWS detects that an instance is unavailable due to an underlying hardware or software issue, *simplified automatic recovery* can automatically restore instance availability by moving the instance from the host with the underlying issue to a different host.

If simplified automatic recovery occurs, AWS sends one of the following events to your AWS Health Dashboard, depending on the outcome:

- Success event: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
- Failure event: `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`

To be notified of these events, you can configure notifications. For more information, see Creating your first notification configuration in AWS User Notifications in the *AWS User Notifications User Guide*. You can also use Amazon EventBridge rules to monitor simplified automatic recovery events.

Simplified automatic recovery is enabled by default on all supported instances during instance launch. However, it can only operate if an instance is in the `running` state, there are no service events listed in the AWS Health Dashboard, and there is available capacity for the instance type. In some situations, such as significant outages, capacity constraints might cause recovery attempts to fail. For more information, see the section called "Troubleshoot simplified automatic recovery failures".

You can disable simplified automatic recovery during or after launch, and re-enable it later if required.

> **Warning**
>
> When AWS recovers your instance due to an underlying hardware or software issue, be aware of the following consequences: data stored in volatile memory (RAM) will be lost and the operating system's uptime will start over from zero. To help protect against data loss, we recommend that you regularly create backups of valuable data. For more information about backup and recovery best practices for EC2 instances, see Best practices for Amazon EC2.
> Automatic instance recovery mechanisms are designed for *individual instances*. For guidance on building a resilient *system*, see Build a resilient system.

Contents

Requirements for enabling simplified automatic recovery

Simplified automatic recovery can be enabled on instances that meet the following criteria:

Instance types

- **General purpose:** A1, M3, M4, M5, M5a, M5n, M5zn, M6a, M6g, M6i, M6in, M7a, M7g, M7i, M7i-flex, M8g, M8i, M8i-flex, T1, T2, T3, T3a, T4g
- **Compute optimized:** C3, C4, C5, C5a, C5n, C6a, C6g, C6gn, C6i, C6in, C7a, C7g, C7gn, C7i, C7i-flex, C8g, C8gn
- **Memory optimized:** R3, R4, R5, R5a, R5b, R5n, R6a, R6g, R6i, R6in, R7a, R7g, R7i, R7iz, R8g, R8i, R8i-flex, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, U7inh-32tb, X1, X1e, X2iezn, X8g
- **Accelerated computing:** G3, G5g, Inf1, P3, VT1
- **High-performance computing:** Hpc6a, Hpc7a, Hpc7g

Tenancy

- Shared
- Dedicated Instance

For more information, see Amazon EC2 Dedicated Instances.

Limitations

Simplified automatic recovery is not supported for instances with the following characteristics:

- Instance size: `metal` instances
- Tenancy: Dedicated Host. For Dedicated Hosts, use Dedicated Host Auto Recovery instead.
- Storage: Instances with instance store volumes
- Networking: Instances using an Elastic Fabric Adapter
- Auto Scaling: Instances that are part of an Auto Scaling group
- Maintenance: Instances currently undergoing a scheduled maintenance event

Configure simplified automatic recovery

Simplified automatic recovery is enabled by default when you launch a supported instance. You can set the automatic recovery behavior to `disabled` during or after launching the instance.

The `default` configuration doesn't enable simplified automatic recovery for an unsupported instance.

Console

To disable simplified automatic recovery at launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then choose **Launch instance**.
3. In the **Advanced details** section, for **Instance auto-recovery**, choose **Disabled**.
4. Configure the remaining instance launch settings as needed and then launch the instance.

To disable simplified automatic recovery after launch

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 then choose **Actions**, **Instance settings**, **Change auto-recovery behavior**.

4. Choose **Off**, and then choose **Save**.

To enable simplified automatic recovery after launch

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 then choose **Actions**, **Instance settings**, **Change auto-recovery behavior**.
4. Choose **Default (On)**, and then choose **Save**.

AWS CLI

To disable simplified automatic recovery at launch

Use the run-instances command with the `--maintenance-options` option.

```
--maintenance-options AutoRecovery=Disabled
```

To disable simplified automatic recovery after launch

Use the modify-instance-maintenance-options command.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery disabled
```

To enable simplified automatic recovery after launch

Use the modify-instance-maintenance-options command.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery default
```

PowerShell

To disable simplified automatic recovery at launch

Use the New-EC2Instance cmdlet.

```
-MaintenanceOptions_AutoRecovery Disabled
```

To disable simplified automatic recovery after launch

Use the Edit-EC2InstanceMaintenanceOption cmdlet.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Disabled
```

To enable simplified automatic recovery after launch

Use the Edit-EC2InstanceMaintenanceOption cmdlet.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Enabled
```

Troubleshoot simplified automatic recovery failures

If simplified automatic recovery fails to recover your instance, consider the following issues:

- AWS service events are running

  Simplified automatic recovery does not operate during service events in the AWS Health Dashboard. You might not receive recovery failure notifications for such events. For the latest service availability information, see the Service health status page.
- Insufficient capacity

  There is temporarily insufficient replacement hardware to migrate the instance.
- Maximum daily recovery attempts reached

  The instance has reached the maximum daily allowance for recovery attempts. Your instance might subsequently be retired if automatic recovery fails and a hardware degradation is determined to be the root cause of the original failed system status check.

If the instance's system status check failure persists despite multiple recovery attempts, see Troubleshoot instances with failed status checks for additional guidance.

Configure CloudWatch action based recovery on an EC2 instance

> **Important**
>
> This section describes how to proactively configure recovery mechanisms on an EC2 instance. These recovery mechanisms are designed to restore instance availability when AWS detects an underlying hardware or software issue that causes a system status check to fail. If you are currently experiencing problems accessing your instance, see Troubleshoot EC2 instances.

If AWS detects that an instance is unavailable due to an underlying hardware or software issue, *CloudWatch action based recovery* can automatically restore instance availability by moving the instance from the host with the underlying issue to a different host.

If CloudWatch action based recovery occurs, AWS sends one of the following events to your AWS Health Dashboard, depending on the outcome:

- Success event: `AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS`
- Failure event: `AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE`

You can configure CloudWatch action based recovery to add recovery actions to Amazon CloudWatch alarms. CloudWatch action based recovery works with the `StatusCheckFailed_System` metric. CloudWatch action based recovery provides to-the-minute recovery response time granularity and Amazon Simple Notification Service (Amazon SNS) notifications of recovery actions and outcomes. These configuration options allow for faster recovery attempts with more granular control over the system status check failure event response compared to simplified automatic recovery. For more information about available CloudWatch options, see Status checks for your instances.

However, CloudWatch action based recovery can only operate if an instance is in the `running` state, there are no service events listed in the AWS Health Dashboard, and there is available capacity for the instance type. In some situations, such as significant outages, capacity constraints might cause recovery attempts to fail. For more information, see the section called “Troubleshoot”.

> **Warning**
>
> When AWS recovers your instance due to an underlying hardware or software issue, be aware of the following consequences: data stored in volatile memory (RAM) and on instance store volumes will be lost, and the operating system's uptime will start over from zero. To help protect against data loss, we recommend that you regularly create backups of valuable data. For more information about backup and recovery best practices for EC2 instances, see Best practices for Amazon EC2.
> Automatic instance recovery mechanisms are designed for *individual instances*. For guidance on building a resilient *system*, see Build a resilient system.

Contents

Requirements for enabling CloudWatch action based recovery

CloudWatch action based recovery can be enabled on instances that meet the following criteria:

Instance types

- **General purpose:** A1, M3, M4, M5, M5a, M5n, M5zn, M6a, M6g, M6i, M6in, M7a, M7g, M7i, M7i-flex, M8g, M8i, M8i-flex, T1, T2, T3, T3a, T4g
- **Compute optimized:** C3, C4, C5, C5a, C5n, C6a, C6g, C6gn, C6i, C6in, C7a, C7g, C7gn, C7i, C7i-flex, C8g, C8gn
- **Memory optimized:** R3, R4, R5, R5a, R5b, R5n, R6a, R6g, R6i, R6in, R7a, R7g, R7i, R7iz, R8g, R8i, R8i-flex, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, U7inh-32tb, X1, X1e, X2idn, X2iedn, X2iezn, X8g
- **Accelerated computing:** G3, G5g, Inf1, P3, VT1
- **High-performance computing:** Hpc6a, Hpc7a, Hpc7g
- **Metal instances:** Any of the above instance types with the metal instance size.
- **If instance store volumes are added at launch:** Then only the following instance types are supported: M3, C3, R3, X1, X1e, X2idn, X2iedn

Tenancy

- Shared
- Dedicated Instance

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

Limitations

CloudWatch action based recovery is not supported for instances with the following characteristics:

- Tenancy: Dedicated Host. For Dedicated Hosts, use Dedicated Host Auto Recovery instead.
- Networking: Instances using an Elastic Fabric Adapter
- Auto Scaling: Instances that are part of an Auto Scaling group
- Maintenance: Instances currently undergoing a scheduled maintenance event

Find a supported instance type

You can view the instance types that support CloudWatch action based recovery.

Console

To view the instance types that support CloudWatch action based recovery

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instance Types**.
3. In the filter bar, add the filter **Auto Recovery support = true**. The **Instance types** table displays all the instance types that support CloudWatch action based recovery.
4. (Optional) Add filters to further scope to specific instance types of interest.

AWS CLI

To view the instance types that support CloudWatch action based recovery

Use the describe-instance-types command with the `auto-recovery-supported` filter.

```
aws ec2 describe-instance-types \
    --filters Name=auto-recovery-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
```

```
    --output text | sort
```

PowerShell

To view the instance types that support CloudWatch action based recovery

Use the Get-EC2InstanceType cmdlet with the `auto-recovery-supported` filter.

```
Get-EC2InstanceType `
    -Filter @{Name="auto-recovery-supported";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

Configure CloudWatch action based recovery

To configure CloudWatch action based recovery for an EC2 instance, create a CloudWatch alarm that monitors the `StatusCheckFailed_System` metric for the specified instance. Set the alarm to trigger when the metric value is **1**, indicating a failed system status check. Configure the alarm action to automatically recover the instance when triggered.

You can configure the alarm using either the Amazon EC2 console or the CloudWatch console. For the instructions, see Add recover actions to Amazon CloudWatch alarms in this user guide, or Adding recover actions to Amazon CloudWatch alarms in the *Amazon CloudWatch User Guide*.

Troubleshoot CloudWatch action based recovery failures

If CloudWatch action based recovery fails to recover your instance, consider the following issues:

- AWS service events are running

  CloudWatch action based recovery does not operate during service events in the AWS Health Dashboard. You might not receive recovery failure notifications for such events. For the latest service availability information, see the Service health status page.
- Insufficient capacity

  There is temporarily insufficient replacement hardware to migrate the instance.
- Maximum daily recovery attempts reached

  The instance has reached the maximum daily allowance for recovery attempts. Your instance might subsequently be retired if automatic recovery fails and a hardware degradation is determined to be the root cause of the original failed system status check.

If the instance's system status check failure persists despite multiple recovery attempts, see Troubleshoot instances with failed status checks for additional guidance.

Use instance metadata to manage your EC2 instance

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata includes the following:

Instance metadata properties

Instance metadata properties are divided into categories, for example, host name, events, and security groups.

Dynamic data

Dynamic data is metadata that's generated when the instance is launched, such as an instance identity document. For more information, see Dynamic data categories.

User data

You can also use instance metadata to access *user data* that you specified when you launched your instance. For example, you can specify parameters for configuring your instance, or include a simple script. You can also 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 an 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 multiple instances using the same `RunInstances` call, 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, so that you can write code that controls what the instances do. For example, the first host might elect itself as the original node in a cluster. For a detailed AMI launch example, see Identify each instance launched in a single request.

> **Important**
>
> Although you can only access instance metadata and user data from within the instance itself, the data is not protected by authentication or cryptographic methods. Anyone who has direct access to the instance, and potentially any software running on the instance,

can view its metadata. Therefore, you should not store sensitive data, such as passwords or long-lived encryption keys, as user data.

Contents

Instance metadata categories

Instance metadata properties are divided into categories. To retrieve instance metadata properties, you specify the category in the request, and the metadata is returned in the response.

When new categories are released, a new instance metadata build is created with a new version number. In the following table, the **Version when category was released** column specifies the build version when an instance metadata category was released. To avoid having to update your code every time Amazon EC2 releases a new instance metadata build, use `latest` instead of the version number in your metadata requests. For more information, see Get the available versions of the instance metadata.

When Amazon EC2 releases a new instance metadata category, the instance metadata for the new category might not be available for existing instances. With Nitro-based instances, 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 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.

Category	Description	Version when category was released
ami-id	The AMI ID used to launch the instance.	1.0
ami-launch-index	If you launch multiple instances using the same RunInstances call, this value indicates the launch order for each instance. The value of the first instance launched is 0. If you launch instances using Auto Scaling or EC2 fleet, this value is always 0.	1.0
ami-manifest-path	The path to the AMI manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is unknown.	1.0
ancestor-ami-ids	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the AMI manifest file contained an ancestor-amis key.	2007-10-10
autoscaling/target-lifecycle-state	Value showing the target Auto Scaling lifecycle state that an Auto Scaling instance is transitioning to. Present when the instance transitions to one of the target lifecycle states after March 10, 2022. Possible values: Detached | InService | Standby | Terminated | Warmed:Hi	2021-07-15

Category	Description	Version when category was released
	`bernated` | `Warmed:Running` | `Warmed:Stopped` | `Warmed:Terminated` . See [Retrieve the target lifecycle state through instance metadata]() in the *Amazon EC2 Auto Scaling User Guide*.	
`block-device-mapping/ami`	The virtual device that contains the root/boot file system.	2007-12-15
`block-device-mapping/ebs`*N*	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 *N* indicates the index of the Amazon EBS volume (such as `ebs1` or `ebs2`).	2007-12-15

Category	Description	Version when category was released
`block-device-mapping/ephemeral N`	The virtual devices for any non-NVMe instance store volumes. The *N* indicates the index of each volume. The number of instance store volumes in the block device mapping might not match the actual number of instance store volumes for the instance. The instance type determines the number of instance store volumes that are available to an instance. If the number of instance store volumes in a block device mapping exceeds the number available to an instance, the additional instance store volumes are ignored.	2007-12-15
`block-device-mapping/root`	The virtual devices or partitions associated with the root devices or partitions on the virtual device, where the root (/ or C:) file system is associated with the given instance.	2007-12-15
`block-device-mapping/swap`	The virtual devices associated with `swap`. Not always present.	2007-12-15
`events/maintenance/history`	If there are completed or canceled maintenance events for the instance, contains a JSON string with information about the events.	2018-08-17

Category	Description	Version when category was released
`events/maintenance/scheduled`	If there are active maintenance events for the instance, contains a JSON string with information about the events. For more information, see [View scheduled events that affect your Amazon EC2 instances]().	2018-08-17
`events/recommendations/rebalance`	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: `{"noticeTime": "2020-11-05T08:22:00Z"}`. This category is available only after the notification is emitted. For more information, see [EC2 instance rebalance recommendations]().	2020-10-27

Category	Description	Version when category was released
`hostname`	If the EC2 instance is using IP-based naming (IPBN), this is the private IPv4 DNS hostname of the instance. If the EC2 instance is using Resource-based naming (RBN), this is the RBN. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). For more information about IPBN and RBN, see [EC2 instance hostnames and domains]().	1.0
`iam/info`	If there is an IAM role associated with the instance, contains information about the last time the instance profile was updated, including the instance's LastUpdated date, InstanceProfileArn, and InstanceProfileId. Otherwise, not present.	2012-01-12
`iam/security-credentials/role-name`	If there is an IAM role associated with the instance, *role-name* is the name of the role, and *role-name* contains the temporary security credentials associated with the role (for more information, see [Retrieve security credentials from instance metadata]()). Otherwise, not present.	2012-01-12

Category	Description	Version when category was released
`identity-credentials/ec2/info`	Information about the credentials in `identity-credentials/ec2/security-credentials/ec2-instance`.	2018-05-23
`identity-credentials/ec2/security-credentials/ec2-instance`	Credentials for the instance identity role that allow on-instance software to identify itself to AWS to support features such as EC2 Instance Connect and AWS Systems Manager Default Host Management Configuration. These credentials have no policies attached, so they have no additional AWS API permissions beyond identifying the instance to the AWS feature. For more information, see Instance identity roles for Amazon EC2 instances.	2018-05-23
`instance-action`	Notifies the instance that it should reboot in preparation for bundling. Valid values: `none` | `shutdown` | `bundle-pending`.	2008-09-01
`instance-id`	The ID of this instance.	1.0
`instance-life-cycle`	The purchasing option of this instance. For more information, see Amazon EC2 billing and purchasing options.	2019-10-01
`instance-type`	The type of instance. For more information, see Amazon EC2 instance types.	2007-08-29

Category	Description	Version when category was released
`ipv6`	The IPv6 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) network interface and the first IPv6 address assigned. If no IPv6 address exists on network interface[0], this item is not set and results in an HTTP 404 response.	2021-01-03
`kernel-id`	The ID of the kernel launched with this instance, if applicable.	2008-02-01
`local-hostname`	In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). If the EC2 instance is using IP-based naming (IPBN), this is the private IPv4 DNS hostname of the instance. If the EC2 instance is using Resource-based naming (RBN), this is the RBN. For more information about IPBN, RBN, and EC2 instance naming, see [EC2 instance hostnames and domains]().	2007-01-19

Category	Description	Version when category was released
`local-ipv4`	The private IPv4 address of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). If this is an IPv6-only instance, this item is not set and results in an HTTP 404 response.	1.0
`mac`	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2011-01-01
`metrics/vhostmd`	No longer available.	2011-05-01
`network/interfaces/macs/mac/device-number`	The unique device number associated with that interface. The device number corresponds to the device name; for example, a `device-number` of 2 is for the eth2 device. This category corresponds to the `DeviceIndex` and `device-index` fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01
`network/interfaces/macs/mac/interface-id`	The ID of the network interface.	2011-01-01

Category	Description	Version when category was released
network/interfaces/macs/mac/ipv4-associations/public-ip	The private IPv4 addresses that are associated with each public IP address and assigned to that interface.	2011-01-01
network/interfaces/macs/mac/ipv6s	The IPv6 addresses assigned to the interface.	2016-06-30
network/interfaces/macs/mac/ipv6-prefix	The IPv6 prefix assigned to the network interface.	
network/interfaces/macs/mac/local-hostname	The private IPv4 DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). If this is a IPv6-only instance, this is the resource-based name. For more information about IPBN and RBN, see EC2 instance hostnames and domains.	2007-01-19
network/interfaces/macs/mac/local-ipv4s	The private IPv4 addresses associated with the interface. If this is an IPv6-only network interface, this item is not set and results in an HTTP 404 response.	2011-01-01
network/interfaces/macs/mac/mac	The instance's MAC address.	2011-01-01
network/interfaces/macs/*mac*/network-card	The index of the network card. Some instance types support multiple network cards.	2020-11-01

Category	Description	Version when category was released
`network/interfaces/macs/mac/owner-id`	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface o wner.	2011-01-01
`network/interfaces/macs/mac/public-hostname`	The interface's public DNS (IPv4). This category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*. If the instance only has a public-IPv6 address and no public-IPv4 address, this item is not set and results in an HTTP 404 response.	2011-01-01
`network/interfaces/macs/mac/public-ipv4s`	The public IP address or Elastic IP addresses associated with the interface. There may be multiple IPv4 addresses on an instance.	2011-01-01
`network/interfaces/macs/mac/security-groups`	Security groups to which the network interface belongs.	2011-01-01
`network/interfaces/macs/mac/security-group-ids`	The IDs of the security groups to which the network interface b elongs.	2011-01-01

Category	Description	Version when category was released
network/interfaces/macs/mac/subnet-id	The ID of the subnet in which the interface resides.	2011-01-01
network/interfaces/macs/mac/subnet-ipv4-cidr-block	The IPv4 CIDR block of the subnet in which the interface resides.	2011-01-01
network/interfaces/macs/mac/subnet-ipv6-cidr-blocks	The IPv6 CIDR block of the subnet in which the interface resides.	2016-06-30
network/interfaces/macs/mac/vpc-id	The ID of the VPC in which the interface resides.	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-block	The primary IPv4 CIDR block of the VPC.	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-blocks	The IPv4 CIDR blocks for the VPC.	2016-06-30
network/interfaces/macs/mac/vpc-ipv6-cidr-blocks	The IPv6 CIDR block of the VPC in which the interface resides.	2016-06-30
placement/availability-zone	The Availability Zone in which the instance launched.	2008-02-01
placement/availability-zone-id	The static Availability Zone ID in which the instance is launched. T he Availability Zone ID is consisten t across accounts. However, it might be different from the Availability Zone, which can vary by account.	2019-10-01

Category	Description	Version when category was released
`placement/group-name`	The name of the placement group in which the instance is launched.	2020-08-24
`placement/host-id`	The ID of the host on which the instance is launched. Applicable only to Dedicated Hosts.	2020-08-24
`placement/partition-number`	The number of the partition in which the instance is launched.	2020-08-24
`placement/region`	The AWS Region in which the instance is launched.	2020-08-24
`product-codes`	AWS Marketplace product codes associated with the instance, if any.	2007-03-01
`public-hostname`	The instance's public DNS (IPv4). This category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*. If the instance only has a public-IPv6 address and no public-IPv4 address, this item is not set and results in an HTTP 404 response.	2007-01-19
`public-ipv4`	The public IPv4 address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19

Category	Description	Version when category was released
public-keys/0/openssh-key	Public key. Only available if supplied at instance launch time.	1.0
ramdisk-id	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
reservation-id	The ID of the reservation.	1.0
security-groups	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.	1.0
services/domain	The domain for AWS resources for the Region.	2014-02-25
services/partition	The partition that the resource is in. For standard AWS Regions, the partition is aws. If you have resources in other partitions, the partition is aws-*partition name* . For example, the partition for resources in the China (Beijing) Region is aws-cn.	2015-10-20

Category	Description	Version when category was released
`spot/instance-action`	The action (hibernate, stop, or terminate) and the approximate time, in UTC, when the action will occur. This item is present only if the Spot Instance has been marked for hibernate, stop, or terminate. For more information, see instance-action.	2016-11-15
`spot/termination-time`	The approximate time, in UTC, that the operating system for your Spot Instance will receive the shutdown signal. This item is present and contains a time value (for example, 2015-01-05T18:02:00Z) only if the Spot Instance has been marked for termination by Amazon EC2. The termination-time item is not set to a time if you terminated the Spot Instance yourself. For more information, see termination-time.	2014-11-05
`tags/instance`	The instance tags associated with the instance. Only available if you explicitly allow access to tags in instance metadata. For more information, see Enable access to tags in instance metadata.	2021-03-23

Dynamic data categories

The following table lists the categories of dynamic data.

Category	Description	Version when category was released
fws/instance-monitoring	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: enabled | disabled	2009-04-04
instance-identity/document	JSON containing instance attributes, such as instance-id, private IP address, etc. See Instance identity documents for Amazon EC2 instances.	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance identity documents for Amazon EC2 instances.	2009-04-04
instance-identity/signature	Data that can be used by other parties to verify its origin and authenticity. See Instance identity documents for Amazon EC2 instances.	2009-04-04

Access instance metadata for an EC2 instance

You can access EC2 instance metadata from inside of the instance itself or from the EC2 console, API, SDKs, or the AWS CLI. To get the current instance metadata settings for an instance from the console or command line, see Query instance metadata options for existing instances.

You can also modify user data for instances with an EBS root volume. The instance must be in the stopped state. For console directions, see Update the instance user data. For a Linux example that uses the AWS CLI, see modify-instance-attribute. For a Windows example that uses the Tools for Windows PowerShell, see the section called “User data and the Tools for Windows PowerShell”.

> **Note**
>
> You are not billed for HTTP requests used to retrieve instance metadata and user data.

Instance metadata access considerations

To avoid problems with instance metadata retrieval, consider the following.

Command format

The command format is different, depending on whether you use Instance Metadata Service Version 1 (IMDSv1) or Instance Metadata Service Version 2 (IMDSv2). By default, you can use both versions of the Instance Metadata Service. To require the use of IMDSv2, see Use the Instance Metadata Service to access instance metadata.

If IMDSv2 is required, IMDSv1 does not work

If you use IMDSv1 and receive no response, it's likely that IMDSv2 is required. To check whether IMDSv2 is required, select the instance to view its details. The **IMDSv2** value indicates either **Required** (you must use IMDSv2) or **Optional** (you can use either IMDSv2 or IMDSv1).

(IMDSv2) Use /latest/api/token to retrieve the token

Issuing PUT requests to any version-specific path, for example `/2021-03-23/api/token`, results in the metadata service returning 403 Forbidden errors. This behavior is intended.

Metadata version

To avoid having to update your code every time Amazon EC2 releases a new instance metadata build, we recommend that you use `latest` in the path, and not the version number.

IPv6 support

To retrieve instance metadata using an IPv6 address, ensure that you enable and use the IPv6 address of the IMDS `[fd00:ec2::254]` instead of the IPv4 address `169.254.169.254`. The instance must be a Nitro-based instance launched in a subnet that supports IPv6.

(Windows) Create custom AMIs using Windows Sysprep

To ensure that IMDS works when you launch an instance from a custom Windows AMI, the AMI must be a standardized image created with Windows Sysprep. Otherwise, the IMDS won't work. For more information, see Create an Amazon EC2 AMI using Windows Sysprep.

In a container environment, consider reconfiguration or increasing the hop limit to 2

The AWS SDKs use IMDSv2 calls by default. If the IMDSv2 call receives no response, some AWS SDKs retry the call and, if still unsuccessful, use IMDSv1. This can result in a delay, especially in a container environment. For those AWS SDKs that *require* IMDSv2, if the hop limit is 1 in a container environment, the call might not receive a response at all because going to the container is considered an additional network hop.

To mitigate these issues in a container environment, consider changing the configuration to pass settings (such as the AWS Region) directly to the container, or consider increasing the hop limit to 2. For information about the hop limit impact, see Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service. For information about changing the hop limit, see Change the PUT response hop limit.

Packets per second (PPS) limit

There is a 1024 packet per second (PPS) limit to services that use link-local addresses. This limit includes the aggregate of Route 53 Resolver DNS Queries, Instance Metadata Service (IMDS) requests, Amazon Time Service Network Time Protocol (NTP) requests, and Windows Licensing Service (for Microsoft Windows based instances) requests.

Additional considerations for user data access

- User data is treated as opaque data: what you specify is what you get back upon retrieval. It is up to the instance to interpret and act on user data.
- User data must be base64-encoded. Depending on the tool or SDK that you're using, the base64-encoding might be performed for you. For example:
  - The Amazon EC2 console can perform the base64-encoding for you or accept base64-encoded input.
  - AWS CLI version 2 performs base64-encoding of binary parameters for you by default. AWS CLI version 1 performs the base64-encoding of the `--user-data` parameter for you.
  - The AWS SDK for Python (Boto3) performs base64-encoding of the `UserData` parameter for you.
- 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 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.
- 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. With Windows instances, 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.
- User data is an instance attribute. If you create an AMI from an instance, the instance user data is not included in the AMI.

Access instance metadata from within an EC2 instance

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.

All of the following are considered instance metadata, but they are accessed in different ways. Select the tab that represents the type of instance metadata you want to access to see more information.

Metadata

Instance metadata properties are divided into categories. For a description of each instance metadata category, see Instance metadata categories.

To access instance metadata properties from within a running instance, get the data from the following IPv4 or IPv6 URIs. These IP addresses are link-local addresses and are valid only from the instance. For more information, see Link-local addresses.

IPv4

```
http://169.254.169.254/latest/meta-data/
```

IPv6

```
http://[fd00:ec2::254]/latest/meta-data/
```

Dynamic data

To retrieve dynamic data from within a running instance, use one of the following URIs.

IPv4

```
http://169.254.169.254/latest/dynamic/
```

IPv6

```
http://[fd00:ec2::254]/latest/dynamic/
```

Examples: Access with cURL

The following examples use cURL to retrieve the high-level instance identity categories.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

Examples: Access with PowerShell

The following examples use PowerShell to retrieve the high-level instance identity categories.

IMDSv2

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

For more information about dynamic data and examples of how to retrieve it, see Instance identity documents for Amazon EC2 instances.

User data

To retrieve user data from an instance, use one of the following URIs. To retrieve user data using the IPv6 address, you must enable it, and the instance must be a Nitro-based instance in a subnet that supports IPv6.

IPv4

```
http://169.254.169.254/latest/user-data
```

IPv6

```
http://[fd00:ec2::254]/latest/user-data
```

A request for user data returns the data as it is (content type `application/octet-stream`). If the instance does not have any user data, the request returns `404 - Not Found`.

Examples: Access with cURL to retrieve comma-separated text

The following examples use cURL to retrieve user data that was specified as comma-separated text.

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

Examples: Access with PowerShell to retrieve comma-separated text

The following examples use PowerShell to retrieve user data that was specified as comma-separated text.

IMDSv2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
-Method PUT -Uri http://169.254.169.254/latest/api/token} -Method GET -uri http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

Examples: Access with cURL to retrieve a script

The following examples use cURL to retrieve user data that was specified as a script.

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

Examples: Access with PowerShell to retrieve a script

The following examples use PowerShell to retrieve user data that was specified as a script.

IMDSv2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

IMDSv1

```
Invoke-RestMethod -uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
```

```
</powershell>
<persist>true</persist>
```

Query instance metadata options for existing instances

You can query the instance metadata options for your existing instances.

Console

To query the instance metadata options 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 your instance and check the following fields:

   - **IMDSv2 –** The value is either **Required** or **Optional**.
   - **Allow tags in instance metadata –** The value is either **Enabled** or **Disabled**.
4. With your instance selected, choose **Actions**, **Instance settings**, **Modify instance metadata options**.

   The dialog box displays whether the instance metadata service is enabled or disabled for the selected instance.

AWS CLI

To query the instance metadata options for an existing instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-id i-1234567898abcdef0 \
    --query 'Reservations[].Instances[].MetadataOptions'
```

PowerShell

To query the instance metadata options for an existing instance using the Tools for PowerShell

Use the Get-EC2Instance Cmdlet.

```
(Get-EC2Instance `
    -InstanceId i-1234567898abcdef0).Instances.MetadataOptions
```

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).

If an IMDSv1 request receives no response, it's likely that IMDSv2 is required.

For requests made using IMDSv2, the following HTTP error codes can be returned:

- `400 - Missing or Invalid Parameters` – The `PUT` request is not valid.
- `401 - Unauthorized` – The `GET` request uses an invalid token. The recommended action is to generate a new token.
- `403 - Forbidden` – The request is not allowed or the IMDS is turned off.
- `404 - Not Found` – The resource is not available or there is no such resource.
- `503` – The request could not be completed. Retry the request.

If the IMDS returns an error, **curl** prints the error message in the output and returns a success status code. The error message is stored in the `TOKEN` variable, which causes **curl** commands that use the token to fail. If you call **curl** with the **-f** option, it returns an error status code in the event of an HTTP server error. If you enable error handling, the shell can catch the error and stop the script.

Query throttling

We throttle queries to the IMDS on a per-instance basis, and we place limits on the number of simultaneous connections from an instance to the IMDS.

If you're using the IMDS 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. For more information about IAM role and security credentials associated with the role, see Retrieve security credentials from instance metadata.

If you are throttled while accessing the IMDS, retry your query with an exponential backoff strategy.

Use the Instance Metadata Service to access instance metadata

You can access instance metadata from a running instance using one of the following methods:

- Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method

  For examples, see Examples for IMDSv2.
- Instance Metadata Service Version 1 (IMDSv1) – a request/response method

  For examples, see Examples for IMDSv1.

By default, you can use either IMDSv1 or IMDSv2, or both.

You can configure the Instance Metadata Service (IMDS) on each instance so that local code or users must use IMDSv2. When you specify that IMDSv2 must be used, IMDSv1 no longer works. For information about how to configure your instance to use IMDSv2, see Configure the Instance Metadata Service options.

The PUT or GET headers are unique to IMDSv2. If these headers are present in the request, then the request is intended for IMDSv2. If no headers are present, it is assumed the request is intended for IMDSv1.

For an extensive review of IMDSv2, see Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service.

Topics

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.

> **Note**
>
> The examples in this section use the IPv4 address of the Instance Metadata Service (IMDS): `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 IMDS is compatible with IMDSv2 commands. The IPv6 address is only accessible on Nitro-based instances in an IPv6-supported subnet (dual stack or IPv6 only).

The following examples use a shell script and IMDSv2 to retrieve the top-level instance metadata items. Each 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` (Linux instances) or `token` (Windows instances)
- Requests the top-level metadata items using the token

Linux example

You can run two separate commands, or combine them.

Separate commands

First, generate a token using the following command.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
```

Then, use the token to generate top-level metadata items using the following command.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

Combined commands

You can store the token and combine the commands. The following example combines the above two commands and stores the session token header in a variable named TOKEN.

> **Note**
>
> If there is an error in creating the token, instead of a valid token, an error message is stored in the variable, and the command will not work.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

After you've created a token, you can reuse it until it expires. In the following example command, which gets the ID of the AMI used to launch the instance, the token that is stored in `$TOKEN` in the previous example is reused.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/ami-id
```

Windows example

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/meta-data/
```

After you've created a token, you can reuse it until it expires. In the following example command, which gets the ID of the AMI used to launch the instance, the token that is stored in `$token` in the previous example is reused.

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
 -Method GET -uri http://169.254.169.254/latest/meta-data/ami-id
```

When you use IMDSv2 to request instance metadata, the request must include the following:

1. Use a PUT request to initiate a session to the instance metadata service. The PUT request returns a token that must be included in subsequent GET requests to the instance metadata service. The token is required to access metadata using IMDSv2.
2. Include the token in all GET requests to the IMDS. When token usage is set to `required`, requests without a valid token or with an expired token receive a `401 - Unauthorized` HTTP error code.
   - 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 IMDS. 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 IMDS connection and throttling limits. For more information, see Query throttling.

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. If you need a bigger hop limit, you can adjust it by using the modify-instance-metadata-options AWS CLI command. For example, you might need a bigger hop limit for backward compatibility with container services running on the instance. For more information, see Modify instance metadata options for existing instances.

Use a supported AWS SDK

To use IMDSv2, your EC2 instances must use an AWS SDK version that supports using IMDSv2. The latest versions of all the AWS SDKs support using IMDSv2.

> **Important**
>
> We recommend that you to stay up to date with SDK releases to keep up with the latest features, security updates, and underlying dependencies. Continued use of an unsupported SDK version is not recommended and is done at your discretion. For more information, see the AWS SDKs and Tools maintenance policy in the *AWS SDKs and Tools Reference Guide*.

The following are the minimum versions that support using IMDSv2:

- AWS CLI – 1.16.289
- AWS Tools for Windows PowerShell – 4.0.1.0
- AWS SDK for .NET – 3.3.634.1
- AWS SDK for C++ – 1.7.229
- AWS SDK for Go – 1.25.38
- AWS SDK for Go v2 – 0.19.0
- AWS SDK for Java – 1.11.678
- AWS SDK for Java 2.x – 2.10.21
- AWS SDK for JavaScript in Node.js – 2.722.0
- AWS SDK for Kotlin – 1.1.4
- AWS SDK for PHP – 3.147.7
- AWS SDK for Python (Botocore) – 1.13.25
- AWS SDK for Python (Boto3) – 1.12.6
- AWS SDK for Ruby – 3.79.0

Examples for IMDSv2

Run the following examples on your Amazon EC2 instance to retrieve the instance metadata for IMDSv2.

On Windows instances, you can use Windows PowerShell or you can install cURL or wget. If you install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the calls and the output might be different from what is described here.

Examples

- Get the available versions of the instance metadata
- Get the top-level metadata items
- Get the values for metadata items
- Get the list of available public keys
- Show the formats in which public key 0 is available
- Get public key 0 (in the OpenSSH key format)
- Get the subnet ID for an instance
- Get the instance tags for an instance

Get the available versions of the instance metadata

This example gets the available versions of the instance metadata. Each version refers to an instance metadata build when new instance metadata categories were released. The instance metadata build versions do not correlate with the Amazon EC2 API versions. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
```

```
...
latest
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

Get the top-level metadata items

This example gets the top-level metadata items. For more information about the items in the response, see Instance metadata categories.

Note that tags are included in this output only if you've allowed access. For more information, see the section called "Enable access to tags in instance metadata".

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
```

```
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

Get the values for metadata items

These examples get the values of some of the top-level metadata items that were obtained in the preceding example. These requests use the stored token that was created using the command in the previous example. The token must not be expired.

cURL

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/reservation-id
r-0efghijk987654321
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

Get the list of available public keys

This example gets the list of available public keys.

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

Show the formats in which public key 0 is available

This example shows the formats in which public key 0 is available.

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
openssh-key
```

Get public key 0 (in the OpenSSH key format)

This example gets public key 0 (in the OpenSSH key format).

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
```

```
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

Get the subnet ID for an instance

This example gets the subnet ID for an instance.

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

Get the instance tags for an instance

If access to instance tags in the instance metadata is turned on, you can get the tags for a instance from instance metadata. For more information, see Retrieve tags from instance metadata.

Examples for IMDSv1

Run the following examples on your Amazon EC2 instance to retrieve the instance metadata for IMDSv1.

On Windows instances, you can use Windows PowerShell or you can install cURL or wget. If you install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the calls and the output might be different from what is described here.

Examples

- Get the available versions of the instance metadata
- Get the top-level metadata items
- Get the values for metadata items
- Get the list of available public keys
- Show the formats in which public key 0 is available
- Get public key 0 (in the OpenSSH key format)
- Get the subnet ID for an instance
- Get the instance tags for an instance

Get the available versions of the instance metadata

This example gets the available versions of the instance metadata. Each version refers to an instance metadata build when new instance metadata categories were released. The instance metadata build versions do not correlate with the Amazon EC2 API versions. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
```

```
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

Get the top-level metadata items

This example gets the top-level metadata items. For more information about the items in the response, see Instance metadata categories.

Note that tags are included in this output only if you've allowed access. For more information, see the section called "Enable access to tags in instance metadata".

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
```

```
services/
tags/
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

Get the values for metadata items

These examples get the values of some of the top-level metadata items that were obtained in the previous example.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id
```

```
r-0efghijk987654321
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/reservation-
id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/local-
hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

Get the list of available public keys

This example gets the list of available public keys.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/ 0=my-public-key
```

Show the formats in which public key 0 is available

This example shows the formats in which public key 0 is available.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
openssh-key
```

Get public key 0 (in the OpenSSH key format)

This example gets public key 0 (in the OpenSSH key format).

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
```

```
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

Get the subnet ID for an instance

This example gets the subnet ID for an instance.

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/network/
interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

Get the instance tags for an instance

If access to instance tags in the instance metadata is turned on, you can get the tags for a instance from instance metadata. For more information, see Retrieve tags from instance metadata.

Transition to using Instance Metadata Service Version 2

If you want to migrate your instances so that local code or users must use Instance Metadata Service Version 2 (IMDSv2), we recommend that you use the following tools and transition path.

Topics

- Tools for helping with the transition to IMDSv2
- Recommended path to requiring IMDSv2

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 CLI and AWS SDKs support IMDSv2. To use IMDSv2, make sure that your EC2 instances have the latest versions of the CLI and SDKs. For information about updating the CLI, see Installing or updating to the latest version of the AWS CLI in the *AWS Command Line Interface User Guide*.

All Amazon Linux 2 and Amazon Linux 2023 software packages support IMDSv2. In Amazon Linux 2023, IMDSv1 is disabled by default.

For the minimum AWS SDK versions that support IMDSv2, see Use a supported AWS SDK.

IMDS Packet Analyzer

The IMDS Packet Analyzer is an open-sourced tool that identifies and logs IMDSv1 calls from your instance's boot phase. This can assist in identifying the software making IMDSv1 calls on EC2 instances, allowing you to pinpoint exactly what you need to update to get your instances ready to use IMDSv2 only. You can run IMDS Packet Analyzer from a command line or install it as a service. For more information, see AWS ImdsPacketAnalyzer on *GitHub*.

CloudWatch

IMDSv2 uses token-backed sessions, while IMDSv1 does not. The `MetadataNoToken` CloudWatch metric tracks the number of calls to the Instance Metadata Service (IMDS) 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.

After you've disabled IMDSv1, you can use the `MetadataNoTokenRejected` CloudWatch metric to track the number of times an IMDSv1 call was attempted and rejected. By tracking this metric, you can ascertain whether your software needs to be updated to use IMDSv2.

For more information, see Instance metrics.

Updates to EC2 APIs and CLIs

For new instances, you can use the RunInstances API to launch new instances that require the use of IMDSv2. For more information, see Configure instance metadata options for new instances.

For existing instances, you can use the ModifyInstanceMetadataOptions API to require the use of IMDSv2. For more information, see Modify instance metadata options for existing instances.

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. The Auto Scaling group launches new instances using the new launch template or launch configuration, but existing instances are not affected. For existing instances in an Auto Scaling group, you can use 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 new launch template or launch configuration.

Use an AMI that configures IMDSv2 by default

When you launch an instance, you can automatically configure it to use IMDSv2 by default (the `HttpTokens` parameter is set to `required`) by launching it with an AMI that is configured with the `ImdsSupport` parameter set to `v2.0`. You can set the `ImdsSupport` parameter to `v2.0` when you register the AMI using the register-image CLI command, or you can modify an existing AMI by using the modify-image-attribute CLI command. For more information, see Configure the AMI.

IAM policies and SCPs

You can use an IAM policy or AWS Organizations service control policy (SCP) to control users as follows:

- Can't launch an instance using the RunInstances API unless the instance is configured to use IMDSv2.
- Can't modify a running instance using the ModifyInstanceMetadataOptions API to re-enable IMDSv1.

The IAM policy or SCP must contain the following IAM condition keys:

- `ec2:MetadataHttpEndpoint`

- `ec2:MetadataHttpPutResponseHopLimit`
- `ec2:MetadataHttpTokens`

If a parameter in the API or CLI call does not match the state specified in the policy that contains the condition key, the API or CLI call fails with an `UnauthorizedOperation` response.

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 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](). For more information on SCPs, see [Service control policies]() in the *AWS Organizations User Guide*.

Recommended path to requiring IMDSv2

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 versions compatible with IMDSv2. For more information about updating the CLI, see [Installing or updating 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. You can use the [IMDS Packet Analyzer]() to identify the software that you need to change to use IMDSv2 requests.

Step 2: Track your transition progress

Track your transition progress by using the CloudWatch metric `MetadataNoToken`. This metric shows the number of IMDSv1 calls to the IMDS on your instances. For more information, see [Instance metrics]().

Step 3: When there is zero IMDSv1 usage

When the CloudWatch metric `MetadataNoToken` records zero IMDSv1 usage, your instances are ready to be fully transitioned to using IMDSv2. At this stage, you can do the following:

- **Account default**

  You can set IMDSv2 to be required as an account default. When an instance is launched, the instance configuration is automatically set to the account default.

  To set the account default, do the following:

  - Amazon EC2 console: On the EC2 Dashboard, under **Account attributes**, **Data protection and security**, for **IMDS defaults**, set **Instance metadata service** to **Enabled** and **Metadata version** to **V2 only (token required)**. For more information, see Set IMDSv2 as the default for the account.
  - AWS CLI: Use the modify-instance-metadata-defaults CLI command and specify `--http-tokens required` and `--http-put-response-hop-limit 2`.
- **New instances**

  When launching a new instance, you can do the following:

  - Amazon EC2 console: In the launch instance wizard, set **Metadata accessible** to **Enabled** and **Metadata version** to **V2 only (token required)**. For more information, see Configure the instance at launch.
  - AWS CLI: Use the run-instances command and specify that IMDSv2 is required.
- **Existing instances**

  For existing instances, you can do the following:

  - Amazon EC2 console: On the **Instances** page, select your instance, choose **Actions**, **Instance settings**, **Modify instance metadata options**, and for **IMDSv2**, choose **Required**. For more information, see Require the use of IMDSv2.
  - AWS CLI: Use the modify-instance-metadata-options CLI command to specify that only IMDSv2 is to be used.

  You can modify the instance metadata options on running instances, and you don't need to restart the instances after modifying the instance metadata options.

Step 4: Check if your instances are transitioned to IMDSv2

You can check if any instances are not yet configured to require the use of IMDSv2, in other words, IMDSv2 is still configured as `optional`. If any instances are still configured as `optional`, you can modify the instance metadata options to make IMDSv2 `required` by repeating the preceding Step 3.

To filter your instances:

- Amazon EC2 console: On the **Instances** page, filter your instances by using the **IMDSv2 = optional** filter. For more information about filtering, see Filter resources using the console. You can also view whether IMDSv2 is required or optional for each instance: In the **Preferences** window, toggle on **IMDSv2** to add the **IMDSv2** column to the **Instances** table.
- AWS CLI: Use the describe-instances command and filter by `metadata-options.http-tokens = optional`, as follows:

```
aws ec2 describe-instances --filters "Name=metadata-options.http-
tokens,Values=optional" --query "Reservations[*].Instances[*].[InstanceId]" --output
 text
```

Step 5: 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 APIs and corresponding CLIs. 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.

Furthermore, after you've disabled IMDSv1, you can use the `MetadataNoTokenRejected` CloudWatch metric to track the number of times an IMDSv1 call was attempted and rejected. If, after disabling IMDSv1, you have software that is not working properly and the `MetadataNoTokenRejected` metric records IMDSv1 calls, it's likely that this software needs to be updated to use IMDSv2.

Limit access to the Instance Metadata Service

You can consider using local firewall rules to disable access from some or all processes to the Instance Metadata Service (IMDS).

For Nitro-based instances, the 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 the IMDS, we recommend that you modify the configuration of the network appliance to drop packets with the destination IPv4 address of the IMDS `169.254.169.254` and, if you enabled the IPv6 endpoint, the IPv6 address of the IMDS `[fd00:ec2::254]`.

Limit IMDS access for Linux instances

Using iptables to limit access

The following example uses Linux iptables and its `owner` module to prevent the Apache webserver (based on its default installation user ID of `apache`) from accessing 169.254.169.254. It uses a *deny rule* to reject all instance metadata requests (whether IMDSv1 or IMDSv2) from any process running as that user.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner
 --uid-owner apache --jump REJECT
```

Or, you can consider only allowing access to particular users or groups, by using *allow rules*. Allow rules might be easier to manage from a security perspective, because they require you to make a decision about what software needs access to instance metadata. If you use *allow rules*, it's less likely you will accidentally allow software to access the metadata service (that you did not intend to have access) if you later change the software or configuration on an instance. You can also combine group usage with allow rules, so that you can add and remove users from a permitted group without needing to change the firewall rule.

The following example prevents access to the IMDS by all processes, except for processes running in the user account `trustworthy-user`.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match
 owner ! --uid-owner trustworthy-user --jump REJECT
```

Note

- To use local firewall rules, you need to adapt the preceding example commands to suit your needs.
- By default, iptables rules are not persistent across system reboots. They can be made to be persistent by using OS features, not described here.

- The iptables `owner` module only matches group membership if the group is the primary group of a given local user. Other groups are not matched.

Using PF or IPFW to limit access

If you are using FreeBSD or OpenBSD, you can also consider using PF or IPFW. The following examples limit access to the IMDS to just the root user.

PF

```
$ block out inet proto tcp from any to 169.254.169.254
```

```
$ pass out inet proto tcp from any to 169.254.169.254 user root
```

IPFW

```
$ allow tcp from any to 169.254.169.254 uid root
```

```
$ deny tcp from any to 169.254.169.254
```

> **Note**
>
> The order of the PF and IPFW commands matter. PF defaults to last matching rule and IPFW defaults to first matching rule.

Limit IMDS access for Windows instances

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 trustworthy-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.

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("Everyone")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $exceptionPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("trustworthy-users")
PS C:\> $ExceptionPrincipalSID =
 $exceptionPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $PrincipalSDDL = "O:LSD:(D;;CC;;;$ExceptionPrincipalSID)(A;;CC;;;
$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service for
 $($blockPrincipal.Value), exception: $($exceptionPrincipal.Value)" -Action block -
Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $PrincipalSDDL
```

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.

Configure the Instance Metadata Service options

The Instance Metadata Service (IMDS) runs locally on every EC2 instance. The *instance metadata options* refer to a set of configurations that control the accessibility and behavior of the IMDS on an EC2 instance.

You can configure the following instance metadata options on each instance:

Instance metadata service (IMDS): `enabled` | `disabled`

You can enable or disable the IMDS on an instance. When disabled, you or any code won't be able to access the instance metadata on the instance.

The IMDS has two endpoints on an instance: IPv4 (`169.254.169.254`) and IPv6 (`[fd00:ec2::254]`). When you enable the IMDS, the IPv4 endpoint is automatically enabled. If you want to enable the IPv6 endpoint, you need to do so explicitly.

IMDS IPv6 endpoint: `enabled` | `disabled`

You can explicitly enable the IPv6 IMDS endpoint on an instance. When the IPv6 endpoint is enabled, the IPv4 endpoint remains enabled. The IPv6 endpoint is only supported on Nitro-based instances in IPv6-supported subnets (dual stack or IPv6 only).

Metadata version: `IMDSv1 or IMDSv2 (token optional)` | `IMDSv2 only (token required)`

When requesting instance metadata, IMDSv2 calls require a token. IMDSv1 calls do not require a token. You can configure an instance to allow either IMDSv1 or IMDSv2 calls (where a token is optional), or to only allow IMDSv2 calls (where a token is required).

Metadata response hop limit: `1–64`

The hop limit is the number of network hops that the PUT response is allowed to make. You can set the hop limit to a minimum of 1 and a maximum of 64. In a container environment, a hop limit of 1 can cause issues. For information about how to mitigate these issues, see the information about container environments under Instance metadata access considerations.

Access to tags in instance metadata: `enabled` | `disabled`

You can enable or disable access to the instance's tags from an instance's metadata. For more information, see View tags for your EC2 instances using instance metadata.

To view an instance's current configuration, see Query instance metadata options for existing instances.

Where to configure instance metadata options

Instance metadata options can be configured at different levels, as follows:

- **Account** – You can set default values for the instance metadata options at the account level for each AWS Region. When an instance is launched, the instance metadata options are automatically set to the account-level values. You can change these values at launch. Account-level default values do not affect existing instances.
- **AMI** – You can set the `imds-support` parameter to `v2.0` when you register or modify an AMI. When an instance is launched with this AMI, the instance metadata version is automatically set to IMDSv2 and the hop limit is set to 2.

- **Instance –** You can change all the instance metadata options on an instance at launch, overriding the default settings. You can also change the instance metadata options after launch on a running or stopped instance. Note that changes may be restricted by an IAM or SCP policy.

For more information, see Configure instance metadata options for new instances and Modify instance metadata options for existing instances.

Order of precedence for instance metadata options

The value for each instance metadata option is determined at instance launch, following a hierarchical order of precedence. The hierarchy, with the highest precedence at the top, is as follows:

- **Precedence 1: Instance configuration at launch –** Values can be specified either in the launch template or in the instance configuration. Any values specified here override values specified at the account level or in the AMI.
- **Precedence 2: Account settings –** If a value is not specified at instance launch, then it is determined by the account-level settings (which are set for each AWS Region). Account-level settings either include a value for each metadata option, or indicate no preference at all.
- **Precedence 3: AMI configuration –** If a value is not specified at instance launch or at the account level, then it is determined by the AMI configuration. This applies only to `HttpTokens` and `HttpPutResponseHopLimit`.

Each metadata option is evaluated separately. The instance can be configured with a mix of direct instance configuration, account-level defaults, and the configuration from the AMI.

You can change the value of any metadata option after launch on a running or stopped instance, unless changes are restricted by an IAM or SCP policy.

Determine values for metadata options – Example 1

In this example, an EC2 instance is launched into a Region where the `HttpPutResponseHopLimit` is set to `1` at the account level. The specified AMI has `ImdsSupport` set to `v2.0`. No metadata options are specified directly on the instance at launch. The instance is launched with the following metadata options:

```
"MetadataOptions": {
    ...
```

```
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    ...
```

These values were determined as follows:

- **No metadata options specified at launch:** During the launch of the instance, specific values for the metadata options were not provided either in the instance launch parameters or in the launch template.
- **Account settings take next precedence:** In the absence of specific values specified at launch, the settings at the account level within the Region take precedence. This means that the default values configured at the account level are applied. In this case, the `HttpPutResponseHopLimit` was set to `1`.
- **AMI settings take last precedence:** In the absence of a specific value specified at launch or at the account level for `HttpTokens` (the instance metadata version), the AMI setting is applied. In this case, the AMI setting `ImdsSupport: v2.0` determined that `HttpTokens` was set to `required`. Note that while the AMI setting `ImdsSupport: v2.0` is designed to set `HttpPutResponseHopLimit: 2`, it was overridden by the account-level setting `HttpPutResponseHopLimit: 1`, which has higher precedence.

Determine values for metadata options – Example 2

In this example, the EC2 instance is launched with the same settings as in the previous Example 1, but with `HttpTokens` set to `optional` directly on the instance at launch. The instance is launched with the following metadata options:

```
"MetadataOptions": {
    ...
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    ...
```

The value for `HttpPutResponseHopLimit` is determined in the same way as in Example 1. However, the value for `HttpTokens` is determined as follows: Metadata options configured on the instance at launch take first precedence. Even though the AMI was configured with `ImdsSupport: v2.0` (in other words, `HttpTokens` is set to `required`), the value specified on the instance at launch (`HttpTokens` set to `optional`) took precedence.

Set the instance metadata version

When an instance is launched, the value for the instance *metadata version* is either `IMDSv1 or IMDSv2 (token optional)` or `IMDSv2 only (token required)`.

At instance launch, you can either manually specify the value for the metadata version, or use the default value. If you manually specify the value, it overrides any defaults. If you opt not to manually specify the value, it will be determined by a combination of default settings, as outlined in the following table.

The table shows how the metadata version for an instance at launch (indicated by **Resulting instance configuration** in column 4) is determined by the settings at the different levels of configuration. The order of precedence is from left to right, where the first column takes the highest precedence, as follows:

- Column 1: **Launch parameter** – Represents the setting on the instance that you manually specify at launch.
- Column 2: **Account level default** – Represents the setting for the account.
- Column 3: **AMI default** – Represents the setting on the AMI.

Launch parameter	Account level default	AMI default	Resulting instance configuration
V2 only (token required)	No preference	V2 only	V2 only
V2 only (token required)	V2 only	V2 only	V2 only
V2 only (token required)	V1 or V2	V2 only	V2 only
V1 or V2 (token optional)	No preference	V2 only	V1 or V2
V1 or V2 (token optional)	V2 only	V2 only	V1 or V2

Launch parameter	Account level default	AMI default	Resulting instance configuration
V1 or V2 (token optional)	V1 or V2	V2 only	V1 or V2
Not set	No preference	V2 only	V2 only
Not set	V2 only	V2 only	V2 only
Not set	V1 or V2	V2 only	V1 or V2
V2 only (token required)	No preference	null	V2 only
V2 only (token required)	V2 only	null	V2 only
V2 only (token required)	V1 or V2	null	V2 only
V1 or V2 (token optional)	No preference	null	V1 or V2
V1 or V2 (token optional)	V2 only	null	V1 or V2
V1 or V2 (token optional)	V1 or V2	null	V1 or V2
Not set	No preference	null	V1 or V2
Not set	V2 only	null	V2 only
Not set	V1 or V2	null	V1 or V2

Use IAM condition keys to restrict instance metadata options

You can use IAM condition keys in an IAM policy or SCP as follows:

- 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

Tasks

- Configure instance metadata options for new instances
- Modify instance metadata options for existing instances

> **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.
> - (Windows only) If your PowerShell version is earlier than 4.0, you must update to Windows Management Framework 4.0 to require the use of IMDSv2.

Configure instance metadata options for new instances

You can configure the following instance metadata options for new instances.

Options

- Require the use of IMDSv2
- Enable the IMDS IPv4 and IPv6 endpoints
- Turn off access to instance metadata
- Allow access to tags in instance metadata

> **Note**
>
> The settings for these options are configured at the account level, either directly in the account or by using a declarative policy. They must be configured in each AWS Region where you want to configure instance metadata options. Using a declarative policy allows you to apply the settings across multiple Regions simultaneously, as well as across multiple accounts simultaneously. When a declarative policy is in use, you can't modify the settings directly within an account. This topic describes how to configure the settings directly within an account. For information about using declarative policies, see Declarative policies in the *AWS Organizations User Guide.*

Require the use of IMDSv2

You can use the following methods to require the use of IMDSv2 on your new instances.

To require IMDSv2

- Set IMDSv2 as the default for the account
- Configure the instance at launch
- Configure the AMI
- Use an IAM policy

Set IMDSv2 as the default for the account

You can set the default version for the instance metadata service (IMDS) at the account level for each AWS Region. This means that when you launch a *new* instance, the instance metadata version is automatically set to the account-level default. However, you can manually override the value at launch or after launch. For more information about how the account-level settings and manual overrides affect an instance, see Order of precedence for instance metadata options.

> **Note**
>
> Setting the account-level default does not reset *existing* instances. For example, if you set the account-level default to IMDSv2, any existing instances that are set to IMDSv1 are not affected. If you want to change the value on existing instances, you must manually change the value on the instances themselves.

You can set the account default for the instance metadata version to IMDSv2 so that all *new* instances in the account launch with IMDSv2 required, and IMDSv1 will be disabled. With this account default, when you launch an instance, the following are the default values for the instance:

- Console: **Metadata version** is set to **V2 only (token required)** and **Metadata response hop limit** is set to **2**.
- AWS CLI: `HttpTokens` is set to `required` and `HttpPutResponseHopLimit` is set to 2.

> **Note**
>
> Before setting the account default to IMDSv2, ensure that your instances do not depend on IMDSv1. For more information, see [Recommended path to requiring IMDSv2](#).

Console

To set IMDSv2 as the default for the account for the specified Region

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. In the navigation pane, choose **EC2 Dashboard**.
4. Under **Account attributes**, choose **Data protection and security**.
5. Next to **IMDS defaults**, choose **Manage**.
6. On the **Manage IMDS defaults** page, do the following:

   a. For **Instance metadata service**, choose **Enabled**.

   b. For **Metadata version**, choose **V2 only (token required)**.

   c. For **Metadata response hop limit**, specify **2** if your instances will host containers. Otherwise, select **No preference**. When no preference is specified, at launch, the value defaults to **2** if the AMI has the setting `ImdsSupport: v2.0`; otherwise it defaults to **1**.

   d. Choose **Update**.

AWS CLI

To set IMDSv2 as the default for the account for the specified Region

Use the modify-instance-metadata-defaults command and specify the Region in which to modify the IMDS account level settings. Include `--http-tokens` set to `required` and `--http-put-response-hop-limit` set to 2 if your instances will host containers. Otherwise, specify `-1` to indicate no preference. When `-1` (no preference) is specified, at launch, the value defaults to 2 if the AMI has the setting `ImdsSupport: v2.0`; otherwise it defaults to 1.

```
aws ec2 modify-instance-metadata-defaults \
    --region us-east-1 \
    --http-tokens required \
    --http-put-response-hop-limit 2
```

The following is example output.

```
{
    "Return": true
}
```

To view the default account settings for the instance metadata options for the specified Region

Use the get-instance-metadata-defaults command and specify the Region.

```
aws ec2 get-instance-metadata-defaults --region us-east-1
```

The following is example output.

```
{
    "AccountLevel": {
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 2
    },
    "ManagedBy": "account"
}
```

The `ManagedBy` field indicates the entity that configured the settings. In this example, `account` indicates that the settings were configured directly in the account. A value of `declarative-policy` would mean the settings were configured by a declarative policy. For more information, see Declarative policies in the *AWS Organizations User Guide*.

To set IMDSv2 as the default for the account for all Regions

Use the modify-instance-metadata-defaults command to modify the IMDS account level settings for all Regions. Include `--http-tokens` set to `required` and `--http-put-response-hop-limit` set to 2 if your instances will host containers. Otherwise, specify `-1` to indicate no preference. When `-1` (no preference) is specified, at launch, the value defaults to 2 if the AMI has the setting `ImdsSupport: v2.0`; otherwise it defaults to 1.

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
            --http-tokens required \
            --http-put-response-hop-limit 2 \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

The following is example output.

```
Region                   Modified
--------------           ---------
ap-south-1               True
eu-north-1               True
eu-west-3                True
...
```

To view the default account settings for the instance metadata options for all Regions

Use the get-instance-metadata-defaults command.

```
echo -e "Region   \t Level          Hops    HttpTokens" ; \
echo -e "-------------- \t ------------   ----    ----------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
```

```
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-instance-metadata-defaults \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

The following is example output.

```
Region          Level          Hops    HttpTokens
--------------  ------------   ----    ----------
ap-south-1      ACCOUNTLEVEL   2       required
eu-north-1      ACCOUNTLEVEL   2       required
eu-west-3       ACCOUNTLEVEL   2       required
...
```

PowerShell

To set IMDSv2 as the default for the account for the specified Region

Use the Edit-EC2InstanceMetadataDefault command and specify the Region in which to modify the IMDS account level settings. Include `-HttpToken` set to `required` and `-HttpPutResponseHopLimit` set to 2 if your instances will host containers. Otherwise, specify `-1` to indicate no preference. When `-1` (no preference) is specified, at launch, the value defaults to 2 if the AMI has the setting `ImdsSupport: v2.0`; otherwise it defaults to 1.

```
Edit-EC2InstanceMetadataDefault `
    -Region us-east-1 `
    -HttpToken required `
    -HttpPutResponseHopLimit 2
```

The following is example output.

```
True
```

To view the default account settings for the instance metadata options for the specified Region

Use the Get-EC2InstanceMetadataDefault command and specify the Region.

```
Get-EC2InstanceMetadataDefault -Region us-east-1 | Format-List
```

The following is example output.

```
HttpEndpoint            :
HttpPutResponseHopLimit : 2
HttpTokens              : required
InstanceMetadataTags    :
```

To set IMDSv2 as the default for the account for all Regions

Use the Edit-EC2InstanceMetadataDefault Cmdlet to modify the IMDS account level settings for all Regions. Include `-HttpToken` set to `required` and `-HttpPutResponseHopLimit` set to 2 if your instances will host containers. Otherwise, specify `-1` to indicate no preference. When `-1` (no preference) is specified, at launch, the value defaults to 2 if the AMI has the setting `ImdsSupport: v2.0`; otherwise it defaults to 1.

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region   = $_
        Modified = (Edit-EC2InstanceMetadataDefault `
                -Region $_ `
                -HttpToken required `
                -HttpPutResponseHopLimit 2)
    }
} | `
Format-Table Region, Modified -AutoSize
```

Expected output

```
Region         Modified
------         --------
ap-south-1         True
eu-north-1         True
eu-west-3          True
...
```

To view the default account settings for the instance metadata options for all Regions

Use the Get-EC2InstanceMetadataDefault Cmdlet.

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region = $_
        HttpPutResponseHopLimit = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpPutResponseHopLimit
        HttpTokens              = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpTokens
    }
} | `
Format-Table -AutoSize
```

Example output

```
Region         HttpPutResponseHopLimit HttpTokens
------         ----------------------- ----------
ap-south-1                           2 required
eu-north-1                           2 required
eu-west-3                            2 required
...
```

Configure the instance at launch

When you launch an instance, you can configure the instance to require the use of IMDSv2 by configuring the following fields:

- Amazon EC2 console: Set **Metadata version** to **V2 only (token required)**.
- AWS CLI: Set `HttpTokens` to `required`.

When you specify that IMDSv2 is required, you must also enable the Instance Metadata Service (IMDS) endpoint by setting **Metadata accessible** to **Enabled** (console) or `HttpEndpoint` to `enabled` (AWS CLI).

In a container environment, when IMDSv2 is required, we recommend setting the hop limit to 2. For more information, see Instance metadata access considerations.

Console

To require the use of IMDSv2 on a new instance

- When launching a new instance in the Amazon EC2 console, expand **Advanced details**, and do the following:
  - For **Metadata accessible**, choose **Enabled**.
  - For **Metadata version**, choose **V2 only (token required)**.
  - (Container environment) For **Metadata response hop limit**, choose **2**.

  For more information, see Advanced details.

AWS CLI

To require the use of IMDSv2 on a new instance

The following run-instances example launches a `c6i.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 requires the instance to use IMDSv2 when requesting instance metadata.

In a container environment, when IMDSv2 is required, we recommend setting the hop limit to 2 with `HttpPutResponseHopLimit=2`.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
 ...
    --metadata-options 
 "HttpEndpoint=enabled,HttpTokens=required,HttpPutResponseHopLimit=2"
```

PowerShell

To require the use of IMDSv2 on a new instance

The following New-EC2Instance Cmdlet example launches a `c6i.large` instance with `MetadataOptions_HttpEndpoint` set to `enabled` and the `MetadataOptions_HttpTokens` parameter to `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 requires the instance to use IMDSv2 when requesting instance metadata.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpTokens required
```

AWS CloudFormation

To specify the metadata options for an instance using AWS CloudFormation, see the AWS::EC2::LaunchTemplate MetadataOptions property in the *AWS CloudFormation User Guide*.

Configure the AMI

When you register a new AMI or modify an existing AMI, you can set the imds-support parameter to v2.0. Instances launched from this AMI will have **Metadata version** set to **V2 only (token required)** (console) or HttpTokens set to required (AWS CLI). With these settings, the instance requires that IMDSv2 is used when requesting instance metadata.

Note that when you set imds-support to v2.0, instances launched from this AMI will also have **Metadata response hop limit** (console) or http-put-response-hop-limit (AWS CLI) set to **2**.

> **Important**
>
> Do not use this parameter unless your AMI software supports IMDSv2. After you set the value to v2.0, you can't undo it. The only way to "reset" your AMI is to create a new AMI from the underlying snapshot.

To configure a new AMI for IMDSv2

Use one of the following methods to configure a new AMI for IMDSv2.

AWS CLI

The following register-image example registers an AMI using the specified snapshot of an EBS root volume as device /dev/xvda. Specify v2.0 for the imds-support parameter so that instances launched from this AMI will require that IMDSv2 is used when requesting instance metadata.

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0123456789example} \
    --architecture x86_64 \
    --imds-support v2.0
```

PowerShell

The following Register-EC2Image Cmdlet example registers an AMI using the specified snapshot of an EBS root volume as device `/dev/xvda`. Specify `v2.0` for the `ImdsSupport` parameter so that instances launched from this AMI will require that IMDSv2 is used when requesting instance metadata.

```
Register-EC2Image `
    -Name 'my-image' `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping  (
    New-Object `
        -TypeName Amazon.EC2.Model.BlockDeviceMapping `
        -Property @{
        DeviceName = '/dev/xvda';
        EBS        = (New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property
 @{
                SnapshotId = 'snap-0123456789example'
                VolumeType = 'gp3'
                } )
        }  ) `
    -Architecture X86_64 `
    -ImdsSupport v2.0
```

To configure an existing AMI for IMDSv2

Use one of the following methods to configure an existing AMI for IMDSv2.

AWS CLI

The following modify-image-attribute example modifies an existing AMI for IMDSv2 only. Specify `v2.0` for the `imds-support` parameter so that instances launched from this AMI will require that IMDSv2 is used when requesting instance metadata.

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --imds-support v2.0
```

PowerShell

The following Edit-EC2ImageAttribute Cmdlet example modifies an existing AMI for IMDSv2 only. Specify `v2.0` for the `imds-support` parameter so that instances launched from this AMI will require that IMDSv2 is used when requesting instance metadata.

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ImdsSupport 'v2.0'
```

Use an IAM policy

You can create an IAM policy that prevents users from launching new instances unless they require IMDSv2 on the new instance.

To enforce the use of IMDSv2 on all new instances by using an IAM policy

To ensure that 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.

Enable the IMDS IPv4 and IPv6 endpoints

The IMDS has two endpoints on an instance: IPv4 (`169.254.169.254`) and IPv6 (`[fd00:ec2::254]`). When you enable the IMDS, the IPv4 endpoint is automatically enabled. The IPv6 endpoint remains disabled even if you launch an instance into an IPv6-only subnet. To enable the IPv6 endpoint, you need to do so explicitly. When you enable the IPv6 endpoint, the IPv4 endpoint remains enabled.

You can enable the IPv6 endpoint at instance launch or after.

Requirements for enabling the IPv6 endpoint

- The selected instance type is a Nitro-based instance.
- The selected subnet supports IPv6, where the subnet is either dual stack or IPv6 only.

Use any of the following methods to launch an instance with the IMDS IPv6 endpoint enabled.

Console

To enable the IMDS IPv6 endpoint at instance launch

- Launch the instance in the Amazon EC2 console with the following specified under **Advanced details**:
  - For **Metadata IPv6 endpoint**, choose **Enabled**.

For more information, see Advanced details.

AWS CLI

To enable the IMDS IPv6 endpoint at instance launch

The following run-instances example launches a `c6i.large` instance with the IPv6 endpoint enabled for the IMDS. To enable the IPv6 endpoint, for the `--metadata-options` parameter, specify `HttpProtocolIpv6=enabled`. When you specify a value for `HttpProtocolIpv6`, you must also set `HttpEndpoint` to `enabled`.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=enabled,HttpProtocolIpv6=enabled"
```

PowerShell

To enable the IMDS IPv6 endpoint at instance launch

The following New-EC2Instance Cmdlet example launches a `c6i.large` instance with the IPv6 endpoint enabled for the IMDS. To enable the IPv6 endpoint, specify `MetadataOptions_HttpProtocolIpv6` as `enabled`. When you specify a value for MetadataOptions_HttpProtocolIpv6, you must also set MetadataOptions_HttpEndpoint to enabled.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpProtocolIpv6 enabled
```

Turn off access to instance metadata

You can turn off access to the instance metadata by disabling the IMDS when you launch an instance. You can turn on access later by re-enabling the IMDS. For more information, see Turn on access to instance metadata.

> **Important**
>
> You can choose to disable the IMDS at launch or after launch. If you disable the IMDS *at launch*, the following might not work:
>
> - You might not have SSH access to your instance. The public-keys/0/openssh-key, which is your instance's public SSH key, will not be accessible because the key is normally provided and accessed from EC2 instance metadata.
> - EC2 user data will not be available and will not run at instance start. EC2 user data is hosted on the IMDS. If you disable the IMDS, you effectively turn off access to user data.
>
> To access this functionality, you can re-enable the IMDS after launch.

Console

To turn off access to instance metadata at launch

- Launch the instance in the Amazon EC2 console with the following specified under **Advanced details**:
  - For **Metadata accessible**, choose **Disabled**.

For more information, see Advanced details.

AWS CLI

To turn off access to instance metadata at launch at launch

Launch the instance with `--metadata-options` set to `HttpEndpoint=disabled`.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=disabled"
```

PowerShell

To turn off access to instance metadata at launch at launch

The following New-EC2Instance Cmdlet example launches an instance with `MetadataOptions_HttpEndpoint` set to `disabled`.

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint disabled
```

AWS CloudFormation

To specify the metadata options for an instance using AWS CloudFormation, see the AWS::EC2::LaunchTemplate MetadataOptions property in the *AWS CloudFormation User Guide*.

Allow access to tags in instance metadata

By default, instance tags are not accessible in the instance metadata. For each instance, you must explicitly allow access. If access is allowed, instance tag *keys* must comply with specific character restrictions, otherwise the instance launch will fail. For more information, see Enable access to tags in instance metadata.

Modify instance metadata options for existing instances

You can modify the instance metadata options for existing instances.

You can also create an IAM policy that prevents users from modifying the instance metadata options on existing instances. To control which 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.

> **Note**
>
> If a declarative policy was used to configure the instance metadata options, you can't modify them directly within the account. For more information, see Declarative policies in the *AWS Organizations User Guide*.

Require the use of IMDSv2

Use one of the following methods to modify the instance metadata options on an existing instance to require that IMDSv2 is used when requesting instance metadata. When IMDSv2 is required, IMDSv1 cannot be used.

> **Note**
>
> Before requiring that IMDSv2 is used, ensure that the instance isn't making IMDSv1 calls. The `MetadataNoToken` CloudWatch metric tracks IMDSv1 calls. When `MetadataNoToken` records zero IMDSv1 usage for an instance, the instance is then ready to require IMDSv2.

Console

To require the use of IMDSv2 on 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.
4. Choose **Actions**, **Instance settings**, **Modify instance metadata options**.
5. In the **Modify instance metadata options** dialog box, do the following:

   a. For **Instance metadata service**, select **Enable**.

   b. For **IMDSv2**, choose **Required**.

c. Choose **Save**.

AWS CLI

To require the use of IMDSv2 on an existing instance

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-1234567890abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

PowerShell

To require the use of IMDSv2 on an existing instance

Use the Edit-EC2InstanceMetadataOption Cmdlet and set the `HttpTokens` parameter to `required`. When you specify a value for `HttpTokens`, you must also set `HttpEndpoint` to `enabled`.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens required `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

Restore the use of IMDSv1

When IMDSv2 is required, IMDSv1 will not work when requesting instance metadata. When IMDSv2 is optional, then both IMDSv2 and IMDSv1 will work. Therefore, to restore IMDSv1, make IMDSv2 optional by using one of the following methods.

Console

To restore the use of IMDSv1 on 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.
4. Choose **Actions**, **Instance settings**, **Modify instance metadata options**.
5. In the **Modify instance metadata options** dialog box, do the following:

    a. For **Instance metadata service**, make sure that **Enable** is selected.

    b. For **IMDSv2**, choose **Optional**.

    c. Choose **Save**.

AWS CLI

To restore the use of IMDSv1 on an instance

You can use the modify-instance-metadata-options CLI command with `http-tokens` set to `optional` to restore the use of IMDSv1 when requesting instance metadata.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens optional \
    --http-endpoint enabled
```

PowerShell

To restore the use of IMDSv1 on an instance

You can use the Edit-EC2InstanceMetadataOption Cmdlet with `HttpTokens` set to `optional` to restore the use of IMDSv1 when requesting instance metadata.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens optional `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

Change the PUT response hop limit

For existing instances, you can change the settings of the PUT response hop limit.

Currently only the AWS CLI and AWS SDKs support changing the PUT response hop limit.

AWS CLI

To change 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-1234567890abcdef0 \
    --http-put-response-hop-limit 3 \
    --http-endpoint enabled
```

PowerShell

To change the PUT response hop limit

Use the Edit-EC2InstanceMetadataOption Cmdlet and set the `HttpPutResponseHopLimit` 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 `HttpPutResponseHopLimit`, you must also set `HttpEndpoint` to `enabled`.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpPutResponseHopLimit 3 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

Enable the IMDS IPv4 and IPv6 endpoints

The IMDS has two endpoints on an instance: IPv4 (`169.254.169.254`) and IPv6 (`[fd00:ec2::254]`). When you enable the IMDS, the IPv4 endpoint is automatically enabled. The IPv6 endpoint remains disabled even if you launch an instance into an IPv6-only subnet. To enable the IPv6 endpoint, you need to do so explicitly. When you enable the IPv6 endpoint, the IPv4 endpoint remains enabled.

You can enable the IPv6 endpoint at instance launch or after.

Requirements for enabling the IPv6 endpoint

- The selected instance type is a Nitro-based instance.

- The selected subnet supports IPv6, where the subnet is either dual stack or IPv6 only.

Currently only the AWS CLI and AWS SDKs support enabling the IMDS IPv6 endpoint after instance launch.

AWS CLI

To enable the IMDS IPv6 endpoint for your instance

Use the modify-instance-metadata-options CLI command and set the `http-protocol-ipv6` parameter to `enabled`. Note that when specifying a value for `http-protocol-ipv6`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options \
 --instance-id i-1234567890abcdef0 \
 --http-protocol-ipv6 enabled \
 --http-endpoint enabled
```

PowerShell

To enable the IMDS IPv6 endpoint for your instance

Use the Edit-EC2InstanceMetadataOption Cmdlet and set the `HttpProtocolIpv6` parameter to `enabled`. Note that when specifying a value for `HttpProtocolIpv6`, you must also set `HttpEndpoint` to `enabled`.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpProtocolIpv6 enabled `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

Turn on access to instance metadata

You can turn on access to instance metadata by enabling the HTTP endpoint of the IMDS on your instance, regardless of which version of the IMDS you are using. You can reverse this change at any time by disabling the HTTP endpoint.

Use one of the following methods to turn on access to instance metadata on an instance.

Console

To turn on access to instance metadata

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance.
4. Choose **Actions**, **Instance settings**, **Modify instance metadata options**.
5. In the **Modify instance metadata options** dialog box, do the following:
   a. For **Instance metadata service**, select **Enable**.
   b. Choose **Save**.

AWS CLI

To turn on access to instance metadata

Use the modify-instance-metadata-options CLI command and set the `http-endpoint` parameter to `enabled`.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint enabled
```

PowerShell

To turn on access to instance metadata

Use the Edit-EC2InstanceMetadataOption Cmdlet and set the `HttpEndpoint` parameter to `enabled`.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

Turn off access to instance metadata

You can turn off access to instance metadata by disabling the HTTP endpoint of the IMDS on your instance, regardless of which version of the IMDS you are using. You can reverse this change at any time by enabling the HTTP endpoint.

Use one of the following methods to turn off access to instance metadata on an instance.

Console

To turn off access to instance metadata

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance.
4. Choose **Actions**, **Instance settings**, **Modify instance metadata options**.
5. In the **Modify instance metadata options** dialog box, do the following:

    a. For **Instance metadata service**, clear **Enable**.

    b. Choose **Save**.

AWS CLI

To turn off access to instance metadata

Use the modify-instance-metadata-options CLI command and set the `http-endpoint` parameter to `disabled`.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint disabled
```

PowerShell

To turn off access to instance metadata

Use the Edit-EC2InstanceMetadataOption Cmdlet and set the `HttpEndpoint` parameter to `disabled`.

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint disabled).InstanceMetadataOptions
```

Allow access to tags in instance metadata

You can allow access to tags in the instance metadata on a running or stopped instance. For each instance, you must explicitly allow access. If access is allowed, instance tag *keys* must comply with specific character restrictions, otherwise you get an error. For more information, see [Enable access to tags in instance metadata](#).

Run commands when you launch an EC2 instance with user data input

When you launch an Amazon EC2 instance, you can pass user data to the instance that is used to perform automated configuration tasks, or to run scripts after the instance starts.

If you're interested in more complex automation scenarios, you might consider AWS CloudFormation. For more information, see [Deploying applications on Amazon EC2 with AWS CloudFormation](#) in the *AWS CloudFormation User Guide*.

On Linux instances, you can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch instance wizard as plain text, as a file (this is useful for launching instances with the command line tools), or as base64-encoded text (for API calls).

On Windows instances, the launch agents handle your user data scripts.

Considerations

- User data is treated as opaque data: what you give is what you get back. It is up to the instance to interpret it.
- User data must be base64-encoded. The Amazon EC2 console can perform the base64-encoding for your or accept base64-encoded input. If you retrieve the user data using instance metadata or the console, it's base64-decoded for you automatically.
- 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 ceil(*n*/3)*4.
- User data is an instance attribute. If you create an AMI from an instance, the instance user data is not included in the AMI.

User data in the AWS Management 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 with the Launch Wizard

You can specify user data when you launch an instance with the Launch Wizard in the EC2 console. To specify user data on launch, follow the procedure for [launching an instance](). The **User data** field is located in the [Advanced details]() section of the launch instance wizard. Enter your PowerShell script in the **User data** field, and then complete the instance launch procedure.

In the following screenshot of the **User data** field, the example script 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. If you leave the **User data has already been base64 encoded** checkbox empty, the Amazon EC2 console performs the base64 encoding for you.

For more information, see Specify instance user data at launch with the Launch Wizard. For a Linux example that uses the AWS CLI, see the section called "User data and the AWS CLI". For a Windows example that uses the Tools for Windows PowerShell, see the section called "User data and the Tools for Windows PowerShell".

View and update the instance user data

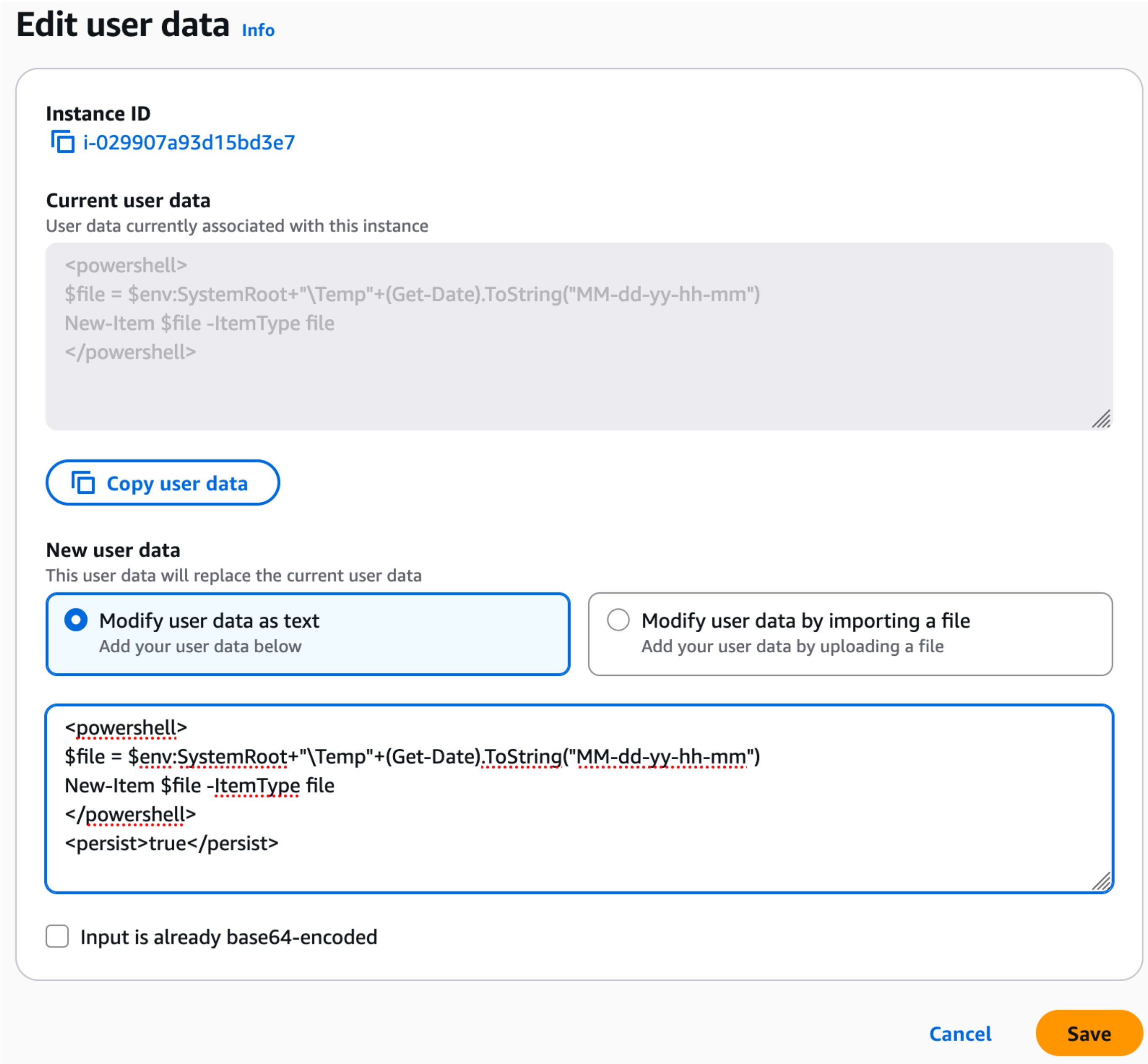
You can view the instance user data for any instance, and you can update the instance user data for a stopped instance.

To update the user data for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
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:



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.

How Amazon EC2 handles user data for Linux instances

The following examples use user data to run commands that set up a LAMP server when the instance launches. In each example, the following tasks are performed:

- The distribution software packages are updated.

- The web server, php, and `mariadb` packages are installed.
- The `httpd` service is started and turned on.
- The user `ec2-user` is added to the apache group.
- The appropriate ownership and file permissions are set for the web directory and the files contained within it.
- A simple web page is created to test the web server and PHP engine.

Contents

Prerequisites

The examples in this topic assume the following:

- Your instance has a public DNS name that is reachable from the internet.
- The security group associated with your instance is configured to allow SSH (port 22) traffic so that you can connect to the instance to view the output log files.
- Your instance is launched using an Amazon Linux AMI. The commands and directives might not work for other Linux distributions. For more information about other distributions, such as their support for cloud-init, see documentation for the specific distribution.

User data and shell scripts

If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch. Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.

> **Important**
>
> By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance. For more information, see How can I utilize user data to automatically run a script with every restart of my Amazon EC2 Linux instance? in the AWS Knowledge Center.

User data shell scripts must start with the `#!` characters and the path to the interpreter you want to read the script (commonly **/bin/bash)**. For an introduction on shell scripting, see the Bash Reference Manual on the *GNU Operating System* website.

Scripts entered as user data are run as the root user, so do not use the **sudo** command in the script. Remember that any files you create will be owned by the root user; if you need a non-root user to have file access, you should modify the permissions accordingly in the script. Also, because the script is not run interactively, you cannot include commands that require user feedback (such as **yum update** without the `-y` flag).

If you use an AWS API, including the AWS CLI, in a user data script, you must use an instance profile when launching the instance. An instance profile provides the appropriate AWS credentials required by the user data script to issue the API call. For more information, see Use instance profiles in the IAM User Guide. The permissions you assign to the IAM role depend on which services you are calling with the API. For more information, see IAM roles for Amazon EC2.

The cloud-init output log file captures console output so it is easy to debug your scripts following a launch if the instance does not behave the way you intended. To view the log file, connect to the instance and open `/var/log/cloud-init-output.log`.

When a user data script is processed, it is copied to and run from `/var/lib/cloud/instances/instance-id/`. The script is not deleted after it is run. Be sure to delete the user data scripts from `/var/lib/cloud/instances/instance-id/` before you create an AMI from the instance. Otherwise, the script will exist in this directory on any instance launched from the AMI.

Update the instance user data

To update the instance user data, you must first stop the instance. If the instance is running, you can view the user data but you cannot modify it.

> **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 modify instance user data

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **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. With the instance still selected, choose **Actions**, **Instance settings**, **Edit user data**.
6. Modify the user data as needed, and then choose **Save**.
7. Start the instance. The new user data is visible on your instance after you start it; however, user data scripts are not run.

User data and cloud-init directives

The cloud-init package configures specific aspects of a new Amazon Linux instance when it is launched; most notably, it configures the `.ssh/authorized_keys` file for the ec2-user so you can log in with your own private key. For more information about the configuration tasks that the cloud-init package performs for Amazon Linux instances, see the following documentation:

- **Amazon Linux 2023 –** Customized cloud-init
- **Amazon Linux 2 –** Using cloud-init on Amazon Linux 2

The cloud-init user directives can be passed to an instance at launch the same way that a script is passed, although the syntax is different. For more information about cloud-init, see https://cloudinit.readthedocs.org/en/latest/index.html.

Important

By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance. For more information, see How can I utilize user data to automatically run a script with every restart of my Amazon EC2 Linux instance? in the AWS Knowledge Center.

Adding these tasks at boot time adds to the amount of time it takes to boot an instance. You should allow a few minutes of extra time for the tasks to complete before you test that your user data directives have completed.

To pass cloud-init directives to an Amazon Linux instance

1. Follow the procedure for launching an instance. The **User data** field is located in the Advanced details section of the launch instance wizard. Enter your cloud-init directive text in the **User data** field, and then complete the instance launch procedure.

   In the examples below, the directives create and configure a web server on Amazon Linux. The `#cloud-config` line at the top is required in order to identify the commands as cloud-init directives.

   AL2023

```
#cloud-config
package_update: true
package_upgrade: all

packages:
- httpd
- mariadb105-server
- php8.1
- php8.1-mysqlnd

runcmd:
- systemctl start httpd
- systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
```

```
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

AL2

```
#cloud-config
package_update: true
package_upgrade: all

packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
- systemctl start httpd
- systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. Allow enough time for the instance to launch and run the directives in your user data, and then check to see that your directives have completed the tasks you intended.

   For this example, in a web browser, enter the URL of the PHP test file the directives created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

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

   You should see the PHP information page. If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For more information, see Configure security group rules.

3. (Optional) If your directives did not accomplish the tasks you were expecting them to, or if you just want to verify that your directives completed without errors, connect to the instance, examine the output log file (`/var/log/cloud-init-output.log`), and look for error

messages in the output. For additional debugging information, you can add the following line to your directives:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends **runcmd** output to `/var/log/cloud-init-output.log`.

User data and the AWS CLI

You can use the AWS CLI to specify, modify, and view the user data for your instance. For information about viewing user data from your instance using instance metadata, see Access instance metadata for an EC2 instance.

On Windows, you can use the AWS Tools for Windows PowerShell instead of using the AWS CLI. For more information, see User data and the Tools for Windows PowerShell .

Example: Specify user data at launch

To specify user data when you launch your instance, use the run-instances command with the `--user-data` parameter. With **run-instances**, the AWS CLI performs base64 encoding of the user data for you.

The following example shows how to specify a script as a string on the command line:

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
    --key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234
 \
    --user-data echo user data
```

The following example shows how to specify a script using a text file. Be sure to use the `file://` prefix to specify the file.

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
    --key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234
 \
    --user-data file://my_script.txt
```

The following is an example text file with a shell script.

```
#!/bin/bash
```

```
yum update -y
service httpd start
chkconfig httpd on
```

Example: Modify the user data of a stopped instance

You can modify the user data of a stopped instance using the modify-instance-attribute command. With **modify-instance-attribute**, the AWS CLI does not perform base64 encoding of the user data for you.

- On a **Linux** computer, use the base64 command to encode the user data.

```
base64 my_script.txt >my_script_base64.txt
```

- On a **Windows** computer, use the certutil command to encode the user data. Before you can use this file with the AWS CLI, you must remove the first (BEGIN CERTIFICATE) and last (END CERTIFICATE) lines.

```
certutil -encode my_script.txt my_script_base64.txt
notepad my_script_base64.txt
```

Use the `--attribute` and `--value` parameters to use the encoded text file to specify the user data. Be sure to use the `file://` prefix to specify the file.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData --value file://my_script_base64.txt
```

Example: Clear the user data of a stopped instance

To delete the existing user data, use the modify-instance-attribute command as follows:

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --user-data Value=
```

Example: View user data

To retrieve the user data for an instance, use the describe-instance-attribute command. With **describe-instance-attribute**, the AWS CLI does not perform base64 decoding of the user data for you.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData
```

The following is example output with the user data base64 encoded.


```
{
    "UserData": {
        "Value":
 "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQpzZXJ2aWNlIGh0dHBkIHN0YXJ0CmNoa2NvbmZpZyBodHRwZCBvbg=="
    },
    "InstanceId": "i-1234567890abcdef0"
}
```


- On a **Linux** computer , use the `--query` option to get the encoded user data and the base64 command to decode it.

  ```
  aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
   userData --output text --query "UserData.Value" | base64 --decode
  ```

- On a **Windows** computer, use the `--query` option to get the coded user data and the certutil command to decode it. Note that the encoded output is stored in a file and the decoded output is stored in another file.

  ```
  aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
   userData --output text --query "UserData.Value" >my_output.txt
  certutil -decode my_output.txt my_output_decoded.txt
  type my_output_decoded.txt
  ```

The following is example output.

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

Combine shell scripts and cloud-init directives

By default, you can include only one content type in user data at a time. However, you can use the `text/cloud-config` and `text/x-shellscript` content-types in a mime-multi part file to include both a shell script and cloud-init directives in your user data.

The following shows the mime-multi part format.

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
cloud-init directives

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
shell script commands
--//--
```

For example, the following user data includes cloud-init directives and a bash shell script. The cloud-init directives creates a file (`/test-cloudinit/cloud-init.txt`), and writes `Created by cloud-init` to that file. The bash shell script creates a file (`/test-userscript/userscript.txt`) and writes `Created by bash shell script` to that file.

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

```
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
runcmd:
- [ mkdir, /test-cloudinit ]
write_files:
- path: /test-cloudinit/cloud-init.txt
content: Created by cloud-init

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
mkdir test-userscript
touch /test-userscript/userscript.txt
echo "Created by bash shell script" >> /test-userscript/userscript.txt
--//--
```

How Amazon EC2 handles user data for Windows instances

On Windows instances, the launch agent performs the tasks related to user data. For more information, see the following:

- EC2Launch v2
- EC2Launch
- EC2Config service

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 an example of running commands on an instance within an Auto Scaling group that works with lifecycle hooks, see Tutorial: Configure user data to retrieve the target lifecycle state through instance metadata in the *Amazon EC2 Auto Scaling User Guide*.

Contents

- Compressed user data
- User data execution
- User data and the Tools for Windows PowerShell

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. 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
- Syntax for Windows PowerShell scripts
- Syntax for YAML configuration scripts
- Base64 encoding

Syntax for batch scripts

Specify a batch script using the `script` tag. Separate the commands using line breaks as shown in the following example.

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
```

By default, user data scripts 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.

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<persist>true</persist>
```

EC2Launch v2 agent

To run an XML user data script as a detached process with the EC2Launch v2 **executeScript** task in the `UserData` stage, add `<detach>true</detach>` to the user data.

> **Note**
>
> The detach tag is not supported by previous launch agents.

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<detach>true</detach>
```

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. The `<powershell>` tag is case-sensitive.

For example:

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
```

```
</powershell>
```

By default, the user data scripts 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-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

You can specify one or more PowerShell arguments with the `<powershellArguments>` tag. If no arguments are passed, EC2Launch and EC2Launch v2 add the following argument by default: `-ExecutionPolicy Unrestricted`.

Example:

```
<powershell>
    $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

EC2Launch v2 agent

To run an XML user data script as a detached process with the EC2Launch v2 **executeScript** task in the `UserData` stage, add `<detach>true</detach>` to the user data.

> **Note**
>
> The detach tag is not supported by previous launch agents.

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

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.

Specify a YAML script with the `executeScript` task.

Example YAML syntax to run a PowerShell script

```
version: 1.0
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
    content: |-
      $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
      New-Item $file -ItemType file
```

Example YAML syntax to run a batch script

```
version: 1.1
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.

```
$UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

The following is an example that decodes using PowerShell.

```
$Script =
 [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData))
```

For more information about base64 encoding, see https://www.ietf.org/rfc/rfc4648.txt.

Compressed user data

EC2Launch v2 supports zipped user data as a method to submit user data that's larger than the 16 KB limit imposed by IMDS. To use this feature, compress your user data script into a `.zip` archive and pass it to your EC2 instance. When EC2Launch v2 detects compressed user data, it automatically unzips the compressed user data script and runs it.

As with standard user data, 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 compressed user data yourself. For more information about the user data size limit and base64 encoding, see Access instance metadata for an EC2 instance.

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.

> **Note**
>
> User data is not enabled to run by default after the initial launch. To enable user data to run when you reboot or start the instance, see Run scripts during subsequent reboots or starts.

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

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
- `Info: Initialize 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

Run scripts during subsequent reboots or starts

When you update instance user data, the updated user data content is automatically reflected in the instance metadata the next time you reboot or start the instance. However, depending on the installed launch agent, additional configuration may be required to configure user data scripts to run on subsequent reboots or starts.

If you choose the **Shutdown with Sysprep** option, user data scripts run the next time the instance starts or reboots, even if you did not enable user data execution for subsequent reboots or starts.

For instructions to enable user data execution, select the tab that matches your launch agent.

EC2Launch v2

Unlike EC2Launch v1, EC2Launch v2 evaluates the user data task on every boot. It is not necessary to manually schedule the user data task. The user data runs based on the included frequency or persist options.

For XML user data scripts

To run user data scripts on every boot, add the `<persist>true</persist>` flag to the user data. If the persist flag is not included, the user data script only runs on the initial boot.

For YAML user data

- To run a task in user data on the initial boot, set the task `frequency` to `once`.

- To run a task in user data on every boot, set the task `frequency` to `always`.

EC2Launch

1. Connect to your Windows instance.
2. Open a PowerShell command window and run one of the following commands:

   Run once

   To run user data once on the next boot, use the `-Schedule` flag.

   ```
   C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   Schedule
   ```

   Run on all subsequent boots

   To run user data on all subsequent boots, use the `-SchedulePerBoot` flag.

   ```
   C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   SchedulePerBoot
   ```
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.

EC2Config

1. Connect to your Windows instance.
2. Open `C:\Program Files\Amazon\Ec2ConfigService\Ec2ConfigServiceSetting.exe`.
3. For **User Data**, select **Enable UserData execution for next service start**.
4. Disconnect from your Windows instance. To run updated scripts the next time the instance is started, stop the instance and update the user data.

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

Access instance metadata for an EC2 instance. For information about user data and the AWS CLI, see User data and the AWS CLI.

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.

```
<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
+DQpSZW5hbWUtQ29tcHV0ZXIgLU5ld05hbWUgdXNlci1kYXRhLXRlc3QNCjwvcG93ZXJzaGVsbD4=
```


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

You can use the Get-EC2Tag command to read the tag value, rename the instance on first boot to match the tag value, and reboot. To run this command successfully, you must have a role with `ec2:DescribeTags` permissions attached to the instance because tag information is retrieved by the API call. For more information on settings permissions by using IAM roles, see Attach an IAM role to an instance.

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
    $instanceId = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" =
 $token} -Method GET -Uri 'http://169.254.169.254/latest/meta-data/instance-id' -
UseBasicParsing
 $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>
```

IMDSv1

```
<powershell>
 $instanceId = (Invoke-WebRequest http://169.254.169.254/latest/meta-data/instance-
id -UseBasicParsing).content
 $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>
```

You can also rename the instance using tags in instance metadata, if your instance is configured to access tags from the instance metadata. For more information, see View tags for your EC2 instances using instance metadata.

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
 $nameValue = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}
 -Method GET -Uri 'http://169.254.169.254/latest/meta-data/tags/instance/Name' -
UseBasicParsing
 $pattern = "^(?![0-9]{1,15}$)[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>
```

IMDSv1

```
<powershell>
 $nameValue = Get-EC2InstanceMetadata -Path /tags/instance/Name
 $pattern = "^(?![0-9]{1,15}$)[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>
```

Identify each instance launched in a single request

This example demonstrates how you can use both user data and instance metadata to configure your Amazon EC2 instances.

> **Note**
>
> The examples in this section use the IPv4 address of the IMDS: `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 IMDS is compatible with IMDSv2 commands. The IPv6 address is only accessible on Nitro-based instances in IPv6-supported subnets (dual stack or IPv6 only).

Alice wants to launch four instances of her favorite database AMI, with the first acting as the original instance and the remaining three acting as replicas. When she launches them, she wants to add user data about the replication strategy for each replica. She is aware that this data will be available to all four instances, so she needs to structure the user data in a way that allows each instance to recognize which parts are applicable to it. She can do this using the `ami-launch-index` instance metadata value, which will be unique for each instance. If she starts more than one instance at the same time, the `ami-launch-index` indicates the order in which the instances were launched. The value of the first instance launched is `0`.

Here is the user data that Alice has constructed.

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

The `replicate-every=1min` data defines the first replica's configuration, `replicate-every=5min` defines the second replica's configuration, and so on. Alice decides to provide this data as an ASCII string with a pipe symbol (`|`) delimiting the data for the separate instances.

Alice launches four instances using the run-instances command, specifying the user data.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 4 \
    --instance-type t2.micro \
    --user-data "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

After they're launched, all instances have a copy of the user data and the common metadata shown here:

- AMI ID: ami-0abcdef1234567890
- Reservation ID: r-1234567890abcabc0
- Public keys: none
- Security group name: default
- Instance type: t2.micro

However, each instance has unique metadata, as shown in the following tables.

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

Metadata	Value
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

Metadata	Value
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice can use the `ami-launch-index` value to determine which portion of the user data is applicable to a particular instance.

1. She connects to one of the instances, and retrieves the `ami-launch-index` for that instance to ensure it is one of the replicas:

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/meta-data/api/
   token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   ```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

For the following steps, the IMDSv2 requests use the stored token from the preceding IMDSv2 command, assuming the token has not expired.

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

2. She saves the `ami-launch-index` as a variable.

   IMDSv2

```
[ec2-user ~]$ ami_launch_index=`curl -H "X-aws-ec2-metadata-token: $TOKEN" 
 http://169.254.169.254/latest/meta-data/ami-launch-index`
```

   IMDSv1

```
[ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-launch-index`
```

3. She saves the user data as a variable.

   IMDSv2

```
[ec2-user ~]$ user_data=`curl -H "X-aws-ec2-metadata-token: $TOKEN" 
 http://169.254.169.254/latest/user-data`
```

   IMDSv1

```
[ec2-user ~]$ user_data=`curl http://169.254.169.254/latest/user-data`
```

4. Finally, Alice uses the **cut** command to extract the portion of the user data that is applicable to that instance.

   IMDSv2

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

IMDSv1

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

Detect whether a host is an EC2 instance

You might need to know whether your application or website is running on an EC2 instance, especially if you have a mixed computing environment. You can use one of the following options to determine whether the host for your application or website is an EC2 instance.

Options

- Inspect the instance identity document
- Inspect the system UUID
- Inspect the system virtual machine generation identifier

Inspect the instance identity document

Each instance has a signed instance identity document that you can verify cryptographically. You can find these documents using the Instance Metadata Service (IMDS).

For more information, see Instance identity documents.

Inspect the system UUID

You can get the system UUID and look in the beginning octet of the UUID for EC2 (in Linux, this might be lowercase `ec2`). This method is quick, but potentially inaccurate because there's a small chance that a system that is not an EC2 instance could have a UUID that starts with these characters. Furthermore, some versions of SMBIOS use the *little-endian* format, which doesn't include EC2 at the beginning of the UUID. This might be the case for EC2 instances that use SMBIOS 2.4 for Windows, or for Linux distributions other than Amazon Linux that have their own implementation of SMBIOS.

Linux example: Get the UUID from DMI (HVM AMIs only)

Use the following command to get the UUID using the Desktop Management Interface (DMI):

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

In the following example output, the UUID starts with "EC2", which indicates that the system is probably an EC2 instance.

```
EC2E1916-9099-7CAF-FD21-012345ABCDEF
```

In the following example output, the UUID is represented in little-endian format.

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

Alternatively, for instances built on the Nitro system, you can use the following command:

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

If the output is an instance ID, as the following example output, the system is an EC2 instance:

```
i-0af01c0123456789a
```

Linux example: Get the UUID from the hypervisor (PV AMIs only)

Use the following command to get the UUID from the hypervisor:

```
[ec2-user ~]$ cat /sys/hypervisor/uuid
```

In the following example output, the UUID starts with "ec2", which indicates that the system is probably an EC2 instance.

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

Windows 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
```

Inspect the system virtual machine generation identifier

A virtual machine generation identifier consists of a unique buffer of 128-bit interpreted as cryptographic random integer identifier. You can retrieve the virtual machine generation identifier to identify your Amazon Elastic Compute Cloud instance. The generation identifier is exposed within the guest operating system of the instance through an ACPI table entry. The value will change if your machine is cloned, copied, or imported into AWS, such as with VM Import/Export.

Example: Retrieve the virtual machine generation identifier from Linux

You can use the following commands to retrieve the virtual machine generation identifier from your instances running Linux.

Amazon Linux 2

1. Update your existing software packages, as necessary, using the following command:

   ```
   sudo yum update
   ```

2. If necessary, source the busybox package with the following command:

   ```
   sudo curl https://www.rpmfind.net/linux/epel/next/8/Everything/x86_64/Packages/
   b/busybox-1.35.0-2.el8.next.x86_64.rpm --output busybox.rpm
   ```

3. If necessary, install the prerequisite packages using the following command:

```
sudo yum install busybox.rpm iasl -y
```

4. Run the following `iasl` command to produce output from the ACPI table:

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

5. Run the following command to review the output of the `iasl` command:

```
cat SSDT2.dsl
```

The output should yield the address space required to retrieve the virtual machine generation identifier:

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
```

```
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
        })
    }
}
}
```

6. (Optional) Elevate your terminal permissions for the remaining steps with the following command:

```
sudo -s
```

7. Use the following command to store the previously gathered address space:

```
VMGN_ADDR=0xFED01000
```

8. Use the following command to iterate through the address space and build the virtual machine generation identifier:

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

9. Retrieve the virtual machine generation identifier from the output file with the following command:

```
cat vmgenid ; echo
```

Your output should be similar to the following:

```
EC2F335D979132C4165896753E72BD1C
```

Ubuntu

1. Update your existing software packages, as necessary, using the following command:

```
sudo apt update
```

2. If necessary, install the prerequisite packages using the following command:

```
sudo apt install busybox iasl -y
```

3. Run the following `iasl` command to produce output from the ACPI table:

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

4. Run the following command to review the output of the `iasl` command:

```
cat SSDT2.dsl
```

The output should yield the address space required to retrieve the virtual machine generation identifier:

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
```

```
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
        })
    }
}
```

```
}
```

5. (Optional) Elevate your terminal permissions for the remaining steps with the following command:

```
sudo -s
```

6. Use the following commands to store the previously gathered address space:

```
VMGN_ADDR=0xFED01000
```

7. Use the following command to iterate through the address space and build the virtual machine generation identifier:

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

8. Retrieve the virtual machine generation identifier from the output file with the following command:

```
cat vmgenid ; echo
```

Your output should be similar to the following:

```
EC2F335D979132C4165896753E72BD1C
```

Example: Retrieve the virtual machine generation identifier from Windows

You can create a sample application to retrieve the virtual machine generation identifier from your instances running Windows. For more information, see Obtaining the virtual machine generation identifier in the Microsoft documentation.

Instance identity documents for Amazon EC2 instances

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 stopped and started, restarted, or launched. You can access the instance identity document for an instance through the Instance Metadata Service (IMDS). For the instructions, see Retrieve the instance identity document.

The instance identity document uses plaintext JSON format. It includes the following information.

Data	Description
`accountId`	The ID of the AWS account that launched the instance.
`architecture`	The architecture of the AMI used to launch the instance (i386 | x86_64 | arm64).
`availabilityZone`	The Availability Zone in which the instance is running.
`billingProducts`	The billing products of the instance.
`devpayProductCodes`	Deprecated.
`imageId`	The ID of the AMI used to launch the instance.
`instanceId`	The ID of the instance.
`instanceType`	The instance type of the instance.
`kernelId`	The ID of the kernel associated with the instance, if applicable.
`marketplaceProductCodes`	The AWS Marketplace product code of the AMI used to launch the instance.
`pendingTime`	The date and time that the instance was launched.
`privateIp`	The private IPv4 address of the instance.
`ramdiskId`	The ID of the RAM disk associated with the instance, if applicable.
`region`	The Region in which the instance is running.
`version`	The version of the instance identity document format.

Retrieve the instance identity document for an EC2 instance

The instance identity document for an Amazon EC2 instance uses a plaintext JSON format. For a description of the contents of an instance identity document, see the section called “Instance identity documents”.

The instance identity document is stored in the instance metadata for the instance, in the `instance-identity/document` dynamic data category. You access the instance identity document for an instance by connecting to the instance and retrieving it from the instance metadata.

You can access instance metadata using the IPv4 address 169.254.169.254 or the IPv6 address fd00:ec2::254. These are Link-local addresses, meaning that you can access them only from the instance. The examples on this page use the IPv4 address of the IMDS: 169.254.169.254. To retrieve instance metadata for EC2 instances over IPv6, use fd00:ec2::254.

To verify the authenticity of an instance identity document after you retrieve it, see Verify instance identity document.

IMDSv2

Linux

Run the following command from your Linux instance to retrieve the instance identity document.

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

Run the following cmdlet from your Windows instance to retrieve the instance identity document.

```
[string]$token = (Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token-ttl-seconds' = '21600'} `
    -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -UseBasicParsing).Content
```

```
(Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $token} `
    -Uri 'http://169.254.169.254/latest/dynamic/instance-identity/document' -
UseBasicParsing).Content
```

IMDSv1

Linux

Run the following command from your Linux instance to retrieve the instance identity document.

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

Run the following cmdlet from your Windows instance to retrieve the instance identity document.

```
(Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/
document).Content
```

The following is example output.

```
{
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
    "availabilityZone" : "us-west-2b",
    "privateIp" : "10.158.112.84",
    "version" : "2017-09-30",
    "instanceId" : "i-1234567890abcdef0",
    "billingProducts" : null,
    "instanceType" : "t2.micro",
    "accountId" : "123456789012",
    "imageId" : "ami-5fb8c835",
    "pendingTime" : "2016-11-19T16:32:11Z",
    "architecture" : "x86_64",
    "kernelId" : null,
    "ramdiskId" : null,
    "region" : "us-west-2"
}
```

Verify the instance identity document for an Amazon EC2 instance

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.

Options

- Option 1: Verify instance identity document using the PKCS7 signature
- Option 2: Verify instance identity document using the base64-encoded signature
- Option 3: Verify instance identity document using the RSA-2048 signature

Option 1: Verify instance identity document using the PKCS7 signature

This topic explains how to verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate.

Linux instances

To verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate

1. Connect to the instance.

2. Retrieve the PKCS7 signature from the instance metadata and add it to a new file named `pkcs7` along with the required header and footer. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   $ echo "-----BEGIN PKCS7-----" >> pkcs7 \
    && TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7 \
    && echo "" >> pkcs7 \
    && echo "-----END PKCS7-----" >> pkcs7
   ```

   IMDSv1

   ```
   $ echo "-----BEGIN PKCS7-----" >> pkcs7 \
    && curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
    >> pkcs7 \
    && echo "" >> pkcs7 \
    && echo "-----END PKCS7-----" >> pkcs7
   ```

3. Find the **DSA** public certificate for your Region in [AWS public certificates for instance identity document signatures] and add the contents to a new file named `certificate`.

4. Use the **OpenSSL smime** command to verify the signature. Include the `-verify` option to indicate that the signature needs to be verified, and the `-noverify` option to indicate that the certificate does not need to be verified.

   ```
   $ openssl smime -verify -in pkcs7 -inform PEM -certfile certificate -noverify | tee
    document
   ```

   If the signature is valid, the `Verification successful` message appears.

   The command also writes the contents of the instance identity document to a new file named `document`. You can compare the contents of the of the instance identity document from the instance metadata with the contents of this file using the following commands.

   ```
   $ openssl dgst -sha256 < document
   ```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

If the signature cannot be verified, contact Support.

Windows instances

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

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-ec2-
   metadata-token-ttl-seconds' = '21600'} -Uri http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} -Uri http://169.254.169.254/latest/
   dynamic/instance-identity/pkcs7).Content)
   ```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Uri
 http://169.254.169.254/latest/dynamic/instance-identity/pkcs7).Content)
```

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

```
PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} -Uri http://169.254.169.254/latest/
dynamic/instance-identity/document).Content)
```

   IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. Find the **DSA** public certificate for your Region in AWS public certificates for instance identity document signatures and add the contents to a new file named `certificate.pem`.
5. Extract the certificate from the certificate file and store it in a variable named `$Store`.

```
PS C:\> $Store =
 [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
Path certificate.pem)))
```

6. Verify the signature.

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
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 your signature cannot be verified, contact AWS Support.

7. Validate the content of the instance identity document.

```
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 can't be validated, contact AWS Support.

Option 2: Verify instance identity document using the base64-encoded signature

This topic explains how to verify the instance identity document using the base64-encoded signature and the AWS RSA public certificate.

Linux instances

To validate the instance identity document using the base64-encoded signature and the AWS RSA public certificate

1. Connect to the instance.
2. Retrieve the base64-encoded signature from the instance metadata, convert it to binary, and add it to a file named `signature`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   $ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/signature | base64 -d >> signature
   ```

   IMDSv1

   ```
   $ curl -s http://169.254.169.254/latest/dynamic/instance-identity/signature |
    base64 -d >> signature
   ```

3. Retrieve the plaintext instance identity document from the instance metadata and add it to a file named `document`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   $ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/document >> document
   ```

   IMDSv1

   ```
   $ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document
    >> document
   ```

4. Find the **RSA** public certificate for your Region in AWS public certificates for instance identity document signatures and add the contents to a new file named `certificate`.
5. Extract the public key from the AWS RSA public certificate and save it to a file named `key`.

   ```
   $ openssl x509 -pubkey -noout -in certificate >> key
   ```

6. Use **OpenSSL dgst** command to verify the instance identity document.

   ```
   $ openssl dgst -sha256 -verify key -signature signature document
   ```

   If the signature is valid, the `Verification successful` message appears.

   The command also writes the contents of the instance identity document to a new file named `document`. You can compare the contents of the of the instance identity document from the instance metadata with the contents of this file using the following commands.

   ```
   $ openssl dgst -sha256 < document
   ```

   ```
   $ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/document | openssl dgst -sha256
   ```

   If the signature cannot be verified, contact Support.

Windows instances

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

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/instance-identity/signature).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/signature).Content)
   ```

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

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. Find the **RSA** public certificate for your Region in [AWS public certificates for instance identity document signatures](#) and add the contents to a new file named `certificate.pem`.

5. Verify the instance identity document.

```
PS C:\> [Security.Cryptography.X509Certificates.X509Certificate2]::new((Resolve-
Path certificate.pem)).PublicKey.Key.VerifyData($Document, 'SHA256', $Signature)
```

If the signature is valid, the command returns `True`. If the signature cannot be verified, contact Support.

Option 3: Verify instance identity document using the RSA-2048 signature

This topic explains how to verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate.

Linux instances

To verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate

1. Connect to the instance.

2. Retrieve the RSA-2048 signature from the instance metadata and add it to a file named `rsa2048` along the required header and footer. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ echo "-----BEGIN PKCS7-----" >> rsa2048 \
 && TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/rsa2048 >> rsa2048 \
 && echo "" >> rsa2048 \
 && echo "-----END PKCS7-----" >> rsa2048
```

IMDSv1

```
$ echo "-----BEGIN PKCS7-----" >> rsa2048 \
```

```
&& curl -s http://169.254.169.254/latest/dynamic/instance-identity/rsa2048
>> rsa2048 \
&& echo "" >> rsa2048 \
&& echo "-----END PKCS7-----" >> rsa2048
```

3. Find the **RSA-2048** public certificate for your Region in [AWS public certificates for instance identity document signatures](#) and add the contents to a new file named `certificate`.
4. Use the **OpenSSL smime** command to verify the signature. Include the `-verify` option to indicate that the signature needs to be verified, and the `-noverify` option to indicate that the certificate does not need to be verified.

   ```
   $ openssl smime -verify -in rsa2048 -inform PEM -certfile certificate -noverify |
    tee document
   ```

   If the signature is valid, the `Verification successful` message appears. If the signature cannot be verified, contact Support.

Windows instances

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 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

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
instance-identity/rsa2048).Content)
```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/rsa2048).Content)
```

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

```
PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
instance-identity/document).Content)
```

IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. Find the **RSA-2048** public certificate for your Region in AWS public certificates for instance identity document signatures and add the contents to a new file named `certificate.pem`.

5. Extract the certificate from the certificate file and store it in a variable named `$Store`.

```
PS C:\> $Store =
 [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
Path certificate.pem)))
```

6. Verify the signature.

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
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 your signature cannot be verified, contact AWS Support.

7. Validate the content of the instance identity document.

```
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 can't be validated, contact AWS Support.

AWS public certificates for instance identity document signatures

The following AWS public certificates can be used to verify the contents of an instance's instance identity document as described in Verify instance identity document.

Ensure that you use the correct certificate for your Region and for the verification procedure that you're using. If you're verifying the PKCS7 signature, use the DSA certificate. If you're verifying the base64-encoded signature, use the RSA certificate. If you're verifying the RSA-2048 signature, use the RSA-2048 certificate.

Expand each Region below to view the Region-specific certificates.

US East (N. Virginia) — us-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
```

```
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUE1y2NIKCU+Rg4uu4u32koG9QEYIwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MzQwMVoXDTI5MDQyODE3MzQwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUE1y2NIKC
U+Rg4uu4u32koG9QEYIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAlxSmwcWnhT4uAeSinJuz+1BTcKhVSWb5jT8pYjQb8ZoZkXXRGb09mvYeU
NeqOBr27rvRAnaQ/9LUQf72+SahDFuS4CMI8nwowytqbmwquqFr4dxA/SDADyRiF
ea1UoMuNHTY49J/1vPomqsVn7mugTp+TbjqCfOJTpu0temHcFA==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALFpzEAVWaQZMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU5MTJaGA8yMTk1MDExNzA4NTkxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjS2vqZu9mEOhOq+0bRpAbCUiapbZMFNQqRg7kTlr7Cf+gDqXKpHPjsng
SfNz+JHQd8WPI+pmNs+q0Z2aTe23klmf2U52KH9/j1k8RlIbap/yFibFTSedmegX
E5r447GbJRsHUmuIIfZTZ/oRlpuIIO5/Vz7SOj22tdkdY2ADp7caZkNxhSP915fk
2jJMTBUOzyXUS2rBU/ulNHbTTeePjcEkvzVYPahD30TeQ+/A+uWUu89bHSQOJR8h
Um4cFApzZgN3aD5j2LrSMu2pctkQwf9CaWyVznqrsGYjYOY66LuFzSCXwqSnFBfv
fFBAFsjCgY24G2DoMyYkF3MyZlu+rwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUrynSPp4uqSECwy+PiO4qyJ8TWSkwgY4GA1UdIwSBhjCBg4AUrynS
Pp4uqSECwy+PiO4qyJ8TWSmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALFpzEAVWaQZMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADW/s8lXijwdP6NkEoH1m9XLrvK4YTqkNfR6
er/uRRgTx2QjFcMNrx+g87gAml11z+D0crAZ5LbEhDMs+JtZYR3ty0HkDk6SJM85
haoJNAFF7EQ/zCp1EJRIkLLsC7bcDL/Eriv1swt78/BB4RnC9W9kSp/sxd5svJMg
N9a6FAplpNRsWAnbP8JBlAP93oJzblX2LQXgykTghMkQO7NaY5hg/H5o4dMPclTK
lYGqlFUCH6A2vdrxmpKDLmTn5//5pujdD2MN0df6sZWtxwZ0osljV4rDjm9Q3VpA
NWIsDEcp3GUB4proOR+C7PNkY+VGODitBOw09qBGosCBstwyEqY=
-----END CERTIFICATE-----
```


US East (Ohio) — us-east-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
```

```
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUVJTc+hOU+8Gk3JlqsX438Dk5c58wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MTE0OVoXDTI5MDQyODE3MTE0OVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUVJTc+hOU
+8Gk3JlqsX438Dk5c58wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAywJQaVNWJqW0R0T0xVOSoN1GLk9x9kKEuN67RN9CLin4dA97qa7Mr5W4P
FZ6vnh5CjOhQBRXV9xJUeYSdqVItNAUFK/fEzDdjf1nUfPlQ3OJ49u6CV01NoJ9m
usvY9kWcV46dqn2bk2MyfTTgvmeqP8fiMRPxxnVRkSzlldP5Fg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM07oeX4xevdMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA2MTAx
MjU4MThaGA8yMTk1MTExNDEyNTgxOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA6v6kGMnRmFDLxBEqXzP4npnL65OO0kmQ7w8YXQygSdmNIoScGSU5wfh9
mZdcvCxCdxgALFsFqPvH8fqiE9ttI0fEfuZvHOs8wUsIdKr0Zz0MjSx3cik4tKET
ch0EKfMnzKOgDBavraCDeX1rUDU0Rg7HFqNAOry3uqDmnqtk00XC9GenS3z/7ebJ
fIBEPAam5oYMVFpX6M6St77WdNE8wEU8SuerQughiMVx9kMB07imeVHBiELbMQ0N
lwSWRL/61fA02keGSTfSp/0m3u+lesf2VwVFhqIJs+JbsEscPxOkIRlzy8mGd/JV
ONb/DQpTedzUKLgXbw7KtO3HTG9iXQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU2CTGYE5fTjx7gQXzdZSGPEWAJY4wgY4GA1UdIwSBhjCBg4AU2CTG
YE5fTjx7gQXzdZSGPEWAJY6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
```

```
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM07oeX4xevdMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANdqkIpVypr2PveqUsAKke1wKCOSuw1UmH9k
xX1/VRoHbrI/UznrXtPQOPMmHA2LKSTedwsJuorUn3cFH6qNs8ixBDrl8pZwfKOY
IBJcTFBbI1xBEFkZoO3wczzo5+8vPQ60RVqAaYb+iCa1HFJpccC3Ovajfa4GRdNb
n6FYnluIcDbmpcQePoVQwX7W3oOYLB1QLN7fE6H1j4TBIsFdO3OuKzmaifQlwLYt
DVxVCNDabpOr6Uozd5ASm4ihPPoEoKo7Ilp0fOT6fZ41U2xWA4+HF/89UoygZSo7
K+cQ90xGxJ+gmlYbLFR5rbJOLfjrgDAb2ogbFy8LzHo2ZtSe60M=
-----END CERTIFICATE-----
```

US West (N. California) — us-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUK2zmY9PUSTR7rc1k2OwPYu4+g7wwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MDI0M1oXDTI5MDQyODE3MDI0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
```

```
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUK2zmY9PU
STR7rc1k2OwPYu4+g7wwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQA1Ng4QmN4n7iPh5CnadSOc0ZfM7by0dBePwZJyGvOHdaw6P6E/vEk76KsC
Q8p+akuzVzVPkU4kBK/TRqLp19wEWoVwhhTaxHjQ1tTRHqXIVlrkw4JrtFbeNM21
GlkSLonuzmNZdivn9WuQYeGe7nUD4w3q9GgiF3CPorJe+UxtbA==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJANNPkIpcyEtIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAzMDdaGA8yMTk1MDQwMzA5MDMwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApHQGvHvq3SVCzDrC7575BW7GWLzcj8CLqYcL3YY7JffupZ7OjcftO57Z
4fo5Pj0CaS8DtPzh8+8vdwUSMbiJ6cDd3ooio3MnCq6DwzmsY+pY7CiI3UVG7KcH
4TriDqr1Iii7nB5MiPJ8wTeAqX89T3SYaf6Vo+4GCb3LCDGvnkZ9TrGcz2CHkJsj
AIGwgopFpwhIjVYm7obmuIxSIUv+oNH0wXgDL029Zd98SnIYQd/njiqkzE+lvXgk
4h4Tu17xZIKBgFcTtWPky+POGu81DYFqiWVEyR2JKKm2/iR1dL1YsT39kbNg47xY
aR129sS4nB5Vw3TRQA2jL0ToTIxzhQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUgepyiONs8j+q67dmcWu+mKKDa+gwgY4GA1UdIwSBhjCBg4AUgepy
iONs8j+q67dmcWu+mKKDa+ihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJANNPkIpcyEtIMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGLFWyutf1u0xcAc+kmnMPqtc/Q6b79VIX0E
tNoKMI2KR8lcV8ZElXDb0NC6v8UeLpe1WBKjaWQtEjL1ifKg9hdY9RJj4RXIDSK7
33qCQ8juF4vep2U5TTBd6hfWxt1Izi88xudjixmbpUU4YKr8UPbmixldYR+BEx0u
B1KJi9l1lxvuc/Igy/xeHOAZEjAXzVvHp8Bne33VVwMiMxWECZCiJxE4I7+Y6fqJ
pLLSFFJKbNaFyXlDiJ3kXyePEZSc1xiWeyRB2ZbTi5eu7vMG4i3AYWuFVLthaBgu
lPfHafJpj/JDcqt2vKUKfur5edQ6j1CGdxqqjawhOTEqcN8m7us=
-----END CERTIFICATE-----
```

US West (Oregon) — us-west-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFx8PxCkbHwpD31bOyCtyz3GclbgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MjM1OVoXDTI5MDQyODE3MjM1OVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFx8PxCkb
HwpD31bOyCtyz3GclbgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBzOl+9Xy1+UsbUBI95HO9mbbdnuX+aMJXgG9uFZNjgNEbMcvx+h8P9IMko
z7PzFdheQQ1NLjsHH9mSR1SyC4m9ja6BsejH5nLBWyCdjfdP3muZM4O5+r7vUa1O
dWU+hP/T7DUrPAIVMOE7mpYa+WPWJrN6BlRwQkKQ7twm9kDalA==
```

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


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALZL3lrQCSTMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAxMzJaGA8yMTk1MDExNzA5MDEzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA02Y59qtAA0a6uzo7nEQcnJ26OKF+LRPwZfixBH+EbEN/Fx0gYy1jpjCP
s5+VRNg6/WbfqAsV6X2VSjUKN59ZMnMY9ALA/Ipz0n00Huxj38EBZmX/NdNqKm7C
qWu1q5kmIvYjKGiadfboU8wLwLcHo8ywvfgI6FiGGsEO9VMC56E/hL6Cohko11LW
dizyvRcvg/IidazVkJQCN/4zC9PUOVyKdhW33jXy8BTg/QH927QuNk+ZzD7HH//y
tIYxDhR6TIZsSnRjz3bOcEHxt1nsidc65mY0ejQty4hy7ioSiapw316mdbtE+RTN
fcH9FPIFKQNBpiqfAW5Ebp3Lal3/+wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU7coQx8Qnd75qA9XotSWT3IhvJmowgY4GA1UdIwSBhjCBg4AU7coQ
x8Qnd75qA9XotSWT3IhvJmqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALZL3lrQCSTMMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFZ1e2MnzRaXCaLwEC1pW/f0oRG8nHrlPZ9W
OYZEWbh+QanRgaikBNDtVTwARQcZm3z+HWSkaIx3cyb6vM0DSkZuiwzm1LJ9rDPc
aBm03SEt5v8mcc7sXWvgFjCnUpzosmky6JheCD4O1Cf8k0olZ93FQnTrbg62OK0h
83mGCDeVKU3hLH97FYoUq+3N/IliWFDhvibAYYKFJydZLhIdlCiiB99AM6Sg53rm
oukS3csyUxZyTU2hQfdjyo1nqW9yhvFAKjnnggiwxNKTTPZzstKW8+cnYwiiTwJN
QpVoZdt0SfbuNnmwRUMi+QbuccXweav29QeQ3ADqjgB0CZdSRKk=
-----END CERTIFICATE-----
```


Africa (Cape Town) — af-south-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7DCCAqwCCQCncbCtQbjuyzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA2MDQxMjQ4MDVaFw00
NTA2MDQxMjQ4MDVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQC12Nr1gMrHcFSZ7S/A
pQBSCMHWmn2qeoQTMVWqe50fnTd0zGFxDdIjKxUK58/8zjWG5uR4TXRzmZpGpmXB
bSufAR6BGqud2LnT/HIWGJAsnX2uOtSyNfCoJigqwhea5w+CqZ6I7iBDdnB4TtTw
```

```
qO6TlnExHFVj8LMkylZgiaE1CQIVAIhdobse4K0QnbAhCL6R2euQzloXAoGAV/21
WUuMz/79Ga0JvQcz1FNy1sT0pU9rU4TenqLQIt5iccn/7EIfNtvVO5TZKulIKq7J
gXZr0x/KIT8zsNweetLOaGehPIYRMPX0vunMMR7hN7qA7W17WZv/76adywIsnDKq
ekfe15jinaX8MsKUdyDK7Y+ifCG4PVhoM4+W2XwDgYQAAoGAIxOKbVgwLxbn6Pi2
6hBOihFv16jKxAQI0hHzXJLV0Vyv9QwnqjJJRfOCy3dB0zicLXiIxeIdYfvqJr+u
hlN8rGxEZYYJjEUKMGvsc0DW85jonXz0bNfcP0aaKH0lKKVjL+OZi5n2kn9wgdo5
F3CVnMl8BUra8A1Tr2yrrE6TVZ4wCQYHKoZIzjgEAwMvADAsAhQfa7MCJZ+/TEY5
AUr0J4wm8VzjoAIUSYZVu2NdRJ/ERPmDfhW5EsjHlCA=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAKumfZiRrNvHMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTExMjcw
NzE0MDVaGA8yMTk5MDUwMjA3MTQwNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFd571nUzVtke3rPyRkYfvs3jh0C0EMzzG72boyUNjnfw1+m0TeFraTLKb9T6F
7TuB/ZEN+vmlYqr2+5Va8U8qLbPF0bRH+FdaKjhgWZdYXxGzQzU3ioy5W5ZM1VyB
7iUsxEAlxsybC3ziPYaHI42UiTkQNahmoroNeqVyHNnBpQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAAJLylWyElEgOpW4B1XPyRVD4pAds8Guw2+krgqkY0HxLCdjosuH
RytGDGN+q75aAoXzW5a7SGpxLxk6Hfv0xp3RjDHsoeP0i1d8MD3hAC5ezxS4oukK
s5gbPOnokhKTMPXbTdRn5ZifCbWlx+bYN/mTYKvxho7b5SVg2o1La9aK
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAIFI+O5A6/ZIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA2MDQx
MjQ4MDRaGA8yMTk4MTEwNzEyNDgwNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy7/WHBBHOrk+20aumT07g8rxrSM0UXgki3eYgKauPCG4Xx//vwQbuZwI
oeVmR9nqnhfij2wOcQdbLandh0EGtbxerete3IoXzd1KXJb11PVmzrzyu5SPBPuP
iCeV4qdjjkXo2YWM6t9YQ911hcG96YSp89TBXFYUh3KLxfqAdTVhuC0NRGhXpyii
j/czo9njofHhqhTr7UEyPun8NVS2QWctLQ86N5zWR3Q0GRoVqqMrJs0cowHTrVw2
9Qr7QBjjBOVbyYmtYxm/DtiKprYV/e6bCAVok015X1sZDd3oCOQNoGlv5XbHJe2o
JFD8GRRy2rkWO/lNwVFDcwec6zC3QwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCE
goqzjpCpmMgCpszFHwvRaSMbspKtK7wNImUjrSBOfBJsfFulyg1Zgn2nDCK7kQhx
```

```
jMJmNIvXbps3yMqQ2cHUkKcKf5t+WldfeT4Vk1Rz6HSA8sd0kgVcIesIaoy2aaXU
VEB/oQziRGyKdN1d4TGYVZXG44CkrzSDvlbmfiTq5tL+kAieznVF3bzHgPZW6hKP
EXC3G/IXrXicFEe6YyE1Rakl62VncYSXiGe/i2XvsiNH3Qlmnx5XS7W0SCN0oAxW
EH9twibauv82DVg1WOkQu8EwFw8hFde9X0Rkiu0qVcuU8lJgFEvPWMDFU5sGB6ZM
gkEKTzMvlZpPbBhg99Jl
-----END CERTIFICATE-----
```


Asia Pacific (Hong Kong) — ap-east-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQCO7MJe5Y3VLjAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDMwMjIxMjFaFw00
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDvQ9RzVvf4MAwGbqfX
blCvCoVb9957OkLGn/04CowHXJ+vTBR7eyIa6AoXltsQXBOmrJswToFKKxT4gbuw
jK7s9QQX4CmTRWcEgO2RXtZSVjOhsUQMh+yf7Ht4OVL97LWnNfGsX2cwjcRWHYgI
7lvnuBNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGKd9FAoGBAOCG
eSNmxpW4QFu4pIlAykm6EnTZKKHT87gdXkAkfoC5fAfOxxhnE2HezZHp9Ap2tMV5
8bWNvoPHvoKCQqwfm+OUBlAxC/3vqoVkKL2mG1KgUH9+hrtpMTkwO3RREnKe7I5O
x9qDimJpOihrL4I0dYvy9xUOoz+DzFAW8+ylWVYpA4GFAAKBgQDbnBAKSxWr9QHY
6Dt+EFdGz6lAZLedeBKpaP53Z1DTO34J0C55YbJTwBTFGqPtOLxnUVDlGiD6GbmC
80f3jvogPR1mSmGsydbNbZnbUEVWrRhe+y5zJ3g9qs/DWmDW0deEFvkhWVnLJkFJ
9pdOu/ibRPH1lE2nz6pK7GbOQtLyHTAJBgcqhkjOOAQDAzAAMC0CFQCoJlwGtJQC
cLoM4p/jtVFOj26xbgIUUS4pDKyHaG/eaygLTtFpFJqzWHc=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICSzCCAbQCCQDtQvkVxRvK9TANBgkqhkiG9w0BAQsFADBqMQswCQYDVQQGEwJV
UzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2VhdHRsZTEYMBYGA1UE
ChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1hem9uYXdzLmNvbTAe
Fw0xOTAyMDMwMzAwMDZaFw0yOTAyMDIwMzAwMDZaMGoxCzAJBgNVBAYTAlVTMRMw
EQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgwFgYDVQQKEw9B
bWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3MuY29tMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1kkHXYTfc7gY5Q55JJhjTieHAgacaQkiR
Pity9QPDE3b+NXDh4UdP1xdIw73JcIIG3sG9RhWiXVCHh6KkuCTqJfPUknIKk8vs
M3RXflUpBe8Pf+P92pxqPMCz1Fr2NehS3JhhpkCZVGxxwLC5gaG0Lr4rFORubjYY
```

```
Rh84dK98VwIDAQABMA0GCSqGSIb3DQEBCwUAA4GBAA6xV9f0HMqXjPHuGILDyaNN
dKcvplNFwDTydVg32MNubAGnecoEBtUPtxBsLoVYXCOb+b5/ZMDubPF9tU/vSXuo
TpYM5Bq57gJzDRaBOntQbX9bgHiUxw6XZWaTS/6xjRJDT5p3S1E0mPI3lP/eJv4o
Ezk5zb3eIf10/sqt4756
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMoxixvs3YssMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA3MjAw
ODQ0NDRaGA8yMTk3MTIyMzA4NDQ0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA4T1PNsOg0FDrGlWePoHeOSmOJTA3HCRy5LSbYD33GFU2eBrOIxoU/+SM
rInKu3GghAMfH7WxPW3etIAZiyTDDU5RLcUq2Qwdr/ZpXAWpYocNc/CEmBFtfbxF
z4uwBIN3/drM0RSbe/wP9EcgmNUGQMMZWeAji8sMtwpOblNWAP9BniUG0Flcz6Dp
uPovwDTLdAYT3TyhzlohKL3f6O48TR5yTaV+3Ran2SGRhyJjfh3FRpP4VC+z5LnT
WPQHN74Kdq35UgrUxNhJraMGCzznolUuoR/tFMwR93401GsM9fVA7SW3jjCGF81z
PSzjy+ArKyQqIpLW1YGWDFk3sf08FQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDK
2/+C3nPMgtyOFX/I3Cyk+Pui44IgOwCsIdNGwuJysdqp5VIfnjegEu2zIMWJSKGO
lMZoQXjffkVZZ97J7RNDW06oB7kj3WVE8a7U4WEOfnO/CbMUf/x99CckNDwpjgW+
K8V8SzAsQDvYZs2KaE+18GFfLVF1TGUYK2rPSZMHyX+v/TIlc/qUceBycrIQ/kke
jDFsihUMLqgmOV2hXKUpIsmiWMGrFQV4AeV0iXP8L/ZhcepLf1t5SbsGdUA3AUY1
3If8s81uTheiQjwY5t9nM0SY/1Th/tL3+RaEI79VNEVfG1FQ8mgqCK0ar4m0oZJl
tmmEJM7xeURdpBBx36Di
-----END CERTIFICATE-----
```

Asia Pacific (Hyderabad) — ap-south-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXjrQ4+XMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAJCKGBBoxIUxqBk94JHhwZZbgvbP0DAOoHENQWxp/98lI7/
```

```
YOfYJOVMJS22aCnHDurofmo5rvNIkgXi7Rztbhu
+lko9rK6DgpmpUwBU0WZtf34aZ2IWNBwHaVhHvWAQf9/46u18dMa2YucK1Wi+Vc+M
+KldrvgXmhym6ErNlzhJyMAkGByqGSM44BAMDLwAwLAIUaaPKxaOHoYvwz709xXpsQueIq+UCFFa/
GpzoDOSokl1057NU/2hnsiW4
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjwLj9CMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+sFcobrjvcAYm0PNRD8f4R1jAzvoLt2+qGeOTAyO1Httj6cmsYN3AP1hN5iYuppFiYsl2eNPa/
CD0Vg0BAfDFlV5rzjpA0j7TJabVh4kj7JvtD+xYMi6wEQA4x6SPONY4OeZ2+8o/
HS8nucpWDVdPRO6ciWUlMhjmDmwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAAy6sgTdRkTqELHBeWj69q60xHyUmsWqHAQ
TGGbYP0yP2qfMl0cCImzRI5W0gn8gogdervfeT7nH5ih0TWEy/QDWfkQ601L4erm4yh4YQq8vcqAPSkf04N
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAIVWfPw/X82fMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MDQx
NDMwMjhaGA8yMjAxMTIwODE0MzAyOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAg29QEFriG+qFEjYW/v62nN70lMJY/Hevx5TtmU/VIYBPQa3HUGTBAbbI
2Tmy8UMpa8kZeaYeI3RAfiQWt0Ws7wUrBuO2Pdp518WDPaJUH7RWEuu1BDDkyZRW
NAMNPCn3ph7Od243IFcLGku7HVekel5poqRpSfojrMasjlf+CvixUeAJbmFoxUHK
kh5unzG2sZyO4wHXcJPQkRf5a8zSTPe9YZP1kXPPEv4p/jTSggaYPxXyS6QVaTlV
zLeLFZ0fesLPMeil3KYQtV7IKLQiEA2F6dxWnxNWQlyMHtdq6PucfEmVx17i/Xza
yNBRo0azY8WUNVkEXrRhp/pU8Nh3GQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU9AO1aZk9RLXk2ZvRVoUxYvQy9uwwgY4GA1UdIwSBhjCBg4AU9AO1
aZk9RLXk2ZvRVoUxYvQy9uyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAIVWfPw/X82fMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADexluMRQRftqViahCnauEWGdMvLCBr8A+Yr
6hJq0guoxEk/lahxR137DnfMPuSbi1Rx5QKo7oBrWfG/zsgQUnF2IwHTzwD+i/2m
XCane6FiS5RpK3lGdILq8ZmlhQk+6iI8yoZLr0LCfTh+CLgIKHOknfR5lFzgzAiF
SI8/Q9mm+uvYtSTZECI6Zh57QZPoETAG/yl+9jiOy2lAelqa/k1i+Qo8gMfOc+Pm
dwY7o6fV+oucgRlsdey6VM45LeyILQqv0RXtVzjuowanzmCCFMjgqiO9oZAWu4Oh
+F3unijELo01vZJs8s2N3KGlo3/jtUFTX6RTKShZlAPLwBi5GMI=
-----END CERTIFICATE-----
```

Asia Pacific (Jakarta) — ap-southeast-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXbVDEikMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAPjuiEx05N3JQ6cVwntJie67D8OuNo4jGRn
+crEtL7YO0jSVB9zGE1ga
+UgRPIaYETL293S8rTJTVgXAqdpBwfaHC6NUzre8U8iJ8FMNnlP9Gw1oUIlgQBjORyynVJexoB31TDZM
+/52g9O/bpq1QqNyKbeIgyBBlc1dAtr1QLnsMAkGByqGSM44BAMDLwAwLAIUK8E6RDIRtwK+9qnaTOBhvO/
njuQCFFocyT1OxK+UDR888oNsdgtif2Sf
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXbVDG2yMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
Vbt0gQ1ebWcur2hSO7PnJifE4OPxQ7RgSAlc4/spJp1sDP+ZrS0LO1ZJfKhXf1R9S3AUwLnsc7b
+IuVXdY5LK9RKqu64nyXP5dx170zoL8loEyCSuRR2fs+04i2QsWBVP+KFNAn7P5L1EHRjkgTO8kjNKviwRV
+OkP9ab5wIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAI4WUy6+DKh0JDSzQEZNyBgNlSoSuC2owtMxCwGB6nBfzzfcekWvs
+87w/g9lNwUnUt0ZHYyh2tuBG6hVJuUEwDJ/z3wDd6wQviLOTF3MITawt9P8siR1hXqLJNxpjRQFZrgHqi
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMtdyRcH51j9MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MDgx
MjM5MTZaGA8yMjAxMDkxMjEyMzkxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvUsKCxoH6KXRYJLeYTWAQfaBQeCwhJaR56mfUeFHJE4g8aFjWkiN4uc1
TvOyYNnIZKTHWmzmulmdinWNbwP0GiROHb/i7ro0HhvnptyycGt8ag8affiIbx5X
7ohdwSN2KJ6G0IKflIx7f2NEI0oAMM/9k+T1eVF+MVWzpZoiDp8frLNkqp8+RAgz
ScZsbRfwv3u/if5xJAvdg2nCkIWDMSHEVPoz0lJo7v0ZuDtWWsL1LHnL5ozvsKEk
+ZJyEi23r+U1hIT1NTBdp4yoigNQexedtwCSr7q36oOdDwvZpqYlkLi3uxZ4ta+a
```

```
01pzOSTwMLgQZSbKWQrpMvsIAPrxoQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU1GgnGdNpbnL3lLF30Jomg7Ji9hYwgY4GA1UdIwSBhjCBg4AU1Ggn
GdNpbnL3lLF30Jomg7Ji9hahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMtdyRcH51j9MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACVl00qQlatBKVeiWMrhpczsJroxDxlZTOba
6wTMZk7c3akb6XMOSZFbGaifkebPZqTHEhDlrClM2j9AIlYcCx6YCrTf4cuhn2mD
gcJN33143eOWSaeRY3ee4j+V9ne98y3kO2wLz95VrRgclPFR8po2iWGzGhwUi+FG
q8dXeCH3N0DZgQsSgQWwmdNQXZZej6RHLU/8In5trHKLY0ppnLBjn/UZQbeTyW5q
RJB3GaveXjfgFUWj2qOcDuRGaikdS+dYaLsi5z9cA3FolHzWxx9MOs8io8vKqQzV
XUrLTNWwuhZy88cOlqGPxnoRbw7TmifwPw/cunNrsjUUOgs6ZTk=
-----END CERTIFICATE-----
```

Asia Pacific (Malaysia) — ap-southeast-5

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQC5X6U+vgoLEDAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMxMjU3NTRaGA8y
MDUwMDEwMzEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBAIzEMPCIPFfOYCg4
BCjKGyl60wOfmwHPzSOXZ3Z2wS/LYNHUtHGwtVNePSTyCu/CuZF6gC9n/wBORtQp
+SSkn+weGc/BmUA1mp/vrN7v+aSCgKJoO+sgpa1PPOgNvUaMw6O5odsZWQCMSjkU
6RTo/PL2v/tMfiCocF4ghvyRC6hvAhUA0VoObKC2IXzXgVvRRupo4qHbcm8CgYAe
bbNuawh3rAxkFvUs9FPzW5E+xllG16Z//61PENKqonmk+zBiBdiIlS1F6ZqmTqkI
z5+qfSt1m3pb3j2W0NT7lEDFvy8Gr6Y2vohCHmL+T1u1Yy4PeqbgrFwcen7y7WoO
/KCV7Y9/ODQMMyuAzT3h5wJNweT7L5MUN8JYpZSi3QOBhAACgYBqaDuG2u6V91Qj
K2wEAElxaaRaNo/ewg/wWDmHYqoeHOR0HfuFCYgASE9f7ULqYtX1VURcgcjw9XN4
BDmPilXvfi04INPTnw4IxFJKDzzC0kVH7esVas982Po8v3megH32H9Rl87r7UG1c
ZEbkSkKVX6YKYg1PR3rfjXgdwVZv/zAJBgcqhkjOOAQDAzAAMC0CFFWeRe2fYW2i
6mMd26Wzbx87Y0DXAhUAoPCnF+5hGJw0jT9aL7QsgcfLi9Y=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAMuBl6rhZCJKMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMx
```

```
MjU3NTRaGA8yMjAzMDYwOTEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFuKydxZsordNH7bLwIluEGOkX7/CdLdpeqkDKEhQkFwzpRxaX4EAlkGh2/o7D
8qneC9cGQhqSG5WVVBrmZG7sfkFOM4m1AtY++kfv+MYto1VFgLk1xJbkpq1r4YeQ
Ul+ZsJYsZpyX/t+g8s7rW0OVcBsYx4L75bf34z38mwK8PQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBADD9C4pWL8RUvF1CJW8kExj35xmozlFlmrKs8Zpi8+Eg6q+W9dgd
xMdH95tgZtmVMDqlvVR+DK0iO1BNpqPjrqWkk2tTLivpS+sGzCE/jCl18Q28Rk71
/A3gLD7Rtbq5TKNvuFCHwYMjrTDHI6aBjIaAlDm4e2/j/OxVtHyZGTre
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANC3xtbPhQ2GMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMx
MjU3NTRaGA8yMjAzMDYwOTEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAt3aMy7Hsp4ySG3mlfi+pdTcZw6H6XNU1Y36fNdi4c+MzinQQbnqMPyt7
QLgU+XCWmcWsVo7GQF6n9N01Rh+UXXUZU4jcXlFocQPCWf9O+IIIIPXkd67kFMUV
HAxCELjfxHbC+I8e7dwOJhmdF4Bfi52Ty8zz0HdE8JDypPkTDlXuGvTgDyW7NP56
I/vlQaXLoYSbcQe5pv2a9gyBaaCM1QoeqwAHAeCNXb9Nuj9ZX3GHGJb3TuqAeKCD
5i9TscCB9XjY6Fx+zfSAobjBZwgLEtLOwJhbZnKmx4gJMaaNFipAjVT2FSS3+yev
eTYBoa1dvhk0ivQyQIPpHmihrmkWuwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBC
fdgyI8GjmCqiHAlh+L1bj0LdNql9zl7RXmOEzsuRdtdMumkxYXX88UtROy3fdi1i
VaEwHdAK8ThzRkesgHza/cXzqCMewaYxujSI6p6G7x99FFeGif1x0FJdj8AoeTL7
4h9bmS/6l4/NL7DJI9G7ovES/hoUA9v9TDhv+vauxXlgfrp0MPecprxBYlrc+DH2
adGCdKCp2lQ2YDKOD9TCEjYIli8XSoyevoWHUjfDYrCrCp8l4s/p7HOgYr8fJBAs
EuVy8211LVz1/X4EMBRNtNjXK9sklsxAOX14NDfBFSS0toxl3K6Tf9t/PviBl95d
hncyDAcFgDCK4w8LLlVW
-----END CERTIFICATE-----
```

Asia Pacific (Melbourne) — ap-southeast-4

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjWF7P2MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
```

```
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAPRXSsQP9E3dw8QXKlrgBgEVCprLHdK/bbrMas0XMu1EhOD
+q
+0PcTr8+iwbtoXlY5MCeatWIp1GrXQjVqsF8vQqxlEuRuYKbR3nq4mWwaeGlx9AG5EjQHRa3GQ44wWHOdof0M3NRI1MP
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjSh40SMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+qWTGAbGsPeMX4hBMjAJUKys2NIRcRZaLM/BCew2FIPVjNtlaj6Gwn9ipU4Mlz3zIwAMWi1AvGMSreppt
+wV6MRtfOjh0Dvj/veJe88aEZJMozNgkJFRS
+WFWsckQeL56tf6kY6QTlNo8V/0CsQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAF7vpPghH0FRo5gu49EArRNPrIvW1egM
wcgkqIwwuXYj+1rhlL+/
iMpQWjdVGEqIZSeXn5fLmdx50eegFCwND837r9e8XYTiQS143Sxt9+Yi6BZ7U7YD8kK9NBWoJxFqUeHdpRCs0O7COjT3
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAN4GTQ64zVs8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTMx
MzMzMDBaGA8yMjAxMTIxNzEzMzMwMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA2BYgeCr+Rk/jIAEDOHS7wJql62vc83QEwjuzk0qOFEReIZzlN1fBRNXK
g0T178Kd3gLYcE59wEFbTe/X5yOA1Lo95x1anSAo7R+Cisf9C2HQuJp+gVb+zx71
lniPF7gHziGpm0M8DdAU/IW+wkZwGbP4z7Hq9+bJ0P2ltvPJ5yxSgkFuDsI9VBHa
CLoprHsCHh2VdP8KcMgQQMmHe1NmBpyTk0ul/aLmQkCQEX6ZIRGOeq228fwlh/t+
Ho+jv87duihVKic6MrL32SlD+maX0LSDUydWdaOLLTGkh7oV7+bFuH6msrXUu+Ur
ZEP1r/MidCWMhfgrFzeTBzOHA97qxQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUcHMd1cHqzmsQ5hpUK3EMLhHdsi4wgY4GA1UdIwSBhjCBg4AUcHMd
1cHqzmsQ5hpUK3EMLhHdsi6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAN4GTQ64zVs8MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAI4PFyVN+7EGSObioiPnv0LL0f70SSzUZJ8p
XO90d4rWea7jIbgZ2AKb+ErynkU9xVg7XQQ5k6KDWgp/4jYFL2dqnt/YAY4PS0un
RSrYElawxLT0BcLn4rcSDC79vQe1xGC5//wDdV6b399COAHRAK6axWYy5w32u9PL
```

```
uw0cIp3Ch8JoNwcgTHKRRGzePmBeR4PNqhHTArG4/dJk6/aUO4OpX0WzI6L67CGY
6Nex3dau+gkLCK93dTEkrXtyXHu4wB0J9zd1w+iQ0SEa9eKc78/NjEsF/FZdGrWC
t57lIMOOXJhQ1kRgSwNeZdQWV1dRakvO6sfcvVYkfj1wAvZvvAw=
-----END CERTIFICATE-----
```


Asia Pacific (Mumbai) — ap-south-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUDLA+x6tTAP3LRTr0z6nOxfsozdMwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE0MTMwMVoXDTI5MDQyODE0MTMwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
```

```
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUDLA+x6tT
AP3LRTr0z6nOxfsozdMwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAZ7rYKoAwwiiH1M5GJbrT/BEk3OO2VrEPw8ZxgpqQ/EKlzMlOs/0Cyrmp7
UYyUgYFQe5nq37Z94rOUSeMgv/WRxaMwrLlLqD78cuF9DSkXaZIX/kECtVaUnjk8
BZx0QhoIHOpQocJUSlm/dLeMuE0+0A3HNR6JVktGsUdv9ulmKw==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPRYyD8TtmC0MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjAzMDcx
MDQ1MDFaGA8yMTk1MDgxMTEwNDUwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA0LSS5I/eCT2PM0+qusorBx67QL26BIWQHd/yF6ARtHBb/1DdFLRqE5Dj
07Xw7eENC+T79mOxOAbeWg91KaODOzw6i9I/2/HpK0+NDEdD6sPKDA1d45jRra+v
CqAjI+nV9Vw91wv7HjMk3RcjWGziM8/hw+3YNIutt7aQzZRwIWlBpcqx3/AFd8Eu
2UsRMSHgkGUW6UzUF+h/U8218XfrauKNGmNKDYUhtmyBrHT+k6J0hQ4pN7fe6h+Z
w9RVHm24BGhlLxLHLmsOIxvbrF277uX9Dxu1HfKfu5D2kimTY7xSZDNLR2dt+kNY
/+iWdIeEFpPT0PLSILt52wP6stF+3QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBI
E6w+WWC2gCfoJO6c9HMyGLMFEpqZmz1n5IcQt1h9iyO7Vkm1wkJiZsMhXpk73zXf
TPxuXEacTX3SOEa07OIMCFwkusO5f6leOyFTynHCzBgZ3U0UkRVZA3WcpbNB6Dwy
h7ysVlqyT9WZd7EOYm5j5oue2G2xdei+6etgn5UjyWm6liZGrcOF6WPTdmzqa6WG
ApEqanpkQd/HM+hUYex/ZS6zEhd4CCDLgYkIjlrFbFb3pJ1OVLztIfSN5J4Oolpu
JVCfIq5u1NkpzL7ys/Ub8eYipbzI6P+yxXiUSuF0v9b98ymczMYjrSQXIf1e8In3
OP2CclCHoZ8XDQcvvKAh
-----END CERTIFICATE-----
```


Asia Pacific (New Zealand) — ap-southeast-6

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQCcP+D4eLFOajAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEyMDMxMjEzMDZaGA8y
MDUwMTIwNDEyMTMwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBANBMmM/LbAX13IeB
```

```
3yr2pMMpiTW1b1go/axNLfcxEmj0fThKk8y0FAT1frN/UPdI9qRInsyBdEZQLiIH
4WHCXk8IidQLh4DED3G9DvSFFC4Gb8ax+UIQY11wfHPWO4+wA9q8liYTg/FxdJMM
tZS9tZAZxjH6+8yfV8Bt0wUd2Y+lAhUAwg0CxCmZPinHtKbFnpMQfKnzjCMCgYAf
i+D7nGVFOWJ8EYOpRYbG0pHkrpwNy+h01csn6IAwQbIfeOG+SF4N3LeuOAC9vXGY
VGh9XNxuasvqkRAzgRmJ/EYM5Xa/YN62EhddRuN8wCOg66MeczUDh3RTb+Tu6HK0
UR6oTSOQ6rxA5aA1qTc7TqI0+pRZv+QYIZK/ozsZzgOBhAACgYAbVm+rOX8J38cG
LxKjUXVM0KB69SksGLEqjO4ps80EcZ/0W3g1hGLnSX/RjYfAS60hbKK1GX9PpcaO
4mDzCTm3dZ6shbae5t8bBU+mUi8BmaK+Pdu3z5J77y4zv22/yXte8ulZ5J0zKP9K
QP3p3pa7XI9NaXpKGT3yZsJ0BWY2xTAJBgcqhkjOOAQDAzAAMC0CFBs4KQmrxHjQ
OHh0r8JqB8UiwlutAhUAwgxid7wT6FNm8Zwq9hioQ2v2K0A=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAMwLuv2/tk9eMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEyMDMx
MjEzMDZaGA8yMjA0MDUwOTEyMTMwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQC9AsBNOUfT8QpguHqMa9KIXP/YgZOmfRAgEmnLawXTC+40bXUq/fwkx+cebv3x
CvUx6d07/iDlD0dm2Hf2HhPFi3OaCI7vc8c3lqVBvcn6yzmwv35aSEvtVm2NHC3j
4/q6vN2tR2d4a1Mf1QhmOVOJ7XiTEbCy2Yv+pmDbk29V6QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAHatyQIy612kio8neI4r5VBg8pgKwW+mFGpAzOQ2NOyLCNoOeD1D
lPRaSipBN7WW6jpIozbz7JG7Kqc3N+piPGGzfjH9gZjqUqfOGrNHB5nsxvMHI2iL
e+DwsOcV7qnEVRgYbmLYOVKJv25wCsM/Tcn/6f6scEX2x+H0hAh7rwiP
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAI/xUfWwSQOtMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEyMDMx
MjEzMDZaGA8yMjA0MDUwOTEyMTMwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArbid9Tnunu0pWCQUqZCkGeiF8hWixi+LBS1AOew2UsT9keAt2rXXGlU/
wSMriKHsvcrfKtuVKj5chArsOyFvagszH5lIKhI+GfWVQEd2aZiOvAJ+iRUQq1+R
gOr2gyjWlf3znZXTcyhHcJoXDhbn5e+kEa3pwS4yKNRpM3P61V4M2702xis08hjP
qMDxwXMa1Fiy4b7nAyNQifgrwb3bCA5k5prmkl/iajaWa1zMMFgLI9vA6UwUSbTK
H9iOuL/xNfMX+Xzq4CCaQYiJ8zdi56kqnB/lrA05NSjOPLskg6sUGkyiDxGK4gzC
```

```
DFrO6oygcdlBaxnyAECGdJcOM6TOmwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBg
8yVPNpctrvSvTazf/Ij8qC6+q6df1SptdYKQkOYWeUVZR7+3uTZ9Yiyq+Bh8Hbyf
NcR0eR6UqhgiB3p5slYNhTvLkpuMuqTpVJFvQtzZycnmKXAQ35G5zCYV0+pkF59L
ch26SukWk0+VP3kIM9VmaIZQKJEBlBLcnbDZyXlHUDnZAYPcxxKQ8uFPkW0GOFMr
ekFCGOUmFSItFXOHy5hD2xmctKNk3l12m6cKydYvONtk/3DovKNnw3c5k40WHQW/
Zdh/LWd4/Jrkan+kQnqm4rkrXF+x3aF0qrKTtHMICvpkZ8g7/588hBs4mZnCmFLa
q/g9WqIcd9SMmBkT6KZ8
-----END CERTIFICATE-----
```


Asia Pacific (Osaka) — ap-northeast-3

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUHTRhxHhBZFOGvTFKxHoy9+f5Hl8wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2NTQwN1oXDTI5MDQyODE2NTQwN1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
```

```
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUHTRhxHhB
ZFOGvTFKxHoy9+f5Hl8wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAUZx7DcYbhWNTD4BNGhr5beruT2OUoGHH9J73UKxwdqeb9bH1LIWhIZO0X
/1mjn3bWBgCwfoS8gjZwsVB6fZbNBRy8urdBZJ87xF/4JPBjt7S9oGx/zthDUYrC
yK0Y0v4G0PgiS81CvYLg09LpmYhLSJbXENlkC04v5yxdKxZxyg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMn1yPk22ditMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA3MTkx
MTEyNThaGA8yMTk2MTIyMjExMTI1OFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArznEYef8IjhrJoazI0QGZkmlmHm/4rEbyQbMNifxjsDE8YWtHNwaM91z
zmyK6Sk/tKlWxcnl3g31iq305ziyFPEewe5Qbwf1iz2cMsvfNBcTh/E6u+mBPH3J
gvGanqUJt6c4IbipdEouIjjnynyVWd4D6erLl/ENijeR1OxVpaqSW5SBK7jms49E
pw3wtbchEl3qsE42Ip4IYmWxqjgaxB7vps91n4kfyzAjUmklcqTfMfPCkzmJCRgp
Vh1C79vRQhmriVKD6BXwfZ8tG3a7mijeDn7kTsQzgO07Z2SAE63PIO48JK8HcObH
tXORUQ/XF1jzi/SIaUJZT7kq3kWl8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBj
ThtO9dLvU2QmKuXAhxXjsIdlQgGG3ZGh/Vke4If1ymgLx95v2Vj9Moxk+gJuUSRL
BzFte3TT6b3jPolbECgmAorjj8NxjC17N8QAAI1d0S0gI8kqkG7V8iRyPIFekv+M
pcai1+cIv5IV5qAz8QOMGYfGdYkcoBjsgiyvMJu/2N2UbZJNGWvcEGkdjGJUYYOO
NaspCAFm+6HA/K7BD9zXB1IKsprLgqhiIUgEaW3UFEbThJT+z8UfHG9fQjzzfN/J
nT6vuY/0RRu1xAZPyh2gr5okN/s6rnmh2zmBHU1n8cbCc64MVfXe2g3EZ9Glq/9n
izPrI09hMypJDP04ugQc
-----END CERTIFICATE-----
```

Asia Pacific (Seoul) — ap-northeast-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
```

```
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUbBSn2UIO6vYk4iNWV0RPxJJtHlgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTEzMzg0NloXDTI5MDQyODEzMzg0NlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUbBSn2UIO
6vYk4iNWV0RPxJJtHlgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAmjTjalG8MGLqWTC2uYqEM8nzI3px1eo0ArvFRsyqQ3fgmWcQpxExqUqRy
l3+2134Kv8dFab04Gut5wlfRtc2OwPKKicmv/IXGN+9bKFnQFjTqif08NIzrDZch
aFT/uvxrIiM+oN2YsHq66GUhO2+xVRXDXVxM/VObFgPERbJpyA==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANuCgCcHtOJhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTQx
NTU3NDRaGA8yMTk1MDIxNzE1NTc0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA66iNv6pJPmGM20W8HbVYJSlKcAg2vUGx8xeAbzZIQdpGfkabVcUHGB6m
Gy59VXDMDlrJckDDk6dxUOhmcX9z785TtVZURq1fua9QosdbTzX4kAgHGdp4xQEs
mO6QZqg5qKjBP6xr3+PshfQ1rB8Bmwg0gXEm22CC7o77+7N7Mu2sWzWbiUR7vil4
9FjWS8XmMNwFTlShp4l1TDTevDWW/uYmC30RThM9S4QPvTZ0rAS18hHVam8BCTxa
LHaVCH/Yy52rsz0hM/FlghnSnK105ZKj+b+KIp3adBL8OMCjgc/Pxi0+j3HQLdYE
32+FaXWU84D2iP2gDT28evnstzuYTQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC1
mA4q+12pxy7By6g3nBk1s34PmWikNRJBwOqhF8ucGRv8aiNhRRye9lokcXomwo8r
KHbbqvtK85l0xUZp/Cx4sm4aTgcMvfJP29jGLclDzeqADIvkWEJ4+xncxSYVlS9x
+78TvF/+8h9U2LnSl64PXaKdxHy2IsHIVRN4GtoaP2Xhpa1S0M328Jykq/571nfN
1WRD1c/fQf1edgzRjhQ4whcAhv7WRRF+qTbfQJ/vDxy8lkiOsvU9XzUaZ0fZSfXX
wXxZamQbONvFcxVHY/0PSiM8nQoUmkkBQuKleDwRWvkoJKYKyr3jvXK7HIWtMrO4
jmXe0aMy3thyK6g5sJVg
-----END CERTIFICATE-----
```

Asia Pacific (Singapore) — ap-southeast-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUSqP6ih+++5KF07NXngrWf26mhSUwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE0MzAxNFoXDTI5MDQyODE0MzAxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUSqP6ih++
+5KF07NXngrWf26mhSUwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAw13BxW11U/JL58j//Fmk7qqtrZTqXmaz1qm2WlIpJpW750MOcP4ux1uPy
eM0RdVZ4jHSMv5gtLAv/PjExBfw9n6vNCk+5GZG4Xec5DoapBZHXmfMo93sjxBFP
4x9rWn0GuwAVO9ukjYPevq2Rerilrq5VvppHtbATVNY2qecXDA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAJVMGw5SHkcvMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
ODU3MTlaGA8yMTk1MDQwMzA4NTcxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAlaSSLfBl7OgmikjLReHuNhVuvM20dCsVzptUyRbut+KmIEEc24wd/xVy
2RMIrydGedkW4tUjkUyOyfET5OAyT43jTzDPHZTkRSVkYjBdcYbe9o/0Q4P7IVS3
XlvwrUu0qo9nSID0mxMnOoF1l8KAqnn10tQ0W+lNSTkasW7QVzcb+3okPEVhPAOq
MnlY3vkMQGI8zX4iOKbEcSVIzf6wuIffXMGHVC/JjwihJ2USQ8fq6oy686g54P4w
ROg415kLYcodjqThmGJPNUpAZ7MOc5Z4pymFuCHgNAZNvjhZDA842Ojecqm62zcm
Tzh/pNMNeGCRYq2EQX0aQtYOIj7bOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU6SSB+3qALorPMVNjToM1Bj3oJMswgY4GA1UdIwSBhjCBg4AU6SSB
+3qALorPMVNjToM1Bj3oJMuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAJVMGw5SHkcvMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAF/0dWqkIEZKg5rca8o0P0VS+tolJJE/FRZO
atHOeaQbWzyac6NEwjYeeV2kY63skJ+QPuYbSuIBLM8p/uTRIvYM4LZYImLGUvoO
```

```
IdtJ8mAzq8CZ3ipdMs1hRqF5GRp8lg4w2QpX+PfhnW47iIOBiqSAUkIr3Y3BDaDn
EjeXF6qS4iPIvBaQQ0cvdddNh/pE33/ceghbkZNTYkrwMyBkQlRTTVKXFN7pCRUV
+L9FuQ9y8mP0BYZa5e1sdkwebydU+eqVzsil98ntkhpjvRkaJ5+Drs8TjGaJWlRw
5WuOr8unKj7YxdL1bv7//RtVYVVi296ldoRUYv4SCvJF11z0OdQ=
-----END CERTIFICATE-----
```

Asia Pacific (Sydney) — ap-southeast-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFxWyAdk4oiXIOC9PxcgjYYh71mwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1MjE0M1oXDTI5MDQyODE1MjE0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
```

```
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFxWyAdk4
oiXIOC9PxcgjYYh71mwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQByjeQe6lr7fiIhoGdjBXYzDfkX0lGGvMIhRh57G1bbceQfaYdZd7Ptc0jl
bpycKGaTvhUdkpMOiV2Hi9dOOYawkdhyJDstmDNKu6P9+b6Kak8He5z3NU1tUR2Y
uTwcz7Ye8Nldx//ws3raErfTI7D6s9m63OX8cAJ/f8bNgikwpw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL2bOgb+dq9rMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAwNTdaGA8yMTk1MDQwMzA5MDA1N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAmRcyLWraysQS8yDC1b5Abs3TUaJabjqWu7d5gHik5Icd6dKl8EYpQSeS
vz6pLhkgO4xBbCRGlgE8LS/OijcZ5HwdrxBiKbicR1YvIPaIyEQQvF5sX6UWkGYw
Ma5IRGj4YbRmJkBybw+AAV9Icb5LJNOMWPi34OWM+2tMh+8L234v/JA6ogpdPuDr
sM6YFHMZ0NWo58MQ0FnEj2D7H58Ti//vFPl0TaaPWaAIRF85zBiJtKcFJ6vPidqK
f2/SDuAvZmyHC8ZBHg1moX9bR5FsU3QazfbW+c+JzAQWHj2AaQrGSCITxCMlS9sJ
l51DeoZBjnx8cnRe+HCaC4YoRBiqIQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU/wHIo+r5U31VIsPoWoRVsNXGxowwgY4GA1UdIwSBhjCBg4AU/wHI
o+r5U31VIsPoWoRVsNXGxoyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL2bOgb+dq9rMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACobLvj8IxlQyORTz/9q7/VJL509/p4HAeve
92riHp6+Moi0/dSEYPeFTgdWB9W3YCNc34Ss9TJq2D7t/zLGGlbI4wYXU6VJjL0S
hCjWeIyBXUZOZKFCb0DSJeUElsTRSXSFuVrZ9EAwjLvHni3BaC9Ve34iP71ifr75
8Tpk6PEj0+JwiijFH8E4GhcV5chB0/iooU6ioQqJrMwFYnwo1cVZJD5v6D0mu9bS
TMIJLJKv4QQQqPsNdjiB7G9bfkB6trP8fUVYLHLsVlIy5lGx+tgwFEYkG1N8IOO/
2LCawwaWm8FYAFd3IZl04RImNs/IMG7VmH1bf4swHOBHgCN1uYo=
-----END CERTIFICATE-----
```

Asia Pacific (Taipei) — ap-east-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQDrb+HArMxssTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
```

```
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQxNDU2MDZaGA8y
MDUwMDUyNTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBAPFb5R5TBUs2N0YW
aJcOjxr4UCDrpVT3x6EeqoGFNAc1yiZw9iv++ff6wsXm0VG5XV1QkWMQcjPfK/2I
TvXh9XEOhsG7vaBHoWkb2gmIxE6C2u8TfoZu0ZmiMceFK7aYz5w/q2J9CmRs38dT
I9qgTq0LYX+VkiNYOQnn7sQXQ8CjAhUAp2Lue9SfYTmubbpH+3j91Sivc+cCgYAi
DiyGU5R5VgiNV3mhuAyiHCft3LEmTIxos/hx/rHzgm+vCMH55tYejIrv9n7YGrIZ
MejOt8bGYxqSf0Ir/aoj3ZfHihxUzwyfyzwH7D8kAodXJNsPUTKi0O0pGl/e+nVD
JhTIFV3AwC8Tt0r2hljbjhOiyhdYy5hxWHa/WokLAQOBhAACgYAkGGHXbFr/JXol
itpS1qfiETWSRkXGHrdSmUkRumgti2y7c57TgXPwCyNlP1sTr1duy/kYDpwSwyqi
qEYfko0b/xWpDEQQPsWBgolpEYP7RIT85IxrTCrC9ohz85n7v6m9oF6BiBQGtZdk
GBN7iqemg85UaF6mXfHvl3tUYjsB2jAJBgcqhkjOOAQDAy8AMCwCFED9Ybn6fEVF
wLNjDvULmki5bPBWAhR64pxsHFRHz0ZQi+SYMWdIpEs5RQ==
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAJp99iy2CTXhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQx
NDU2MDZaGA8yMjAzMTAyOTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDVdclgpcIIsyGG51ag8srK1UMfZ+Ioh9E4PLP+cIWQcdh31fv2kV/whKaqfmif
NefBIuHrTPl6L+mJQbsfeA+TiEx1Jamg4lwQfhgQKQCmjZVMaHtK+AY57TVN6FBf
HXxT2MXzktsKxAQRGeSTFQYrHRzSYvu+GkA+37lqyWRfcwIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBABM+WL9MSb7z3bNOTMkM7ufPucvj+LQ+zEAJiuG0rcB8GI2DhGB5
8ohDPUGlq1cP7vYmDkWG/FrLqx7qgIzkbE3zfb62JpdOuX73oHgCmikMfq8ifoS1
CUxMQ36PDZBFNPzGZxv5kEORqhmGBkYCoIUCQ43jh3aALR/A/rfzfij8
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJt5hgMuWbpMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQx
NDU2MDZaGA8yMjAzMTAyOTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
```

```
CgKCAQEA91FXs9Wk9kMI3lkNhjC69C6RTBpSvdLWSohPDlJKXFoaGYIDtyW1ViwO
FORAKz0koarVBx1D4fNKCxf+mhYAf//4Pxt31GzjehHNs2YVGimwhq/SNr+3AuIO
PibA+qScXamaiknE1Pxb4jisbBu2PMAG9OxU2dESNwiqBMvdJvg30YNTUITZ6r0e
MPjR1Y/EFVx/SLHW6wdvwqhNi9hz8HHdThzjCbCQPvmoLBJIKXIfosKqvauSezt1
KJSoS2jMx1OjJB2bBlQTZC8vF0WFNPoiH4jXyJ6+wMUMNSSd5b3uL5stVTMzuIz4
39Hd5YW41rmAy+6jmpGp2eXn6Td0rQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAs
gcgv+Xr67Cj08jKdAUT4GuCoGxveOFAjSDjs5y0cD+kXIT/Yj/+HKFOx5cwThT8F
TxNQ/oVuo3bGeuFEX799H0rT9Jupy0WiE1Vx81mk2M/1EDtns437edM4ofiWR8tR
Q1VNiRgz6Crr0nFgNoQa9HmR6/M+bImxevHJkWB3skvnxtA+lpKGwJs8jo6fA7AW
jr00opK0r731iBd1z/5hjIjkHn5NizAdiJEh3lVY/A3WS82aygW8LvYnKZeaBfZJ
J4NgxeflxocLFKBJD+hyf5u1hhvl7Mn/aGr5VI8/0F2oUqYo+xsHLT3keNghywO4
NNcyLcD9u16HVXhRhGvo
-----END CERTIFICATE-----
```


Asia Pacific (Thailand) — ap-southeast-7

DSA


```
-----BEGIN CERTIFICATE-----
MIIC8TCCAq8CCQCOEEMWiJIJpTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIxNTI4NTZaGA8y
MDUwMDQxMzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASsGByqGSM44BAEwggEeAoGBAJWKfrIlz9+/U4wm
NDJPX00avUTdksBKX5RwHkY4o0///2G2HWwiBbxw8F1U+RzCXr80gSoe4+/SCSg3
uDuhJyzN5r4jr7c590CAgsiGdWERs7l4gLQCP504feU4TEjJoq9A8MfRZeqMj7Ug
fT5InTUW7S1/r98ddG/rBHMsJwEbAhUAneIckZ1YeyGcCeL32lujMhj+g68CgYA+
RsyTXCy3Tug5aHun51IfcG2d+pn5K/tv/N3WUR18RplVctrLhwIOUoAsDWPWtxNV
sODetezAyo759CK43JAmYGzXKbRFUHm3n46jP88tSdhuSeJhc/D4l5/0+2L7ndXp
L3+W4N05NdiKSGI8e8t452wTZv/RODivPZ3RtuCDywOBhQACgYEAjVS1Huwzdn1J
+kpd2Rcbe1BAGkmvk5sUu0KhyttqIB1kxTxeWZgsdO8REZSC5gJOwkcGFvXnb3DY
+Ms45r0e0s0rx2FFYjqMLwyRpK9wUjJfSXeJMa9iLQEXuyzBz6zPgfemXbS3zNq/
eoJ9ztIwjB9DMoKL+E1vSLsTGhehqRowCQYHKoZIzjgEAwMxADAuAhUAmljDM3c9
hf0j4XbMjjpnzrx1xhkCFQCASR9pgFNgLK8y6Kojj+P1KJkrSQ==
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAIuIHAhL0xWcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIx
NTI4NTZaGA8yMjAzMDkxNzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCajgAeOauwvqGDLrHvxujnZ1BnkMzwjrycMUTkj8jqNtWoDQWUJVNPZJILosEU
VwK2I3oNkEsx/ryl9XfXcNNceoYfVEPzkTzozrZyuOG66FWtUU1LKeJ7h9/rX0Zd
9lZEokrdr6dLPt9FsHWaK5ExlUnWBjNltcQLkkKqoeYaFwIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAE4G5G+FvKTsX3T7BEcId7f5LSCc2J9gZRDiWn2oTr40CrBMOzJT
KsWr9W89YXW3gaGWltzc0WCwYQbJZgAkuEAZItJjbhdnns87ZbsFO+NZhc6gDtjA
WC3dPlSB9b6rfVoVW9O6Xwa7iNXZo8ddYVJ/ZOIv/totUz9qJt4DmmKk
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANaQIrYcijxaMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIx
NTI4NTZaGA8yMjAzMDkxNzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAxtMGK4r6WerVtMfpRRCY3DMP9Q/s4jnRIqolPaxeT99BAFp03HXyOrTz
WjggEHArRGGeDowFkL7Vj+QNY7ran2lnYVlAQ7Ow6OLl6WttT6lMsgsLUIcsED6G
eJ1Ko+ovT1qvmuush1U5R2pHkcCJ1IT2s6uQff85l066K7dCFpaoA1QHUk9zACHj
i0DtWuTAWBXirAZSGlmtP6uj1Aw6y5kACDTjIvZJeW/kmswfApTKaJ56eNkJsUBU
NpKwt3um1BMZduy0cjUv0Sc3dIbLNyfOdyPn8tX5u5ck0EPdtBB5WjEh71IXdZJD
oaBREV7AHg5ERPQsm0BqSvMQt09KiwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
qJyZbv/GUQnm8dEyVYgilQr3Yu35n8sfTwwZ1OJzIVce7NBPWePMfYii/tvGoKNp
KwFa/Vugbn1EwqpTEKqkPE1dOZvnUa97XxhYkoW5U7sgdKCANi1KWjywn7MiJ3Eg
j4gdqYK8wxvHi21ppqr572UO77ZuA8YMBoBT/CyQzWYUSmbqwKnnzaQBAZeO2iAk
VHuNU+9UsntNB676gRl9ag3Wfxq3yx5Ee1CeQf+US3HJn/pKk1H8dExXmBHvHwO6
GKUVNPN1rxfjTiaSt8wuO80uElAnyHIM3VOR8rJO7PKsobyEeJV0WI1hURO+wxpL
h3IsW7iBrFVvhX5xx7ZU
-----END CERTIFICATE-----
```


Asia Pacific (Tokyo) — ap-northeast-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
```

```
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULgwDh7TiDrPPBJwscqDwiBHkEFQwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTEyMjMxMFoXDTI5MDQyODEyMjMxMFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IULgwDh7Ti
DrPPBJwscqDwiBHkEFQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBtjAglBde1t4F9EHCZOj4qnY6Gigy07Ou54i+lR77MhbpzE8V28Li9l+YT
QMIn6SzJqU3/fIycIro1OVY1lHmaKYgPGSEZxBenSBHfzwDLRmC9oRp4QMe0BjOC
gepj1lUoiN7OA6PtA+ycNlsP0oJvdBjhvayLiuM3tUfLTrgHbw==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL9KIB7Fgvg/MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAwMjVaGA8yMTk1MDExNzA5MDAyNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAz0djWUcmRW85C5CiCKPFiTIvj6y2OuopFxNE5d3Wtab10bm06vnXVKXu
tz3AndG+Dg0zIL0gMlU+QmrSR0PH2PfV9iejfLak9iwdm1WbwRrCEAj5VxPe0Q+I
KeznOtxzqQ5Wo5NLE9bA61sziUAFNVsTFUzphEwRohcekYyd3bBC4v/RuAjCXHVx
40z6AIksnAOGN2VABMlTeMNvPItKOCIeRLlllSqXX1gbtL1gxSW40JWdF3WPB68E
e+/1U3F7OEr7XqmNODOL6yh92QqZ8fHjG+afOL9Y2Hc4g+P1nk4w4iohQOPABqzb
MPjK7B2Rze0f9OEc51GBQu13kxkWWQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU5DS5IFdU/QwYbikgtWvkU3fDwRgwgY4GA1UdIwSBhjCBg4AU5DS5
IFdU/QwYbikgtWvkU3fDwRihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL9KIB7Fgvg/MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAG/N7ua8IE9IMyno0n5T57erBvLTOQ79fIJN
Mf+mKRM7qRRsdg/eumFft0rLOKo54pJ+Kim2cngCWNhkzctRHBV567AJNt4+ZDG5
hDgV0IxWO1+eaLE4qzqWP/9VrO+p3reuumgFZLVpvVpwXBBeBFUf2drUR14aWfI2
L/6VGINXYs7uP8v/2VBS7r6XZRnPBUy/R4hv5efYXnjwA9gq8+a3stC2ur8m5ySl
faKSwE4H32OyAyaZWH4gpwUdbUlYgPHtm/ohRtiWPrN7KEG5Wq/REzMIjZCnxOfS
6KR6PNjlhxBsImQhmBvz6j5PLQxOxBZIpDoiK278e/1Wqm9LrBc=
-----END CERTIFICATE-----
```

Canada (Central) — ca-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
```

```
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUIrLgixJJB5C4G8z6pZ5rB0JU2aQwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1MzU0M1oXDTI5MDQyODE1MzU0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUIrLgixJJ
B5C4G8z6pZ5rB0JU2aQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBHiQJmzyFAaSYs8SpiRijIDZW2RIo7qBKb/pI3rqK6yOWDlPuMr6yNI81D
IrKGGftg4Z+2KETYU4x76HSf0s//vfH3QA57qFaAwddhKYy4BhteFQl/Wex3xTlX
LiwI07kwJvJy3mS6UfQ4HcvZy219tY+0iyOWrz/jVxwq7TOkCw==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJNKhJhaJOuMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA3Mjkx
MTM3MTdaGA8yMTk2MDEwMjExMzcxN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAhDUh6j1ACSt057nSxAcwMaGr8Ez87VA2RW2HyY8l9XoHndnxmP50Cqld
+26AJtltlqHpI1YdtnZ6OrVgVhXcVtbvte0lZ3ldEzC3PMvmISBhHs6A3SWHA9ln
InHbToLX/SWqBHLOX78HkPRaG2k0COHpRy+fG9gvz8HCiQaXCbWNFDHZev9OToNI
xhXBVzIa3AgUnGMalCYZuh5AfVRCEeALG60kxMMC8IoAN7+HG+pMdqAhJxGUcMO0
LBvmTGGeWhi04MUZWfOkwn9JjQZuyLg6B1OD4Y6s0LB2P1MovmSJKGY4JcF8Qu3z
xxUbl7Bh9pvzFR5gJN1pjM2n3gJEPwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAJ
UNKM+gIIHNk0G0tzv6vZBT+o/vt+tIp8lEoZwaPQh1121iw/I7ZvhMLAigx7eyvf
IxUt9/nf8pxWaeGzi98RbSmbap+uxYRynqe1p5rifTamOsguuPrhVpl12OgRWLcT
rjg/K60UMXRsmg2w/cxV45pUBcyVb5h6Op5uEVAVq+CVns13ExiQL6kk3guG4+Yq
```

```
LvP1p4DZfeC33a2Rfre2IHLsJH5D4SdWcYqBsfTpf3FQThH0l0KoacGrXtsedsxs
9aRd7OzuSEJ+mBxmzxSjSwM84Ooh78DjkdpQgv967p3d+8NiSLt3/n7MgnUy6WwB
KtDujDnB+ttEHwRRngX7
-----END CERTIFICATE-----
```


Canada West (Calgary) — ca-west-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAYPouptUMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAMITzTJUa6cBsIfdHN69zW/
aHjUB4r1ZfKb1FMhIp9EZtEf5n+06oXjUG2+dKRSlFQeEK333ehNZsPd6uqey6TYKtHpFb5XRLS8BpqB
+7gnbAd0CBZM5o4NWesSQ1GLnTdQcGZkYG/
QESkbadoCXQTifCujJE682hTDLIVt1d4ewwCQYHKoZIzjgEAwMvADAsAhRJc4gRS/HWTkCR2MESaQEe/
jOMNQIUNoTwLvuPrmGPupPlGiHe0veZiO8=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAYPou9weMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
v4XBVH13ZCMgqlRHMqV8AWI5iO6gFn2A9sN3AZXTMqwtZeiDdebq3k6Wt7ieYvpXTg0qvgsjQIovRZWaBDBJy9x8C2hW
+w9lMQjFHkJ7Jy/
PHCJ69EzebQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAGe9Snkz1A6rHBH6/5kDtYvtPYwhx2sXNxztbhkXErFk40Nw5l4
gvDVtWG7qyb6fAqgoisyAbk8K9LzxSim2S1nmT9vD84B/t/VvwQBylc
+ej8kRxMH7fquZLp7IXfmtBzyUqu6Dpbne+chG2
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALyTn5IHrIZjMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMzEyMDcx
NTM3MDFaGA8yMjAzMDUxMzE1MzcwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1GP5os424BJmGPCKOSgOclP7lzUiB85du03M4hfjzSOszsBpmBGFDLz1
owYHtIxlq3+Vi1Lt5Qlx3id/ov1QyaBPFWXVeklHVXy9vieCcI3TdjGjTl1W/8MM
m3X26QPcsnHM/Kk2wJ7s186MrqmdSsp3SCPpxv4vEG2Q9yR2bXY41hpc2rWlW8qU
D0JGXluvmmAdFnto20l1XWZ6xFen1h60DRugek/ufCbN+ljKy0xLqPoavH0Ybjsb
UpsAsBs7phaoN+X/5hIERfbp5LfVnqq54pNG5KNu4KynfW9+kA/WS4cJ6FTTN5t+
y0P1HvcL+BL2RuDy6T2bB21xw5WqtQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQURTvu/Dd4zDnmS5G5CfVLnmUBNOswgY4GA1UdIwSBhjCBg4AURTvu
/Dd4zDnmS5G5CfVLnmUBNOuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALyTn5IHrIZjMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFt523A3Aug6/F8xxyITgA8gkU0btFhlXNSP
U4U2OQ9n0tWI9WqnKNWH3KBxwY5EPitU6b3LM4xc9lDWpz7h2Pto+WhxP9LVKe6f
r8r7teTLCVZ7cfYZHzHg+flZjVpAgzE5BVfrRlj3QKpv0hYT3J1wMtI++Vorq5Nf
aPjzedehJLhmZVALwnfqfLrgv6/gmraP9Vmoa8U4D6AljNiQGYaLwyoPoRm3bUs2
v1Mh9GkEQ1b9+1pFXcqgzJJTGRuiPCyPbECI79FAnx5JM/CkGJV8HlOmjIW1qkK1
Y2qT7wzErrKLJyB53Pw15BdIM1onbDAQreZb0yZQLdoEl/tx7Uk=
-----END CERTIFICATE-----
```


China (Beijing) — cn-north-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
```

```
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAOtrM5XLDSjCMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQx
MDAxNDJaGA8yMTk1MDExNzEwMDE0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvVBz+WQNdPiM9S+aUULOQEriTmNDUrjLWLr7SfaOJScBzis5D5ju0jh1
+qJdkbuGKtFX5OTWTm8pWhInX+hIOoS3exC4BaANoa1A3o6quoG+Rsv72qQf8LLH
sgEi6+LMlCN9TwnRKOToEabmDKorss4zFl7VSsbQJwcBSfOcIwbdRRaW9Ab6uJHu
79L+mBR3Ea+G7vSDrVIA8goAPkae6jY9WGw9KxsOrcvNdQoEkqRVtHo4bs9fMRHU
Etphj2gh4ObXlFN92VtvzD6QBs3CcoFWgyWGvzg+dNG5VCbsiiuRdmii3kcijZ3H
Nv1wCcZoEAqH72etVhsuvNRC/xAP8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA8
ezx5LRjzUU9EYWYhyYIEShFlP1qDHs7F4L46/5lc4pL8FPoQm5CZuAF31DJhYi/b
fcV7i3n++/ymQbCLC6kAg8DUB7NrcROll5ag8d/JXGzcTCnlDXLXx1905fPNa+jI
0q5quTmdmiSi0taeaKZmyUdhrB+a7ohWdSdlokEIOtbH1P+g5yll3bI2leYE6Tm8
```

```
LKbyfK/532xJPqO9abx4Ddn89ZEC6vvWVNDgTsxERg992Wi+/xoSw3XxkgAryIv1
zQ4dQ6irFmXwCWJqc6kHg/M5W+z60S/94+wGTXmp+19U6Rkq5jVMLh16XJXrXwHe
4KcgIS/aQGVgjM6wivVA
-----END CERTIFICATE-----
```

China (Ningxia) — cn-northwest-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
```

```
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPu4ssY3BlzcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEyMDMy
MTI5MzJaGA8yMTk1MDUwODIxMjkzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsOiGi4A6+YTLzCdIyP8b8SCT2M/6PGKwzKJ5XbSBoL3gsnSWiFYqPg9c
uJPNbiy9wSA9vlyfWMd90qvTfiNrT6vewP813QdJ3EENZOx4ERcf/Wd22tV72kxD
yw1Q3I1OMH4bOItGQAxU5OtXCjBZEEUZooOkU8RoUQOU2Pql4NTiUpzWacNutAn5
HHS7MDc4lUlsJqbN+5QW6fFrcNG/0Mrib3JbwdFUNhrQ5j+Yq5h78HarnUivnX/3
Ap+oPbentv1qd7wvPJu556LZuhfqI0TohiIT1Ah+yUdN5osoaMxTHKKtf/CsSJ1F
w3qXqFJQA0VWsqjFyHXFI32I/GOupwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCn
Um00QHvUsJSN6KATbghowLynHn3wZSQsuS8E0COpcFJFxP2SV0NYkERbXu0n/Vhi
yq5F8v4/bRA2/xpedLWmvFs7QWlomuXhSnYFkd33Z5gnXPb9vRkLwiMSw4uXls35
qQraczUJ9EXDhrv7VmngIk9H3YsxYrlDGEqh/oz4Ze4ULOgnfkauanHikk+BUEsg
/jsTD+7e+niEzJPihHdsvKFDlud5pakEzyxovHwNJ1GS2I//yxrJFIL91mehjqEk
RLPdNse7N6UvSnuXcOokwu6l6kfzigGkJBxkcq4gre3szZFdCQcUioj7Z4xtuTL8
YMqfiDtN5cbD8R8ojw9Y
-----END CERTIFICATE-----
```

Europe (Frankfurt) — eu-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
```

```
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFD5GsmkxRuecttwsCG763m3u63UwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1NTUyOVoXDTI5MDQyODE1NTUyOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFD5Gsmkx
RuecttwsCG763m3u63UwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBBh0WaXlBsW56Hqk588MmJxsOrvcKfDjF57RgEDgnGnQaJcStCVWDO9UYO
JX2tdsPw+E7AjDqjsuxYaotLn3Mr3mK0sNOXq9BljBnWD4pARg89KZnZI8FN35HQ
O/LYOVHCknuPL123VmVRNs51qQA9hkPjvw21UzpDLxaUxt9Z/w==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAKD+v6LeR/WrMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTA4MTlaGA8yMTk1MDExNzA5MDgxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAka8FLhxs1cSJGK+Q+q/vTf8zVnDAPZ3U6oqppOW/cupCtpwMAQcky8DY
Yb62GF7+C6usniaq/9W6xPn/3o//wti0cNt6MLsiUeHqNl5H/4U/Q/fR+GA8pJ+L
npqZDG2tFi1WMvvGhGgIbScrjR4VO3TuKy+rZXMYvMRk1RXZ9gPhk6evFnviwHsE
jV5AEjxLz3duD+u/SjPp1vloxe2KuWnyC+EKInnka909sl4ZAUh+qIYfZK85DAjm
GJP4W036E9wTJQF2hZJrzsiB1MGyC1WI9veRISd30izZZL6VVXLXUtHwVHnVASrS
zZDVpzj+3yD5hRXsvFigGhY0FCVFnwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUxC2l6pvJaRflgu3MUdN6zTuP6YcwgY4GA1UdIwSBhjCBg4AUxC2l
6pvJaRflgu3MUdN6zTuP6YehYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAKD+v6LeR/WrMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAIK+DtbUPppJXFqQMv1f2Gky5/82ZwgbbfXa
HBeGSii55b3tsyC3ZW5ZlMJ7Dtnr3vUkiWbV1EUaZGOUlndUFtXUMABCb/coDndw
CAr53XTv7UwGVNe/AFO/6pQDdPxXn3xBhF0mTKPrOGdvYmjZUtQMSVb9lbMWCFfs
w+SwDLnm5NF4yZchIcTs2fdpoyZpOHDXy0xgxO1gWhKTnYbaZOxkJvEvcckxVAwJ
obF8NyJla0/pWdjhlHafEXEN8lyxyTTyOa0BGTuYOBD2cTYYynauVKY4fqHUkr3v
Z6fboaHEd4RFamShM8uvSu6eEFD+qRmvqlcodbpsSOhuGNLzhOQ=
-----END CERTIFICATE-----
```

Europe (Ireland) — eu-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUakDaQ1Zqy87Hy9ESXA1pFC116HkwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MTgxMFoXDTI5MDQyODE2MTgxMFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUakDaQ1Zq
y87Hy9ESXA1pFC116HkwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQADIkn/MqaLGPuK5+prZZ5Ox4bBZLPtreO2C7r0pqU2kPM2lVPyYYydkvP0
lgSmmsErGu/oL9JNztDe2oCA+kNy17ehcsf8cw0uP861czNFKCeU8b7FgBbL+sIm
qi33rAq6owWGi/5uEcfCR+JP7W+oSYVir5r/yDmWzx+BVH5S/g==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOrmqHuaUt0vMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTA2MTlaGA8yMTk1MDQwMzA5MDYxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjE7nVu+aHLtzp9FYV25Qs1mvJ1JXD7J0iQ1Gs/RirW9a5ZECCtc4ssnf
zQHq2JRVr0GRchvDrbm1HaP/avtFQR/Thvfltwu9AROVT22dUOTvERdkNzveoFCy
hf52Rqf0DMrLXG8ZmQPPXPDFAv+sVMWCDftcChxRYZ6mP9O+TpgYNT1krD5PdvJU
7HcXrkNHDYqbsg8A+Mu2hzl0QkvUET83Csg1ibeK54HP9w+FsD6F5W+6ZSHGJ88l
FI+qYKs7xsjJQYgXWfEt6bbckWs1kZIaIOyMzYdPF6ClYzEec/UhIe/uJyUUNfpT
VIsI5OltBbcPF4c7Y20jOIwwI2SgOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUF2DgPUZivKQR/Zl8mB/MxIkjZDUwgY4GA1UdIwSBhjCBg4AUF2Dg
PUZivKQR/Zl8mB/MxIkjZDWhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOrmqHuaUt0vMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGm6+57W5brzJ3+T8/XsIdLTuiBSe5ALgSqI
qnO5usUKAeQsa+kZIJPyEri5i8LEodh46DAF1RlXTMYgXXxl0YggX88XPmPtok17
```

```
l4hib/D9/lu4IaFIyLzYNSzsETYWKWoGVe7ZFz60MTRTwY2u8YgJ5dec7gQgPSGj
avB0vTIgoW41G58sfw5b+wjXCsh0nROon79RcQFFhGnvup0MZ+JbljyhZUYFzCli
31jPZiKzqWa87xh2DbAyvj2KZrZtTe2LQ48Z4G8wWytJzxEeZdREe4NoETf+Mu5G
4CqoaPR05KWkdNUdGNwXewydb3+agdCgfTs+uAjeXKNdSpbhMYg=
-----END CERTIFICATE-----
```


Europe (London) — eu-west-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUCgCV/DPxYNND/swDgEKGiC5I+EwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MjkxNFoXDTI5MDQyODE2MjkxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
```

```
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUCgCV/DPx
YNND/swDgEKGiC5I+EwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQATPu/sOE2esNa4+XPEGKlEJSgqzyBSQLQc+VWo6FAJhGG9fp7D97jhHeLC
5vwfmtTAfnGBxadfAOT3ASkxnOZhXtnRna460LtnNHm7ArCVgXKJo7uBn6ViXtFh
uEEw4y6p9YaLQna+VC8Xtgw6WKq2JXuKzuhuNKSFaGGw9vRcHg==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANBx0E2bOCEPMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA4MTEx
NDU2NDJaGA8yMTk2MDExNTE0NTY0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArYS3mJLGaMrh2DmiPLbqr4Z+xWXTzBWCjOwpsuHE9H6dWUUyl2Bgnu+Z
d8QvW306Yleec45M4F2RA3J4hWHtShzsMlOJVRt+YulGeTf9OCPr26QmIFfs5nD4
fgsJQEry2MBSGA9Fxq3Cw6qkWcrOPsCR+bHOU0XykdKl0MnIbpBf0kTfciAupQEA
dEHnM2J1L2iI0NTLBgKxy5PXLH9weX20BFauNmHH9/J07OpwL20SN5f8TxcM9+pj
Lbk8h1V4KdIwVQpdWkbDL9BCGlYjyadQJxSxz1J343NzrnDM0M4h4HtVaKOS7bQo
Bqt2ruopLRCYgcuFHck/1348iAmbRQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBG
wujwU1Otpi3iBgmhjMClgZyMMn0aQIxMigoFNqXMUNx1Mq/e/Tx+SNaOEAu0n2FF
aiYjvY0/hXOx75ewzZvM7/zJWIdLdsgewpUqOBH4DXFhbSk2TxggSPb0WRqTBxq5
Ed7F7+7GRIeBbRzdLqmISDnfqey8ufW0ks51XcQNomDIRG5s9XZ5KHviDCar8FgL
HngBCdFI04CMagM+pwTO9XN1Ivt+NzUj208ca3oP1IwEAd5KhIhPLcihBQA5/Lpi
h1s3170z1JQ1HZbDrH1pgp+8hSI0DwwDVb3IIH8kPR/J0Qn+hvOl2HOpaUg2Ly0E
pt1RCZe+W7/dF4zsbqwK
-----END CERTIFICATE-----
```


Europe (Milan) — eu-south-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqwCCQCMElHPdwG37jAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA0MjkyMDM1MjJaFw00
NTA0MjkyMDM1MjJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
```

```
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQDAkoL4YfdMI/MrQ0oL
NPfeEk94eiCQA5xNOnU7+2eVQtEqjFbDADFENh1p3sh9Q9OoheLFH8qpSfNDWn/0
ktCS909ApTY6Esx1ExjGSeQq/U+SC2JSuuTT4WFMKJ63a/czMtFkEPPnVIjJJJmT
HJSKSsVUgpdDIRvJXuyB0zdB+wIVALQ3OLaVGdlPMNfS1nD/Yyn+32wnAoGAPBQ3
7XHg5NLOS4326eFRUT+4ornQFjJjP6dp3pOBEzpImNmZTtkCNNUKE4Go9hv5T4lh
R0pODvWv0CBupMAZVBP9ObplXPCyEIZtuDqVa7ukPOUpQNgQhLLAqkigTyXVOSmt
ECBj9tu5WNP/x3iTZTHJ+g0rhIqpgh012UwJpKADgYQAAoGAV1OEQPYQUg5/M3xf
6vE7jKTxxyFWEyjKfJK7PZCzOIGrE/swgACy4PYQW+AwcUweSlK/Hx2OaZVUKzWo
wDUbeu65DcRdw2rSwCbBTU342sitFo/iGCV/Gjf+BaiAJtxniZze7J1ob8vOBeLv
uaMQmgOYeZ5e0fl04GtqPl+lhcQwCQYHKoZIzjgEAwMwADAtAhQdoeWLrkm0K49+
AeBK+j6m2h9SKQIVAIBNhS2a8cQVABDCQXVXrc0tOmO8
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAOZ3GEIaDcugMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTEwMjQx
NTE5MDlaGA8yMTk5MDMyOTE1MTkwOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCjiPgW3vsXRj4JoA16WQDyoPc/eh3QBARaApJEc4nPIGoUolpAXcjFhWplo2O+
ivgfCsc4AU9OpYdAPha3spLey/bhHPRi1JZHRNqScKP0hzsCNmKhfnZTIEQCFvsp
DRp4zr91/WS06/flJFBYJ6JHhp0KwM81XQG59lV6kkoW7QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAGLLrY3P+HH6C57dYgtJkuGZGT2+rMkk2n81/abzTJvsqRqGRrWv
XRKRXlKdM/dfiuYGokDGxiC0Mg6TYy6wvsR2qRhtXW1OtZkiHWcQCnOttz+8vpew
wx8JGMvowtuKB1iMsbwyRqZkFYLcvH+Opfb/Aayi20/ChQLdI6M2R5VU
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAO/+DgYF78KwMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA0Mjky
MDM1MjJaGA8yMTk4MTAwMjIwMzUyMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAv1ZLV+Z/P6INq+R1qLkzETBg7sFGKPiwHekbpuB6lrRxKHhj8V9vaReM
lnv1Ur5LAPpMPYDsuJ4WoUbPYAqVqyMAo7ikJHCCM1cXgZJefgN6z9bpS+uA3YVh
V/0ipHh/X2hc2S9wvxKWiSHu6Aq9GVpqL035tJQD+NJuqFd+nXrtcw4yGtmvA6wl
5Bjn8WdsP3xOTKjrByYY1BhXpP/f1ohU9jE9dstsRXLa+XTgTPWcWdCS2oRTWPGR
```

```
c5Aeh47nnDsyQfP9gLxHeYeQItV/BD9kU/2Hn6mnRg/B9/TYH8qzlRTzLapXp4/5
iNwusrTNexGl8BgvAPrfhjDpdgYuTwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQB7
5ya11K/hKgvaRTvZwVV8GlVZt0CGPtNvOi4AR/UN6TMm51BzUB5nurB4z0R2MoYO
Uts9sLGvSFALJ4otoB77hyNpH3drttU1CVVwal/yK/RQLSon/IoUkaGEbqalu+mH
nYad5IG4tEbmepX456XXcO58MKmnczNbPyw3FRzUZQtI/sf94qBwJ1Xo6XbzPKMy
xjL57LHIZCssD+XPifXay69OFlsCIgLim11HgPkRIHEOXLSf3dsW9r+4CjoZqB/Z
jj/P4TLCxbYCLkvglwaMjgEWF40Img0fhx7yT2X92MiSrs3oncv/IqfdVTiN8OXq
jgnq1bf+EZEZKvb6UCQV
-----END CERTIFICATE-----
```

Europe (Paris) — eu-west-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUaC9fX57UDr6u1vBvsCsECKBZQyIwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MzczOFoXDTI5MDQyODE2MzczOFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
```

```
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUaC9fX57U
Dr6u1vBvsCsECKBZQyIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCARv1bQEDaMEzYI0nPlu8GHcMXgmgA94HyrXhMMcaIlQwocGBs6VILGVhM
TXP2r3JFaPEpmXSQNQHvGA13clKwAZbni8wtzv6qXb4L4muF34iQRHF0nYrEDoK7
mMPR8+oXKKuPO/mv/XKo6XAV5DDERdSYHX5kkA2R9wtvyZjPnQ==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALWSfgHuT/ARMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA1MzEx
MTE4MTZaGA8yMTk2MTEwMzExMTgxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy5V7KDqnEvF3DrSProFcgu/oL+QYD62b1U+Naq8aPuljJe127Sm9WnWA
EBdOSASkOaQ9fzjCPoG5SGgWKxYoZjsevHpmzjVv9+Ci+F57bSuMbjgUbvbRIFUB
bxQojVoXQPHgK5v433ODxkQ4sjRyUbf4YV1AFdfU7zabC698YgPVOExGhXPlTvco
8mlc631ubw2g52j0lzaozUkHPSbknTomhQIvO6kUfX0e0TDMH4jLDG2ZIrUB1L4r
OWKG4KetduFrRZyDHF6ILZu+s6ywiMicUd+2UllDFC6oas+a8D11hmO/rpWU/ieV
jj4rWAFrsebpn+Nhgy96iiVUGS2LuQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDE
iYv6FQ6knXCg+svlcaQG9q59xUC5z8HvJZ1+SxzPKKC4PKQdKvIIfE8GxVXqlZG1
cl5WKTFDMapnzb9RV/DTaVzWx3cMYT77vm1Hl1XGjhx611CGcENH1egI3lOTILsa
+KfopuJEQQ9TDMAIkGjhA+KieU/U5Ctv9fdej6d0GC6OEuwKkTNzPWue6UMq8d4H
2xqJboWsE1t4nybEosvZfQJcZ8jyIYcYBnsG13vCLM+ixjuU5MVVQNMY/gBJzqJB
V+U0QiGiuT5cYgY/QihxdHt99zwGaE0ZBC7213NKrlNuLSrqhDI2NLu8NsExqOFy
OmY0v/xVmQUQl26jJXaM
-----END CERTIFICATE-----
```


Europe (Spain) — eu-south-2

DSA


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

```
MIIC8DCCAq
+gAwIBAgIGAXjwLk46MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAGG2m8EKmaf5qQqj3Z
+rzSaTaXE3B/R/4A2VuGqRYR7MljPtwdmU6/3CPjCACcZmTIcOAKbFiDHqadQgBZXfzGpzw8Zo
+eYmmk5fXycgnj57PYH1dIWU6I7mCbAah5MZMcmHaTmIsomGrhcnWB8d8qOU7oZ0UWK4lbiAQs1MihoUwCQYHKoZIzjg
WmbaU7YM5GwCFCvIJOesO5hZ8PHC52dAR8WWC6oe
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjwLkiaMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
VvR1+45Aey5zn3vPk6xBm5o9grSDL6D2iAuprQnfVXn8CIbSDbWFhA3fi5ippjKkh3sl8VyCvCOUXKdOaNrYBrPRkrdH
+3m/
rxIUZ2IK1fDlC6sWAjddf6sBrV2w2a78H0H8EwuwiSgttURBjwJ7KPPJCqaqrQIDAQABMA0GCSqGSIb3DQEBBQUAA4GB
+FzqQDzun/
iMMzcFucmLMl5BxEblrFXOz7IIuOeiGkndmrqUeDCykztLku45s7hxdNy4ltTuVAaE5aNBdw5J8U1mRvsKvHLy2ThH6h
+hBgiphYp84DUbWVYeP8YqLEJSqscKscWC
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALWSmO6DvSpQMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
MzU4NDNaGA8yMjAxMTIyMjEzNTg0M1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuAAhuSpsHC00/fD2zNlBDpNLRndi9qbHsNeuz3WqN7Samj2aSrM2hS+i
hUxx0BspZj0tZCOsbpPZ+i74NOEQtFeqQoEGvKhB1nJiF4y5I81HDhs5qHvoIivm
7rbbik3zgm1PqS/DmDjVQaXPcD31Rd9ILwBmWEwJqHigyNVlxYtCzTQcrlBrvNZM
dnNgCDAdX/HBEFxx9O12xeu0bSt0s+PJWZ1RTbYrNe7LIH6ntUqHxP/ziQ5trXEZ
uqy7aWk1L8uK4jmyNph0lbaqBa3Y6pYmU1nC27UE4i3fnPB0LSiAr+SrwVvX1g4z
ilo8kr+tbIF+JmcgYLBvO8Jwp+EUqQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUwvGzKJL9A5LReJ4Fxo5K6I20xcowgY4GA1UdIwSBhjCBg4AUwvGz
KJL9A5LReJ4Fxo5K6I20xcqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
```

```
b24gV2ViIFNlcnZpY2VzIExMQ4IJALWSmO6DvSpQMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAJAZd31jyoTGLawAD2+v/vQsaB9vZIx5EImi
G8YGkd61uFWeNhAmtrwyE/i6FDSIphDrMHBkvw/D3BsqK+Ev/JOK/VYuaYDx/8fp
H4cwp9jC57CXzdIDREWNf6M9PsHFg2WA9XNNtClOZL5WJiJwel8eDSg+sqJUxE0l
MW+QChq/20F6niyaRK4bXrZq14as7h+F9u3A9xHEOVP7Zk9C2ehrBXzCMLSDt3GV
fEuMea2RxMhozwz34Hkdb6j18qoCfygubulovRNQjKw/cEmgPR16KfZPP5caILVt
9qkYPvePmbiVswZDee73cDymJYxLqILpOZwyXvUH8StiH42FHZQ=
-----END CERTIFICATE-----
```


Europe (Stockholm) — eu-north-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUN1c9U6U/xiVDFgJcYKZB4NkH1QEwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MDYwM1oXDTI5MDQyODE2MDYwM1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
```

```
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUN1c9U6U/
xiVDFgJcYKZB4NkH1QEwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBTIQdoFSDRHkpqNPUbZ9WXR2O5v/9bpmHojMYZb3Hw46wsaRso7STiGGX/
tRqjIkPUIXsdhZ3+7S/RmhFznmZc8e0bjU4n5vi9CJtQSt+1u4E17+V2bF+D3h/7
wcfE0l3414Q8JaTDtfEf/aF3F0uyBvr4MDMd7mFvAMmDmBPSlA==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALc/uRxg++EnMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
NDAwMTFaGA8yMTk3MDkxMzE0MDAxMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzwCGJEJIxqtr2PD2a1mA6LhRzKhTBa1AZsg3eYfpETXIVlrpojMfvVoN
qHvGshWLgrGTT6os/3gsaADheSaJKavxwX3X6tJA8fvEGqr3a1C1MffH9hBWbQqC
LbfUTAbkwis4GdTUwOwPjT1Cm3u9R/VzilCNwkj7iQ65AFAI8Enmsw3UGldEsop4
yChKB3KW3WI0FTh0+gD0YtjrqqYJxpGOYBpJp5vwdd3fZ4t1vidmDMs7liv4f9Bx
p0oSmUobU4GUlFhBchK1DukICVQdnOVzdMonYm7s+HtpFbVHR8yf6QoixBKGdSal
mBf7+y0ixjCn0pnC0VLVooGo4mi17QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
4ONZiixgk2sjJctwbyD5WKLTH6+mxYcDw+3y/F0fWz561YORhP2FNnPOmEkf0Sl/
Jqk4svzJbCbQeMzRoyaya/46d7UioXMHRZam5IaGBhOdQbi97R4VsQjwQj0RmQsq
yDueDyuKTwWLK9KnvI+ZA6e6bRkdNGflK4N8GGKQ+fBhPwVELkbT9f16OJkezeeN
S+F/gDADGJgmPXfjogICb4Kvshq0H5Lm/xZlDULF2g/cYhyNY6EOI/eS5m1I7R8p
D/m6WoyZdpInxJfxW616OMkxQMRVsruLTNGtby3u1g6ScjmpFtvAMhYejBSdzKG4
FEyxIdEjoeO1jhTsck3R
-----END CERTIFICATE-----
```


Europe (Zurich) — eu-central-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXiKJnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
```

```
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAYNjaCNg/
cfgQOl1BUj5ClUulqwZ9Q+SfDzPZh9D2C0VbiRANiZoxrV8RdgmzzC5T7VcriVwjwvta2Ch//
b+sZ86E5h0XWWr+BeEjD9cu3eDj12XB5sWEbNHNx49p5Tmtu5r2LDtlL8X/
Rpfalu2Z2OJgjFJWGf7hRwxe456n
+lowCQYHKoZIzjgEAwMvADAsAhRChsLcj4U5CVb2cp5M0RE1XbXmhAIUeGSnH+aiUQIWmPEFja+itWDufIk=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjSGFGiMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
opKZAUusJx2hpgU3pUHhlp9ATh/VeVD582jTd9IY
+8t5MDa6Z3fGliByEiXz0LEHdi8MBacLREu1TwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAILlpoE3k9o7KdALAxsFJNit
+g3RMzdbiFM+7MA63Nv5fsf+0xgcjSNBElvPCDKFvTJl4QQhToy056llO5GvdS9RK
+H8xrP2mrqngApoKTApv93vHBixgFSn5KrczRO0YSm3OjkqbydU7DFlmkXXR7GYE+5jbHvQHYiT1J5sMu
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALvT012pxTxNMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
NTEyMDdaGA8yMjAxMTIyMjE1MTIwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAyn+Lsnq1ykrfYlZkk6aAAYNReNd9Iw8AUwCBkgOr2eBiBBepYxHwU85N
++moQ+j0EV2VaahBeTLShGZZS1HsyK8+cYT2QzpghIoamcYhrPXyIxlWiRQlaqSg
OFiE9bsqL3rCF5Vz+tOiTe5W/7ojfOFls6++g7ZpobwJlpMbuJepqyeHMPyjvO5A
age81lJewc4bxo2ntaW0HCqNksqfYB78j6X6kn3PFpX7FaYAwZA+Xx6C7UCY7rNi
UdQzfAo8htfJi4chz7frpUdQ9kl3IOQrsLshBB5fFUjl09NiFipCGBwi+8ZMeSn1
5qwBI01BWXPfG7WX6Owyjhmh6JtE1wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU8HN4vvJrsZgPQeksMBgJb9xR1yYwgY4GA1UdIwSBhjCBg4AU8HN4
vvJrsZgPQeksMBgJb9xR1yahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALvT012pxTxNMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGlHYDtcHpfBvdHx9HeQE8HgNugJUPdEqxun
t9U33p8VFrs+uLPtrOd9HDJEGvvs5h84EUie/oGJxRt7V1Vlid1PvHf6cRmpjgqY
```

```
YdggAVkZtY/PnFVmzf2bMVlSQPrqCl7U0zaw2Kvnj4zgX0rZyCetgrRZSUSxotyp
978Wy9ccXwVSeYG/YAr5rJpS6ZH7eRQvUY0IzwFNeaOPgOTEVpcjWlV6+MQEvsEx
W85q+s6AVr49eppEx8SLJsl0C23yB+L+t32tAveQImRWtJMpzZ5cxh/sYgDVeoC0
85H1NK/7H9fAzT1cPu1oHSnB0xYzzHGOAmXmusMfwUk8fL1RQkE=
-----END CERTIFICATE-----
```


Israel (Tel Aviv) — il-central-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq+gAwIBAgIGAX0QPi
+9MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYX
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAbazCL5XXyPmcw3+oMYQUF5/9YogW6D0FZbYuyPgjOoUwWdl6fjlzWca
pq+l1ezuK2DFOzNTEyPEwwCQYHKoZIzjgEAwMvADAsAhRt1jkpXsvrS
+xTo2M9h2s2uLAhEQIUOZ2FcnTSrshF2EIdixZZwtNv66Q=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAX0QQGVLMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+S8vOy5hpLoRe4RkOrY0cM3bN07GdEMlin5mU0y1t8y3ct4YewvmkgT42kTyMM
+t1K4S0xsqjXxxS716uGYh7eWtkxrCihj8AbXN/6pa095h
+7TZyl2n83keiNUzM2KoqQVMwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBADwA6VVEIIZD2YL00F12po40xDLzIc9XvqFPS
FmU7H8s62/jD6cOR1A1cClIyZUe1yT1ZbPySCs43J+Thr8i8FSRxzDBSZZi5foW
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOVp1h2I9wW7MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTUx
MjQ0MTJaGA8yMjAxMTIxOTEyNDQxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13PkyWvl61iV/SYfO1UFO76UpDfPm2SF/Rz/o33cm699X++EYPxTnoEc
vmWeS0I7eDXc4OCUiToG0sEx0klE0CX1Z1tK6qJ+zgWQLZ9SZEC9H0NsSA6LhrHu
Nq0dzeK3LjhdfcX46/4GqdiptpdTuM4m/hOQ5yx4JMQ/n1sdpv4M5VLRWwWW9Lem
ufb79Id7O9SispxgRsz1KXIjp7N9S4BY7itSXz97uSyzTqEjWZ6mDUhTu3t21GKC
6flALGTTrG2yghEhz53rkvLsvwzjPSSlT6LIf0mrRPzHaf+EdaKoasElE1SHh+ZH
9mI81HywpE+HZ+W+5hBCvjYp90Y1fwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU58tN2JO+yEGq5JbIXxGi4vRVPyIwgY4GA1UdIwSBhjCBg4AU58tN
2JO+yEGq5JbIXxGi4vRVPyKhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOVp1h2I9wW7MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANBN0e1EqNy4+IU2yQzMJ+Wy5ZIOtTP6GSBR
7muVY1bDeAwtNTEOpwgrZVlC7/xq5Q0LC1y0Z70hHXEf8au7qStaAoUtxzvhTAZI
NC0lwoFU56UFw4N0vZII17iqEfoqRC4PpI3OxqEJHFy0VLlvAzJoKB4QLLqDAYVA
LXCiOLoVT+y9tRYsxw5MyO0Bi6fxQIIADl2bE9xkunTN1JkkwqO3LxNy/ryz4QWR
8K7jHUItifv4h/hxBKpHEquN8CkdvM9oeGl7I8PFrSFEpGr1euDXY0euZzzYiDBV
m6GpTJgzpVsEuIX52dPcPemwQncoIfZyhWDW85MJUnby2WTEcFo=
-----END CERTIFICATE-----
```


Mexico (Central) — mx-central-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq8CCQD4QwfTExrgxTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDExMTQwOTQ4MjRaGA8y
MDUwMTExNTA5NDgyNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASwGByqGSM44BAEwggEfAoGBAK+pJEOtD2RFlCbf
oAzONffZa42FB8G6OA/30DT81xWvB2Hnmgw7LfiELYqgfuck3YSDffNDcmiFTJuX
AU9u0NtbbxcOytcdUq4HInpHiB85I6WWWKYB66aGbvYowUPpqBZkASb1i/pYAh5W
nNYvx408008tTzqpMfZDchJHzWqBAhUA0ogoJzTw2/4pKhz9aqTsRCzRVPsCgYEA
qklRUcuU0du/bT/M6kWxYvrTGhO9KXQe+7RtbaIq4dWRsCBnO4smDY/GmI9H8ew1
LRJ9AcLGmxDm795CvVZkHNcht7PDAREagWmz2YvhLA+ev6U0RpfdlBXCck2p1CxQ
LMtoFO7DThksHIQBtnlQdHDvguKEZQz/Iobhne6Kb3wDgYQAAoGAQSTlOqFq6RtR
Jvp406XhGl+eO9SQUpevpzG3Dzbdy6EQ8PBJDOHHJvxbebvpnQssh0CnQfTkSw26
jwPy9V5zRC0ZezwnRTyGSJwiErUKDGeVekEAoNJL1USy8jKkgBajlskZrR+0JHDN
Jv3UwYIPplc4ZS2f2E7btrtlaWt/P70wCQYHKoZIzjgEAwMvADAsAhRXljPWKWYf
61DkDVdgPPHsS2/LuwIUejcGV9WuS3uPvJ6lmn4opxUGBZw=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICPzCCAaigAwIBAgIUCmzpTTMBQYItpMC2VDYsZfIAS7IwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcMB1NlYXR0bGUxIDAeBgNVBAoMF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQwMjEyNTAzNloXDTI5MDQwMTEyNTAzNlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYXR0bGUxIDAe
BgNVBAoMF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDfvOnCzm1iN58Nm7k6ehoy6v0lnnFI617D6CY3bfuq01RCdEQL
96+pYawJieTH8JAQKj02CAa3AeaqdXTE/pDhI/YKLreeMb4K68WMn24Wjjs6oxjB
bAmsKXtt9ihKHGBFNUhgFrNFYyA2i7ieJviwpHjQ/XgXiG2u1/t/4VydUwIDAQAB
MA0GCSqGSIb3DQEBCwUAA4GBAL5+vvj4lhaE+J5tuCqV3XJzDd97lsD4le2O2uGw
P0sGdUcRAdxzU3Bwq/hhtzNWnfwo0aCEQKmLM7xyd3nUa0VvKXLq+DDuayipWINr
OATnNxFRe99d38qHTR1dgkjZdKbbtnl6O4fgM57tVEuQJd/N4ILl9jaRcJ5Ip+9t
3y5t
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALxuE00HoJomMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEwMzEx
MTE0MzNaGA8yMjA0MDQwNjExMTQzM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13xWucALo8M/TXbZJgHrqFqFON91dSLPE/eLbmeIJbP1pb9ICd33qKAx
HlKSrXI9b9YS3U1Pl0bF3ZgfeE/x4Y0KDDZwzpf07H8IgRittULJoNLYVKCJXWPq
Ky1qvDJX3653dUbUu9eAdVCTRgk7eKpPBLAmW27+pgAGzEYrVV3u2AvqNTonvfTU
sPgEVnAL1J490pNM85KtFynxFTWGigHkd3BHidxmLrTH4I4eRxNZ9q/3gsDW+zKt
jQlpM7JzZa2OqxsF5YQDh1fF52Emqsr+ufLeGqDLOgT1QWcqpz57AX8GqZpgZULo
itCRNXbQDzZY9FxIgpiFJv3y/qYYDQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCt
MTeH4EgqjJEjb1qm5tzXYurLprVrEVQ+PhGXJFJd3xAZyeaDVYy7kio08E2xhmHd
HtkBDty2KnOHsTQmeAJCci7d4tYXZ/1qe341wwm9OoFcO8jhIndx6FXJCgQUY4dL
AAr9HQJFWG5dMZgbi1Zuhxdio3sSoOBjL2p7QIsGNkITvCDIs/H0/szpJnyyyIqu
wmUhSel5hdy5MwOsyUKVGnAdaS5Vd9oL4kLszS9nBZ7ny6BC9odIkFAdGqQ5vM4z
vcbfOq14hjatQmJgJhksN/0Dpl78Gheq0pIhP8LTkAOEG2832nQLzCa3oxSk8otG
GJXkzzyQjse+13r8+yNJ
-----END CERTIFICATE-----
```

Middle East (Bahrain) — me-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCVWIgSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDcwojQfgWdV1QliO0B
8n6cLZ38VE7ZmrjZ9OQV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q
PHlP1WGL8IZ34BUgRTtG4TVolvp0smjkMvyRu5hIdKtzjV93Ccx15gVgyk+o1IEG
fZ2Kbw/Dd8JfoPS7KaSCmJKxXQIVAIZbIaDFRGa2qcMkW2HWASyNDl7bAoGBANtz
IdhfMq+l2I5iofY2oj3HI21Kj3LtZrWEg3W+/4rVhL3lTm0Nne1rl9yGujrjQwy5
Zp9V4A/w9w2O10Lx4K6hj34Eefy/aQnZwNdNhv/FQP7Az0fju+Yl6L13OOHQrL0z
Q+9cF7zEosekEnBQx3v6psNknKgD3Shgx+GO/LpCA4GFAAKBgQCVS7m77nuNAlZ8
wvUqcooxXMPkxJFl54NxAsAul9KP9KN4svm0O3Zrb7t2FOtXRM8zU3TqMpryq1o5
mpMPsZDg6RXo9BF7Hn0DoZ6PJTamkFA6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr
12AztQ8bFWsrTgTzPE3p6U5ckcgV1TAJBgcqhkjOOAQDAy8AMCwCFB2NZGWm5EDl
86ayV3c1PEDukgQIAhQow38rQkN/VwHVeSW9DqEshXHjuQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDPDCCAqWgAwIBAgIJAMl6uIV/zqJFMA0GCSqGSIb3DQEBCwUAMHIxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0dGxlMSAw
HgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwRZWMyLmFt
YXpvbmF3cy5jb20wIBcNMTkwNDI2MTQzMjQ3WhgPMjE5ODA5MjkxNDMyNDdaMHIx
CzAJBgNVBAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0
dGxlMSAwHgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwR
ZWMyLmFtYXpvbmF3cy5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALVN
CDTZEnIeoX1SEYqq6k1BV0ZlpY5y3KnoOreCAE589TwS4MX5+8Fzd6AmACmugeBP
Qk7Hm6b2+g/d4tWycyxLaQlcq81DB1GmXehRkZRgGeRge1ePWd1TUA0I8P/QBT7S
gUePm/kANSFU+P7s7u1NNl+vynyi0wUUrw7/wIZTAgMBAAGjgdcwgdQwHQYDVR0O
BBYEFILtMd+T4YgH1cgc+hVsVOV+480FMIGkBgNVHSMEgZwwgZmAFILtMd+T4YgH
1cgc+hVsVOV+480FoXakdDByMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGlu
Z3RvbjEQMA4GA1UEBwwHU2VhdHRsZTEgMB4GA1UECgwXQW1hem9uIFdlYiBTZXJ2
aWNlcyBMTEMxGjAYBgNVBAMMEWVjMi5hbWF6b25hd3MuY29tggkAyXq4hX/OokUw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOBgQBhkNTBIFgWFd+ZhC/LhRUY
4OjEiykmbEp6hlzQ79T0Tfbn5A4NYDI2icBP0+hmf6qSnIhwJF6typyd1yPK5Fqt
NTpxxcXmUKquX+pHmIkK1LKDO8rNE84jqxrxRsfDi6by82fjVYf2pgjJW8R1FAw+
```

```
mL5WQRFexbfB5aXhcMo0AA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANZkFlQR2rKqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTAyMDUx
MzA2MjBaGA8yMTk4MDcxMTEzMDYyMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy4Vnit2eBpEjKgOKBmyupJzJAiT4fr74tuGJNwwa+Is2vH12jMZn9Il1
UpvvEUYTIboIgISpf6SJ5LmV5rCv4jT4a1Wm0kjfNbiIlkUi8SxZrPypcw24m6ke
BVuxQZrZDs+xDUYIZifTmdgD50u5YE+TLg+YmXKnVgxBU6WZjbuK2INohi71aPBw
2zWUR7Gr/ggIpf635JLU3KIBLNEmrkXCVSnDFlsK4eeCrB7+UNak+4BwgpuykSGG
Op9+2vsuNqFeU1l9daQeG9roHR+4rIWSPa0opmMxv5nctgypOrE6zKXx2dNXQldd
VULv+WH7s6Vm4+yBeG8ctPYH5GOo+QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBs
ZcViiZdFdpcXESZP/KmZNDxB/kktlIEIhsQ+MNn29jayE5oLmtGjHj5dtA3XNKlr
f6PVygVTKbtQLQqunRT83e8+7iCZMKI5ev7pITUQVvTUwI+Fc01JkYZxRFlVBuFA
WGZO+98kxCS4n6tTwVt+nSuJr9BJRVC17apfHBgSS8c5OWna0VU/Cc9ka4eAfQR4
7pYSDU3wSRE01cs30q34lXZ629IyFirSJ5TTOIc0osNL7vwMQYj8HOn4OBYqxKy8
ZJyvfXsIPh0Na76PaBIs6ZlqAOflLrjGzxBPiwRM/XrGmF8ze4KzoUqJEnK13O6A
KHKgfiigQZ1+gv5FlyXH
-----END CERTIFICATE-----
```

Middle East (UAE) — me-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXhqnnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAW+csuHsWp/7/
pv8CTKFwxsYudxuR6rbWaHCykIeAydXL9AWnphK6yp1ODEMBFl68Xq8Hp23sOWyf8moOhqCom9+0+ovuUFdpvCie86bp
TOZU568Ty1ff3dDWbdRzeNQRHodRG+XEQSizMkAreeWt4kBa+PUwCQYHKoZIzjgEAwMvADAsAhQD3Z
+XGmzKmgaLgGcVX/Qf1+Tn4QIUH1cgksBSVKbWj81tovBMJeKgdYo=
```

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

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjRrnDjMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
KyA6zyruJQrYy00a6wqLA7eeUzk3bMiTkLsTeDQfrkaZMfBAjGaaOymRo1C3qzE4rIenmahvUplu9ZmLwL1idWXMRX2R
+d2SeoKOKQWoc2UOFZMHYxDue7zkyk1CIRaBukTeY13/
RIrlc6X61zJ5BBtZXlHwayjQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBABTqTy3R6RXKPW45FA+cgo7YZEj/
Cnz5YaoUivRRdX2A83BHuBTvJE2+WX00FTEj4hRVjameE1nENoO8Z7fUVloAFDlDo69fhkJeSvn51D1WRrPnoWGgEfr1
B+Wqm3kVEz/QNcz6npmA6
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM4h7b1CVhqqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MTEx
MDE1MDNaGA8yMjAxMDkxNTEwMTUwM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApybTWfMOhSoMpqPo72eqAmnn1dXGZM+G8EoZXzwHwT/+IHEXNB4q5N6k
tudYLre1bJxuzEw+iProSHjmb9bB9YscRTofjVhBlt35Fc+i8BaMeH94SR/eE8QO
mll8gnLNW3d62lyuhzuyv1e5wVlRqzYw+X2zRH4/wRDOCOpzjKoHIgyPKsMgwsw5
aTZhNMsGxZN9dbkf0iCGeQLDytwU/JTh/HqvSr3VfUOapTJJiyAxoCtZWgp1/7wC
RvOCSMRJobpUqxZgl/VsttwNkikSFz1wGkcYeSQvk+odbnYQckA8tdddoVI56eD4
qtREQvfpMAX5v7fcqLexl5d5vH8uZQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUOadrbTs+OhzwoAgUJ7RqQNdwufkwgY4GA1UdIwSBhjCBg4AUOadr
bTs+OhzwoAgUJ7RqQNdwufmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM4h7b1CVhqqMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAICTdAOGEOnII8HaGCpCB8us/hGFaLptJaAf
D5SJAyVy66/mdfjGzE1BKkKxnbxemEVUIzbRidOnyilB+pKwN3edAjTZtWdpVA0V
R/G/qQPmcVljtycBz4VC6SuOUYflGzLH1GZ6GJWbuDtFzw8r7HGdRN1wrEPe3UF2
sMpuVezqnRUdvVRoVQP4jFgNsE7kNvtn2NiPhb/CtrxpcwIQ7r6YeoHcBSheuV1Z
xZDHynC3KUprQGx1+Z9QqPrDf180MaoqAlTl4+W6Pr2NJYrVUFGS/ivYshMg574l
CPU6r4wWZSKwEUXq4BInYX6z6iclp/p/J5QnJp2mAwyi6M+I13Y=
-----END CERTIFICATE-----
```

South America (São Paulo) — sa-east-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUX4Bh4MQ86Roh37VDRRX1MNOB3TcwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2NDYwOVoXDTI5MDQyODE2NDYwOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUX4Bh4MQ8
6Roh37VDRRX1MNOB3TcwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBnhocfH6ZIX6F5K9+Y9V4HFk8vSaaKL5ytw/P5td1h9ej94KF3xkZ5fyjN
URvGQv3kNmNJBoNarcP9I7JIMjsNPmVzqWawyCEGCZImoARxSS3Fc5EAs2PyBfcD
9nCtzMTaKO09Xyq0wqXVYn1xJsE5d5yBDsGrzaTHKjxo61+ezQ==
```

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

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMcyoxx4U0xxMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU4MDJaGA8yMTk1MDExNzA4NTgwMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAw45IhGZVbQcy1fHBqzROhO8CsrDzxj/WP4cRbJo/2DAnimVrCCDs5O86
FA39Zo1xsDuJHDlwMKqeXYXkJXHYbcPWc6EYYAnR+PlLG+aNSOGUzsy202S03hT0
B20hWPCqpPp39itIRhG4id6nbNRJOzLm6evHuepMAHR4/OV7hyGOiGaV/v9zqiNA
pMCLhbh2xk0PO35HCVBuWt3HUjsgeks2eEsu9Ws6H3JXTCfiqp0TjyRWapM29OhA
cRJfJ/d/+wBTz1fkWOZ7TF+EWRIN5ITEadlDTPnF1r8kBRuDcS/lIGFwrOOHLo4C
cKoNgXkhTqDDBDu6oNBb2rS0K+sz3QIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUqBy7D847Ya/w321Dfr+rBJGsGTwwgY4GA1UdIwSBhjCBg4AUqBy7
D847Ya/w321Dfr+rBJGsGTyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMcyoxx4U0xxMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACOoWSBf7b9AlcNrl4lr3QWWSc7k90/tUZal
PlT0G3Obl2x9T/ZiBsQpbUvs0lfotG0XqGVVHcIxF38EbVwbw9KJGXbGSCJSEJkW
vGCtc/jYMHXfhx67Szmftm/MTYNvnzsyQQ3v8y3Rdah+xe1NPdpFrwmfL6xe3pFF
cY33KdHA/3PNLdn9CaEsHmcmj3ctaaXLFIzZhQyyjtsrgGfTLvXeXRokktvsLDS/
YgKedQ+jFjzVJqgr4NjfY/Wt7/8kbbdhzaqlB5pCPjLLzv0zp/XmO6k+JvOePOGh
JzGk5t1QrSju+MqNPFk3+1O7o910Vrhqw1QRB0gr1ExrviLbyfU=
-----END CERTIFICATE-----
```

AWS GovCloud (US-East) — us-gov-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
```

```
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULVyrqjjwZ461qelPCiShB1KCCj4wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDUwNzE1MjIzNloXDTI5MDUwNjE1MjIzNlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCpohwYUVPH9I7Vbkb3WMe/JB0Y/bmfVj3VpcK445YBRO9K80al
esjgBc2tAX4KYg4Lht4EBKccLHTzaNi51YEGX1aLNrSmxhz1+WtzNLNUsyY3zD9z
vwX/3k1+JB2dRA+m+Cpwx4mjzZyAeQtHtegVaAytkmqtxQrSCexBxvqRqQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU1ZXneBYnPvYXkHVlVjg7918V
gE8wgZkGA1UdIwSBkTCBjoAU1ZXneBYnPvYXkHVlVjg7918VgE+hYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IULVyrqjjw
Z461qelPCiShB1KCCj4wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBfAL/YZv0y3zmVbXjyxQCsDloeDCJjFKIu3ameEckeIWJbST9LMto0zViZ
puIAf05x6GQiEqfBMk+YMxJfcTmJB4Ebaj4egFlslJPSHyC2xuydHlr3B04INOH5
Z2oCM68u6GGbj0jZjg7GJonkReG9N72kDva/ukwZKgq8zErQVQ==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALPB6hxFhay8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
MjMyNDlaGA8yMTk3MDkxMzEyMzI0OVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAva9xsI9237KYb/SPWmeCVzi7giKNron8hoRDwlwwMC9+uHPd53UxzKLb
pTgtJWAPkZVxEdl2Gdhwr3SULoKcKmkqE6ltVFrVuPT33La1UufguT9k8ZDDuO9C
```

```
hQNHUdSVEuVrK3bLjaSsMOS7Uxmnn7lYT990IReowvnBNBsBlcabfQTBV04xfUG0
/m0XUiUFjOxDBqbNzkEIblW7vK7ydSJtFMSljga54UAVXibQt9EAIF7B8k9l2iLa
mu9yEjyQy+ZQICTuAvPUEWe6va2CHVY9gYQLA31/zU0VBKZPTNExjaqK4j8bKs1/
7dOV1so39sIGBz21cUBec1o+yCS5SwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBt
hO2W/Lm+Nk0qsXW6mqQFsAou0cASc/vtGNCyBfoFNX6aKXsVCHxq2aq2TUKWENs+
mKmYu1lZVhBOmLshyllh3RRoL3Ohp3jCwXytkWQ7ElcGjDzNGc0FArzB8xFyQNdK
MNvXDi/ErzgrHGSpcvmGHiOhMf3UzChMWbIr6udoDlMbSIO7+8F+jUJkh4Xl1lKb
YeN5fsLZp7T/6YvbFSPpmbn1YoE2vKtuGKxObRrhU3h4JHdp1Zel1pZ6lh5iM0ec
SD11SximGIYCjfZpRqI3q50mbxCd7ckULz+UUPwLrfOds4VrVVSj+x0ZdY19Plv2
9shw5ez6Cn7E3IfzqNHO
-----END CERTIFICATE-----
```

AWS GovCloud (US-West) — us-gov-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUe5wGF3jfb7lUHzvDxmM/ktGCLwwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDUwNzE3MzAzMloXDTI5MDUwNjE3MzAzMlowXDELMAkGA1UEBhMCVVMx
```

```
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCpohwYUVPH9I7Vbkb3WMe/JB0Y/bmfVj3VpcK445YBRO9K80al
esjgBc2tAX4KYg4Lht4EBKccLHTzaNi51YEGX1aLNrSmxhz1+WtzNLNUsyY3zD9z
vwX/3k1+JB2dRA+m+Cpwx4mjzZyAeQtHtegVaAytkmqtxQrSCexBxvqRqQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU1ZXneBYnPvYXkHVlVjg7918V
gE8wgZkGA1UdIwSBkTCBjoAU1ZXneBYnPvYXkHVlVjg7918VgE+hYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUe5wGF3jf
b7lUHzvDxmM/ktGCLwwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCbTdpx1Iob9SwUReY4exMnlwQlmkTLyA8tYGWzchCJOJJEPfsW0ryy1A0H
YIuvyUty3rJdp9ib8h3GZR71BkZnNddHhy06kPs4p8ewF8+d8OWtOJQcI+ZnFfG4
KyM4rUsBrljpG2aOCm12iACEyrvgJJrS8VZwUDZS6mZEnn/lhA==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANCOF0Q6ohnuMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTAx
OTQyNDdaGA8yMTk1MDIxMzE5NDI0N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzIcGTzNqie3f1olrrqcfzGfbymSM2QfbTzDIOG6xXXeFrCDAmOq0wUhi
3fRCuoeHlKOWAPu76B9os71+zgF22dIDEVkpqHCjBrGzDQZXXUwOzhm+PmBUI8Z1
qvbVD4ZYhjCujWWzrsX6Z4yEK7PEFjtf4M4W8euw0RmiNwjy+knIFa+VxK6aQv94
lW98URFP2fD84xedHp6ozZlr3+RZSIFZsOiyxYsgiwTbesRMI0Y7LnkKGCIHQ/XJ
OwSISWaCddbu59BZeADnyhl4f+pWaSQpQQ1DpXvZAVBYvCH97J1oAxLfH8xcwgSQ
/se3wtn095VBt5b7qTVjOvy6vKZazwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA/
S8+a9csfASkdtQUOLsBynAbsBCH9Gykq2m8JS7YE4TGvqlpnWehz78rFTzQwmz4D
fwq8byPkl6DjdF9utqZ0JUo/Fxelxom0h6oievtBlSkmZJNbgc2WYm1zi6ptViup
Y+4S2+vWZyg/X1PXD7wyRWuETmykk73uEyeWFBYKCHWsO9sI+62O4Vf8Jkuj/cie
1NSJX8fkervfLrZSHBYhxLbL+actVEo00tiyZz8GnhgWx5faCY38D/k4Y/j5Vz99
7lUX/+fWHT3+lTL8ZZK7fOQWh6NQpI0wTP9KtWqfOUwMIbgFQPoxkP00TWRmdmPz
WOwTObEf9ouTnjG9OZ20
-----END CERTIFICATE-----
```


Precision clock and time synchronization on your EC2 instance

A consistent and accurate time reference on your Amazon EC2 instance is crucial for many server tasks and processes. Time stamps in system logs play an essential role in identifying when issues occurred and the chronological order of events. When you use the AWS CLI or an AWS SDK to make requests from your instance, these tools sign requests on your behalf. If your instance's date and time settings are inaccurate, it can result in a discrepancy between the date in the signature and the date of the request, leading to AWS rejecting your requests.

To address this important aspect, Amazon offers the Amazon Time Sync Service, which is accessible from all EC2 instances and used by various AWS services. The service uses a fleet of satellite-connected and atomic reference clocks in each AWS Region to deliver accurate and current time readings of the Coordinated Universal Time (UTC) global standard.

For the best performance, we recommend using the local Amazon Time Sync Service on your EC2 instances. For a backup to the local Amazon Time Sync Service on your instances, or to connect resources outside of Amazon EC2 to the Amazon Time Sync Service, you can use the public Amazon Time Sync Service located at `time.aws.com`. The public Amazon Time Sync Service, like the local Amazon Time Sync Service, automatically smears any leap seconds that are added to UTC. The public Amazon Time Sync Service is supported globally by our fleet of satellite-connected and atomic reference clocks in each AWS Region.

Hardware packet timestamping

You can enable hardware packet timestamping on your instance to add a 64-bit nanosecond-precision timestamp to every incoming network packet. Because hardware packet timestamping occurs at the hardware level—before the packet reaches the kernel, socket, or application layer—you bypass any delays added by software timestamping. The underlying reference clock for hardware timestamping is the Amazon Time Sync Service PTP hardware clock.

Benefits

Hardware packet timestamping provides the following benefits:

- Improves event ordering, which can also be used to determine the actual order in which packets arrive at your EC2 instance, ensuring fair packet processing.
- Measures one-way network latency.
- Increases distributed transaction speed with higher precision and accuracy compared to most on-premises solutions.

Prerequisites and configuration

To enable hardware packet timestamping, your instance must meet the following prerequisites:

- Must be a Linux instance.
- Meet the requirements to support the PTP hardware clock.

For the configuration instructions, see Hardware Packet Timestamping on the **Linux kernel driver for Elastic Network Adapter (ENA) family** page on *GitHub*.

Leap seconds

Leap seconds, introduced in 1972, are occasional one-second adjustments to UTC time to factor in irregularities in the earth's rotation in order to accommodate differences between International Atomic Time (TAI) and solar time (Ut1). To manage leap seconds on behalf of customers, we designed leap second smearing within the Amazon Time Sync Service. For more information, see Look Before You Leap – The Coming Leap Second and AWS.

Leap seconds are going away, and we are in full support of the decision made at the 27th General Conference on Weights and Measures to abandon leap seconds by or before 2035.

To support this transition, we still plan on smearing time during a leap second event when accessing the Amazon Time Sync Service over the local NTP connection or our public NTP pools (`time.aws.com`). The PTP hardware clock, however, does not provide a smeared time option. In the event of a leap second, the PTP hardware clock will add the leap second following UTC standards. Leap-smeared and leap second time sources are the same in most cases. But because they differ during a leap second event, we do not recommend using both smeared and non-smeared time sources in your time client configuration during a leap second event.

Topics

Related resources

- AWS Compute Blog: It's About Time: Microsecond-Accurate Clocks on Amazon EC2 Instances
- AWS Cloud Operations & Migrations Blog: Manage Amazon EC2 instance clock accuracy using Amazon Time Sync Service and Amazon CloudWatch – Part 1
- (Linux) https://chrony-project.org/

Set the time reference on your EC2 instance to use the local Amazon Time Sync Service

The local Amazon Time Sync Service either uses the Network Time Protocol (NTP), or provides a local Precision Time Protocol (PTP) hardware clock on supported instances. The PTP hardware clock supports either an NTP connection (Linux and Windows instances), or a direct PTP connection (Linux instances only). The NTP and direct PTP connections use the same highly accurate time source, but the direct PTP connection is more accurate than the NTP connection. The NTP connection to the Amazon Time Sync Service supports leap smearing while the PTP connection to the PTP hardware clock does not smear time. For more information, see Leap seconds.

Your instances can access the local Amazon Time Sync Service as follows:

- Through NTP at the following IP address endpoints:
  - IPv4: `169.254.169.123`
  - IPv6: `fd00:ec2::123` (Only accessible on Nitro-based instances.)
- (Linux only) Through a direct PTP connection to connect to a local PTP hardware clock:
  - `PHC0`

Amazon Linux AMIs, Windows AMIs, and most partner AMIs configure your instance to use the NTP IPv4 endpoint by default. This is the recommended setting for most customer workloads. No further configuration is required for instances launched from these AMIs unless you want to use the IPv6 endpoint or connect directly to the PTP hardware clock.

NTP and PTP connections do not require any VPC configuration changes, and your instance does not require access to the internet.

Considerations

- There is a 1024 packet per second (PPS) limit to services that use link-local addresses. This limit includes the aggregate of Route 53 Resolver DNS Queries, Instance Metadata Service (IMDS) requests, Amazon Time Service Network Time Protocol (NTP) requests, and Windows Licensing Service (for Microsoft Windows based instances) requests.
- Only Linux instances can use a *direct PTP connection* to connect to the local PTP hardware clock. Windows instances use NTP to connect to the local PTP hardware clock.

Contents

Connect to the IPv4 endpoint of the Amazon Time Sync Service

Your AMI might already have configured the Amazon Time Sync Service by default. Otherwise, use the following procedures to configure your instance to use the local Amazon Time Sync Service through the IPv4 endpoint.

For help troubleshooting issues, see Troubleshoot NTP synchronization issues on Linux instances or Troubleshoot time issues on Windows instances.

Amazon Linux

AL2023 and recent versions of Amazon Linux 2 are configured to use the Amazon Time Sync Service IPv4 endpoint by default. If you confirm that your instance is already configured, you can skip the following procedure.

To verify that chrony is configured to use the IPv4 endpoint

Run the following command. In the output, the line that starts with ^* indicates the preferred time source.

```
chronyc sources -v | grep -F ^*
^* 169.254.169.123               3   4   377    14    +12us[+9653ns] +/-  290us
```

To configure chrony to connect to the IPv4 endpoint on older versions of Amazon Linux 2

1. Connect to your instance and uninstall the NTP service.

```
[ec2-user ~]$ sudo yum erase 'ntp*'
```

2. Install the `chrony` package.

```
[ec2-user ~]$ sudo yum install chrony
```

3. Open the `/etc/chrony.conf` file using a text editor (such as **vim** or **nano**). Add the following line before any other `server` or `pool` statements that may be present in the file, and save your changes:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

4. Restart the `chrony` daemon (`chronyd`).

```
[ec2-user ~]$ sudo service chronyd restart
```

```
Starting chronyd:                                          [  OK  ]
```

> **Note**
>
> On RHEL and CentOS (up to version 6), the service name is `chrony` instead of `chronyd`.

5. To configure `chronyd` to start at each system boot, use the `chkconfig` command.

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

6. Verify that `chrony` is using the `169.254.169.123` IPv4 endpoint to synchronize the time.

```
[ec2-user ~]$ chronyc sources -v | grep -F ^*
```

In the output, `^*` indicates the preferred time source.

```
^* 169.254.169.123               3   6    17     43    -30us[ -226us] +/-   287us
```

7. Verify the time synchronization metrics that are reported by chrony.

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID    : A9FEA97B (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed Nov 22 13:18:34 2017
System time     : 0.000000626 seconds slow of NTP time
Last offset     : +0.002852759 seconds
RMS offset      : 0.002852759 seconds
Frequency       : 1.187 ppm fast
Residual freq   : +0.020 ppm
Skew            : 24.388 ppm
Root delay      : 0.000504752 seconds
Root dispersion : 0.001112565 seconds
Update interval : 64.4 seconds
Leap status     : Normal
```

Ubuntu

To configure chrony to connect to the IPv4 endpoint on Ubuntu

1. Connect to your instance and use apt to install the chrony package.

```
ubuntu:~$ sudo apt install chrony
```

Note

If necessary, update your instance first by running sudo apt update.

2. Open the /etc/chrony/chrony.conf file using a text editor (such as **vim** or **nano**). Add the following line before any other server or pool statements that are already present in the file, and save your changes:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

3. Restart the chrony service.

```
ubuntu:~$ sudo /etc/init.d/chrony restart
```

```
Restarting chrony (via systemctl): chrony.service.
```

4. Verify that chrony is using the 169.254.169.123 IPv4 endpoint to synchronize the time.

```
ubuntu:~$ chronyc sources -v | grep -F ^*
```

In the output, the line starting with ^* indicates the preferred time source.

```
^* 169.254.169.123               3   6    17    12    +15us[  +57us] +/-  320us
```

5. Verify the time synchronization metrics that are reported by chrony.

```
ubuntu:~$ chronyc tracking
```

```
Reference ID    : 169.254.169.123 (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed Nov 29 07:41:57 2017
System time     : 0.000000011 seconds slow of NTP time
Last offset     : +0.000041659 seconds
RMS offset      : 0.000041659 seconds
Frequency       : 10.141 ppm slow
Residual freq   : +7.557 ppm
Skew            : 2.329 ppm
Root delay      : 0.000544 seconds
Root dispersion : 0.000631 seconds
Update interval : 2.0 seconds
Leap status     : Normal
```

SUSE Linux

Starting with SUSE Linux Enterprise Server 15, chrony is the default implementation of NTP.

To configure chrony to connect to IPv4 endpoint on SUSE Linux

1. Open the /etc/chrony.conf file using a text editor (such as **vim** or **nano**).
2. Verify that the file contains the following line:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

If this line is not present, add it.

3. Comment out any other server or pool lines.
4. Open YaST and enable the chrony service.

Windows

Starting with the August 2018 release, Windows AMIs use the Amazon Time Sync Service by default. No further configuration is required for instances launched from these AMIs and you can skip the following procedures.

If you're using an AMI that doesn't have the Amazon Time Sync Service configured by default, first verify your current NTP configuration. If your instance is already using the IPv4 endpoint of the Amazon Time Sync Service, no further configuration is required. If your instance is not using the Amazon Time Sync Service, then complete the procedure to change the NTP server to use the Amazon Time Sync Service.

To verify the NTP configuration

1. From your instance, open a Command Prompt window.
2. Get the current NTP configuration by typing the following command:

```
w32tm /query /configuration
```

This command returns the current configuration settings for the Windows instance and will show if you're connected to the Amazon Time Sync Service.

3. (Optional) Get the status of the current configuration by typing the following command:

```
w32tm /query /status
```

This command returns information such as the last time the instance synced with the NTP server and the poll interval.

To change the NTP server to use the Amazon Time Sync Service

1. From the Command Prompt window, run the following command:

```
w32tm /config /manualpeerlist:169.254.169.123 /syncfromflags:manual /update
```

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` IPv4 endpoint.

Default 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 use the Amazon Time Sync Service by default. This time service is accessible from any AWS Region at the 169.254.169.123 IPv4 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 (15 minutes).

Registry path	Key name	Data
HKLM:\System\CurrentControlSet\services\w32time\Config	UpdateInterval	120
HKLM:\System\CurrentControlSet\services\w32time\Parameters	NtpServer	169.254.169.123,0x9
HKLM:\System\CurrentControlSet\services\w32time\Parameters	Type	NTP
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	Enabled	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	InputProvider	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	SpecialPollInterval	900

Connect to the IPv6 endpoint of the Amazon Time Sync Service

This section explains how the steps described in Connect to the IPv4 endpoint of the Amazon Time Sync Service differ if you are configuring your instance to use the local Amazon Time Sync Service through the IPv6 endpoint. It doesn't explain the entire Amazon Time Sync Service configuration process.

The IPv6 endpoint is only accessible on Nitro-based instances.

We don't recommend using both the IPv4 and IPv6 endpoint entries together. The IPv4 and IPv6 NTP packets come from the same local server for your instance. Configuring both IPv4 and IPv6 endpoints is unnecessary and will not improve the accuracy of the time on your instance.

Linux

Depending on the Linux distribution you're using, when you reach the step to edit the `chrony.conf` file, you'll be using the IPv6 endpoint of the Amazon Time Sync Service (`fd00:ec2::123`) rather than the IPv4 endpoint (`169.254.169.123`):

```
server fd00:ec2::123 prefer iburst minpoll 4 maxpoll 4
```

Save the file and verify that chrony is using the `fd00:ec2::123` IPv6 endpoint to synchronize time:

```
[ec2-user ~]$ chronyc sources -v
```

In the output, if you see the `fd00:ec2::123` IPv6 endpoint, the configuration is complete.

Windows

When you reach the step to change the NTP server to use the Amazon Time Sync Service, you'll be using the IPv6 endpoint of the Amazon Time Sync Service (`fd00:ec2::123`) rather than the IPv4 endpoint (`169.254.169.123`):

```
w32tm /config /manualpeerlist:fd00:ec2::123 /syncfromflags:manual /update
```

Verify that your new settings are using the `fd00:ec2::123` IPv6 endpoint to synchronize time:

```
w32tm /query /configuration
```

In the output, verify that `NtpServer` displays the `fd00:ec2::123` IPv6 endpoint.

Connect to the PTP hardware clock

The PTP hardware clock is part of the [AWS Nitro System], so it is directly accessible on [supported bare metal and virtualized EC2 instances] without using any customer resources.

The NTP endpoints for the PTP hardware clock are the same as those for the regular Amazon Time Sync Service. If your instance has a PTP hardware clock and you configured the NTP connection (to either the IPv4 or IPv6 endpoint), your instance time is automatically sourced from the PTP hardware clock over NTP.

For Linux instances, you can configure a *direct* PTP connection, which will give you more accurate time than the NTP connection. Windows instances only support an NTP connection to the PTP hardware clock.

Requirements

The PTP hardware clock is available on an instance when the following requirements are met:

- Supported AWS Regions: US East (N. Virginia), US East (Ohio), Asia Pacific (Malaysia), Asia Pacific (Thailand), Asia Pacific (Tokyo), and Europe (Stockholm)
- Supported Local Zones: US East (New York City)
- Supported instance families:
  - **General purpose:** M7a, M7g, M7i
  - **Memory optimized:** R7a, R7g, R7i
  - **Storage optimized:** I8g, I8ge
- (Linux only) ENA driver version 2.10.0 or later installed on a supported operating system. For more information about supported operating systems, see the driver prerequisites on *GitHub*.

(Linux only) Configure a direct PTP connection to the PTP hardware clock

This section describes how to configure your Linux instance to use the local Amazon Time Sync Service through the PTP hardware clock using a direct PTP connection. It requires adding a server entry for the PTP hardware clock in the `chrony` configuration file.

To configure a direct PTP connection to the PTP hardware clock (Linux instances only)

1. **Install prerequisites**

   Connect to your Linux instance and do the following:

   a. Install the Linux kernel driver for Elastic Network Adapter (ENA) version 2.10.0 or later.
   b. Enable the PTP hardware clock.

   For the installation instructions, see Linux kernel driver for Elastic Network Adapter (ENA) family on *GitHub*.

2. **Verify ENA PTP device**

   Verify that the ENA PTP hardware clock device shows up on your instance.

```
[ec2-user ~]$ for file in /sys/class/ptp/*; do echo -n "$file: "; cat "$file/clock_name"; done
```

Expected output

```
/sys/class/ptp/ptp<index>: ena-ptp-<PCI slot>
```

Where:

- *index* is the kernel-registered PTP hardware clock index.
- *PCI slot* is the ENA ethernet controller PCI slot. This is the same slot as shown in `lspci | grep ENA`.

Example output

```
/sys/class/ptp/ptp0: ena-ptp-05
```

If `ena-ptp-<PCI slot>` is not in the output, the ENA driver was not correctly installed. Review step 1 in this procedure for installing the driver.

3. **Configure PTP symlink**

   PTP devices are typically named `/dev/ptp0`, `/dev/ptp1`, and so on, with their index depending on the hardware initialization order. Creating a symlink ensures that applications like chrony consistently reference the correct device, regardless of index changes.

   The latest Amazon Linux 2023 AMIs include a `udev` rule that creates the `/dev/ptp_ena` symlink, pointing to the correct `/dev/ptp` entry associated with the ENA host.

   First check if the symlink is present by running the following command.

```
[ec2-user ~]$ ls -l /dev/ptp*
```

   Example output

```
crw------- 1 root root 245, 0 Jan 31 2025 /dev/ptp0
lrwxrwxrwx 1 root root      4 Jan 31 2025 /dev/ptp_ena -> ptp0
```

Where:

- /dev/ptp*<index>* is the path to the PTP device.
- /dev/ptp_ena is the constant symlink, which points to the same PTP device.

If the /dev/ptp_ena symlink is present, skip to Step 4 in this procedure. If it's missing, do the following:

a. Add the following udev rule.

```
[ec2-user ~]$ echo "SUBSYSTEM==\"ptp\", ATTR{clock_name}==\"ena-ptp-*\",
 SYMLINK += \"ptp_ena\"" | sudo tee -a /etc/udev/rules.d/53-ec2-network-
interfaces.rules
```

b. Reload the udev rule, either by rebooting the instance, or by running the following command.

```
[ec2-user ~]$ sudo udevadm control --reload-rules && udevadm trigger
```

4. **Configure chrony**

chrony must be configured to use the /dev/ptp_ena symlink instead of directly referencing /dev/ptp*<index>*.

a. Edit /etc/chrony.conf using a text editor and add the following line anywhere in the file.

```
refclock PHC /dev/ptp_ena poll 0 delay 0.000010 prefer
```

b. Restart chrony.

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

5. **Verify chrony configuration**

Verify that chrony is using the PTP hardware clock to synchronize the time on this instance.

```
[ec2-user ~]$ chronyc sources
```

Expected output

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
#* PHC0                          0   0   377    1   +2ns[ +1ns] +/-   5031ns
```

In the output that's returned, * indicates the preferred time source. PHC0 corresponds to the PTP hardware clock. You might need to wait a few seconds after restarting chrony for the asterisk to appear.

Set the time reference on your EC2 instance or any internet-connected device to use the public Amazon Time Sync Service

You can set your instance, or any internet-connected device such as your local computer or an on-prem server, to use the public Amazon Time Sync Service, which is accessible over the internet at `time.aws.com`. You can use the public Amazon Time Sync Service as a backup for the local Amazon Time Sync Service and to connect resources outside of AWS to the Amazon Time Sync Service.

> **Note**
>
> For the best performance, we recommend using the *local* Amazon Time Sync Service on your instances, and only using the *public* Amazon Time Sync Service as a backup.

Use the instructions for the operating system of your instance or device.

Linux

To set your Linux instance or device to use the public Amazon Time Sync Service using chrony or ntpd

1. Edit `/etc/chrony.conf` (if you use chrony) or `/etc/ntp.conf` (if you use ntpd) using a text editor as follows:

a. To prevent your instance or device from trying to mix smeared and non-smeared servers, remove or comment out lines starting with `server` except any existing connection to the local Amazon Time Sync Service.

> **Important**
>
> If you're setting your EC2 instance to connect to the public Amazon Time Sync Service, do not remove the following line which sets your instance to connect to the local Amazon Time Sync Service. The local Amazon Time Sync Service is a more direct connection and will provide better clock accuracy. The public Amazon Time Sync Service should only be used as a backup.
>
> ```
> server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
> ```

b. Add the following line to connect to the public Amazon Time Sync Service.

```
pool time.aws.com iburst
```

2. Restart the daemon using one of the following commands.

   - chrony

   ```
   sudo service chronyd force-reload
   ```

   - ntpd

   ```
   sudo service ntp reload
   ```

macOS

To set your macOS instance or device to use the public Amazon Time Sync Service

1. Open **System Preferences**.
2. Choose **Date & Time**, and then choose the **Date & Time** tab.
3. To make changes, choose the lock icon, and enter your password when prompted.
4. For **Set date and time automatically**, enter **`time.aws.com`**.

Windows

To set your Windows instance or device to use the public Amazon Time Sync Service

1. Open the **Control Panel**.
2. Choose the **Date and Time** icon.
3. Choose the **Internet Time** tab. This tab is not be available if your PC is part of a domain. In this case, it will synchronize time with the domain controller. You can configure the controller to use the public Amazon Time Sync Service.
4. Choose **Change settings**.
5. Select the checkbox for **Synchronize with an Internet time server**.
6. Next to **Server**, enter **`time.aws.com`**.

To set your Windows Server instance or device to use the public Amazon Time Sync Service

- Follow Microsoft's instructions to update your registry.

Compare timestamps for your Linux instances

If you're using the Amazon Time Sync Service, you can compare the timestamps on your Amazon EC2 Linux instances with ClockBound to determine the true time of an event. ClockBound measures the clock accuracy of your EC2 instance, and allows you to check if a given timestamp is in the past or future with respect to your instance's current clock. This information is valuable for determining the order and consistency of events and transactions across EC2 instances, independent of each instance's geographic location.

ClockBound is an open source daemon and library. To learn more about ClockBound, including installation instructions, see ClockBound on *GitHub*.

ClockBound is only supported for Linux instances.

If you're using the direct PTP connection to the PTP hardware clock, your time daemon, such as chrony, will underestimate the clock error bound. This is because a PTP hardware clock does not pass the correct error bound information to chrony, the way that NTP does. As a result, your clock synchronization daemon assumes the clock is accurate to UTC and thus has an error bound of `0`. To measure the full error bound, the Nitro System calculates the error bound of the PTP hardware clock, and makes it available to your EC2 instance over the ENA driver `sysfs` filesystem. You can read this directly as a value, in nanoseconds.

To retrieve the PTP hardware clock error bound

1. First get the correct location of the PTP hardware clock device by using one of the following commands. The path in the command is different depending on the AMI used to launch the instance.

   - For Amazon Linux 2:

     ```
     cat /sys/class/net/eth0/device/uevent | grep PCI_SLOT_NAME
     ```

   - For Amazon Linux 2023:

     ```
     cat /sys/class/net/ens5/device/uevent | grep PCI_SLOT_NAME
     ```

   The output is the PCI slot name, which is the location of the PTP hardware clock device. In this example, the location is `0000:00:03.0`.

   ```
   PCI_SLOT_NAME=0000:00:03.0
   ```

2. To retrieve the PTP hardware clock error bound, run the following command. Include the PCI slot name from the previous step.

   ```
   cat /sys/bus/pci/devices/0000:00:03.0/phc_error_bound
   ```

   The output is the clock error bound of the PTP hardware clock, in nanoseconds.

To calculate the correct clock error bound at a specific point in time when using the direct PTP connection to the PTP hardware clock, you must add the clock error bound from chrony or ClockBound at the time that chrony polls the PTP hardware clock. For more information about measuring and monitoring clock accuracy, see Manage Amazon EC2 instance clock accuracy using Amazon Time Sync Service and Amazon CloudWatch – Part 1.

Change the time zone of your instance

Amazon EC2 instances are set to the UTC (Coordinated Universal Time) time zone by default. You can change the time on an instance to the local time zone or to another time zone in your network.

Use the instructions for your instance's operating system.

Linux

> **Important**
>
> This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To change the time zone on Amazon Linux

1. View the system's current time zone setting.

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

2. List the available time zones.

   ```
   [ec2-user ~]$ timedatectl list-timezones
   ```

3. Set the chosen time zone.

   ```
   [ec2-user ~]$ sudo timedatectl set-timezone America/Vancouver
   ```

4. (Optional) Confirm that the current time zone is updated to the new time zone by running the **timedatectl** command again.

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

Windows

To change the time zone on a Windows 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 in the following format:

```
display name
time zone ID
```

3. Locate the time zone ID to assign to the instance.
4. Example: Assign the UTC time zone:

```
tzutil /s "UTC"
```

   Example: Assign Pacific Standard Time:

```
tzutil /s "Pacific Standard Time"
```

When you change the time zone on a Windows instance, you must ensure that the time zone persists through system restarts. Otherwise, when the instance restarts, it reverts back to using UTC time. You can persist your time zone 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 3 in the following procedure. If the key is not set, follow 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. (Optional) Verify that the instance saved the key successfully using the following command:

```
reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
\TimeZoneInformation" /s
```

This command returns the subkeys for the **TimeZoneInformation** registry key. You should see the **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      0xffffffc4
    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
```

Manage device drivers for your EC2 instance

Device drivers are software components that communicate with the virtualized hardware for your Amazon EC2 instance. To prevent system errors, performance issues, and other unexpected behavior, it's important to keep your drivers up-to-date. That's especially true for drivers that can have a strong impact on system performance depending on your usage, such as networking, graphics, and storage device drivers. New driver releases can include defect fixes or introduce expanded functionality that you might want to take advantage of for instances that are currently running.

Network drivers

Linux distributions can incorporate networking features like Elastic Network Adapter (ENA) or Elastic Fabric Adapter (EFA) within the kernel. However, the timing may vary for implementation of kernel driver features within the different distributions.

ENA and EFA Linux kernel drivers are available from the Amazon Drivers GitHub repository. For more information and links to the available drivers, see [Amazon Drivers](#) on GitHub.

For more information about ENA drivers, see Enable enhanced networking with ENA on your EC2 instances. For more information about EFA drivers, see **Getting started** topics in the Elastic Fabric Adapter for AI/ML and HPC workloads on Amazon EC2 section of this guide.

To install or update networking drivers on Windows instances, see the following topics:

- Install the ENA driver on Windows
- Install the latest AWS PV drivers

  For more information, see Paravirtual drivers for Windows instances.

Note

EFA is not supported on Windows instances.

Graphics drivers

To install or update graphics drivers, see the following topics:

- AMD drivers for your EC2 instance
- NVIDIA drivers for your Amazon EC2 instance

Storage device drivers

To install or update storage drivers, see the following topics:

- For Linux instances, see Install or upgrade the NVMe driver in the *Amazon EBS User Guide*.
- For Windows instances, see AWS NVMe drivers.

AMD drivers for your EC2 instance

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 NVIDIA drivers instead.

Contents

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
- AMD Advanced Media Framework
- Video Acceleration API
- DirectX 9 and later
- Microsoft Hardware Media Foundation Transform

AMIs with the AMD driver installed

AWS offers different Amazon Machine Images (AMIs) 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. Only the following operating system versions support AMD drivers:

- Amazon Linux 2 with kernel version 5.4
- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04
- Windows Server 2016

- Windows Server 2019
- Windows Server 2022

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 End User License Agreement.

Install the AMD driver on your Amazon Linux 2 Linux instance

1. Connect to your Linux instance.
2. Install the AWS CLI on your Linux instance and configure default credentials. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*.

   > **Important**
   >
   > Your user or role must have the permissions granted that contains the **AmazonS3ReadOnlyAccess** policy. For more information, see AWS managed policy: AmazonS3ReadOnlyAccess in the *Amazon Simple Storage Service User Guide*.

3. Install kernel 5.4

   ```
   $ sudo amazon-linux-extras disable kernel-5.10
   $ sudo amazon-linux-extras enable kernel-5.4
   $ sudo yum install -y kernel
   ```

4. Install **gcc** and **make**, if they are not already installed.

   ```
   $ sudo yum install gcc make
   ```

5. Update your package cache and get the package updates for your instance.

   ```
   $ sudo amazon-linux-extras install epel -y
   $ sudo yum update -y
   ```

6. Reboot the instance.

   ```
   $ sudo reboot
   ```

7. Reconnect to the instance after it reboots.
8. Download the latest AMD driver.

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

9. Extract the file.

```
$ tar -xf amdgpu-pro-*rhel*.tar.xz
```

10. Change to the folder for the extracted driver.
11. Run the self install script to install the full graphics stack.

```
$ ./amdgpu-pro-install -y --opencl=pal,legacy
```

12. Reboot the instance.

```
$ sudo reboot
```

13. Confirm that the driver is functional.

```
$ sudo dmesg | grep amdgpu
```

The response should look like the following:

```
Initialized amdgpu
```

Install the AMD driver on your Ubuntu Linux instance

1. Connect to your Linux instance.
2. Update your package cache and get the package updates for your instance.

```
$ sudo apt-get update --fix-missing && sudo apt-get upgrade -y
```

3. Install **gcc** and **make**, if they are not already installed.

```
$ sudo apt install build-essential -y
```

4. Install Linux firmware and kernel modules

```
$ sudo apt install linux-firmware linux-modules-extra-aws -y
```

5. Reboot instance

```
$ sudo reboot
```

6. Reconnect to the instance after it reboots.
7. Install the AMD Linux driver package
   - For Ubuntu 20.04:

   ```
   $ wget https://repo.radeon.com/.preview/afe3e25b8f1beff0bb312e27924d63b5/amdgpu-install/5.4.02.01/ubuntu/focal/amdgpu-install_5.4.02.01.50402-1_all.deb
   $ sudo dpkg --add-architecture i386
   $ sudo apt install ./amdgpu-install_5.4.02.01.50402-1_all.deb
   ```

   - For later Ubuntu versions go to Linux® Drivers for AMD Radeon™ Graphics and download the latest Ubuntu package and install it.

   ```
   $ sudo apt install ./amdgpu-install_{version-you-downloaded}.deb
   ```

8. Run the self install script to install the full graphics stack.

```
$ amdgpu-install --usecase=workstation --vulkan=pro -y
```

9. Reboot the instance.

```
$ sudo reboot
```

10. Confirm that the driver is functional.

```
$ sudo dmesg | grep amdgpu
```

The response should look like the following:

```
Initialized amdgpu
```

Install the AMD driver on your Windows instance

1. Connect to your Windows instance and open a PowerShell window.
2. Configure default credentials for the AWS Tools for Windows PowerShell on your Windows instance. For more information, see Getting Started with the AWS Tools for Windows PowerShell in the *AWS Tools for PowerShell User Guide*.

   Important

   Your user or role must have the permissions granted that contains the **AmazonS3ReadOnlyAccess** policy. For more information, see AWS managed policy: AmazonS3ReadOnlyAccess in the *Amazon Simple Storage Service User Guide*.

3. Set the key prefix according to your version of Windows:

   - Windows 10 and Windows 11

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS10"
   ```

   - Windows Server 2016

   ```
   $KeyPrefix = "archives"
   ```

   - Windows Server 2019

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS_2K19" # use "archives" for Windows Server 2016
   ```

   - Windows Server 2022

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS_2K22"
   ```

4. Download the drivers from Amazon S3 to your desktop using the following PowerShell commands.

```
$Bucket = "ec2-amd-windows-drivers"
$LocalPath = "$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
    }
}
```

5. Unzip the downloaded driver file and run the installer using the following PowerShell commands.

```
Expand-Archive $LocalFilePath -DestinationPath "$home\Desktop\AMD\$KeyPrefix" -
Verbose
```

   Now, check the content of the new directory. The directory name can be retrieved using the `Get-ChildItem` PowerShell command.

```
Get-ChildItem "$home\Desktop\AMD\$KeyPrefix"
```

   The output should be similar to the following:

```
Directory: C:\Users\Administrator\Desktop\AMD\latest

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       10/13/2021  12:52 AM                210414a-365562C-Retail_End_User.2
```

   Install the drivers:

```
pnputil /add-driver $home\Desktop\AMD\$KeyPrefix\*.inf /install /subdirs
```

6. Follow the instructions to install the driver and reboot your instance as required.
7. To verify that the GPU is working properly, check Device Manager. You should see "AMD Radeon Pro V520 MxGPU" listed as a display adapter.
8. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol, [Amazon DCV](https://docs.aws.amazon.com/dcv/).

NVIDIA drivers for your Amazon EC2 instance

An instance with an attached NVIDIA GPU, such as a P- or G- series instance types, 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 AWS AMI with the driver pre-installed.

To install AMD drivers on an instance with an attached AMD GPU, such as a G4ad instance, see AMD drivers instead.

Contents

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, GRID, and gaming drivers

- OpenCL, OpenGL, and Vulkan
- NVIDIA CUDA and related libraries (for example, cuDNN, TensorRT, nvJPEG, and cuBLAS)
- NVENC for video encoding and NVDEC for video decoding
- Windows-only APIs: DirectX, Direct2D, DirectX Video Acceleration, DirectX Raytracing

Available drivers by instance type

The following table summarizes the supported NVIDIA drivers for each GPU instance type.

Instance type	Tesla driver	GRID driver	Gaming driver
G3	Yes	Yes	No
G4dn	Yes	Yes	Yes
G5	Yes	Yes	Yes
G5g	Yes [1]	No	No

Instance type	Tesla driver	GRID driver	Gaming driver
G6	Yes	Yes	Yes
G6e	Yes	Yes	Yes
G6f	No	Yes	No
Gr6	Yes	Yes	No
Gr6f	No	Yes	No
P2	Yes	No	No
P3	Yes	No	No
P4d	Yes	No	No
P4de	Yes	No	No
P5	Yes	No	No
P5e	Yes	No	No
P5en	Yes	No	No
P6-B200	Yes	No	No
P6e-GB200	Yes	No	No

[1] This Tesla driver also supports optimized graphics applications specific to the ARM64 platform

Installation options

Use one of the following options to get the NVIDIA drivers required for your GPU instance.

Options

- Option 1: AMIs with NVIDIA drivers installed
- Option 2: Public NVIDIA drivers
- Option 3: GRID drivers (G6, Gr6, G6e, G6f, Gr6f, G5, G4dn, and G3 instances)

- Option 4: NVIDIA gaming drivers (G6, G6e, G5, and G4dn instances)

Option 1: AMIs with NVIDIA drivers installed

AWS and NVIDIA offer different Amazon Machine Images (AMIs) that come with the NVIDIA drivers installed.

- Marketplace offerings with the Tesla driver
- Marketplace offerings with the GRID driver
- Marketplace offerings with the Gaming driver

To review considerations that are dependent on your operating system (OS) platform, choose the tab that applies to your AMI.

Linux

To update the driver version installed using one of these AMIs, you must uninstall the NVIDIA packages from your instance to avoid version conflicts. Use this command to uninstall the NVIDIA packages:

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

The CUDA toolkit package has dependencies on the NVIDIA drivers. Uninstalling the NVIDIA packages erases the CUDA toolkit. You must reinstall the CUDA toolkit after installing the NVIDIA driver.

Windows

If you create a custom Windows AMI using one of the AWS Marketplace offerings, the AMI must be a standardized image created with Windows Sysprep to ensure that the GRID driver works. For more information, see Create an Amazon EC2 AMI using Windows Sysprep.

Option 2: Public NVIDIA drivers

If the AWS Marketplace AMIs described in Option 1: AMIs with NVIDIA drivers installed don't fit your use case, you can install the public drivers and bring your own license. To install a public driver, download it from the NVIDIA site as described in this section.

Step 1: Download a public NVIDIA driver

Log on to your instance and download the 64-bit NVIDIA driver appropriate for the instance type from http://www.nvidia.com/Download/Find.aspx. For **Product Type**, **Product Series**, and **Product**, use the options in the following table.

Instance	Product type	Product series	Product	Minimum driver version
G3	Tesla	M-Class	M60	--
G4dn	Tesla	T-Series	T4	--
G5	Tesla	A-Series	A10	470.00 or later
G5g[1]	Tesla	T-Series	NVIDIA T4G	470.82.01 or later
G6	Tesla	L-Series	L4	525.0 or later
G6e	Tesla	L-Series	L40S	535.0 or later
Gr6	Tesla	L-Series	L4	525.0 or later
P2	Tesla	K-Series	K80	--
P3	Tesla	V-Series	V100	--
P4d	Tesla	A-Series	A100	--
P4de	Tesla	A-Series	A100	--
P5	Tesla	H-Series	H100	530 or later
P5e	Tesla	H-Series	H200	550 or later
P5en	Tesla	H-Series	H200	550 or later
P6-B200[2]	Tesla	HGX-Series	B200	570 or later
P6e-GB200	Tesla	HGX-Series	B200	570 or later

[1] The operating system for G5g instances is Linux aarch64.

[2] For P6-B200 instance types, there are additional installation requirements to configure NVIDIA Fabric Manager.

Step 2: Install the public driver

Follow the install instructions for your instance type operating system.

Linux

1. To install the NVIDIA Tesla driver on a Linux distribution, see the NVIDIA Driver Installation Guide.
2. P6-B200 instance types require installation and configuration of additional packages that come bundled with the NVIDIA CUDA Toolkit. For more information, see Configure Fabric Manager for P6-B200 instance types.

Windows

To install the NVIDIA driver on Windows, follow these steps:

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 display adapter named **Microsoft Basic Display Adapter** that is marked with a warning icon using Device Manager. Install these Windows features: **Media Foundation** and **Quality Windows Audio Video Experience**.

   > **Important**
   >
   > Don't disable the display adapter named **Microsoft Remote Display Adapter**. If **Microsoft Remote Display Adapter** is disabled your connection might be interrupted and attempts to connect to the instance after it has rebooted might fail.

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 on Amazon EC2 instances.

Configure Fabric Manager for P6-B200 instance types

The P6-B200 platform is unique in that it exposes Mellanox ConnectX 7 network interface cards (NICs) to the instance as PCIe devices. These CX7 NICs do not act as typical network interfaces but instead function as NVSwitch bridges providing a control path to initialize and configure the NVFabric, which is the NVLink topology of the GPU interconnect.

To fully initialize the system, the NVIDIA Fabric Manager must configure `NVFabric` and establish the NVSwitch topology. This enables InfiniBand kernel modules to communicate with the CX7 devices.

Installation and configuration steps vary slightly depending on the instance operating system. Follow the instructions on the *NVIDIA Developer* website to get the [CUDA Toolkit]() as either a local or network installer. package for your instance. After you've installed the toolkit, configure NVIDIA Fabric Manager as follows:

Amazon Linux 2023

1. Install NVIDIA Link Subnet Manager and `ibstat`.

```
sudo dnf install nvlink5
```

2. Enable automatic loading of the Infiniband module on startup.

```
echo "ib_umad" | sudo tee -a /etc/modules-load.d/modules.conf
```

3. Enable the NVIDIA Fabric Manager service on startup.

```
sudo dnf install nvidia-fabric-manager
sudo systemctl start nvidia-fabricmanager.service
```

4. Reboot the instance.

```
sudo reboot
```

Ubuntu

1. Install the `mlx5_ib` kernel module.

```
sudo apt install linux-modules-extra-$(uname -r) -y
```

2. Install NVIDIA Link Subnet Manager.

```
sudo apt install nvlsm
```

3. Install `ibstat`.

```
sudo apt install infiniband-diags
```

4. Enable automatic loading of the Infiniband module on startup.

```
echo "ib_umad" | sudo tee -a /etc/modules-load.d/modules.conf
```

5. Enable the NVIDIA Fabric Manager service on startup.

```
sudo apt install nvidia-fabricmanager
sudo systemctl enable nvidia-fabricmanager
```

6. Reboot the instance.

```
sudo reboot
```

Option 3: GRID drivers (G6, Gr6, G6e, G6f, Gr6f, G5, G4dn, and G3 instances)

These downloads are available to AWS customers only. By downloading, in order to adhere to requirements of the AWS solution referred to in the NVIDIA GRID Cloud End User License Agreement (EULA), you agree to use the downloaded software only to develop AMIs for use with the NVIDIA L4, NVIDIA L40S, NVIDIA A10G, NVIDIA Tesla T4, or NVIDIA Tesla M60 hardware. You can use the GRID drivers to both create and use AMIs within the AWS environment. Upon installation of the software, you are bound by the terms of the NVIDIA GRID Cloud End User License Agreement. For information about the version of the NVIDIA GRID driver for your operating system, see the NVIDIA Virtual GPU (vGPU) Software on the NVIDIA website.

Considerations

- G6f and Gr6f instances require GRID 18.4 or later.
- G6e instances require GRID 17.4 or later.
- G6 and Gr6 instances require GRID 17.1 or later.
- G5 instances require GRID 13.1 or later (or GRID 12.4 or later).

- G3 instances require AWS provided DNS resolution for GRID licensing to work.
- IMDSv2 is only supported with NVIDIA driver version 14.0 or greater.
- For Windows instances, if you launch your instance from a custom Windows AMI, the AMI must be a standardized image created with Windows Sysprep to ensure that the GRID driver works. For more information, see Create an Amazon EC2 AMI using Windows Sysprep.
- GRID 17.0 and later do not support Windows Server 2019.
- GRID 14.2 and later do not support Windows Server 2016.
- GRID 17.0 and later is not supported with G3 instances.
- For Linux instances, you might need to install or update packages, such as gcc, if the NVIDIA installer fails with an error message. The specifics depend on the versions of the operating system and the kernel. For more information, see the NVIDIA Enterprise Support Portal.

Prerequisites

- (Linux) Verify that the AWS CLI is installed on your instance and configured with default credentials. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*.
- (Windows) Configure default credentials for the AWS Tools for Windows PowerShell on your instance. For more information, see Getting started with the AWS Tools for Windows PowerShell in the *AWS Tools for PowerShell User Guide*.
- Your user or role must have the permissions granted that contains the **AmazonS3ReadOnlyAccess** policy.

Amazon Linux 2023

To install the NVIDIA GRID driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

   ```
   [ec2-user ~]$ sudo dnf install gcc make
   ```

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers packages.

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra
```

6. Download the GRID driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

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

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

   When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

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

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. If you are using NVIDIA vGPU software version 14.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the [NVIDIA documentation](#).

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. Reboot the instance.

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

12. (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 Amazon 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 on your Amazon EC2 GPU-based instances.

Amazon Linux 2

To install the NVIDIA GRID driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

```
[ec2-user ~]$ sudo yum install gcc make
```

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel that is running.

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. Download the GRID driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

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

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

   If you are using Amazon Linux 2 with kernel version 5.10, use the following command to install the GRID driver.

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

   When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

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

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. If you are using NVIDIA vGPU software version 14.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. Reboot the instance.

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

12. (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 Amazon 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 on your Amazon EC2 GPU-based instances.

CentOS 7 and Red Hat Enterprise Linux 7

To install the NVIDIA GRID driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

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

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel that you are running.

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. Disable the nouveau open source driver for NVIDIA graphics cards.

   a. Add nouveau to the /etc/modprobe.d/blacklist.conf blacklist file. Copy the following code block and paste it into a terminal.

   ```
   [ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   b. Edit the /etc/default/grub file and add the following line:

   ```
   GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
   ```

   c. Rebuild the Grub configuration.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

7. Download the GRID driver installation utility using the following command:

   ```
   [ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
   ```

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

   ```
   [ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
   ```

8. Add permissions to run the driver installation utility using the following command.

   ```
   [ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
   ```

9. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

   ```
   [ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
   ```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

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

```
[ec2-user ~]$ nvidia-smi -q | head
```

11. If you are using NVIDIA vGPU software version 14.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

12. Reboot the instance.

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

13. (Optional) Depending on your use case, you might complete the following optional steps. If you do not require this functionality, do not complete these steps.

    a. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol Amazon 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 on your Amazon EC2 GPU-based instances.

    c. Install the GUI desktop/workstation package.

    ```
    [ec2-user ~]$ sudo yum groupinstall -y "Server with GUI"
    ```

CentOS Stream 8 and Red Hat Enterprise Linux 8

To install the NVIDIA GRID driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

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

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel that you are running.

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. Download the GRID driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

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

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

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

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. If you are using NVIDIA vGPU software version 14.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. Reboot the instance.

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

12. (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 Amazon 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 on your Amazon EC2 GPU-based instances.

    c. Install the GUI workstation package.

    ```
    [ec2-user ~]$ sudo dnf groupinstall -y workstation
    ```

Rocky Linux 8

To install the NVIDIA GRID driver on your Linux instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

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

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel that you are running.

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. Download the GRID driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

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

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

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

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. If you are using NVIDIA vGPU software version 14.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. Reboot the instance.

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

12. (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 Amazon 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 on your Amazon EC2 GPU-based instances.

Ubuntu and Debian

To install the NVIDIA GRID driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

```
[ec2-user ~]$ sudo apt-get install -y gcc make
```

3. (Ubuntu) Upgrade the `linux-aws` package to receive the latest version.

```
$ sudo apt-get upgrade -y linux-aws
```

(Debian) Upgrade package to receive the latest version.

```
$ sudo apt-get upgrade -y
```

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

```
$ sudo reboot
```

5. Reconnect to your instance after it has rebooted.
6. Install the kernel headers package for the version of the kernel you are currently running.

```
$ sudo apt-get install -y linux-headers-$(uname -r) linux-modules-extra-$(uname -r)
```

7. Disable the `nouveau` open source driver for NVIDIA graphics cards.

   a. Add `nouveau` to the `/etc/modprobe.d/blacklist.conf` blacklist file. Copy the following code block and paste it into a terminal.

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   b. Edit the `/etc/default/grub` file and add the following line:

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

c. Rebuild the Grub configuration.

```
$ sudo update-grub
```

8. Download the GRID driver installation utility using the following command:

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

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

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

9. Add permissions to run the driver installation utility using the following command.

```
$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

   When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

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

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

12. If you are using NVIDIA vGPU software version 14.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

13. Reboot the instance.

```
$ sudo reboot
```

14. (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 Amazon 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 on your Amazon EC2 GPU-based instances.

    c. Install the GUI desktop/workstation package.

    ```
    $ sudo apt-get install -y lightdm ubuntu-desktop
    ```

Windows operating systems

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.

```
$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 Windows versions in the bucket by removing the `-KeyPrefix $KeyPrefix` option. For information about the version of the NVIDIA GRID driver for your operating system, see the [NVIDIA Virtual GPU (vGPU) Software] on the NVIDIA website.

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`.

G5 instances require GRID 13.1 or later (or GRID 12.4 or later).

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

Run the following PowerShell commands to create the registry value to disable the licensing page in the control panel. The AWS Tools for PowerShell in AWS Windows AMIs defaults to the 32-bit version and this command fails. Instead, use the 64-bit version of PowerShell included with the operating system.

```
New-Item -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name GridLicensing
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" -
Name "NvCplDisableManageLicensePage" -PropertyType "DWord" -Value "1"
```

Command Prompt

Run the following registry command to create the registry value to disable the licensing page in the control panel. You can run it using the Command Prompt window or a 64-bit version of PowerShell.

```
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" /v
 NvCplDisableManageLicensePage /t REG_DWORD /d 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, Amazon 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 on your Amazon EC2 GPU-based instances.

Option 4: NVIDIA gaming drivers (G6, G6e, G5, and 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 L4, NVIDIA L40S, NVIDIA A10G, NVIDIA Tesla T4, or NVIDIA Tesla M60 hardware. You can use the GRID drivers to both create and use AMIs within the AWS environment. Upon installation of the software, you are bound by the terms of the NVIDIA GRID Cloud End User License Agreement.

Considerations

- G3 instances require AWS provided DNS resolution for GRID licensing to work.
- IMDSv2 is only supported with NVIDIA driver version 495.x or greater.

Prerequisites

- (Linux) Verify that the AWS CLI is installed on your instance and configured with default credentials. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*.
- Your user or role must have the permissions granted that contains the **AmazonS3ReadOnlyAccess** policy.

Amazon Linux 2023

To install the NVIDIA gaming driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

```
[ec2-user ~]$ sudo dnf install gcc make
```

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

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

4. Reconnect to your instance after it is rebooted.
5. Install the kernel headers packages.

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra kernel-devel-
$(uname -r) kernel-headers-$(uname -r) dkms
```

6. Download the gaming driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. Extract the gaming driver installation utility from the downloaded `.zip` archive.

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. Add permissions to run the driver installation utility using the following command:

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

10. Use the following command to create the required configuration file.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
 vGamingMarketplace=2
 EOF
```

11. Use the following command to download and rename the certification file.

    - For version 460.39 or later:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
    ```

    - For version 440.68 to 445.48:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
    ```

    - For earlier versions:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
    ```

12. If you are using NVIDIA driver version 510.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. Reboot the instance.

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

14. Verify the NVIDIA Gaming license using the following command.

```
[ec2-user ~]$ nvidia-smi.exe -q
```

In the output, search for `vGPU Software Licensed Product`.

15. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol [Amazon DCV](https://docs.aws.amazon.com/dcv/).

Amazon Linux 2

To install the NVIDIA gaming driver on your instance

1. Connect to your instance. Update your package cache and get the package updates for your instance.

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

2. Install **gcc** and **make**, if they are not already installed.

```
[ec2-user ~]$ sudo yum install gcc make
```

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

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

4. Reconnect to your instance after it is rebooted.
5. Install the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. Download the gaming driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

   Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. Extract the gaming driver installation utility from the downloaded `.zip` archive.

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. Add permissions to run the driver installation utility using the following command:

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

If you are using Amazon Linux 2 with kernel version 5.10, use the following command to install the NVIDIA gaming drivers.

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

10. Use the following command to create the required configuration file.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. Use the following command to download and rename the certification file.

- For version 460.39 or later:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- For version 440.68 to 445.48:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- For earlier versions:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. If you are using NVIDIA driver version 510.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. Reboot the instance.

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

14. Verify the NVIDIA Gaming license using the following command.

```
[ec2-user ~]$ nvidia-smi.exe -q
```

    In the output, search for `vGPU Software Licensed Product`.

15. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol Amazon DCV.

CentOS 7 and Red Hat Enterprise Linux 7

To install the NVIDIA gaming driver on your instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.

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

2. Update your package cache and get the package updates for your instance.

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

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. Disable the `nouveau` open source driver for NVIDIA graphics cards.

   a. Add `nouveau` to the `/etc/modprobe.d/blacklist.conf` blacklist file. Copy the following code block and paste it into a terminal.

      ```
      [ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
      blacklist vga16fb
      blacklist nouveau
      blacklist rivafb
      blacklist nvidiafb
      blacklist rivatv
      EOF
      ```

   b. Edit the `/etc/default/grub` file and add the following line:

      ```
      GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
      ```

   c. Rebuild the Grub configuration.

      ```
      [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
      ```

7. Download the gaming driver installation utility using the following command:

   ```
   [ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
   ```

   Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

   ```
   [ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
   ```

8. Extract the gaming driver installation utility from the downloaded `.zip` archive.

   ```
   [ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
   ```

9. Add permissions to run the driver installation utility using the following command:

   ```
   [ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
   ```

10. Run the installer using the following command:

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

11. Use the following command to create the required configuration file.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

12. Use the following command to download and rename the certification file.

    - For version 460.39 or later:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
    ```

    - For version 440.68 to 445.48:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
    ```

    - For earlier versions:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
    ```

13. If you are using NVIDIA driver version 510.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
```

14. Reboot the instance.

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

15. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol [Amazon DCV](). If you do not require this functionality, do not complete this step.

CentOS Stream 8 and Red Hat Enterprise Linux 8

To install the NVIDIA gaming driver on your instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.

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

2. Update your package cache and get the package updates for your instance.

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

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. Download the gaming driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

   Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. Extract the gaming driver installation utility from the downloaded `.zip` archive.

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. Add permissions to run the driver installation utility using the following command:

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. Run the installer using the following command:

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

   When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

10. Use the following command to create the required configuration file.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. Use the following command to download and rename the certification file.

   - For version 460.39 or later:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

   - For version 440.68 to 445.48:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

   - For earlier versions:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. If you are using NVIDIA driver version 510.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
```

13. Reboot the instance.

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

14. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol Amazon DCV.

Rocky Linux 8

To install the NVIDIA gaming driver on your instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.

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

2. Update your package cache and get the package updates for your instance.

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

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

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

4. Reconnect to your instance after it has rebooted.
5. Install the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo dnf install -y unzip elfutils-libelf-devel libglvnd-devel
 kernel-devel-$(uname -r)
```

6. Download the gaming driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. Extract the gaming driver installation utility from the downloaded `.zip` archive.

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. Add permissions to run the driver installation utility using the following command:

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. Run the installer using the following command:

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

   When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

10. Use the following command to create the required configuration file.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. Use the following command to download and rename the certification file.

    - For version 460.39 or later:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
    ```

    - For version 440.68 to 445.48:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
    ```

    - For earlier versions:

    ```
    [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
    ```

12. If you are using NVIDIA driver version 510.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. Reboot the instance.

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

14. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol [Amazon DCV](https://docs.aws.amazon.com/dcv/).

Ubuntu and Debian

To install the NVIDIA gaming driver on your instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.

```
$ sudo apt-get install -y gcc make build-essential
```

2. Update your package cache and get the package updates for your instance.

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

3. Upgrade the `linux-aws` package to receive the latest version.

```
$ sudo apt-get upgrade -y linux-aws
```

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

```
$ sudo reboot
```

5. Reconnect to your instance after it has rebooted.
6. Install the kernel headers package for the version of the kernel you are currently running.

```
$ sudo apt install -y unzip dkms linux-headers-$(uname -r)
```

7. Disable the `nouveau` open source driver for NVIDIA graphics cards.

   a. Add `nouveau` to the `/etc/modprobe.d/blacklist.conf` blacklist file. Copy the following code block and paste it into a terminal.

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

b. Edit the `/etc/default/grub` file and add the following line:

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

c. Rebuild the Grub configuration.

```
$ sudo update-grub
```

8. Download the gaming driver installation utility using the following command:

```
$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

9. Extract the gaming driver installation utility from the downloaded `.zip` archive.

```
$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

10. Add permissions to run the driver installation utility using the following command:

```
$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

11. Run the installer using the following command:

```
$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

12. Use the following command to create the required configuration file.

```
$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

13. Use the following command to download and rename the certification file.

    - For version 460.39 or later:

    ```
    $ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
    ```

    - For version 440.68 to 445.48:

    ```
    $ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
    ```

    - For earlier versions:

    ```
    $ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
    gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
    ```

14. If you are using NVIDIA driver version 510.x or greater on the G4dn, G5, or G5g instances, disable GSP with the following commands. For more information about why this is required, see the NVIDIA documentation.

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

15. Reboot the instance.

```
$ sudo reboot
```

16. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol Amazon DCV. If you do not require this functionality, do not complete this step.

Windows operating systems

Before you install an NVIDIA gaming driver on your instance, you must ensure that the following prerequisites are met in addition to the considerations mentioned for all gaming drivers.

- If you launch your Windows instance using a custom Windows AMI, the AMI must be a standardized image created with Windows Sysprep to ensure that the gaming driver works. For more information, see Create an Amazon EC2 AMI using Windows Sysprep.
- Configure default credentials for the AWS Tools for Windows PowerShell on your Windows instance. For more information, see Getting Started with the AWS Tools for Windows PowerShell in the *AWS Tools for PowerShell User Guide*.

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.

```
$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 if you change the value of the `$KeyPrefix` variable from *"windows/latest"* to *"windows"*.

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. Use one of the following methods to register the driver.

Version 527.27 or above

Create the following registry key with the 64-bit version of PowerShell, or the Command Prompt window.

key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global`

name: vGamingMarketplace

type: DWord

value: 2

PowerShell

Run the following PowerShell command to create this registry value. The AWS Tools for PowerShell in AWS Windows AMIs defaults to the 32-bit version and this command fails. Instead, use the 64-bit version of PowerShell included with the operating system.

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global"
 -Name "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

Command Prompt

Run 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.

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global" /v
 vGamingMarketplace /t REG_DWORD /d 2
```

Earlier versions

Create the following registry key with the 64-bit version of PowerShell, or the Command Prompt window.

key: `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global`

name: vGamingMarketplace

type: DWord

value: 2

PowerShell

Run the following PowerShell command to create this registry value. The AWS Tools for PowerShell in AWS Windows AMIs defaults to the 32-bit version and this command fails. Instead, use the 64-bit version of PowerShell included with the operating system.

```
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name
 "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

Command Prompt

Run the following registry command to create this registry key with the Command Prompt window. You can also use this command in the 64-bit version of PowerShell.

```
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global" /v vGamingMarketplace /t
 REG_DWORD /d 2
```

5. Run the following command in PowerShell. This downloads the certification file, renames the file `GridSwCert.txt`, and moves the file to the Public Documents folder on your system drive. Typically, the folder path is `C:\Users\Public\Documents`.

   - For version 460.39 or later:

   ```
   Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
   Archive/GridSwCertWindows_2024_02_22.cert" -OutFile "$Env:PUBLIC\Documents
   \GridSwCert.txt"
   ```

   - For version 445.87:

   ```
   Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
   Archive/GridSwCert-Windows_2020_04.cert" -OutFile "$Env:PUBLIC\Documents
   \GridSwCert.txt"
   ```

   - For earlier versions:

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2019_09.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

6. Reboot your instance.
7. Locate the `nvidia-smi.exe` file on the instance.

```
Get-ChildItem -Path C:\ -Recurse -Filter "nvidia-smi.exe"
```

Verify the NVIDIA Gaming license using the following command. Replace *path* with the name of the folder in the output from the previous command.

```
C:\Windows\System32\DriverStore\FileRepository\path\nvidia-smi.exe -q
```

The output should be similar to the following.

```
vGPU Software Licensed Product
Product Name                  : NVIDIA Cloud Gaming
License Status                : Licensed (Expiry: N/A)
```

8. (Optional) To help take advantage of the single display of up to 4K resolution, set up the high-performance display protocol Amazon 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.

Install the CUDA toolkit on Linux

Follow these steps to install the CUDA toolkit on Linux:

1. Connect to your Linux instance.
2. Open the NVIDIA website and select the version of CUDA that you need.
3. Select the architecture, distribution, and version for the operating system on your instance. For **Installer Type**, select **runfile (local)**.
4. Follow the instructions to download the install script.
5. Add run permissions to the install script that you downloaded using the following command.

```
[ec2-user ~]$ chmod +x downloaded_installer_file
```

6. Run the install script as follows to install the CUDA toolkit and add the CUDA version number to the toolkit path.

```
[ec2-user ~]$ sudo sh downloaded_installer_file --silent --override --toolkit --
samples --toolkitpath=/usr/local/cuda-version --samplespath=/usr/local/cuda --no-
opengl-libs
```

7. (Optional) Set the default CUDA version as follows.

```
[ec2-user ~]$ sudo ln -s /usr/local/cuda-version /usr/local/cuda
```

Install the CUDA toolkit on Windows

Follow these steps to install the CUDA toolkit on Windows:

To install the CUDA toolkit

1. Connect to your Windows instance.
2. Open the NVIDIA website 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 the ENA driver on EC2 Windows instances

If your instance isn't based on one of the latest Windows Amazon Machine Images (AMIs) that Amazon provides, use the following procedure to install the current ENA driver on your instance. You should perform this update at a time when it's convenient to reboot your instance. If the install script doesn't automatically reboot your instance, we recommend that you reboot the instance as the final step.

If you use an instance store volume to store data while the instance is running, that data is erased when you stop the instance. Before you stop your 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.

Prerequisites

To install or upgrade the ENA driver, your Windows instance must meet the following prerequisites:

- PowerShell version 3.0 or later is installed.
- The commands shown in this section must run in the 64-bit version of PowerShell. Do not use the x86 version of PowerShell. That is the 32-bit version of the shell, and is not supported for these commands.

Step 1: Back up your data

We recommend that you create a backup AMI, in case you're not able to roll back your changes through the **Device Manager**. To create a backup AMI with the AWS Management Console, follow these steps:

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 the driver upgrade, and choose **Stop instance** from the **Instance state** menu.
4. After the instance is stopped, select the instance again. To create your backup, choose **Image and templates** from the **Actions** menu, then choose **Create image**.
5. To restart your instance, choose **Start instance** from the **Instance state** menu.

Step 2: Install or upgrade your ENA driver

You can install or upgrade your ENA driver with AWS Systems Manager Distributor, or with PowerShell cmdlets. For further instructions, select the tab that matches the method that you want to use.

Systems Manager Distributor

You can use the Systems Manager Distributor feature to deploy packages to your Systems Manager managed nodes. With Systems Manager Distributor, you can install the ENA driver package once, or with scheduled updates. For more information about how to install the ENA driver package (`AwsEnaNetworkDriver`) with Systems Manager Distributor, see Install or update packages in the *AWS Systems Manager User Guide*.

PowerShell

This section covers how to download and install ENA driver packages on your instance with PowerShell cmdlets.

Option 1: Download and extract the latest version

1. Connect to your instance and log in as the local administrator.
2. Use the **invoke-webrequest** cmdlet to download the latest driver package:

```
PS C:\> invoke-webrequest https://ec2-windows-drivers-downloads.s3.amazonaws.com/ENA/Latest/AwsEnaNetworkDriver.zip -outfile $env:USERPROFILE\AwsEnaNetworkDriver.zip
```

> **Note**
>
> If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

Alternatively, you can download the latest driver package from a browser window on your instance.

3. Use the **expand-archive** cmdlet to extract the zip archive that you downloaded to your instance:

```
PS C:\> expand-archive $env:userprofile\AwsEnaNetworkDriver.zip -
DestinationPath $env:userprofile\AwsEnaNetworkDriver
```

Option 2: Download and extract a specific version

1. Connect to your instance and log in as the local administrator.
2. Download the ENA driver package for the specific version you want from the version link in the ENA Windows driver version history table.
3. Extract the zip archive to your instance.

Install the ENA driver with PowerShell

The install steps are the same whether you've downloaded the latest driver or a specific version. To install the ENA driver, follow these steps.

1. To install the driver, run the `install.ps1` PowerShell script from the `AwsEnaNetworkDriver` directory on your instance. If you get an error, make sure that you're using PowerShell 3.0 or later.
2. If the installer doesn't automatically reboot your instance, run the **Restart-Computer** PowerShell cmdlet.

```
PS C:\> Restart-Computer
```

Step 3 (optional): Verify the ENA driver version after installation

To ensure that the ENA driver package was successfully installed on your instance, you can verify the new version as follows:

1. Connect to your instance and log in as the local administrator.

2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.

   Note

   ENA adapters all use the same driver. If you have multiple ENA adapters, you can select any one of them to update the driver for all of the ENA adapters.

6. To verify the current version that's installed, open the **Driver** tab and check the **Driver Version**. If the current version doesn't match your target version, see Troubleshoot the Elastic Network Adapter Windows driver.

Roll back an ENA driver installation

If anything goes wrong with the installation, you might need to roll back the driver. Follow these steps to roll back to the previous version of the ENA driver that was installed on your instance.

1. Connect to your instance and log in as the local administrator.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.

   Note

   ENA adapters all use the same driver. If you have multiple ENA adapters, you can select any one of them to update the driver for all of the ENA adapters.

6. To roll back the driver, open the **Driver** tab and choose **Roll Back Driver**. This opens the **Driver Package rollback** window.

> **Note**
>
> If the **Driver** tab doesn't show the **Roll Back Driver** action, or if the action is unavailable, it means that the Driver Store on your instance doesn't contain the previously installed driver package. To troubleshoot this issue, see Troubleshooting scenarios, and expand the **Unexpected ENA driver version installed** section. For more information about the device driver package selection process, see How Windows selects a driver package for a device on the *Microsoft documentation website*.

Track ENA Windows driver version releases

Windows AMIs include the ENA Windows driver to enable enhanced networking.

For Windows Server versions 2016 and above, we recommend that you use the latest driver version. For earlier versions of Windows Server, refer to the following table to determine which ENA driver version to use.

Windows Server version	ENA driver version
Windows Server 2012 R2	2.6.0 and earlier
Windows Server 2012	2.6.0 and earlier
Windows Server 2008 R2	2.2.3 and earlier

ENA Windows driver version history

The following table summarizes the changes for each release.

Driver version	Details	Release date
2.11.0	**New features** •	August 1, 2025

Driver version	Details	Release date
	Added support for dynamic queue allocation with scaling up to 64 I/O queues, subject to instance type limits. **Bug fixes** • Fixed missing handling of error conditions when retrieving the physical DMA address width from the device.	
2.10.0	**Bug fixes** • Fixed improper handling of memory allocation failures during queue initialization to prevent unexpected reboots. • Fixed possible misconfiguration of network offloads where device capabilities were ignored.	June 24, 2025

Driver version	Details	Release date
2.9.0	**New features** • Added support for asynchronous reset requests initiated by the device. • Added support for handling the maximum large LLQ depth value provided by the device. • Added Event ID `58001` in the Windows Event Viewer to enhance visibility into unexpected power state transitions caused by device misconfiguration. **Bug fixes** • Fixed improper handling of memory allocation failures during device initialization to prevent unexpected reboots. • Fixed an issue in the interrupt service routine that could enqueue a DPC during device stop, preventing unexpected reboots.	December 12, 2024
2.8.0	**Bug fixes** • Fixed a race condition in the complete flow of egress network buffer list (NBL) processing, which could lead to memory corruption caused by attempting to release an NBL that was already released. • Fixed misdetection of L3 protocol when disabling all LSO and checksum offloads that could lead to unexpected behavior.	September 30, 2024

Driver version	Details	Release date
2.7.0	**New features** • Removed support for Windows Server 2012 (Windows 8) and Windows Server 2012 R2 (Windows 8.1). These operating system versions have reached the end of support from AWS. Driver install will fail on Windows Server 2012 and earlier. • Added support for offloading IPv6 Tx checksum calculation to the device. • Added wide Low Latency Queuing (LLQ) support. This is dynamically enabled based on device recommend ation. You can override this setting with the new "WideLLQ" registry key. • Added reporting for packet drops resulting from Rx overrun, which indicates insufficient space in the Rx ring for incoming packets. • Added support for suboptimal configuration notificat ions from the device. See event ID `59000` in the Windows event viewer. **Bug fixes** • Avoid unnecessary device reset caused by Tx packets with headers that exceed the maximum Low Latency Queuing (LLQ) header size.	May 1, 2024

Driver version	Details	Release date

2.6.0	**New features**
	• Adds the following network performance metrics for instance types that support ENA Express.
	  • `ena_srd_mode`
	  • `ena_srd_tx_pkts`
	  • `ena_srd_eligible_tx_pkts`
	  • `ena_srd_rx_pkts`
	  • `ena_srd_resource_utilization`
	• Adds `conntrack_allowance_available` network performance metric for Nitro based instance types.
	• Adds new adapter reset reason due to detection of RX data corruption.
	• Updates driver logging infrastructure.
	
	Bug fixes
	• Prevents adapter reset in the event that CPU starvatio n causes a network performance metrics update to fail.
	• Prevents false detection of an interruption to the device heartbeat.
	•
		June 20, 2023

Driver version	Details	Release date
	Fixes driver installation script to support downgrade operation. • Fixes the receive error count statistic.	
2.5.0	**Announcement** ENA Windows driver version 2.5.0 has been rolled back due to failure to initialize on the Windows domain controller. Windows Client and Windows Server are unaffected.	February 17, 2023

Driver version	Details	Release date
2.4.0	**New features** • Adds support for Windows Server 2022. • Removes support for Windows Server 2008 R2. • Sets Low Latency Queuing (LLQ) to always on to improve performance on sixth generation Amazon EC2 instances. **Bug fixes** • Fixes a failure to publish network performance metrics to the Performance Counters for Windows (PCW) system. • Fixes a memory leak during the registry key reading operation. • Prevents an infinite reset loop in the event of an unrecoverable error during the adapter reset process.	April 28, 2022

Driver version	Details	Release date
2.2.4	**Announcement** ENA Windows driver version 2.2.4 has been rolled back due to potential performance degradation on the sixth generation EC2 instances. We recommend that you downgrade the driver, using one of the following methods: • **Install the previous version** 1. Download the previous version package from the link in this table (version 2.2.3). 2. Run the **install.ps1** PowerShell installation script. For more details for pre- and post-installation steps see Enable enhanced networking on Windows. • **Use Amazon EC2 Systems Manager for a bulk update** • Perform a bulk update via SSM document `AWS-ConfigureAWSPackage`, with the following parameters: • **Name**: AwsEnaNetworkDriver • **Version**: 2.2.3	October 26, 2021

Driver version	Details	Release date
2.2.3	**New features** • Adds support for new Nitro cards with up to 400 Gbps instance networking. **Bug fixes** • Fixes race condition between system time change and system time query by the ENA driver, which causes false-positive detection of HW unresponsiveness. **Announcement** Windows ENA driver version 2.2.3 is the final version that supports Windows Server 2008 R2. Currently available instance types that use ENA will continue to be supported on Windows Server 2008 R2, and the drivers are available by download. No future instance types will support Windows Server 2008 R2, and you cannot launch, import, or migrate Windows Server 2008 R2 images to future instance types.	March 25, 2021

Driver version	Details	Release date
2.2.2	**New features** • Adds support to query network adapter performance metrics with CloudWatch and the Performance Counters for Windows consumers. **Bug fixes** • Fixes performance issues on bare metal instances.	December 21, 2020
2.2.1	**New features** • Adds a method to allow the host to query the Elastic Network Adapter for network performance metrics.	October 1, 2020

Driver version	Details	Release date
2.2.0	**New features** • Adds support for next generation hardware types. • Improves instance start time after resuming from stop-hibernate, and eliminates false positive ENA error messages. **Performance optimizations** • Optimizes processing of inbound traffic. • Improves shared memory management in low resource environment. **Bug fixes** • Avoids system crash upon ENA device removal in rare scenario where driver fails to reset.	August 12, 2020
2.1.5	**Bug fixes** • Fixes occasional network adapter initialization failure on bare metal instances.	June 23, 2020

Driver version	Details	Release date
2.1.4	**Bug fixes** • Prevent connectivity issues caused by corrupted LSO packet metadata arriving from the network stack. • Prevent system crash caused by a rare race condition that results in accessing an already released packet memory.	November 25, 2019
2.1.2	**New features** • Added support for vendor ID report to allow OS to generate MAC-based UUIDs. **Bug fixes** • Improved DHCP network configuration performance during initialization. • Properly calculate L4 checksum on inbound IPv6 traffic when the maximum transmission unit (MTU) exceeds 4K. • General improvements to driver stability and minor bug fixes.	November 4, 2019

Driver version	Details	Release date
2.1.1	**Bug fixes** • Prevent drops of highly fragmented TCP LSO packets arriving from operating system. • Properly handle Encapsulating Security Payload (ESP) protocol within the IPSec in IPv6 networks.	September 16, 2019

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.
		July 1, 2019

Driver version	Details	Release date
	Bug fixes • Fix crash due to invalid parsing of non-contiguous Tx headers. • Fix driver v1.5 crash during the elastic network inte rface 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.	

Driver version	Details	Release date
1.5.0	**Updates** • 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
1.0.8	The initial release. Included in AMIs for Windows Server 2008 R2, Windows Server 2012 RTM, Windows Server 2012 R2, and Windows Server 2016.	July 2016

Subscribe to ENA Windows driver release notifications from Amazon SNS

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

Subscribe to EC2 notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select 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 where you want notifications sent.
   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.

Unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Subscriptions**.
3. Select the checkbox for the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

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.

	Red Hat PV	Citrix PV	AWS PV
Instance type	Not supported for all instance types. If you specify an unsupported instance type, the instance is impaired.	Supported for Xen instance types.	Supported for Xen instance types.
Attached volumes	Supports up to 16 attached volumes.	Supports more than 16 attached volumes.	Supports more than 16 attached volumes.

	Red Hat PV	Citrix PV	AWS PV
Network	The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.		The driver automatically configures jumbo frames on the network adapter when on a compatible instance type. When the instance is in a cluster placement group, this offers better network performance between instances that are in the cluster placement group. For more information, see [Placement groups for your Amazon EC2 instances]().

The following table shows which PV drivers you should run on each version of Windows Server on Amazon EC2.

Windows Server version	PV driver version
Windows Server 2025	Not supported
Windows Server 2022	AWS PV latest version
Windows Server 2019	AWS PV latest version
Windows Server 2016	AWS PV latest version
Windows Server 2012 R2	AWS PV version 8.4.3
Windows Server 2012	AWS PV version 8.4.3
Windows Server 2008 R2	AWS PV version 8.3.5
Windows Server 2008	Citrix PV 5.9
Windows Server 2003	Citrix PV 5.9

Contents

AWS PV drivers

The AWS PV drivers are stored in the `%ProgramFiles%\Amazon\Xentools` directory. This directory also contains public symbols and a command line tool, `xenstore_client.exe`, that enables you to access entries in XenStore. For example, the following PowerShell command returns the current time from the Hypervisor:

```
PS C:\> [DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl
 AWSXenStoreBase).XenTime).ToString("hh:mm:ss")
11:17:00
```

The AWS PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`. These driver components are as follows: xenbus, xeniface, xennet, xenvbd, and xenvif.

AWS PV 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

- Use AWS Systems Manager to automatically update the PV drivers. For more information, see Walkthrough: Automatically Update PV Drivers on EC2 Windows Instances in the *AWS Systems Manager User Guide*.
- 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 AWS PV drivers, or upgrading a domain controller, see Upgrade Windows Server instances (AWS PV upgrade) manually.

AWS PV driver package history

The following table shows the changes to AWS PV drivers for each driver release.

Package version	Details	Release date
8.6.0	• Stability fixes to XenStore interactions.	27 May 2025
8.5.0	• Stability fixes to address rare cases of crashes during network device detachment. • Stability fixes to address rare cases of crashes during EBS volume detachment. • Fixed bugs in the package installer. • Updated the PV installer to use `Pnputil`.	31 October 2024
8.4.3	Fixed bugs in the package installer to improve the upgrade experience. This is the last version that can run on Windows Server 2012 and 2012 R2. This version is available for download, however it is no longer supported since Windows Server 2012 and 2012 R2 have reached end of support.	24 January 2023
8.4.2	Stability fixes to address race condition.	13 April 2022
8.4.1	Improved package installer.	7 January 2022
8.4.0	• Stability fixes to address rare cases of stuck disk IO. • Stability fixes to address rare cases of crashes during EBS volume detachment. • Added feature to distribute load across multiple cores for workloads that leverage more than 20,000 IOPS and experience degradation due to bottlenecks. To enable this feature, see Workloads that leverage more than 20,000 disk IOPS experience degradation due to CPU bottlenecks.	2 March 2021
8.3.5	Improved package installer. This is the last version that can run on Windows Server 2008 R2. This version is available for download but no longer	7 January 2022

Package version	Details	Release date
	supported. Windows Server 2008 R2 has reached end-of-life, and is no longer supported by Microsoft.	
8.3.4	Improved reliability of network device attachment.	4 August 2020
8.3.3	• Update to XenStore-facing component to prevent bug check during error-handling paths. • Update to storage component to avoid crashes when an invalid SRB is submitted. To update this driver on Windows Server 2008 R2 instances, you must first verify that the appropriate patches are installed to address the following Microsoft Security Advisory: Microsoft Security Advisory 3033929.	4 February 2020
8.3.2	Enhanced reliability of networking components.	30 July 2019
8.3.1	Improved performance and robustness of storage component.	12 June 2019
8.2.7	Improved efficiency to support migrating to latest generation instance types.	20 May 2019
8.2.6	Improved efficiency of crash dump path.	15 January 2019
8.2.5	Additional security enhancements. PowerShell installer now available in package.	12 December 2018
8.2.4	Reliability improvements.	2 October 2018

Package version	Details	Release date
8.2.3	Bug fixes and performance improvements. Report EBS volume ID as disk serial number for EBS volumes. This enables cluster scenarios such as S2D.	29 May 2018
8.2.1	Network and storage performance improvements plus multiple robustness fixes. To verify that this version has been installed, refer to the following Windows registry value: `HKLM\Software\Amazon\PVDriver\Version 8.2.1` .	8 March 2018
7.4.3	Added support for Windows Server 2016. Stability fixes for all supported Windows OS versions. *AWS PV driver version 7.4.3's signature expires on March 29, 2019. We recommend updating to the latest AWS PV driver.	18 Nov 2016
7.4.2	Stability fixes for support of X1 instance type.	2 Aug 2016
7.4.1	• 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.	12 July 2016
7.3.2	• Improved logging and diagnostics. • Stability fix in AWS PV Storage driver. In some cases disks may not surface in Windows after reattaching the disk to the instance. • Added support for Windows Server 2012.	24 June 2015

Package version	Details	Release date
7.3.1	TRIM update: Fix related to TRIM requests. This fix stabilizes instances and improves instance performance when managing large numbers of TRIM requests.	
7.3.0	TRIM support: The AWS PV driver now sends TRIM requests to the hypervisor. Ephemeral disks will properly process TRIM requests given the underlying storage supports TRIM (SSD). Note that EBS-based storage does not support TRIM as of March 2015.	
7.2.5	• Stability fix in AWS PV Storage drivers: In some cases the AWS PV driver could dereference invalid memory and cause a system failure. • Stability fix while generating a crash dump: In some cases the AWS PV driver could get stuck in a race condition when writing a crash dump. Before this release, the issue could only be resolved by forcing the driver to stop and restart which lost the memory dump.	
7.2.4	Device ID persistence: This driver fix masks the platform PCI device ID and forces the system to always surface the same device ID, even if the instance is moved. More generally, the fix affects how the hypervisor surfaces virtual devices. The fix also includes modifications to the co-installer for the AWS PV drivers so the system persists mapped virtual devices.	
7.2.2	• Load the AWS PV drivers in Directory Services Restore Mode (DSRM) mode: Directory Services Restore Mode is a safe mode boot option for Windows Server domain controllers. • Persist device ID when virtual network adapter device is reattached: This fix forces the system to check the MAC address mapping and persist the device ID. This fix ensures that adapters retain their static settings if the adapters are reattached.	

Package version	Details	Release date
7.2.1	• Run in safe mode: Fixed an issue where the driver would not load in safe mode. Previously the AWS PV Drivers would only instantiate in normal running systems. • Add disks to Microsoft Windows Storage Pools: Previously we synthesized page 83 queries. The fix disabled page 83 support. Note this does not affect storage pools that are used in a cluster environment because PV disks are not valid cluster disks.	
7.2.0	Base: The AWS PV base version.	

Citrix PV drivers

The Citrix PV drivers are stored in the `%ProgramFiles%\Citrix\XenTools` (32-bit instances) or `%ProgramFiles(x86)%\Citrix\XenTools` (64-bit instances) directory.

The Citrix PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services`. These driver components are as follows: xenevtchn, xeniface, xennet, Xennet6, xensvc, xenvbd, and xenvif.

Citrix also has a driver component named XenGuestAgent, which runs as a Windows service. It handles tasks such as shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

If you are encountering networking errors while performing certain workloads, you may need to disable the TCP offloading feature for the Citrix PV driver. For more information, see TCP offloading.

Red Hat PV drivers

Red Hat 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 Red Hat drivers can fail to boot and become inaccessible. We recommend upgrading Red Hat drivers to Citrix PV drivers, and then upgrade Citrix PV drivers to AWS PV drivers.

The source files for the Red Hat drivers are in the `%ProgramFiles%\RedHat` (32-bit instances) or `%ProgramFiles(x86)%\RedHat` (64-bit instances) directory. The two drivers are `rhelnet`, the Red Hat Paravirtualized network driver, and `rhelscsi`, the Red Hat SCSI miniport driver.

Subscribe to notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. You can subscribe to these notifications.

> **Note**
>
> You must specify the Region in which the SNS topic was created.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, you can unsubscribe. For more information, see [Delete an SNS topic and subscription](#).

Console

To subscribe to notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select 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.

AWS CLI

To subscribe to notifications

Use the following command.

```
aws sns subscribe \
    --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers \
    --region us-east-1 \
    --protocol email \
    --notification-endpoint YourUserName@YourDomainName.ext
```

PowerShell

To subscribe to notifications

Use the following command.

```
Connect-SNSNotification `
    -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers' `
    -Region us-east-1 `
    -Protocol email `
    -Endpoint "YourUserName@YourDomainName.ext"
```

Upgrade PV drivers on EC2 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 **Device Manager** and view **Network Adapters**. Check whether the PV driver is one of the following:

- AWS PV Network Device
- Citrix PV Ethernet Adapter
- Red Hat PV NIC Driver

System requirements

Be sure to check the `readme.txt` file in the download for system requirements.

Contents

Upgrade Windows Server instances (AWS PV upgrade) with Distributor

You can use Distributor, a capability of AWS Systems Manager, to install or upgrade the AWS PV driver package. The installation or upgrade can be performed one time, or you can install or update it on a schedule. The `In-place update` option for **Installation Type** isn't supported for this Distributor package.

> **Important**
>
> If your instance is a domain controller, see Upgrade a domain controller (AWS PV upgrade). The upgrade process for domain controller instances is different than standard editions of Windows.

1. We recommend that you create a backup in case you need to roll back your changes.

   > **Tip**
   >
   > Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For more information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

   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 the instance using Remote Desktop. For more information, see the section called “Connect using an RDP client”.
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. For the instructions for how to install or upgrade the AWS PV driver package using Distributor, see the procedures in Install or update packages in the *AWS Systems Manager User Guide*.
5. For **Name**, choose **AWSPVDriver**.
6. For **Installation type**, select **Uninstall and reinstall**.
7. Configure the other parameters for the package as necessary and run installation or upgrade using the referenced procedure in Step 4.

   After running the Distributor package, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes.
8. After the upgrade is complete, and the instance passes both health checks in the Amazon EC2 console, verify that the new driver was installed by connecting to the instance using Remote Desktop.
9. After you are connected, run the following PowerShell command:

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

10. 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 Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters noted in Step 3.

If you previously disabled TCP offloading 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 might need to reapply the static IP address or DNS configuration after upgrading AWS PV drivers.

Upgrade Windows Server instances (AWS PV upgrade) manually

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, Windows Server 2019, or Windows Server 2022. This upgrade is not available for Red Hat drivers, or for other versions of Windows Server.

Some older versions of Windows Server can't use the latest drivers. To verify which driver version to use for your operating system, see the driver version table in the Paravirtual drivers for Windows instances page.

> **Important**
>
> If your instance is a domain controller, see Upgrade a domain controller (AWS PV upgrade). The upgrade process for domain controller instances is different than standard editions of Windows.

To upgrade AWS PV drivers manually

1. We recommend that you create a backup in case you need to roll back your changes.

   > **Tip**
   >
   > Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For more information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

   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 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 drivers to your instance using one of the following options:

   - **Browser** – Download the latest driver package to the instance and extract the zip archive.
   - **PowerShell** – Run the following commands:

```
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
```

   If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

5. 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 Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters noted in Step 3.

If you previously disabled TCP offloading 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 might need to 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 ensure that your FSMO roles remain operational during the upgrade, we recommend that you transfer those roles to other domain controllers before you start the upgrade. For more information, see How to view and transfer FSMO roles on the *Microsoft Learn* website.

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 in the Microsoft documentation.
2. Run the following command to configure Windows to boot into Directory Services Restore Mode (DSRM).

> **Warning**
>
> Before running this command, confirm that you know the DSRM password. You'll need this information so that you can log in to your instance after the upgrade is complete and the instance automatically reboots.

```
bcdedit /set {default} safeboot dsrepair
```

PowerShell:

```
PS C:\> bcdedit /set "{default}" safeboot dsrepair
```

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 **`DisableDCCheck`** to the registry:

```
reg add HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck /t
 REG_SZ /d true
```

4. Download the latest driver package to the instance and extract the zip archive.
5. 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.
6. 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 username in the following format *hostname*\administrator. For example, Win2k12TestBox\administrator.
7. Run the following command to remove the DSRM boot configuration:

```
bcdedit /deletevalue safeboot
```

8. Reboot the instance.
9. 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.
10. Run the following command to delete **DisableDCCheck** from the registry:

```
reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck
```

> **Note**
>
> If you previously disabled TCP offloading 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 (Red Hat to Citrix PV upgrade)

Before you start upgrading your Red Hat 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.
- 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 an Amazon EBS-backed AMI.

> **Tip**
>
> Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For more information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

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.

To upgrade Red Hat 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 using RDP.
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 Red Hat software. Your instance will be rebooted.

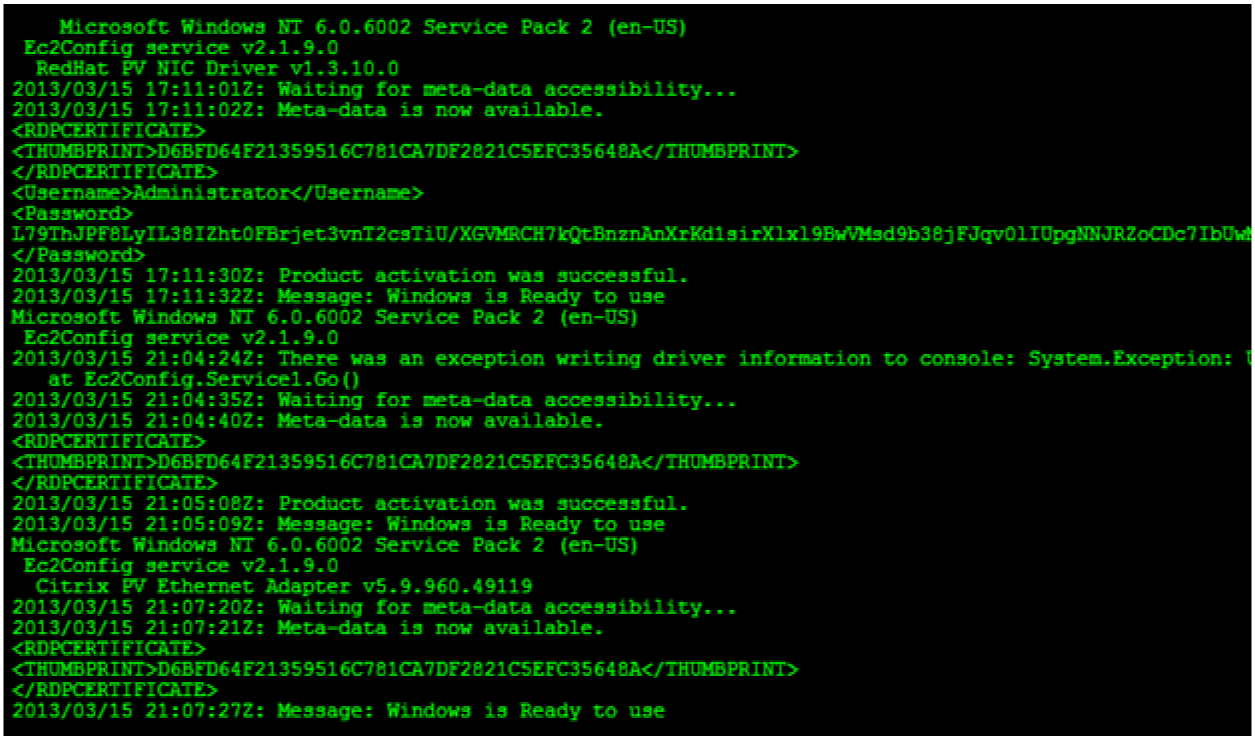
> **Note**
>
> If you do not see the uninstaller dialog box, choose **Red Hat Paravirtualize** in the Windows taskbar.

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.

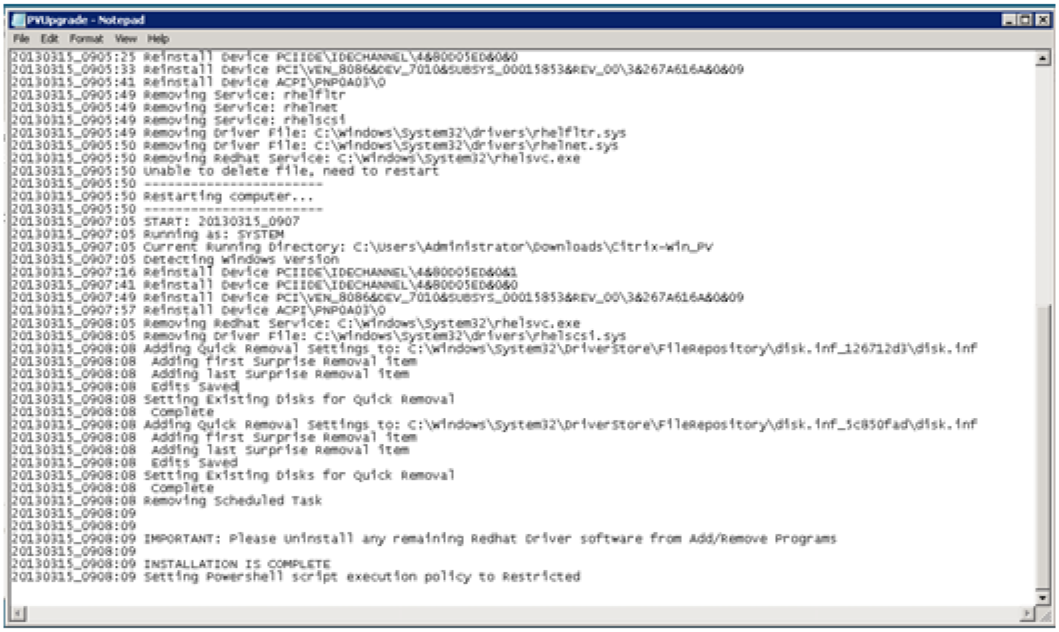


```
    Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
 Ec2Config service v2.1.9.0
  RedHat PV NIC Driver v1.3.10.0
2013/03/15 17:11:01Z: Waiting for meta-data accessibility...
2013/03/15 17:11:02Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
<Username>Administrator</Username>
<Password>
L79ThJPF8LyIL38IZht0FBrjet3vnT2csTiU/XGVMRCH7kQtBnznAnXrKd1sirXlx19BwVMsd9b38jFJqv0lIUpgNNJRZoCDc7IbUw
</Password>
2013/03/15 17:11:30Z: Product activation was successful.
2013/03/15 17:11:32Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
 Ec2Config service v2.1.9.0
2013/03/15 21:04:24Z: There was an exception writing driver information to console: System.Exception:
   at Ec2Config.Service1.Go()
2013/03/15 21:04:35Z: Waiting for meta-data accessibility...
2013/03/15 21:04:40Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:05:08Z: Product activation was successful.
2013/03/15 21:05:09Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
 Ec2Config service v2.1.9.0
  Citrix PV Ethernet Adapter v5.9.960.49119
2013/03/15 21:07:20Z: Waiting for meta-data accessibility...
2013/03/15 21:07:21Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:07:27Z: Message: Windows is Ready to use
```

8. Connect to your instance and log in as the local administrator.

9. Close the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box.

10. Confirm that the installation is complete. Navigate to the `Citrix-WIN_PV` folder that you extracted earlier, open the `PVUpgrade.log` file, and then check for the text `INSTALLATION IS COMPLETE`.



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 an Amazon EBS-backed AMI](#).

> **Tip**
>
> Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For more

information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

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 using RDP.
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 on Windows instances

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

> **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 features 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

There are two methods for restoring network and storage connectivity to an instance affected by this issue. Choose one of the following methods:

Method	Prerequisites	Procedure Overview
Method 1 - Enhanced networking	Enhanced networking is only available in a virtual private	You change the server instance type to a C3

Method	Prerequisites	Procedure Overview
	cloud (VPC) which requires a C3 instance type. If the server does not currently use the C3 instance type, then you must temporarily change it.	instance. Enhanced networkin g then enables you to connect to the affected instance and fix the problem. After you fix the problem, you change the instance back to the original instance type. This method is typically faster than Method 2 and less likely to result in user error. You will incur additiona l charges as long as the C3 instance is running.
Method 2 - Registry configura tion	Ability to create or access a second server. Ability to change Registry settings.	You detach the root volume from the affected instance, attach it to a different instance, connect, and make changes in the Registry. You will incur additiona l charges as long as the additional server is running. This method is slower than Method 1, but this method has worked in situations where Method 1 failed to resolve the problem.

Method 1 - Enhanced networking

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. 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.
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.

4. Choose **Launch instances** and create a temporary Windows Server 2008 or Windows Server 2012 instance in the same Availability Zone as the affected instance. Do not create a Windows Server 2012 R2 instance.

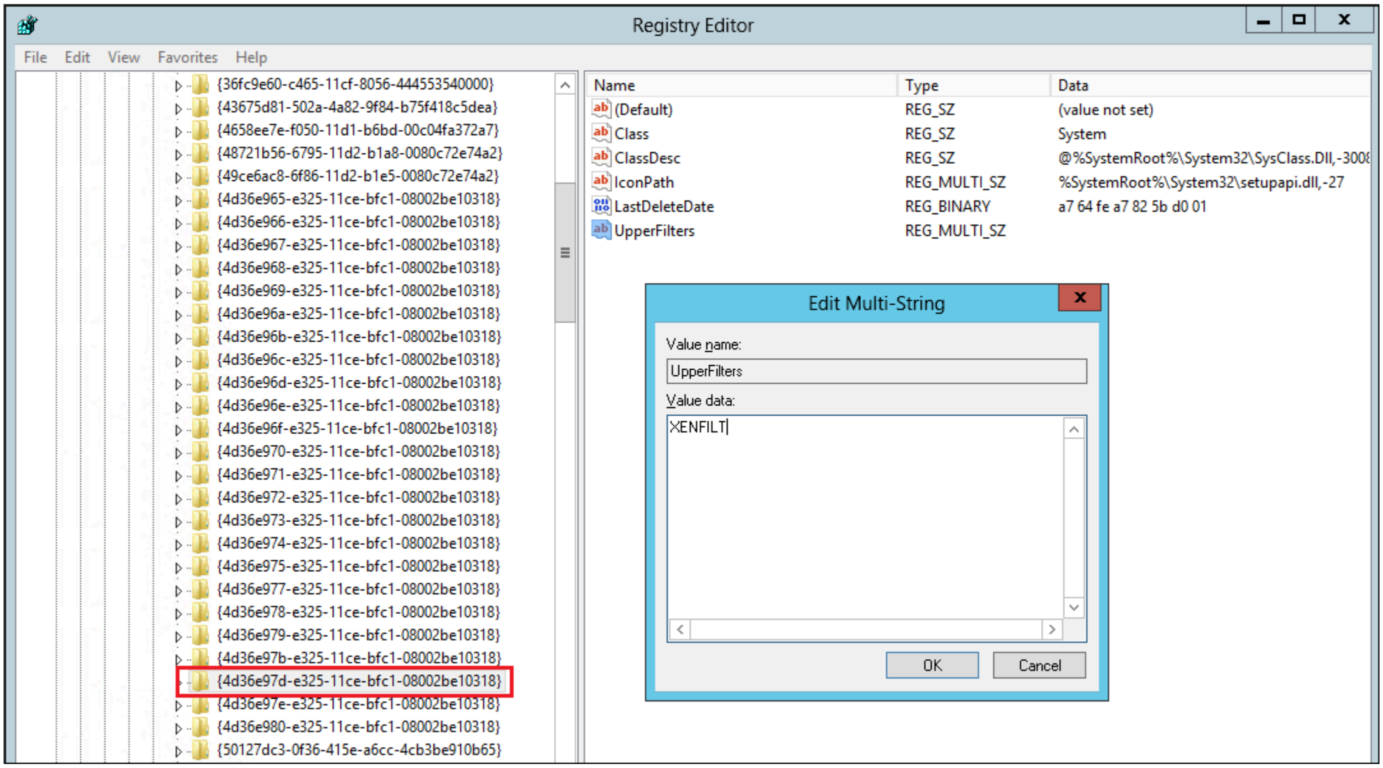
   > **Important**
   >
   > If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. Detach the volume and then attach the volume 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.
8. On the temporary instance, open the **Run** dialog box, type **`regedit`**, and press Enter.
9. In the Registry Editor navigation pane, choose **HKEY_Local_Machine**, and then from the **File** menu choose **Load Hive**.
10. In the **Load Hive** dialog box, navigate to *Affected Volume*\Windows\System32\config\System and type a temporary name in the **Key Name** dialog box. For example, enter OldSys.
11. In the navigation pane of the Registry Editor, locate the following keys:

    HKEY_LOCAL_MACHINE*`your_temporary_key_name`*\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

    HKEY_LOCAL_MACHINE*`your_temporary_key_name`*\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. For each key, double-click **UpperFilters**, enter a value of XENFILT, and then choose **OK**.



13. Locate the following key:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS\Parameters

14. Create a new string (REG_SZ) with the name ActiveDevice and the following value:

    PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. Locate the following key:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS

16. Change the **Count** from 0 to 1.

17. Locate and delete the following keys:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\xenvbd\StartOverride

    HKEY_LOCAL_MACHINE *your_temporary_key_name*\ControlSet001\Services\xenfilt\StartOverride

18. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened the Registry Editor.

19. From the **File** menu, choose **Unload Hive**.
20. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
21. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your Windows Server 2012 R2 instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
22. Start the instance.
23. Connect to the instance using Remote Desktop and then download the AWS PV Drivers Upgrade package to the instance.
24. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

    After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.
25. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history.
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.

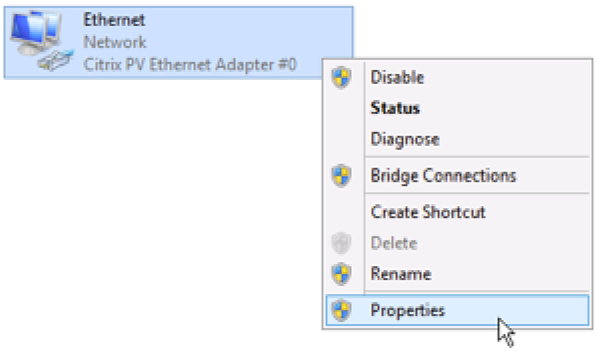
By default, TCP offloading is enabled for the Citrix PV drivers in Windows AMIs. If you encounter transport-level errors or packet transmission errors (as visible on the Windows Performance Monitor)—for example, when you're running certain SQL workloads—you may need to disable this feature.

> **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**.
4. Choose **Change adapter settings**.
5. Right-click **Citrix PV Ethernet Adapter #0** and select **Properties**.



6. In the **Local Area Connection Properties** dialog box, choose **Configure** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
7. On the **Advanced** tab, disable each of the properties, except for **Correct TCP/UDP Checksum Value**. To disable a property, select it from **Property** and choose **Disabled** from **Value**.
8. Choose **OK**.
9. Run the following commands from a Command Prompt window.

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

10. Reboot the instance.

Time synchronization

Prior to the release of the 2013.02.13 Windows AMI, the Citrix Xen guest agent could set the system time incorrectly. This can cause your DHCP lease to expire. If you have issues connecting to your instance, you might need to update the agent.

To determine whether you have the updated Citrix Xen guest agent, check whether the `C:\Program Files\Citrix\XenGuestAgent.exe` file is from March 2013. If the date on this file is earlier than that, update the Citrix Xen guest agent service. For more information, see Upgrade your Citrix Xen guest agent service.

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 it 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 EC2 Windows instances.
   - You have RDP access to the instance. For steps to connect to your Windows instance using RDP, see Connect to your Windows instance using an RDP client.
   - 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.

10Note whether core `0` or a different core is experiencing a heavy load.

3. **Choose which configuration to apply**

Configuration name	When to apply this configuration	Notes
Default configuration	Workload is driving less than 20,000 IOPS, or other configurations did not improve performance or stability.	For this configuration, IO occurs on a few cores, which may benefit smaller workloads by increasing cache locality and reducing context switching.
Allow driver to choose whether to distribute completion	Workload is driving more than 20,000 IOPS and moderate or high load is observed on core `0`.	This configuration is recommended for all Xen instances using PV 8.4.0 or later and leveraging more than 20,000 IOPS,

Configuration name	When to apply this configuration	Notes
		whether or not problems are encountered.
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 or 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.

```
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 `NotiferDistributed` 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 `NotiferDistributed` registry key.

```
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.

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000002 -Name NotifierDistributed
```

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name DpcRedirection
```

4. Reboot your instance.

AWS NVMe drivers

Amazon EBS volumes and instance store volumes are exposed as NVMe block devices on Nitro-based instances. To fully utilize the performance and capabilities of Amazon EBS features for volumes exposed as NVMe block devices, the instance must have the AWS NVMe driver installed. All current generation AWS Windows AMIs come with the AWS NVMe driver installed by default.

For more information about EBS and NVMe, see Amazon EBS and NVMe in the *Amazon EBS User Guide*. For more information about SSD instance store and NVMe, see SSD instance store volumes for EC2 instances.

Linux instances

The following AMIs include the required NVMe drivers:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 or later with `linux-aws` kernel

> **Note**
>
> AWS Graviton-based instance types require Ubuntu 18.04 or later with `linux-aws` kernel

- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later

To confirm that your instance has the NVMe driver

You can confirm that your instance has the NVMe driver using the following command.

- Amazon Linux, RHEL, CentOS, and SUSE Linux Enterprise Server

  ```
  $ modinfo nvme
  ```

  If the instance has the NVMe driver, the command returns information about the driver.

- Amazon Linux 2 and Ubuntu

  ```
  $ ls /sys/module/ | grep nvme
  ```

  If the instance has the NVMe driver, the command returns the installed drivers.

To update the NVMe driver

If your instance has the NVMe driver, you can update the driver to the latest version using the following procedure.

1. Connect to your instance.
2. Update your package cache to get necessary package updates as follows.

   - For Amazon Linux 2, Amazon Linux, CentOS, and Red Hat Enterprise Linux:

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

- For Ubuntu and Debian:

```
[ec2-user ~]$ sudo apt-get update -y
```

3. Ubuntu 16.04 and later include the `linux-aws` package, which contains the NVMe and ENA drivers required by Nitro-based instances. Upgrade the `linux-aws` package to receive the latest version as follows:

```
[ec2-user ~]$ sudo apt-get install --only-upgrade -y linux-aws
```

   For Ubuntu 14.04, you can install the latest `linux-aws` package as follows:

```
[ec2-user ~]$ sudo apt-get install linux-aws
```

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

```
sudo reboot
```

5. Reconnect to your instance after it has rebooted.

Windows instances

PowerShell

If you did not launch your instance from one of the latest AWS Windows AMIs provided by Amazon, use the following procedure to install the current AWS NVMe driver on your instance. Reboot is required for this install. Either the install script will reboot your instance or you must reboot it as the final step.

Prerequisites

- PowerShell version 3.0 or later is installed.
- The commands shown in this section must run in the 64-bit version of PowerShell. Do not use the `x86` version of PowerShell. That is the 32-bit version of the shell, and is not supported for these commands.

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 the drivers to your instance using one of the following options:

   - **Browser** – Download the latest driver package to the instance and extract the zip archive.
   - **PowerShell** – Run the following commands:

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/
   NVMe/Latest/AWSNVMe.zip -outfile $env:USERPROFILE\nvme_driver.zip
   Expand-Archive $env:userprofile\nvme_driver.zip -DestinationPath
    $env:userprofile\nvme_driver
   ```

   If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:

   ```
   [Net.ServicePointManager]::SecurityProtocol =
    [Net.SecurityProtocolType]::Tls12
   ```

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.

a. (Optional) Starting with AWS NVMe version `1.5.0`, Small Computer System Interface (SCSI) persistent reservations are supported for Windows Server 2016 and later. This feature adds support for Windows Server Failover Clustering with shared Amazon EBS storage. By default, this feature isn't enabled during installation.

   You can enable the feature when running the `install.ps1` script to install the driver by specifying the `EnableSCSIPersistentReservations` parameter with a value of `$true`.

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $true
```

   You can disable the feature when running the `install.ps1` script to install the driver by specifying the `EnableSCSIPersistentReservations` parameter with a value of `$false`.

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $false
```

b. Starting with AWS NVMe `1.5.0`, the `install.ps1` script always installs the `ebsnvme-id` tool with the driver.

   (Optional) For versions `1.4.0`, `1.4.1`, and `1.4.2`, the `install.ps1` script allows you to specify whether the `ebsnvme-id` tool should be installed with the driver.

   i. To install the `ebsnvme-id` tool, specify `InstallEBSNVMeIdTool ‘Yes’`.

   ii. If you don't want to install the tool, specify `InstallEBSNVMeIdTool ‘No’`.

      If you don't specify `InstallEBSNVMeIdTool`, and the tool is already present at `C:\ProgramData\Amazon\Tools`, the package will upgrade the tool by default. If the tool is not present, `install.ps1` will not upgrade the tool by default.

      If you don't want to install the tool as part of the package, and want to install it later, you can find the latest version or the tool in the driver package. Alternatively, you can download version `1.0.0` from Amazon S3:

      Download the `ebsnvme-id` tool.

5. If the installer does not reboot your instance, reboot the instance.

Distributor

You can use Distributor, a capability of AWS Systems Manager, to install the NVMe driver package one time or with scheduled updates.

To install the latest AWS NVMe driver

1. For the instructions for how to install the NVMe driver package using Distributor, see the procedures in [Install or update packages] in the *Amazon EC2 Systems Manager User Guide*.
2. For **Installation Type**, select **Uninstall and reinstall**.
3. For **Name**, choose **AWSNVMe**.
4. (Optional) For **Additional Arguments**, you can customize the installation by specifying values. The values must be formatted using valid JSON syntax. For examples of how to pass additional arguments for the `aws configure` package, see the [Command document plugin reference].

   a. Starting with AWS NVMe `1.5.0`, the driver supports SCSI persistent reservations for Windows Server 2016 and later. By default, this feature isn't enabled during installation.

      - To enable this feature, specify `{"SSM_EnableSCSIPersistentReservations": "true"}`.
      - If you don't want to enable this feature, specify `{"SSM_EnableSCSIPersistentReservations": "false"}`.

   b. Starting with AWS NVMe `1.5.0`, the `install.ps1` script will always install the `ebsnvme-id` tool.

      (Optional) For versions `1.4.0`, `1.4.1`, and `1.4.2`, the `install.ps1` script allows you to specify whether the ebsnvme-id tool should be installed with the driver.

      - To install the ebsnvme-id tool, specify `{"SSM_InstallEBSNVMeIdTool": "Yes"}`.
      - If you don't want to install the tool, specify `{"SSM_InstallEBSNVMeIdTool": "No"}`.

        If `SSM_InstallEBSNVMeIdTool` is not specified for **Additional Arguments**, and the tool is already present at `C:\ProgramData\Amazon\Tools`, the package will upgrade the tool by default. If the tool is not present, the package will not upgrade the tool by default.

If you don't want to install the tool as part of the package, and want to install it later, you can find the latest version of the tool in the driver package. Alternatively, you can download version `1.0.0` from Amazon S3:

Download the `ebsnvme-id` tool.

5. If the installer does not reboot your instance, reboot the instance.

Configure SCSI persistent reservations for Windows instances

After AWS NVMe driver version `1.5.0` or later has been installed, you can enable or disable SCSI persistent reservations using the Windows registry for Windows Server 2016 and later. You must reboot the instance for these registry changes to take effect.

You can enable SCSI persistent reservations with the following command which sets the `EnableSCSIPersistentReservations` to a value of `1`.

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 1
```

You can disable SCSI persistent reservations with the following command which sets the `EnableSCSIPersistentReservations` to a value of `0`.

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 0
```

AWS NVMe Windows driver version history

The following table shows which AWS NVMe drivers run on each version of Windows Server on Amazon EC2.

Windows Server version	AWS NVMe driver version
Windows Server 2025	latest version

Windows Server version	AWS NVMe driver version
Windows Server 2022	latest version
Windows Server 2019	latest version
Windows Server 2016	latest version
Windows Server 2012 R2	version 1.5.1 and earlier
Windows Server 2012	version 1.5.1 and earlier
Windows Server 2008 R2	version 1.3.2 and earlier
Windows Server 2008	version 1.3.2 and earlier

The following table describes the released versions of the AWS NVMe driver.

Package version	Driver version	Details	Release date
1.6.0	1.6.0	• Updated the install script to use PnPUtil. • Updated ebsnvme-id.exe to use NVMe IOCTL.	25 October 2024
1.5.1	1.5.0	Fixed the install script to create a folder for the `ebsnvme-id` tool if it is not present.	17 November 2023
1.5.0	1.5.0	Added support for Small Computer System Interface (SCSI) persistent reservations for instances running Windows Server 2016 and later. The ebsnvme-id tool (`ebsnvme-id.exe`) is now installed by default.	31 August 2023
1.4.2	1.4.2	Fixed a bug where the AWS NVMe driver did not support instance store volumes on D3 instances.	16 March 2023

Package version	Driver version	Details	Release date
1.4.1	1.4.1	Reports Namespace Preferred Write Granularity (NPGW) for EBS volumes that support this optional NVMe feature. For more information, see section 8.25, "Improving Performance through I/O Size and Alignment Adherence," in the [NVMe Base Specification, version 1.4]().	20 May 2022
1.4.0	1.4.0	• Added support for IOCTLs that allow applications to interact with NVMe devices. This support allows applications to get `IdentifyController`, `IdentifyNamespace`, and `NameSpace` list from the NVMe device. For more information, see [Protocol-specific queries]() in the Microsoft documentation. • The 1.4.0 driver version and the latest `ebsnvme-id` tool (`ebsnvme-id.exe`) are combined in a single package. This combination allows you to install both driver and tool from a single package. For more details, see [AWS NVMe drivers](). • Bug fixes and reliability improvements.	23 November 2021
[1.3.2]()	1.3.2	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. This is the last version that can run on Windows Server 2008 and 2008 R2. This version is available for download but no longer supported. Windows Server 2008 and 2008 R2 has reached end-of-life, and is no longer supported by Microsoft.	10 September 2019

Package version	Driver version	Details	Release date
1.3.1	1.3.1	Reliability Improvements.	21 May 2019
1.3.0	1.3.0	Device optimization improvements.	31 August 2018
1.2.0	1.2.0	Performance and reliability improvements for AWS NVMe devices on all supported instances, including bare metal instances.	13 June 2018
>1.0.0	>1.0.0	AWS NVMe driver for supported instance types running Windows Server.	12 February 2018

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

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select this Region because the SNS notifications that you are subscribing to are in this Region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:

    a. For **TopicARN**, copy the following Amazon Resource Name (ARN):

       arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers

    b. For **Protocol**, choose `Email`.

    c. For **Endpoint**, type an email address that you can use to receive the notifications.

    d. Choose **Create subscription**.

6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Subscriptions**.
3. Select the checkbox for the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

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'
```

Configure your Amazon EC2 Windows instance

After you've launched a Windows instance, you can log in as an administrator to perform additional configuration for Windows features and system settings. EC2 Windows troubleshooting utilities can help you troubleshoot issues on your instance.

You can configure Windows launch agents and other Windows-specific features as follows.

Windows launch agents

Each AWS Windows AMI (and many other AMIs that are available on the AWS Marketplace) includes a Windows launch agent that's pre-configured with default settings. Launch agents perform tasks during instance startup and run if an instance is stopped and later started, or restarted.

EC2 Fast Launch for Windows

Every Amazon EC2 Windows instance must go through the standard Windows operating system (OS) launch steps, which include several reboots, and often take 15 minutes or longer to complete. Amazon EC2 Windows Server AMIs that have the EC2 Fast Launch feature enabled complete some of those steps and reboots in advance to reduce the time it takes to launch an instance.

Windows-specific system settings

The following list includes some system settings that apply only for Windows operating systems:

Change the Windows Administrator password

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 must use the Administrator account and provide the default password. When you connect to an instance the first time, we recommend that you change the Administrator password from its default value.

Add Windows System components

Windows Server operating systems include many optional components. Including all optional components in each AWS Windows Server AMI is not practical. Instead, we provide installation media EBS snapshots that have the necessary files to configure or install components on your Windows instance.

Install WSL on Windows

Windows Subsystem for Linux (WSL) is a free download that you can install on your Windows instance. By installing WSL, you can run native Linux command line tools directly on your Windows instance and use the Linux tools for scripting, alongside your traditional Windows desktop. You can easily swap between Linux and Windows on a single Windows instance, which you might find useful in a development environment.

AWS device drivers for Windows instances

You can update the AWS device drivers for your Windows instances. For more information, see the section called "Manage device drivers".

The following table summarizes the supported drivers for Nitro-based instances by Windows version.

Version	Storage driver	Enhanced networking driver
Windows Server 2025	AWS NVMe latest version	ENA latest version
Windows Server 2022	AWS NVMe latest version	ENA latest version
Windows Server 2019	AWS NVMe latest version	ENA latest version
Windows Server 2016	AWS NVMe latest version	ENA latest version
Windows Server 2012 R2	AWS NVMe version 1.5.1	ENA version 2.6.0
Windows Server 2008 R2	AWS NVMe version 1.3.2	ENA version 2.2.3

The following table summarizes the supported drivers for Xen-based instances by Windows version.

Version	Storage driver	Enhanced networking driver
Windows Server 2022	AWS PV latest version	• ENA latest version [1] • Intel VF [2] • AWS PV latest version [3]
Windows Server 2019	AWS PV latest version	• ENA latest version [1] • Intel VF [2] • AWS PV latest version [3]
Windows Server 2016	AWS PV latest version	• ENA latest version [1]

Version	Storage driver	Enhanced networking driver
		• Intel VF [2] • AWS PV latest version [3]
Windows Server 2012 R2	AWS PV version 8.4.3	• ENA version 2.6.0 [1] • Intel VF [2] • AWS PV version 8.4.3 [3]
Windows Server 2008 R2	AWS PV version 8.3.5	• ENA version 2.2.3 [1] • Intel VF [2] • AWS PV version 8.3.5 [3]

[1] For instance types G3, H1, I3, `m4.16xlarge`, P2, P3, P3dn, and R4.

[2] For instance types C3, C4, D2, I2, M4 (excluding `m4.16xlarge`), and R3.

[3] For instance types C1, M1, M2, M3, T1, T2, X1, and X1e.

Windows launch agents on Amazon EC2 Windows instances

Each AWS Windows AMI includes a Windows launch agent that's pre-configured with default settings. Launch agents perform tasks during instance startup and run if an instance is stopped and later started, or restarted. For information about a specific agent, see the detail pages in the following list.

For more information about AWS Windows AMIs, see the AWS Windows AMI reference.

- Use the EC2Launch v2 agent to perform tasks during EC2 Windows instance launch
- Use the EC2Launch v1 agent to perform tasks during EC2 Windows instance launch
- Use the EC2Config service to perform tasks during EC2 legacy Windows operating system instance launch

Content

- Configure DNS Suffix for EC2 Windows launch agents
- Subscribe to EC2 Windows launch agent notifications
- Windows Service administration for EC2Launch v2 and EC2Config agents

Compare Amazon EC2 launch agents

The following table shows the major functional differences between EC2Config, EC2Launch v1, and EC2Launch v2.

Feature	EC2Config	EC2Launch v1	EC2Launch v2
Run as	Windows Service	PowerShell Scripts	Windows Service
Supports	Legacy OS only	Windows Server versions: • 2016 • 2019 (LTSC and SAC)	Windows Server versions: • 2016 • 2019 (LTSC and SAC) • 2022 • 2025
Configuration file	XML	JSON	JSON/YAML
Set Administrator username	No	No	Yes
Compressed user data	No	No	Yes
Local user data baked on AMI	No	No	Yes, configurable
Task configuration in user data	No	No	Yes
Configurable wallpaper	No	No	Yes

Feature	EC2Config	EC2Launch v1	EC2Launch v2
Customize task run order	No	No	Yes
Configurable tasks	15	9	20 at launch
Supports Windows Event Viewer	Yes	No	Yes
Number of Event Viewer event types	2	0	30

> **Note**
>
> EC2Config documentation is provided for historical reference only. The operating system versions it runs on are no longer supported by Microsoft. We strongly recommend that you upgrade to the latest launch service.

Configure DNS Suffix for EC2 Windows launch agents

With Amazon EC2 launch agents, you can configure a list of DNS suffixes that Windows instances use for domain name resolution. The launch agents override the standard Windows settings in the `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` registry key by adding the following values to the DNS suffix search list:

- The domain of the instance
- The suffixes that result from devolution of the instance domain
- NV domain
- The domains specified by each network interface cards

All launch agents support DNS suffix configuration. For more information, see your specific launch agent version:

- For information about the `setDnsSuffix` task and how to configure DNS suffixes in EC2Launch v2, see setDnsSuffix .

- For information about DNS suffix list setup and how to enable or disable devolution for EC2Launch v1, see [Configure the EC2Launch v1 agent on your Windows instance]().
- For information about DNS suffix list setup and how to enable or disable devolution for EC2Config, see [EC2Config settings files]().

Domain name devolution

Domain name devolution is an Active Directory behavior that allows computers in a child domain to access resources in the parent domain without using a fully qualified domain name. By default, domain name devolution continues until there are only two nodes left in the domain name progression.

Launch agents perform devolution on the domain name if the instance is connected to a domain, and add the results to the DNS suffix search list that's maintained in the **`System\CurrentControlSet\Services\Tcpip\Parameters\SearchList`** registry key. The agents use the settings from the following registry keys, to determine the devolution behavior.

- **`System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution`**
  - When not set, disables devolution
  - When set to `1`, enables devolution (default)
  - When set to `0`, disables devolution
- **`System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`**
  - When not set, use level of 2 (default)
  - When set to `3` or greater, use value to set level

When you disable devolution or change your devolution settings to a higher level, the `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` registry key stil contains the suffixes that were added previously. They are not automatically removed. You can manually update the list, or you can clear the list and let your agent run through the process to set up the new list.

> **Note**
>
> To clear the DNS suffix list from the registry, you can run the following command.

```
PS C:\> Invoke-CimMethod -ClassName Win32_NetworkAdapterConfiguration -
MethodName "SetDNSSuffixSearchOrder" -Arguments @{ DNSDomainSuffixSearchOrder =
 $null } | Out-Null
```

Devolution examples

The following examples show domain name progression through the devolution process.

`corp.example.com`

- Progresses to `example.com`

`locale.region.corp.example.com`

1. Progresses to `region.corp.example.com`
2. Progresses to `corp.example.com`
3. Progresses to `example.com`

`locale.region.corp.example.com` with a setting of `DomainNameDevolutionLevel=3`

1. Progresses to `region.corp.example.com`
2. Progresses to `corp.example.com`. The progression stops here, due to the level setting.

Subscribe to EC2 Windows launch agent notifications

Amazon SNS can notify you when new versions of the EC2 launch agents are released. Use the following procedure to subscribe to these notifications.

Subscribe to EC2Config notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select 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) that matches the agent you want to receive notifications for:

      - **EC2Launch v2**:

        ```
        arn:aws:sns:us-east-1:309726204594:amazon-ec2launch-v2
        ```

      - **EC2Launch or EC2Config**:

        ```
        arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config
        ```

   b. For **Protocol**, choose `Email`.

   c. For **Endpoint**, enter the email address where you want to receive the notifications.

   d. Choose **Create subscription**.
6. You'll receive a email asking you to confirm your subscription. Open the email and follow the directions to complete your subscription.

Whenever a new version of the launch agent is released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

Unsubscribe from launch agent 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**.

Windows Service administration for EC2Launch v2 and EC2Config agents

If you've logged into your instance as a user with administrative rights, you can manage the EC2Launch v2 and EC2Config launch agents just as you would any other Windows service. EC2Launch v1 is a set of PowerShell scripts that is managed via scheduled task by default. This section covers service administration for EC2Launch v2 and EC2Config.

To apply updated settings to your instance, you can stop and restart the EC2Launch v2 agent or the EC2Config service launch agent from the Microsoft Management Console (MMC) interface for Services. Similarly, when you install a new version of the launch agent, you must stop the agent first, then restart it when the installation is complete.

> **Note**
>
> You must open the MMC Services interface as an administrator to select these actions. To do this, you can select **Run as administrator** from the context menu. Alternatively, to open the interface using your keyboard, follow these steps:
>
> 1. Using the `Tab` key or arrow keys, select the **Services** menu item from the **Administrative Tools** menu.
> 2. Use the following keyboard combination to open as an administrator: `Ctrl` + `Shift` + `Enter`.

The following procedures list steps to stop and start the launch agent on your instance.

Stop the launch agent

1. Launch and connect to your Windows instance.
2. Select **Administrative Tools** from the Windows **Start** menu.
3. Open the **Services** console as an administrator, as described at the beginning of this section.
4. In the list of services, select the agent that's running on your instance (**EC2Launch** or **EC2Config**), then choose **Stop** from the **Action** menu. Alternatively, you can use the context menu to stop the agent.

Restart the launch agent

1. Launch and connect to your Windows instance.
2. Select **Administrative Tools** from the Windows **Start** menu.
3. Open the **Services** console as an administrator, as described at the beginning of this section.
4. In the list of services, select the agent that's running on your instance (**EC2Launch** or **EC2Config**), then choose **Start** or **Restart** from the **Action** menu. Alternatively, you can use the context menu to restart the agent.

If you don't need to update the configuration settings, create your own AMI, or use AWS Systems Manager, you can delete or uninstall the launch agent.

Delete

Deleting a service removes its registry subkey.

Uninstall

Uninstalling a service removes the files, the registry subkey, and any shortcuts to the service.

Delete the launch agent

1. Launch and connect to your Windows instance.
2. Start a Windows Command Prompt window.
3. Run one of the following commands to delete the launch agent.

   - Run the following command to delete the EC2Launch or EC2Launch v2:

     ```
     sc delete ec2launch
     ```

   - Run the following command to delete the EC2Config service:

     ```
     sc delete ec2config
     ```

Uninstall the launch agent

1. Launch and connect to your Windows instance.
2. Choose **Windows System**, then **Control Panel** from the Windows **Start** menu.
3. Choose **Programs and Features** to open the list of programs that are installed on your instance.
4. Select your launch agent from the list (**Amazon EC2Launch** or **EC2ConfigService**), then choose **Uninstall** from the **File** menu. Alternatively, you can use the context menu.

   > **Note**
   >
   > You can see what launch agent version is installed in the **Version** column.

Use the EC2Launch v2 agent to perform tasks during EC2 Windows instance launch

All supported instances of Amazon EC2 that are launched from AWS Windows Server 2022 and Windows Server 2025 AMIs include the EC2Launch v2 launch agent (`EC2Launch.exe`) by default. We also provide Windows Server 2016 and 2019 AMIs with EC2Launch v2 installed as the default launch agent. These AMIs are provided in addition to the Windows Server 2016 and 2019 AMIs that include EC2Launch v1. You can search for Windows AMIs that include EC2Launch v2 by default by entering the following prefix in your search from the **AMIs** page in the Amazon EC2 console: `EC2LaunchV2-Windows_Server-*`.

To compare launch agent version features, see Compare Amazon EC2 launch agents.

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`). For more information about file locations, see EC2Launch v2 directory structure.

EC2Launch v2 publishes Windows event logs to help you troubleshoot errors and set triggers. For more information, see Windows event logs.

Supported OS versions

The EC2Launch v2 agent supports the following Windows Server operating system (OS) versions:

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019 (Long-Term Servicing Channel and Semi-Annual Channel)
- Windows Server 2016

Tasks that run by default

The EC2Launch v2 agent runs the following tasks one time only by default during the initial instance launch. Tasks are organized according to the order in which they run within their launch stage.

`Boot` stage

- extendRootPartition

`PreReady` stage

- activateWindows
- setDnsSuffix
- setAdminAccount
- setWallpaper

`PostReady` stage

- startSsm

EC2Launch v2 concepts

The following concepts are useful to understand when considering EC2Launch v2.

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 instance configuration 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. For more information about file locations, see EC2Launch v2 directory structure.

Frequency

Task frequency determines when tasks should run, depending on the boot context. Most tasks have only one allowed frequency. You can specify a frequency for `executeScript` tasks.

You will see the following frequencies in the EC2Launch v2 task configuration.

- 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

Stage

A stage is a logical grouping of tasks that the EC2Launch v2 agent runs. Some tasks can run only in a specific stage. Others can run in multiple stages. When using `agent-config.yml`, you must specify a list of stages, and a list of tasks to run within each stage.

The service runs stages in the following order:

Stage 1: Boot

Stage 2: Network

Stage 3: PreReady

Windows is ready

After the PreReady stage completes, the service sends the `Windows is ready` message to the Amazon EC2 console.

Stage 4: PostReady

User data runs during the *PostReady* stage. Some script versions run before the `agent-config.yml` file *PostReady* stage, and some run after, as follows:

Before `agent-config.yml`

- YAML user data version 1.1
- XML user data

After `agent-config.yml`

- YAML user data version 1.0 (legacy version for backwards compatibility)

For example stages and tasks, see [Example: `agent-config.yml`](#).

When you use user data, you must specify a list of tasks for the launch agent to run. The stage is implied. For example tasks, see [Example: user data](#).

EC2Launch v2 runs the list of tasks in the order that you specify in `agent-config.yml` and in user data. Stages run sequentially. The next stage starts after the previous stage completes. Tasks also run sequentially.

Task

You can invoke a task to perform an action on an instance. You can configure tasks in the `agent-config.yml` file or through user data. For a list of available tasks for EC2Launch v2, see EC2Launch v2 tasks. For task configuration schema and details, see EC2Launch v2 task configuration.

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 How Amazon EC2 handles user data for Windows instances.

EC2Launch v2 task overview

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 from the configuration to the disk.
- Run scripts specified in the EC2Launch v2 config file or from `user-data`. Scripts from `user-data` can be plain text or zipped and provided as a base64 format.
- Run a program with given arguments.
- Set the computer name.
- Send instance information to the Amazon EC2 console.
- Send the RDP certificate thumbprint to the Amazon EC2 console.

- Dynamically extend the operating system partition to include any unpartitioned space.
- Run user data. For more information about specifying user data, see EC2Launch v2 task configuration.
- Set non-persistent static routes to reach the metadata service and AWS KMS servers.
- Set non-boot partitions to `mbr` or `gpt`.
- Start the Systems Manager 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.

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
- `EC2AgentTelemetry.dll` – DLL for supporting EC2 agent telemetry
- `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 default tasks run by the service. Permission to create files in this directory is restricted to the administrator account to prevent privilege escalation.
- `log` – Instance logs

  Logs for the service (`agent.log`), console (`console.log`), performance (`bench.log`), errors (`err.log`), and telemetry (`telemetry.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.
- `wallpaper` – Wallpaper

  This wallpaper images is stored in this directory.

Telemetry

Telemetry is additional information that helps AWS to better understand your requirements, diagnose issues, and deliver features to improve your experience with AWS services.

EC2Launch v2 version `2.1.592` and later collect telemetry, such as usage metrics and errors. This data is collected from the Amazon EC2 instance on which EC2Launch v2 runs. This includes all Windows AMIs owned by AWS.

The following types of telemetry are collected by EC2Launch v2:

- **Usage information** — agent commands, install method, and scheduled run frequency.
- **Errors and diagnostic information** — agent installation error codes, run error codes, and error call stacks.

Examples of collected data from version 2.0.592 through 2.1.1:

```
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentCommandErrorCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AdminPasswordTypeCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IpConflictDetectionCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentErrorLocation=addroutes.go:49
```

Starting with version 2.2.63, EC2 Agent telemetry data is formatted as a JSON object:

```
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2" ... }
```

Telemetry is enabled by default. You can disable telemetry collection at any time.

Disable telemetry on an instance

To disable telemetry for a single instance, you can either set a system environment variable, or use the MSI to modify the installation.

To disable telemetry by setting a system environment variable, run the following command as an administrator.

```
setx /M EC2LAUNCH_TELEMETRY 0
```

To disable telemetry using the MSI, run the following command after you download the MSI.

```
msiexec /i ".\AmazonEC2Launch.msi" Remove="Telemetry" /q
```

More topics for EC2Launch v2

- Install the latest version of EC2Launch v2
- Configure EC2Launch v2 settings for Windows instances
- Task definitions for EC2Launch v2 startup tasks
- Troubleshoot issues with the EC2Launch v2 agent
- EC2Launch v2 version histories

Install the latest version of EC2Launch v2

You can use one of the following methods to install the EC2Launch v2 agent on your EC2 instance:

- Download the agent from Amazon S3 and install with Windows PowerShell. For download URLs, see EC2Launch v2 downloads on Amazon S3.
- Install with SSM Distributor.
- Install from an EC2 Image Builder component when you create a custom image.
- Launch your instance from an AMI that has EC2Launch v2 pre-installed.

> **Warning**
>
> AmazonEC2Launch.msi uninstalls previous versions of the EC2 launch services, such as EC2Launch (v1) and EC2Config.

For install steps, select the tab that matches your preferred method.

PowerShell

To install the latest version of EC2Launch v2 agent with Windows PowerShell, follow these steps.

1. Create your local directory.

```
New-Item -Path "$env:USERPROFILE\Desktop\EC2Launchv2" -ItemType Directory
```

2. Set the URL for your download location. Run the following command with the Amazon S3 URL you'll use. For download URLs, see [EC2Launch v2 downloads on Amazon S3]

```
$Url = "Amazon S3 URL/AmazonEC2Launch.msi"
```

3. Use the following compound command to download the agent and run the install

```
$DownloadFile = "$env:USERPROFILE\Desktop\EC2Launchv2\" + $(Split-Path -Path
 $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadFile
msiexec /i "$DownloadFile"
```

> **Note**
>
> If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

4. The **msiexec** command installs EC2Launch v2 in the following location on Windows Server instances: `%ProgramFiles%\Amazon\EC2Launch`. To verify that the install ran, you can check the local file system on your instance.

AWS Systems Manager Distributor

To configure automatic updates for EC2Launch v2 with AWS Systems Manager Quick Setup, see Automatically install and update with Distributor Quick Setup.

You can also perform a one-time install of the `AWSEC2Launch-Agent` package from AWS Systems Manager Distributor. For instructions on how to install a package from Systems Manager Distributor, see Install or update packages in the *AWS Systems Manager User Guide*.

EC2 Image Builder component

You can install the `ec2launch-v2-windows` component when you build a custom image with EC2 Image Builder. For instructions on how to build a custom image with EC2 Image Builder, see [Create an image pipeline using the EC2 Image Builder console wizard](#) in the *EC2 Image Builder User Guide*.

AMI

EC2Launch v2 is preinstalled by default on AMIs for the Windows Server 2022 and above operating systems:

- Windows_Server-*version*-English-Full-Base
- Windows_Server-*version*-English-Core-Base
- Windows_Server-*version*-English-Core-EKS_Optimized
- Windows Server *version* AMIs with all other languages
- Windows Server *version* AMIs with SQL installed

EC2Launch v2 is also preinstalled on the following Windows Server AMIs. You can find these AMIs from the Amazon EC2 console, or by using the following search prefix: `EC2LaunchV2-` in the AWS CLI.

- EC2LaunchV2-Windows_Server-2019-English-Core-Base
- EC2LaunchV2-Windows_Server-2019-English-Full-Base
- EC2LaunchV2-Windows_Server-2016-English-Core-Base
- EC2LaunchV2-Windows_Server-2016-English-Full-Base

Automatically install and update EC2Launch v2 with AWS Systems Manager Distributor Quick Setup

With AWS Systems Manager Distributor Quick Setup, you can set up automatic updates for EC2Launch v2. The following process sets up a Systems Manager Association on your instance that automatically updates the EC2Launch v2 agent at a frequency that you specify. The Association that the Distributor Quick Setup creates can include instances within an AWS account and Region, or instances within an AWS Organization. For more information about setting up an organization, see [Tutorial: Creating and configuring an organization](#) in the *AWS Organizations User Guide*.

Before you begin, make sure that your instances meet all of the prerequisites.

Prerequisites

To set up automatic updates with Distributor Quick Setup, your instances must meet the following prerequisites.

- You have at least one running instance that supports EC2Launch v2. See supported operating systems for EC2Launch v2.
- You've performed the Systems Manager set-up tasks on your instances. For more information, see Setting up Systems Manager in the *AWS Systems Manager User Guide*.
- EC2Launch v2 must be the only launch agent installed on your instance. If you have more than one launch agent installed, your Distributor Quick Setup configuration will fail. Before you configure EC2Launch v2 with a Distributor Quick Setup, uninstall EC2Config or EC2Launch v1 launch agents, if they exist.

Configure Distributor Quick Setup for EC2Launch v2

To create a configuration for EC2Launch v2 with Distributor Quick Setup, use the following settings when you complete the steps for Distributor package deployment:

- **Software packages:** Amazon EC2Launch v2 agent.
- **Update frequency:** Select a frequency from the list.
- **Targets:** Choose from the available deployment options.

To check the status of your configuration, navigate to the Systems Manager Quick Setup **Configurations** tab in the AWS Management Console.

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose **Quick Setup**.
3. In the **Configurations** tab, select the row that's associated with the configuration that you created. The **Configurations** tab lists your configurations, and includes a summary of key details, such as **Region**, **Deployment status**, and **Association status**.

> **Note**
>
> The association name for every EC2Launch v2 Distributor configuration begins with the following prefix: `AWS-QuickSetup-Distributor-EC2Launch-Agent-`.

4. To view details, select the configuration and choose **View details**.

For more information and troubleshooting steps, see Troubleshooting Quick Setup results in the *AWS Systems Manager User Guide*.

EC2Launch v2 downloads on Amazon S3

To install the latest version of EC2Launch v2, download the installer from the following location:

- https://s3.amazonaws.com/amazon-ec2launch-v2/windows/amd64/latest/AmazonEC2Launch.msi

Configure install options

When you install or upgrade EC2Launch v2, you can configure installation options with the EC2Launch v2 install dialog or with the **msiexec** command in a command line shell.

The first time the EC2Launch v2 installer runs on an instance, it initializes launch agent settings on your instance as follows:

- It creates the local path and writes the launch agent file to it. This is sometimes referred to as a *clean install*.
- It creates the `EC2LAUNCH_TELEMETRY` environment variable if it doesn't already exist, and sets it based on your configuration.

For configuration details, select the tab that matches the configuration method that you'll use.

Amazon EC2Launch Setup dialog

When you install or upgrade EC2Launch v2, you can configure the following installation options through the EC2Launch v2 install dialog.

Basic Install options

Send Telemetry

When you include this feature in the setup dialog, the installer sets the `EC2LAUNCH_TELEMETRY` environment variable to a value of `1`. If you disable **Send Telemetry**, the installer sets the environment variable to a value of `0`.

When the EC2Launch v2 agent runs, it reads the `EC2LAUNCH_TELEMETRY` environment variable to determine whether to upload telemetry data. If the value equals `1`, it uploads the data. Otherwise, it doesn't upload.

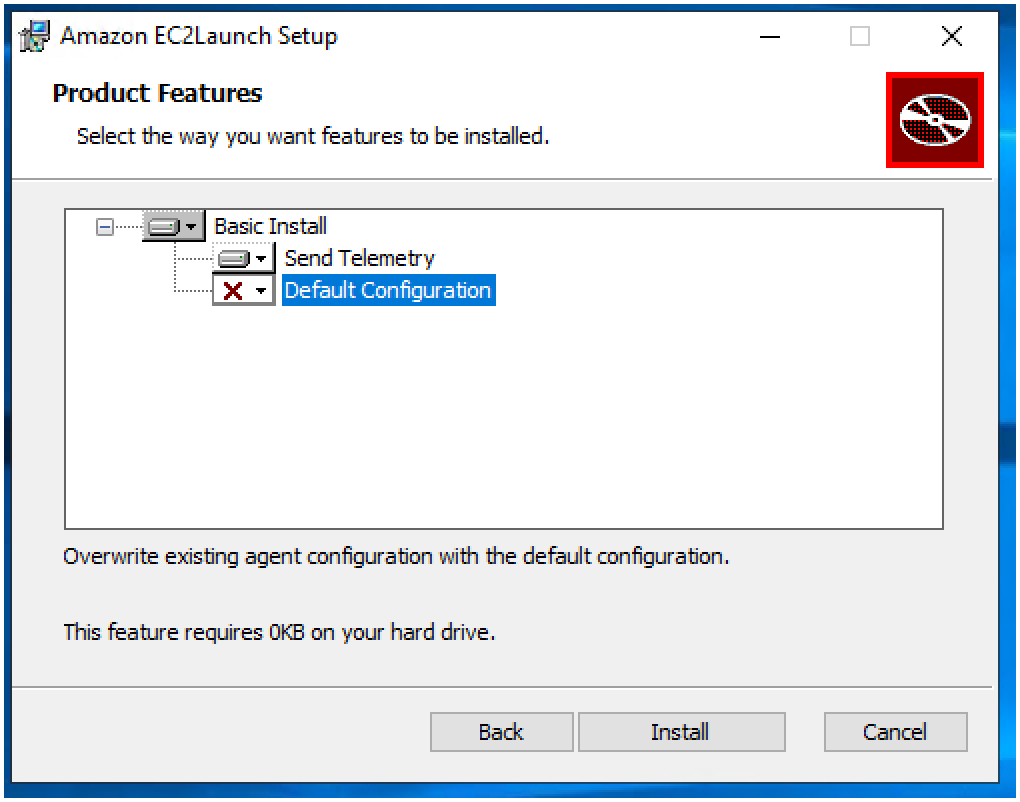
Default configuration

The default configuration for EC2Launch v2 is to overwrite the local launch agent if it exists already. The first time you run an install on an instance, the default configuration performs a clean install. If you disable the default configuration on the initial install, the installation fails.

If you run the install again on the instance, you can disable the default configuration to perform an upgrade that doesn't replace the `%ProgramData%/Amazon/EC2Launch/config/agent-config.yml` file.

Example: Upgrade EC2Launch v2 with telemetry

The following example shows the EC2Launch v2 setup dialog configured to upgrade the current installation and enable telemetry. This configuration performs an install without replacing the agent configuration file, and sets the `EC2LAUNCH_TELEMETRY` environment variable to a value of `1`.



Command line

When you install or upgrade EC2Launch v2, you can configure the following installation options with the **msiexec** command in a command line shell.

`ADDLOCAL` parameter values

Basic (required)

Install the launch agent. If this value is not present in the `ADDLOCAL` parameter, the installation ends.

Clean

When you include the `Clean` value in the `ADDLOCAL` parameter, the installer writes the agent configuration file to the following location: `%ProgramData%/Amazon/EC2Launch/config/agent-config.yml`. If the agent configuration file already exists, it overwrites the file.

When you leave the `Clean` value out of the `ADDLOCAL` parameter, the installer performs an upgrade that doesn't replace the agent configuration file.

Telemetry

When you include the `Telemetry` value in the `ADDLOCAL` parameter, the installer sets the `EC2LAUNCH_TELEMETRY` environment variable to a value of `1`.

When you leave the `Telemetry` value out of the `ADDLOCAL` parameter, the installer sets the environment variable to a value of `0`.

When the EC2Launch v2 agent runs, it reads the `EC2LAUNCH_TELEMETRY` environment variable to determine whether to upload telemetry data. If the value equals `1`, it uploads the data. Otherwise, it doesn't upload.

Example: install EC2Launch v2 with telemetry

```
& msiexec /i "C:\Users\Administrator\Desktop\EC2Launchv2\AmazonEC2Launch.msi"
 ADDLOCAL="Basic,Clean,Telemetry" /q
```

Verify the EC2Launch v2 version

Use one of the following procedures to verify the version of EC2Launch v2 that is installed on your instances.

PowerShell

Verify the installed version of EC2Launch v2 with Windows PowerShell, as follows.

1. Launch an instance from your AMI and connect to it.
2. Run the following command in PowerShell to verify the installed version of EC2Launch v2:

```
& "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" version
```

Windows Control Panel

Verify the installed version of EC2Launch v2 in the Windows Control Panel, as follows.

1. Launch an instance from your AMI and connect to it.
2. Open the Windows Control Panel and choose **Programs and Features**.

3. Look for `Amazon EC2Launch` in the list of installed programs. Its version number appears in the **Version** column.

To view the latest updates for the AWS Windows AMIs, see Windows AMI version history in the *AWS Windows AMI Reference*.

For the latest version of EC2Launch v2, see EC2Launch v2 version history.

You can receive notifications when new versions of the EC2Launch v2 service are released. For more information, see Subscribe to EC2 Windows launch agent notifications.

Configure EC2Launch v2 settings for Windows instances

This section contains information about how to configure settings for EC2Launch v2.

Topics include:

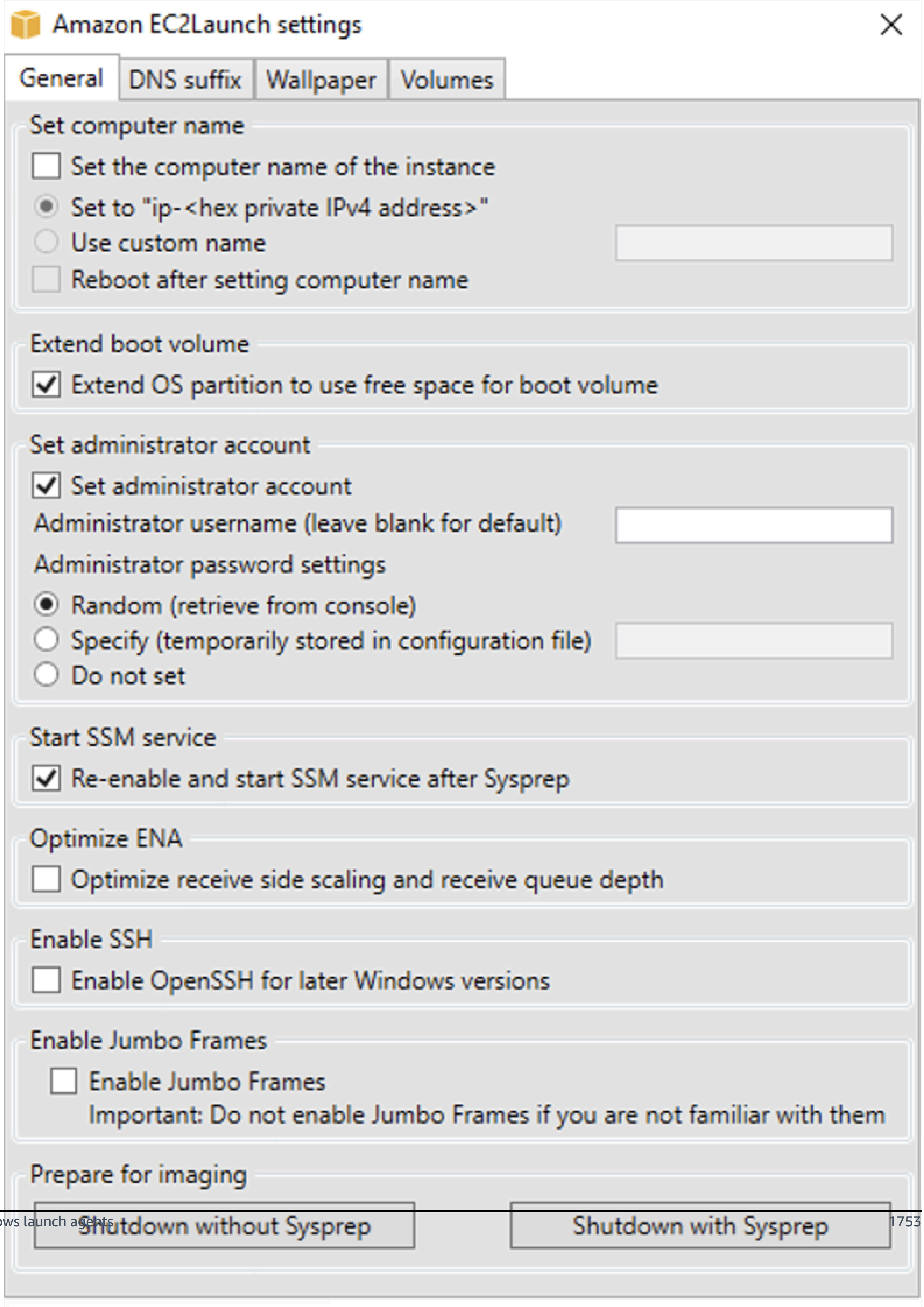
- Change settings using the EC2Launch v2 settings dialog box
- Configure EC2Launch v2 using the CLI
- EC2Launch v2 task configuration
- EC2Launch v2 exit codes and reboots
- EC2Launch v2 and Sysprep

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.

> **Note**
>
> If you improperly configure custom tasks in the agent-config.yml file, and you attempt to open the Amazon EC2Launch settings dialog box, you will receive an error. For example schema, see Example: `agent-config.yml`.

1. Launch and connect to your Windows instance.
2. From the Start menu, choose **All Programs**, and then navigate to **EC2Launch settings**. Before you choose **Shutdown with Sysprep** or **Shutdown without Sysprep**, make sure that you save any changes that you want to apply when you run the shutdown.


Amazon EC2Launch settings
General
DNS suffix
Wallpaper
Volumes
Set computer name
Set the computer name of the instance
Set to "ip-<hex private IPv4 address>"
Use custom name
Reboot after setting computer name
Extend boot volume
Extend OS partition to use free space for boot volume
Set administrator account
Set administrator account
Administrator username (leave blank for default)
Administrator password settings
Random (retrieve from console)
Specify (temporarily stored in configuration file)
Do not set
Start SSM service
Re-enable and start SSM service after Sysprep
Optimize ENA
Optimize receive side scaling and receive queue depth
Enable SSH
Enable OpenSSH for later Windows versions
Enable Jumbo Frames
Enable Jumbo Frames
Important: Do not enable Jumbo Frames if you are not familiar with them
Prepare for imaging
Shutdown without Sysprep
Shutdown with Sysprep

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. `Do not set`

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, 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 using Run Command in the *AWS Systems Manager User Guide*.

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 Receive side scaling CPU affinity.

f. **Enable SSH**

This setting enables OpenSSH for later Windows versions to allow for remote system administration.

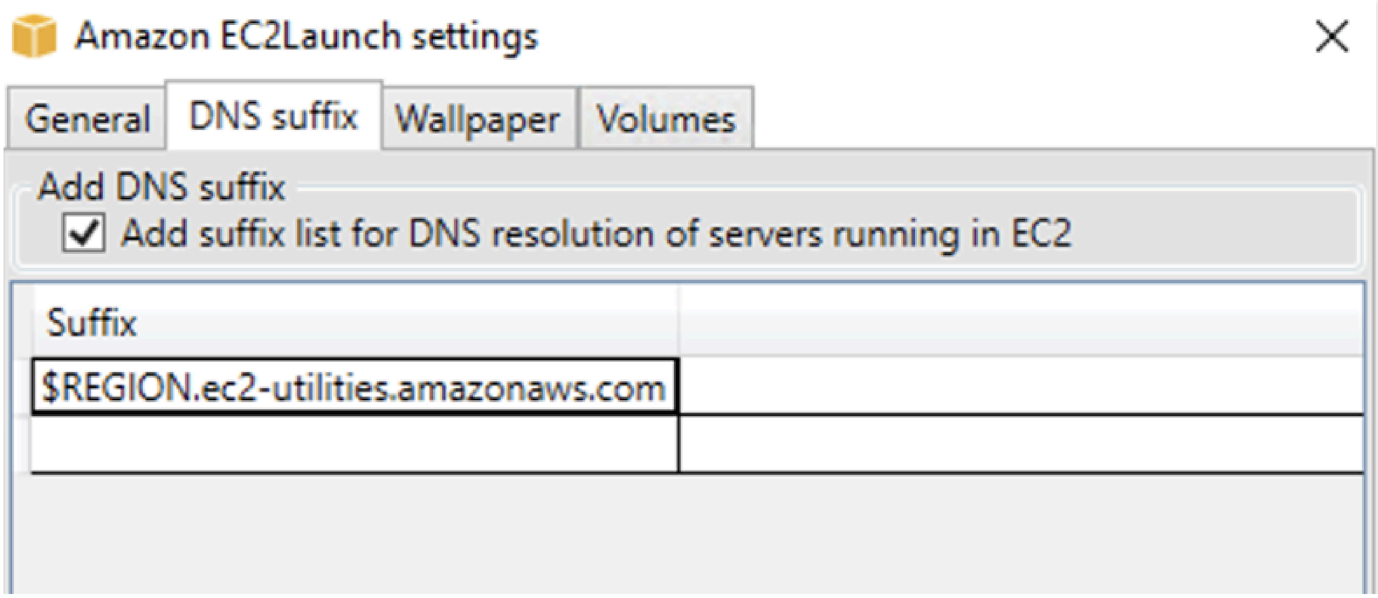
g. **Enable Jumbo Frames**

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).

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.

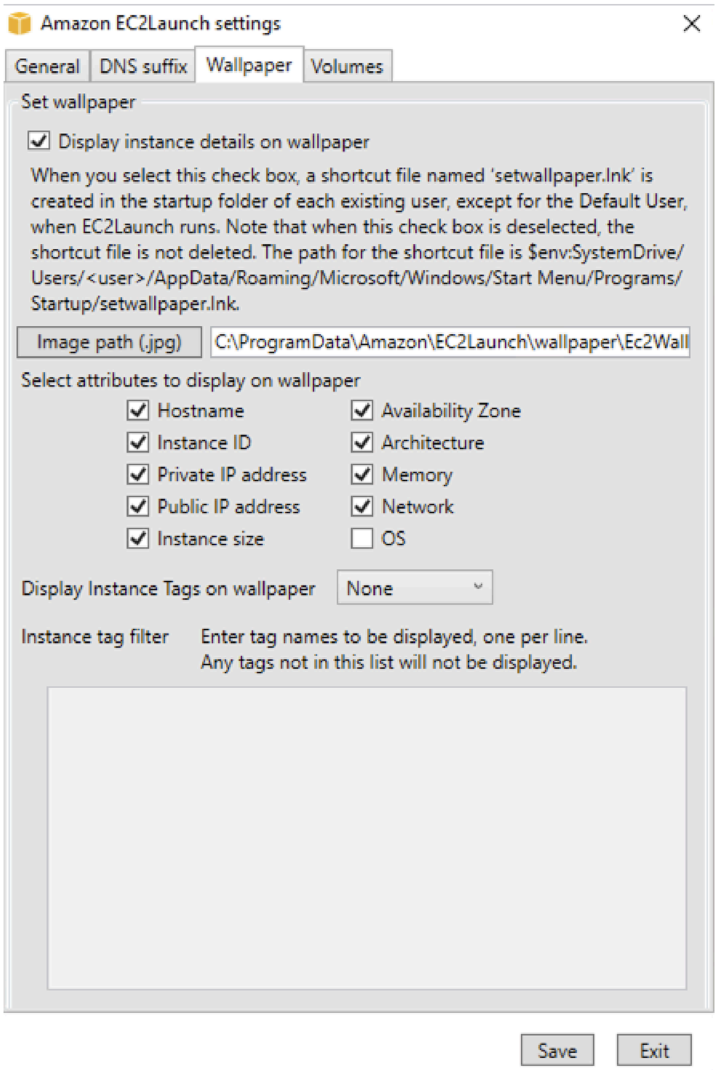


5. On the **Wallpaper** tab, you can configure your instance wallpaper with a background image, and specify instance details for the wallpaper to display. Amazon EC2 generates the details each time you log in.

   You can configure your wallpaper with the following controls.

   - **Display instance details on wallpaper** – This checkbox activates or deactivates instance detail display on the wallpaper.
   - **Image path (.jpg)** – Specify the path to the image to use as the wallpaper background.
   - **Select attributes to display on wallpaper** – Select the check boxes for the instance details that you want to appear on the wallpaper. Clear the check boxes for previously selected instance details that you want to remove from the wallpaper.
   - **Display Instance Tags on wallpaper** – Select one of the following settings to display instance tags on the wallpaper:
     - **None** – Don't display any instance tags on the wallpaper.
     - **Show all** – Display all instance tags on the wallpaper.
     - **Show filtered** – Display specified instance tags on the wallpaper. When you select this setting, you can add instance tags that you want to display on your wallpaper in the **Instance tag filter** box.

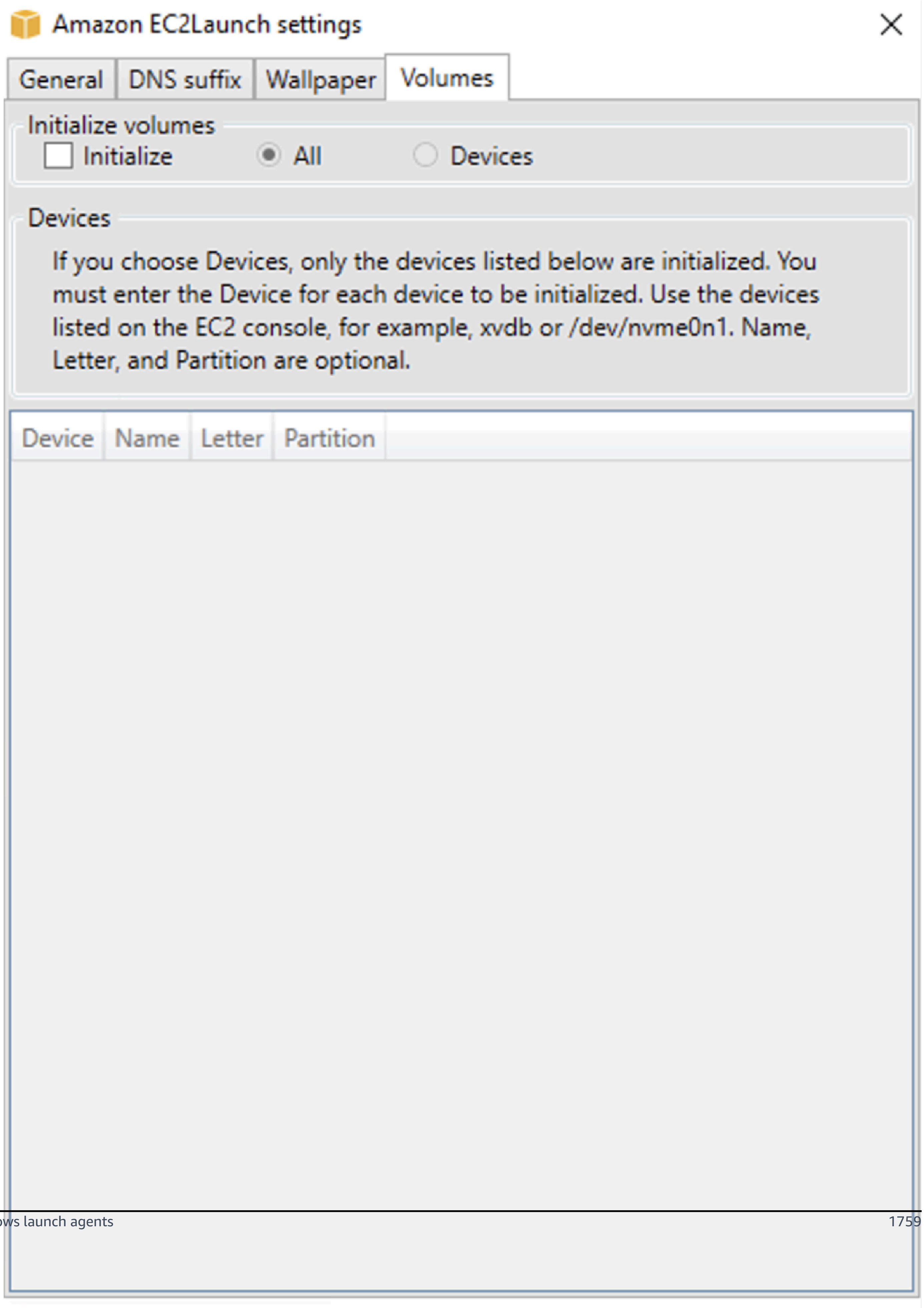
> **Note**
>
> You must enable tags in metadata to show tags on the wallpaper. For more information about instance tags and metadata, see [View tags for your EC2 instances using instance metadata](#).



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.
Device
Name
Letter
Partition

The following is an example configuration YAML file created from the settings entered in the EC2Launch dialog.

```
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
```

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
- get-agent-config
- list-volumes
- reset
- run
- status
- sysprep
- validate
- version
- wallpaper

collect-logs

Collects log files for EC2Launch, zips the files, and places them in a specified directory.

Example

```
ec2launch collect-logs -o C:\Mylogs.zip
```

Usage

`ec2launch 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

```
ec2launch get-agent-config -f json
```

Usage

`ec2launch get-agent-config [flags]`

Flags

`-h`, `--help`

help for `get-agent-config`

`-f`, `--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

The main goal of this task is to reset the agent for the next time that it runs. To do that, the **reset** command deletes all of the agent state data for EC2Launch v2 from the local `EC2Launch` directory (see EC2Launch v2 directory structure). Reset optionally deletes the service and Sysprep logs.

Script behavior depends on what mode the agent runs the scripts in – inline, or detached.

Inline (default)

The EC2Launch v2 agent runs scripts one at a time (`detach: false`). This is the default setting.

> **Note**
>
> When your inline script issues a **reset** or **sysprep** command, it runs immediately and resets the agent. The current task finishes, then the agent shuts down without running any further tasks.
> For example, if the task that issues the command would have been followed by a `startSsm` task (included by default after user data runs), the task doesn't run and the Systems Manager service never starts.

Detached

The EC2Launch v2 agent runs scripts concurrently with other tasks (`detach: true`).

> **Note**
>
> When your detached script issues a **reset** or **sysprep** command, those commands wait for the agent to finish before they run. Tasks after the executeScript will still run.

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

`-h`, `--help`

help for `run`

status

Gets the status of the EC2Launch v2 agent. Optionally blocks the process until the agent is finished. The process exit code determines the agent state:

- `0` –the agent ran and was successful.
- 1 – the agent ran and failed.
- 2 – the agent is still running.
- 3 – the agent is in an unknown state. The agent state is not running or stopped.
- 4 – an error occurred when attempting to retrieve the agent state.
- 5 – the agent is not running and the status of the last known run is unknown. This could mean one of the following:

- both the `state.json` and `previous-state.json` are deleted.
- the `previous-state.json` is corrupted.

This is the agent state after running the `reset` command.

Example:

```
ec2launch status -b
```

Usage

`ec2launch status [flags]`

Flags

`-b`,`--block`

blocks the process until the agent finishes running

`-h`,`--help`

help for `status`

sysprep

The main goal of this task is to reset the agent for the next time that it runs. To do that, the **sysprep** command resets the agent state, updates the `unattend.xml` file, disables RDP, and runs Sysprep.

Script behavior depends on what mode the agent runs the scripts in – inline, or detached.

Inline (default)

The EC2Launch v2 agent runs scripts one at a time (`detach: false`). This is the default setting.

> **Note**
>
> When your inline script issues a **reset** or **sysprep** command, it runs immediately and resets the agent. The current task finishes, then the agent shuts down without running any further tasks.

For example, if the task that issues the command would have been followed by a `startSsm` task (included by default after user data runs), the task doesn't run and the Systems Manager service never starts.

Detached

The EC2Launch v2 agent runs scripts concurrently with other tasks (`detach: true`).

> **Note**
>
> When your detached script issues a **reset** or **sysprep** command, those commands wait for the agent to finish before they run. Tasks after the executeScript will still run.

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\EC2Launch\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.

Syntax

```
ec2launch wallpaper ^
--path="C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg" ^
--all-tags ^
--
attributes=hostName,instanceId,privateIpAddress,publicIpAddress,instanceSize,availabilityZone,a
```

Inputs

Parameters

--allowed-tags [*`tag-name-1`*, *`tag-name-n`*]

(Optional) Base64 encoded JSON array of instance tag names to display on the wallpaper. You can use this tag or the `--all-tags`, but not both.

--attributes *`attribute-string-1`*, *`attribute-string-n`*

(Optional) A comma-separated list of `wallpaper` attribute strings to apply settings to the wallpaper.

[--path | -p] *`path-string`*

(Required) Specifies the `wallpaper` background image file path.

Flags

--all-tags

(Optional) Displays all of the instance tags on the wallpaper. You can use this tag or the `--allowed-tags`, but not both.

[--help | -h]

Displays help for the **wallpaper** command.

EC2Launch v2 task configuration

This section includes the configuration schema, tasks, details, and examples for `agent-config.yml` and user data.

Tasks and examples

- Schema: agent-config.yml
- Configure EC2Launch v2 user data scripts that run during launch or reboot

Schema: `agent-config.yml`

The structure of the `agent-config.yml` file is shown below. Note that a task cannot be repeated in the same stage. For task properties, see the task descriptions that follow.

Document structure: agent-config.yml

JSON

```
{
 "version": "1.1",
 "config": [
  {
   "stage": "string",
   "tasks": [
    {
     "task": "string",
     "inputs": {
      ...
     }
    },
    ...
   ]
  },
  ...
 ]
}
```

YAML

```
version: 1.1
config:
- stage: string
  tasks:
  - task: string
 inputs:
    ...
  ...
...
```

Example: `agent-config.yml`

The following example shows settings for the `agent-config.yml` configuration file.

```
version: 1.1
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
```

Configure EC2Launch v2 user data scripts that run during launch or reboot

The following JSON and YAML examples show the document structure for user data. Amazon EC2 parses each task named in the `tasks` array that you specify in the document. Each task has its own set of properties and requirements. For details, see the Task definitions for EC2Launch v2 startup tasks.

> **Note**
>
> A task must only appear once in the user data tasks array.

Document structure: user data

JSON

```
{
 "version": "1.1",
 "tasks": [
  {
   "task": "string",
   "inputs": {
    ...
   },
  },
  ...
 ]
}
```

YAML

```
version: 1.1
tasks:
- task: string
  inputs:
    ...
...
```

Example: user data

For more information about user data, see [How Amazon EC2 handles user data for Windows instances](#).

The following YAML document example shows a PowerShell script that EC2Launch v2 runs as user data to create a file.

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
```

```
    content: |-
      New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

You can use an XML format for the user data that's compatible with previous versions of the launch agent. EC2Launch v2 runs the script as an `executeScript` task in the `UserData` stage. To conform with EC2Launch v1 and EC2Config behavior, the user data script runs as an attached/ inline process by default.

You can add optional tags to customize how your script runs. For example, to run the user data script when the instance reboots in addition to one time when the instance launches, you can use the following tag:

`<persist>true</persist>`

Example:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

You can specify one or more PowerShell arguments with the `<powershellArguments>` tag. If no arguments are passed, EC2Launch v2 adds the following argument by default: `-ExecutionPolicy Unrestricted`.

Example:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

To run an XML user data script as a detached process, add the following tag to your user data.

`<detach>true</detach>`

Example:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

> **Note**
>
> The detach tag is not supported on previous launch agents.

Change log: user data

The following table lists changes for user data, and cross-references them to the EC2Launch v2 agent version that applies.

User data version	Details	Introduced in
1.1	• User data tasks run before the `PostReady` stage in the agent config file. • Runs user data before starting the Systems Manager Agent (same behavior as EC2Launch v1 and EC2Config).*	EC2Launch v2 version 2.0.1245
1.0	• Will be deprecated. • User data tasks run after the `PostReady` stage in the agent config file. This is not backwards compatible with EC2Launch v1. • Impacted by a race condition between Systems Manager Agent start and user data tasks.	EC2Launch v2 version 2.0.0

* When used with the default `agent-config.yml` file.

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 an XML 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 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 an Amazon EBS-backed AMI.

Task definitions for EC2Launch v2 startup tasks

Each task that EC2Launch v2 runs during launch or statup has its own set of properties and requirements. Task details include settings for how often a task runs – once, or always, what stage of the agent boot process it runs in, syntax, and YAML document examples. For more information, review the task details shown in this reference.

EC2Launch v2 Tasks

- activateWindows
- enableJumboFrames
- enableOpenSsh
- executeProgram
- executeScript
- extendRootPartition
- initializeVolume
- optimizeEna
- setAdminAccount
- setDnsSuffix
- setHostName
- setWallpaper
- startSsm
- sysprep
- writeFile

activateWindows

Activates Windows against a set of AWS KMS servers. Activation is skipped if the instance is detected as Bring-Your-Own-License (BYOL).

Frequency — once

AllowedStages — `[PreReady]`

Inputs —

`activation`: (map)

`type`: (string) activation type to use, set to `amazon`

Example

```
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).

Frequency — always

AllowedStages — `[PostReady, UserData]`

Inputs — none

Example

```
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 and later versions.

```
task: enableOpenSsh
```

executeProgram

Runs a program with optional arguments and a specified frequency.

Stages: You can run the `executeProgram` task during the `PreReady`, `PostReady`, and `UserData` stages.

Frequency: configurable, see *Inputs*.

Inputs

This section contains one or more programs for the **executeProgram** task to run (inputs). Each input can include the following configurable settings:

frequency (string)

(Required) Specify exactly one of the following values:

- `once`
- `always`

path (string)

(Required) The file path for the executable to run.

arguments (list of strings)

(Optional) A comma separated list of arguments to provide to the program as input.

runAs (string)

(Required) Must be set to `localSystem`

Output

All of the tasks write logfile entries to the `agent.log` file. Additional output from the `executeProgram` task is stored separately in a dynamically named folder, as follows:

`%LocalAppData%\Temp\`*`EC2Launch#########`*`\`*`outputfilename.tmp`*

The exact path to the output files is included in the `agent.log` file, for example:

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\ExecuteProgramInputs.tmp
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

Output files for the `executeProgram` task

`ExecuteProgramInputs.tmp`

Contains the path for the executable, and all of the input parameters that the `executeProgram` task passes to it when it runs.

`Output.tmp`

Contains runtime output from the program that the `executeProgram` task runs.

`Err.tmp`

Contains runtime error messages from the program that the `executeProgram` task runs.

Examples

The following examples show how to run an executable file from a local directory on an instance with the `executeProgram` task.

Example 1: Setup executable with one argument

This example shows an `executeProgram` task that runs a setup executable in quiet mode.

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\Users\Administrator\Desktop\setup.exe
      arguments: ['-quiet']
```

Example 2: VLC executable with two arguments

This example shows an `executeProgram` task that runs a VLC executable file with two arguments passed as input parameters.

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\vlc-3.0.11-win64.exe
      arguments: ['/L=1033','/S']
      runAs: localSystem
```

executeScript

Runs a script with optional arguments and a specified frequency. Script behavior depends on what mode the agent runs the scripts in – inline, or detached.

Inline (default)

The EC2Launch v2 agent runs scripts one at a time (`detach: false`). This is the default setting.

> **Note**
>
> When your inline script issues a **reset** or **sysprep** command, it runs immediately and resets the agent. The current task finishes, then the agent shuts down without running any further tasks.
> For example, if the task that issues the command would have been followed by a `startSsm` task (included by default after user data runs), the task doesn't run and the Systems Manager service never starts.

Detached

The EC2Launch v2 agent runs scripts concurrently with other tasks (`detach: true`).

> **Note**
>
> When your detached script issues a **reset** or **sysprep** command, those commands wait for the agent to finish before they run. Tasks after the executeScript will still run.

Stages: You can run the `executeScript` task during the `PreReady`, `PostReady`, and `UserData` stages.

Frequency: configurable, see *Inputs*.

Inputs

This section contains one or more scripts for the **executeScript** task to run (inputs). Each input can include the following configurable settings:

frequency (string)

(Required) Specify exactly one of the following values:

- `once`
- `always`

type (string)

(Required) Specify exactly one of the following values:

- `batch`
- `powershell`

arguments (list of strings)

(Optional) A list of string arguments to pass to the shell (not to the PowerShell script). This parameter isn't supported for `type: batch`. If no arguments are passed, EC2Launch v2 adds the following argument by default: `-ExecutionPolicy Unrestricted`.

content (string)

(Required) Script content.

runAs (string)

(Required) Specify exactly one of the following values:

- `admin`
- `localSystem`

detach (Boolean)

(Optional) The EC2Launch v2 agent defaults to run scripts one at a time (`detach: false`). To run the script concurrently with other tasks, set the value to `true` (`detach: true`).

> **Note**
>
> Script exit codes (including `3010`) have no effect when `detach` is set to `true`.

Output

All of the tasks write logfile entries to the `agent.log` file. Additional output from script that the `executeScript` task runs is stored separately in a dynamically named folder, as follows:

`%LocalAppData%\Temp\`*`EC2Launch#########`*`\`*`outputfilename.ext`*

The exact path to the output files is included in the `agent.log` file, for example:

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\UserScript.ps1
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

Output files for the `executeScript` task

`UserScript.`*`ext`*

Contains the script that the `executeScript` task ran. The file extension depends on the type of script you specified in the `type` parameter for the `executeScript` task, as follows:

- If the type is `batch`, then the file extension is `.bat`.
- If the type is `powershell`, then the file extension is `.ps1`.

`Output.tmp`

Contains runtime output from the script that the `executeScript` task runs.

`Err.tmp`

Contains runtime error messages from the script that the `executeScript` task runs.

Examples

The following examples show how to run an inline script with the `executeScript` task.

Example 1: Hello world output text file

This example shows an `executeScript` task that runs a PowerShell script to create a "Hello world" text file on the `C:` drive.

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: admin
      content: |-
        New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

```
        Set-Content 'C:\PowerShellTest.txt' "Hello world"
```

Example 2: Run two scripts

This example shows that the `executeScript` task can run more than one script, and the script type doesn't necessarily need to match.

The first script (`type: powershell`) writes a summary of the processes that are currently running on the instance to a text file located on the `C:` drive.

The second script (`batch`) writes the system information to the `Output.tmp` file.

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |
        Get-Process | Out-File -FilePath C:\Process.txt
    - frequency: always
      type: batch
      runAs: localSystem
      content: |
        systeminfo
```

Example 3: Idempotent system configuration with reboots

This example shows an `executeScript` task that runs an idempotent script to perform the following system configuration with a reboot between each step:

- Rename the computer.
- Join the computer to the domain.
- Enable Telnet.

The script ensures that each operation runs one time only. This prevents a reboot loop and makes the script idempotent.

```
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 empty volumes that are attached to the instance so that they're activated and partitioned. The launch agent skips initialization if it detects that the volume is not empty. A volume is considered empty if the first 4 KiB of the volume are empty, or if the volume doesn't have a Windows-recognizable drive layout.

The `letter` input parameter is always applied when this task runs, regardless of whether the drive is already initialized.

The `initializeVolume` task performs the following actions.

- Set disk attributes `offline` and `readonly` to false.
- Create a partition. If no partition type is specified in the `partition` input parameter, the following defaults apply:
  - If the disk size is smaller than 2 TB, set the partition type to `mbr`.
  - If the disk size is 2 TB or larger, set the partition type to `gpt`.
- Format the volume as NTFS.
- Set the volume label as follows:
  - Use the value of the `name` input parameter, if specified.
  - If the volume is ephemeral, and no name was specified, set the volume label to `Temporary Storage Z`.
- If the volume is ephemeral (SSD or HDD – not Amazon EBS), create an `Important.txt` file at the root of the volume with the following content:

```
This is an 'Instance Store' disk and is provided at no additional charge.

*This disk offers increased performance since it is local to the host
*The number of Instance Store disks available to an instance vary by instance type
*DATA ON THIS DRIVE WILL BE LOST IN CASES OF IMPAIRMENT OR STOPPING THE INSTANCE.
 PLEASE ENSURE THAT ANY IMPORTANT DATA IS BACKED UP FREQUENTLY

For more information, please refer to: Instance store temporary block storage for EC2
 instances.
```

- Set the drive letter to the value specified in the `letter` input parameter.

Stages: You can run the `initializeVolume` task during the `PostReady` and `UserData` stages.

Frequency: always.

Inputs

You can configure runtime parameters as follows:

devices (list of maps)

(Conditional) Configuration for each device that the launch agent initializes. This is required if the `initialize` input parameter is set to `devices`.

- **device (string, required) –** Identifies the device during instance creation. For example, `xvdb`, `xvdf`, or `\dev\nvme0n1`.
- **letter (string, optional) –** One character. The drive letter to assign.
- **name (string, optional) –** The volume name to assign.
- **partition (string, optional) –** Specify one of the following values for the type of partition to create, or let the launch agent default based on volume size:
  - mbr
  - gpt

initialize (string)

(Required) Specify exactly one of the following values:

- `all`
- `devices`

Examples

The following examples show sample input configurations for the `initializeVolume` task.

Example 1: Initialize two volumes on an instance

This example shows an `initializeVolume` task that initializes two secondary volumes on an instance. The device named `DataVolume2` in the example is ephemeral.

```
task: initializeVolume
inputs:
  initialize: devices
  devices:
  - device: xvdb
    name: DataVolume1
    letter: D
    partition: mbr
  - device: /dev/nvme0n1
    name: DataVolume2
    letter: E
    partition: gpt
```

Example 2: Initialize EBS volumes attached to an instance

This example shows an `initializeVolume` task that initializes all empty EBS volumes that are attached to the instance.

```
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

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

```
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. For more information about how launch agents set DNS suffixes, see Configure DNS Suffix for EC2 Windows launch agents.

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

```
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

Creates the `setwallpaper.lnk` shortcut file in the startup folder of each existing user except for `Default User`. This shortcut file runs when the user logs in for the first time after instance boot. It sets up the instance with a custom wallpaper that displays the instance attributes.

The shortcut file path is:

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/
Startup/setwallpaper.lnk
```

> **Note**
>
> When you remove the `setWallpaper` task, it does not delete this shortcut file. For more information, see `setWallpaper` task is not enabled but the wallpaper resets at reboot.

Stages: You can configure wallpaper during the `PreReady` and `UserData` stages.

Frequency: `always`

Wallpaper configuration

You can use the following settings to configure your wallpaper.

Inputs

Input parameters that you provide, and attributes that you can set to configure your wallpaper:

attributes (list of strings)

(Optional) You can add one or more of the following attributes to your wallpaper:

- `architecture`
- `availabilityZone`
- `hostName`
- `instanceId`
- `instanceSize`
- `memory`
- `network`
- `privateIpAddress`
- `publicIpAddress`

instanceTags

(Optional) You can use exactly one of the following options for this setting.

- **AllTags** (string) – Add all instance tags to your wallpaper.

```
instanceTags: AllTags
```

- **instanceTags** (list of strings) – Specify a list of instance tag names to add to your wallpaper. For example:

```
instanceTags:
  - Tag 1
  - Tag 2
```

path (string)

(Required) The filename path of the local .jpg format image file to use for your wallpaper image.

Example

The following example shows wallpaper configuration inputs that set the file path for the wallpaper background image, along with instance tags named `Tag 1` and `Tag 2`, and attributes that include the host name, instance ID, and private and public IP addresses for the instance.

```
task: setWallpaper
inputs:
```

```
  path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
  attributes:
  - hostName
  - instanceId
  - privateIpAddress
  - publicIpAddress
instanceTags:
  - Tag 1
  - Tag 2
```

Note

You must enable tags in metadata to show tags on the wallpaper. For more information about instance tags and metadata, see View tags for your EC2 instances using instance metadata.

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
```

Troubleshoot issues with the EC2Launch v2 agent

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

- Windows event logs
- EC2Launch v2 console log output

Common troubleshooting scenarios

This section shows common troubleshooting scenarios and steps for resolution.

Scenarios

- Service fails to set the wallpaper
- Service fails to run user data
- Service runs a task only one time
- Service fails to run a task
- Service runs user data more than once
- Scheduled tasks from EC2Launch v1 fail to run after migration to EC2Launch v2
- Service initializes an EBS volume that is not empty
- setWallpaper task is not enabled but the wallpaper resets at reboot
- Service stuck in running status
- Invalid agent-config.yml prevents opening EC2Launch v2 settings dialog box
- task:executeScript should be unique and only invoked once

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` file, or use the EC2Launch v2 settings dialog box. 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.

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 (see step 6) or in the `agent-config.yml`.

`setWallpaper` task is not enabled but the wallpaper resets at reboot

The `setWallpaper` task creates the `setwallpaper.lnk` shortcut file in the startup folder of each existing user. This shortcut file runs when the user logs in for the first time after instance boot. It sets up the instance with a custom wallpaper that displays the instance attributes. Removing the `setWallpaper` task does not delete this shortcut file. You must manually delete this file or delete it using a script.

The shortcut path is:

`$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/setwallpaper.lnk`

Resolution

Manually delete this file, or delete it using a script.

Example PowerShell script to delete shortcut file

```
foreach ($userDir in (Get-ChildItem "C:\Users" -Force -Directory).FullName)
{
 $startupPath = Join-Path $userDir -ChildPath "AppData\Roaming\Microsoft\Windows\Start
 Menu\Programs\Startup"
 if (Test-Path $startupPath)
 {
  $wallpaperSetupPath = Join-Path $startupPath -ChildPath "setwallpaper.lnk"
  if (Test-Path $wallpaperSetupPath)
  {
   Remove-Item $wallpaperSetupPath -Force -Confirm:$false
  }
 }
}
```

Service stuck in running status

Description

EC2Launch v2 is blocked, with log messages (`agent.log`) similar to the following:

```
2022-02-24 08:08:58 Info:
 ******************************************************************************************
```

```
2022-02-24 08:08:58 Info: EC2Launch Service starting
2022-02-24 08:08:58 Info: Windows event custom log exists: Amazon EC2Launch
2022-02-24 08:08:58 Info: ACPI SPCR table not supported. Bailing Out
2022-02-24 08:08:58 Info: Serial port is in use. Waiting for Serial Port...
2022-02-24 08:09:00 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:02 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:04 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:06 Info: ACPI SPCR table not supported. Use default console port.
```

Possible cause

SAC is enabled and using the serial port. For more information, see Use SAC to troubleshoot your Windows instance.

Resolution

Try the following steps to resolve this issue:

- Disable the service that is using the serial port.
- If you want the service to continue to use the serial port, write custom scripts to perform launch agent tasks and invoke them as scheduled tasks.

Invalid `agent-config.yml` prevents opening EC2Launch v2 settings dialog box

Description

EC2Launch v2 settings attempts to parse the `agent-config.yml` file before it opens the dialog box. If the YAML configuration file does not follow the supported schema, the dialog box will show the following error:

```
Unable to parse configuration file agent-config.yml. Review configuration
file. Exiting application.
```

Resolution

1. Verify that the configuration file follows the supported schema.
2. If you want to start from scratch, copy the default configuration file into `agent-config.yml`. You can use the example `agent-config.yml` provided in the Task Configuration section.
3. You can also start over by deleting `agent-config.yml`. EC2Launch v2 settings generates an empty configuration file.

`task:executeScript should be unique and only invoked once`

Description

A task cannot be repeated in the same stage.

Resolution

Some tasks must be input as an array, such as **executeScript** and **executeProgram**. For an example of how to write the script as an array, see **executeScript**.

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.

Event	Least significant digit
`Success`	`. . .0`
`Informational`	`. . .1`
`Warning`	`. . .2`
`Error`	`. . .3`

Service-related events that are generated when the service starts or stops include a single digit event identifier.

Event	Single digit identifier
`Success`	`0`
`Informational`	`1`

Event	Single digit identifier
`Warning`	2
`Error`	3

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
- Event ID examples
- Windows event log schema

Event ID format

The following table shows the format of an EC2Launch v2 event identifier.

3	2 1	0
S	T	L

The letters and numbers in the table represent the following event type and definitions.

Event type	Definition
S (Stage)	0 - Service-level message 1 - Boot 2 - Network 3 - PreReady

Event type	Definition
	5 - Windows is Ready 6 - PostReady 7 - User Data
T (Task)	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](#).
L (Level of the event)	0 - Success 1 - Informational 2 - Warning 3 - Error

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

MessageId/Event Id	Event message
`. . .0`	`Success`
`. . .1`	`Informational`

MessageId/Event Id	Event message
. . .2	Warning
. . .3	Error
x	EC2Launch service-level logs
0	EC2Launch service exited successfully
1	EC2Launch service informational logs
2	EC2Launch service warning logs
3	EC2Launch service error logs
10	Replace state.json with previous-state.json
100	Serial Port
200	Sysprep
300	PrimaryNic
400	Metadata
x000	Stage (1 digit), Task (2 digits), Status (1 digit)
1000	Boot
1010	Boot - extend_root_partition
2000	Network
2010	Network - add_routes
3000	PreReady

MessageId/Event Id	Event message
3010	PreReady - activate_windows
3020	PreReady - install_egpu_manager
3030	PreReady - set_monitor_on
3040	PreReady - set_hibernation
3050	PreReady - set_admin_account
3060	PreReady - set_dns_suffix
3070	PreReady - set_wallpaper
3080	PreReady - set_update_schedule
3090	PreReady - output_log
3100	PreReady - enable_open_ssh
5000	Windows is Ready to use
6000	PostReadyLocalData
7000	PostReadyUserData
6010/7010	PostReadyLocal/UserData - set_wallpaper
6020/7020	PostReadyLocal/UserData - set_update_schedule
6030/7030	PostReadyLocal/UserData - set_hostname
6040/7040	PostReadyLocal/UserData - execute_program

MessageId/Event Id	Event message
6050/7050	PostReadyLocal/UserData - execute_script
6060/7060	PostReadyLocal/UserData - manage_package
6070/7070	PostReadyLocal/UserData - initialize_volume
6080/7080	PostReadyLocal/UserData - write_file
6090/7090	PostReadyLocal/UserData - start_ssm
7100	PostReadyUserData - enable_open_ssh
6110/7110	PostReadyLocal/UserData - enable_jumbo_frames

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. For more information about instance console output and how to access it, see the section called "Instance console output".

Outputs

- EC2Launch v2 console log output
- EC2Launch v2 console log messages

EC2Launch v2 console log output

The following is sample console log output for EC2Launch v2. Some values in this example are substituted with representative text surrounded by curly braces.

```
2025/07/22 21:26:53Z: Windows sysprep configuration complete.
2025/07/22 21:26:53Z: Message: Waiting for access to metadata...
2025/07/22 21:26:53Z: Message: Meta-data is now available.
2025/07/22 21:26:53Z: AMI Origin Version: 2024.12.13
2025/07/22 21:26:53Z: AMI Origin Name: Windows_Server-2022-English-Full-Base
2025/07/22 21:26:53Z: OS: Microsoft Windows NT 10.0.20348
2025/07/22 21:26:53Z: OsVersion: 10.0
2025/07/22 21:26:53Z: OsProductName: Windows Server 2022 Datacenter
2025/07/22 21:26:53Z: OsBuildLabEx: 20348.1.amd64fre.fe_release.210507-1500
2025/07/22 21:26:53Z: OsCurrentBuild: 20348
2025/07/22 21:26:53Z: OsReleaseId: 2009
2025/07/22 21:26:53Z: Language: en-US
2025/07/22 21:26:53Z: TimeZone: UTC
2025/07/22 21:26:53Z: Offset: UTC +0000
2025/07/22 21:26:53Z: Launch: EC2 Launch v2.2.63
2025/07/22 21:26:53Z: AMI-ID: ami-1234567890abcdef1
2025/07/22 21:26:53Z: Instance-ID: i-1234567890abcdef0
2025/07/22 21:26:54Z: Instance Type: t3.xlarge
2025/07/22 21:26:54Z: Driver: AWS NVMe Driver v1.6.0.35
2025/07/22 21:26:54Z: SubComponent: 1.6.0.35; EnableSCSIPersistentReservations: 0
2025/07/22 21:26:54Z: Driver: AWS PV Driver Package v8.5.0
2025/07/22 21:26:55Z: Driver: Amazon Elastic Network Adapter v2.8.0.0
2025/07/22 21:26:55Z: HOSTNAME: EC2AMAZ-9FJG5CC
2025/07/22 21:26:55Z: RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
2025/07/22 21:26:55Z: RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
2025/07/22 21:26:56Z: SSM: Amazon SSM Agent v3.3.2746.0
2025/07/22 21:26:57Z: User data format: no_user_data
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentCommandErrorCode=1
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AdminPasswordTypeCode=0
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentErrorLocation=execute_windows.go:410
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IpConflictDetectionCode=0
2025/07/22 21:26:57Z: Message: Windows is Ready to use
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2", ...}
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2", ...}
```

EC2Launch v2 console log messages

The following is a list of all of the EC2Launch v2 console log messages.

```
Error EC2Launch service is stopping. {error message}
```

Stopped service error details:

- `Error setting up EC2Launch agent folders`
- `See instance logs for detail`
- `Error stopping service`
- `Error initializing service`

```
Windows sysprep configuration complete
```

```
Invalid administrator username: {invalid username}
```

```
Invalid administrator password
Username: {username}
Password: <Password>{encrypted password}</Password>
```

The following message is an information block that contains AMI details:

```
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}
HOSTNAME: {computer name}
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 installed.
Unable to get service status for vmms
Microsoft-Hyper-V is {status}
SSM: Amazon SSM Agent {version}
AWS VSS Version: {version}
```

```
Windows sysprep configuration complete
Windows is being configured. SysprepState is {state}
Windows is still being configured. SysprepState is {state}
Windows is Ready to use
Waiting for meta-data accessibility...
Meta-data is now available.
Still waiting for meta-data accessibility...
Failed to find primary network interface...retrying...
User data format: {format}
```

EC2Launch v2 telemetry messages include the launch telemetry property values. Starting with version 2.2.63, EC2 agent telemetry data is formatted as a JSON object.

```
EC2LaunchTelemetry: {telemetry property}
```

```
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2" ... }
```

EC2Launch v2 version histories

Version histories

- EC2Launch v2 version history

EC2Launch v2 version history

To ensure that you have the latest launch agent installed, see Install the latest version of EC2Launch v2. You can receive notifications when new versions of the EC2Launch v2 agent are released. For more information, see Subscribe to EC2 Windows launch agent notifications.

The following table describes the released versions of EC2Launch v2.

Version	Details	Release date
2.2.63	• Updated launch agent to improve [Telemetry]().	July 29, 2025
2.1.1	• Added full support for IPv6.	May 14, 2025
2.0.2107	• Enhanced route addition to handle scenarios when IPv4 or IPv6 addresses are not available on the interface.	March 27, 2025
2.0.2081	• Fixed an issue where RDP certificate information was not properly retrieved or validated. Added functionality to automatically start the Remote Desktop Services if needed. • Adjusted EC2Launch v2 service permissions to fix an issue that occurs when querying the service status.	February 4, 2025
2.0.2046	• Updated the wallpaper path in the `agent-config.yml` file to use the default operating system wallpaper path. • Added telemetry to monitor the locations where agent errors occurred. • Updated agent log messaging.	October 3, 2024
2.0.1981	• Updated `EC2Launch.exe` CLI command error messages for non-Administrator users.	August 6, 2024
2.0.1948	• Added telemetry to monitor usage of admin password options. • Modified EC2Launch permissions.	July 1, 2024
2.0.1924	• Updated the EC2Launch Settings UI.	June 10, 2024

Version	Details	Release date
	• Updated the wallpaper CLI command. • Updated the EC2Launch installer.	
2.0.1914	• Add routes with unspecified gateway addresses (`0.0.0.0` for IPv4 or `::` for IPv6). • Always add both IPv4 and IPv6 routes. • Fixed an issue where the `Administrator` username was added to the `agent-config.yml` file when it wasn't specified. • Modified EC2Launch v2 permissions.	June 5, 2024
2.0.1881	• Added an encrypted password option to `setAdminAccount` task. • Added CLI command to encrypt static password in agent-config.yml. • Fixed an issue where XML user data doesn't add PowerShell arguments when it runs with Administrator permissions. For more details, see How Amazon EC2 handles user data for Windows instances. • Adjusted PowerShell arguments for the `executeScript` task and user data scripts when they run with `LocalSystem` permissions. When arguments are empty, the agent uses the following default value: `-ExecutionPolicy Unrestricted`. • Prevented printing duplicate driver versions to the console log.	May 8, 2024

Version	Details	Release date
2.0.1815	• Adjusted error handling to fail on critical setup issues before sysprep. • Fixed an issue where wallpaper and hostname tasks could use an incorrect IP address on instances with multiple IP addresses assigned to the primary network interface. • Wallpaper and hostname tasks changed to get private IP from IMDS first, then fail back to WMI if IMDS is disabled. • Fixed an issue with the `initializeVolume` task where `sc1` volumes failed to initialize due to a transient error.	March 6, 2024
2.0.1739	• Fixed an issue that prevented exit codes from being captured by `executeScript` tasks that were run as the Windows Administrator user.	January 17, 2024
2.0.1702	• Restricted `Telemetry.log` permissions to `read-execute` only for standard users. • Configured the EC2Launch Windows service to restart on start-up failure. • Made `add-routes` failures actionable by logging `route.exe` `stderr` output. • Fixed an issue that occurs when route metrics are outside of the range [1, 9999]. • Added wallpaper support to several new instance types. • Fixed an issue caused by user data scripts that run as the Windows Administrator user and send output to `stderr`.	January 4, 2024

Version	Details	Release date
2.0.1643	• Updated the `ebsnvme-id.exe` tool to version 1.1.0.7. • Fixed an issue with receive side scaling (RSS) and receive queue depth settings on metal instance types that begin with 'metal-*', such as metal-48x1. • Removed telemetry event that reports on XML userdata commands that block the agent. • Updated `setDnsSuffix` task to limit domain name devolution based on registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` . • Added a public task and CLI that adds network routes. • *Note* – This is the last version to officially support Windows Server 2012. • *Note* – This is the last version to officially support 32-bit operating systems.	October 4, 2023
2.0.1580	• Changed the way that the launch agent handles errors when you modify log file permissions. • Added a timeout for connecting to the serial port. The time out allows the launch agent to continue running if the serial port is in use.	September 5, 2023

Version	Details	Release date

2.0.1521	• Deprecated the –`block` flag of the `EC2Launch.exe` **reset** and **sysprep** commands.
	• Updated `EC2Launch.exe` to detect and handle the **reset** and **sysprep** commands that are used in inline `executeScript` tasks. Those commands cause the agent to stop running after the `executeScript` task runs them.
	• Updated XML userdata scripts to run inline by default.
	• Enable XML userdata scripts to run detached with the new `detach` tag. For more details, see User data scripts.
	• Made the following changes to the agent log.
	  ◦ Updated agent log messages.
	  ◦ Removed `executeScript` content and output from the agent log.
	  ◦ Removed `executeProgram` arguments and output from the agent log.
	• Made the following changes to the console log.
	  ◦ Added `EnableSCSIPersistentReservations` value to the console log.
		July 3, 2023

Version	Details	Release date
2.0.1303	• Added additional error handling and log lines when adding network routes. • Allowed `executeScript` and `executeProgram` tasks in the PreReady stage. • Updated `executeProgram` task to generate output files similar to the output from the executeScript task. For more information, see executeProgram. • Added telemetry to monitor usage of blocking agent co mmands in XML user data.	May 3, 2023
2.0.1245	• Improved visibility into crashes by logging crash call stacks in clear text. • Added the EventLog service as a startup dependency to fix a crash when the Amazon EC2Launch service starts up faster t han the EventLog service. • Made XML user data run before PostReady stage from the agent config file (like EC2Launch v1 and EC2Config). • Added YAML user data version 1.1 to make user data run before PostReady stage from the agent config file (YAML user data version 1.0 runs after PostReady stage from the agent config file).	March 8, 2023

Version	Details	Release date
2.0.1173	• Adds an optional feature to display instance tags on wallp aper. For more information, see setWallpaper . • Adds error handling when the security group for Elastic Graphics is not properly set up. • Fixes a timeout when the Instance Metadata Service is not enabled.	February 6, 2023
2.0.1121	• Fixes an issue where a 404 error is printed to the wallpaper when no public IPv4 address is assigned. • Fixes an issue where the volume's file system is formatted as `RAW` instead of `NTFS` when its device's drive letter is set to `D`. • Fixes an issue where NVMe SSD volumes are incorrectly identified as EBS volumes. • Fixes an error when activating Windows when IMDS is disab led.	January 4, 2023

Version	Details	Release date
2.0.1082	• Fixes an issue where the `setWallpaper : privateIpAddress` field is blank when IMDS is disabled. • Fixes an issue with setting the hostname to the private IPv4 address when IMDS is disabled. • Fixes an issue with initializing volumes on Windows Server 2012. • Fixes an issue with setting jumbo frames. • Fixes an error when no SSH key is specified at instance launch. • Fixes an error on Windows Server 2012 when Windows does not have a 'ReleaseId' registry key.	December 7, 2022
2.0.1011	• Fixes logic for finding network adapter when PnPDeviceID is empty.	November 11, 2022
2.0.1009	• Uses PCI segment information to select the console port.	November 8, 2022
2.0.982	• Adds retry logic to get RDP information. • Fixes errors during volume initialization on `d2.8xlarge` instances. • Fixes issue where an incorrect network adapter can be selected after a reboot. • Removes false alarm error message when ACPI SPCR is unavailable.	October 31, 2022

Version	Details	Release date
2.0.863	• Updates IMDS wait logic to make only IMDSv2 requests. • Adds logic to assign drive letter to volumes that are already initialized but not mounted. • Prints a more specific error message when key pair type is not supported. • Fixes 3010 reboot code bug. • Adds check for invalid base64-encoded user data.	July 6, 2022
2.0.698	• Fixes typo in log output when executing scripts.	January 30, 2022
2.0.674	• Telemetry uploads the enabled/disabled privacy control. • Fixes `index out of bounds` bug. • Removes wallpaper shortcuts during `sysprep`.	November 15, 2021
2.0.651	• Adds logic to uninstall legacy agents during EC2Launch v2 installation. • Fixes `list-volume` CLI issue when root volume is not listed as volume 0.	October 7, 2021
2.0.592	• Fixes bug to correctly report stage status. • Removes false alarm error messages when log files are closed. • Adds telemetry.	August 31, 2021

Version	Details	Release date
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 (XML 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

Version	Details	Release date
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. • Fixes issue with changing DNS suffix search order. • Adds check conditions to better report invalid user data.	December 4, 2020
2.0.153	Adds Sysprep functionality in UserData.	November 3, 2020

Version	Details	Release date
2.0.146	• Fixes issue with RootExtend on non-English AMIs. • Grants users group write permission to log files. • Creates MS Reserved partition for GPT volumes. • Adds list-volumes command and volume dropdown in Amazon EC2Launch settings. • Adds get-agent-config command for printing agent-config.yml file in yaml or json format. • Erases static password if no public key detected.	October 6, 2020
2.0.124	• Adds option to display OS version on wallpaper. • Initializes encrypted EBS volumes. • Adds routes for VPCs with no local DNS name.	September 10, 2020
2.0.104	• Creates DNS suffix search list if it does not exist. • Skips Hibernation if not requested.	August 12, 2020
2.0.0	Initial release.	June 30, 2020

Use the EC2Launch v1 agent to perform tasks during EC2 Windows instance launch

Amazon managed AMIs for Windows Server 2016 and 2019 include a set of Windows Powershell scripts called EC2Launch. EC2Launch performs tasks during the initial instance boot. For information about the EC2Launch versions that are included in AWS Windows AMIs, see the AWS Windows AMI Reference.

> **Note**
>
> The latest launch agent for Windows Server 2016 and later operating system versions is EC2Launch v2, which replaces both EC2Config and EC2Launch, and comes pre-installed on AWS Windows Server 2016 and 2019 AMIs with names that begin with `EC2LaunchV2-Windows_Server-*`. You can also manually install and configure the agent on Windows Server 2016 and 2019. For more information, see Install EC2Launch v2.
> To use EC2Launch with IMDSv2, the version must be 1.3.2002730 or later.

You can use the following Windows PowerShell command to verify the installed version of EC2Launch.

```
Test-ModuleManifest -Path "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1" | Select Version
```

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 to the private IPv4 address of the instance.
- 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 Run commands when you launch an EC2 instance with user data input.
- 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.

The following tasks help to maintain backward compatibility with the EC2Config service. You can also configure EC2Launch to perform these tasks during startup:

- Initialize secondary EBS volumes.
- Send Windows Event logs to the EC2 console logs.
- Send the *Windows is ready to use* message to the EC2 console.

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.
- `Library` — Contains shared libraries for EC2 launch agents.
- `Log` — Contains subdirectories for scripts and the log files that are generated by the scripts.

Telemetry

Telemetry is additional information that helps AWS to better understand your requirements, diagnose issues, and deliver features to improve your experience with AWS services.

EC2Launch version `1.3.2003498` and later collect telemetry, such as usage metrics and errors. This data is collected from the Amazon EC2 instance on which EC2Launch runs. This includes all Windows AMIs owned by AWS.

The following types of telemetry are collected by EC2Launch:

- **Usage information** — agent commands, install method, and scheduled run frequency.
- **Errors and diagnostic information** — agent installation and run error codes.

Examples of collected data:

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

Telemetry is enabled by default. You can disable telemetry collection at any time. If telemetry is enabled, EC2Launch sends telemetry data without additional customer notifications.

Your choice to enable or disable telemetry is collected.

You can opt in or out of telemetry collection. Your selection to opt in or out of telemetry is collected to ensure that we adhere to your telemetry option.

Telemetry visibility

When telemetry is enabled, it appears in the Amazon EC2 console output as follows:

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

Disable telemetry on an instance

To disable telemetry by setting a system environment variable, run the following command as an administrator:

```
setx /M EC2LAUNCH_TELEMETRY 0
```

To disable telemetry during installation, run `install.ps1` as follows:

```
. .\install.ps1 -EnableTelemetry:$false
```

More topics for EC2Launch

- Install the latest version of EC2Launch
- Configure the EC2Launch v1 agent on your Windows instance
- EC2Launch version history

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.

To download and install the latest version of EC2Launch using PowerShell

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.

To install the latest version of EC2Launch using PowerShell, run the following commands from a PowerShell window as an administrator:

```
mkdir $env:USERPROFILE\Desktop\EC2Launch
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/EC2-Windows-Launch.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/install.ps1"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
& $env:USERPROFILE\Desktop\EC2Launch\install.ps1
```

Note

If you receive an error when downloading the file, and you are using Windows Server 2016, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

Verify the installation by checking the launch agent. Run the following commands from a PowerShell window as an administrator:

```
Import-Module C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Ec2Launch.psm1
Import-LocalizedData -BaseDirectory C:\ProgramData\Amazon\EC2-Windows\Launch\Module\ -FileName 'Ec2Launch.psd1' -BindingVariable moduleManifest
$moduleManifest.Get_Item('ModuleVersion')
```

Configure the EC2Launch v1 agent on your Windows instance

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

- Schedule EC2Launch to run on every boot
- Initialize drives and map drive letters
- Send Windows event logs to the EC2 console
- Send Windows is ready message after a successful boot

Configure initialization tasks

Specify settings in the `LaunchConfig.json` file to enable or disable the following initialization tasks:

- Set the computer name to the instance private IPv4 address.
- Set the monitor to always stay on.
- Set up new wallpaper.
- Add DNS suffix list.

> **Note**
>
> This adds a DNS suffix lookup for the following domain and configures other standard suffixes. For more information about how launch agents set DNS suffixes, see Configure DNS Suffix for EC2 Windows launch agents.
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- Extend the boot volume size.
- Set the administrator password.

To configure initialization settings

1. On the instance to configure, open the following file in a text editor: `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json`.
2. Update the following settings as needed and save your changes. Provide a password in `adminPassword` only if `adminPasswordtype` is `Specify`.

```
{
 "setComputerName": false,
```

```
 "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.

```
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:

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

2. Or, run the executable with the following command:

```
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, see the section called "User data scripts".

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. For more information about getting volume details in Windows, see Get-Volume in the Microsoft documentation.

To map drive letters to volumes

1. Open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` file in a text editor.
2. Specify the following volume settings and save your changes:

```
{
 "driveLetterMapping": [
  {
   "volumeName": "sample volume",
   "driveLetter": "H"
  }
 ]
}
```

3. Open Windows PowerShell and use the following command to run the EC2Launch script that initializes the disks:

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

To initialize the disks each time the instance boots, add the `-Schedule` flag as follows:

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -Schedule
```

Send Windows event logs to the EC2 console

Specify settings in the `EventLogConfig.json` file to send Windows Event logs to EC2 console logs.

To configure settings to send Windows Event logs

1. On the instance, open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogConfig.json` file in a text editor.
2. Configure the following log settings and save your changes:

```
{
 "events": [
  {
   "logName": "System",
   "source": "An event source (optional)",
   "level": "Error | Warning | Information",
   "numEntries": 3
  }
 ]
}
```

3. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task each time the instance boots.

```
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.

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -Schedule
```

EC2Launch version history

> **Important**
>
> Beginning January 1, 2025, only the last two versions are supported. When a new version is released, the oldest previously supported version will automatically be marked private and will no longer be available for download.

To download and install the latest version of EC2Launch, see Install the latest version of EC2Launch.

You can receive notifications when new versions of the EC2Launch agent are released. For more information, see Subscribe to EC2 Windows launch agent notifications.

The following EC2Launch agent versions are supported and are available for download.

Version	Details	Release date
1.3.2005119	•	11 February 2025

Version	Details	Release date
	Fixed an issue where `Invoke-Userdata` would fail when called without any parameters.	
1.3.2005065	• Fixed an issue where RDP certificate information was not properly retrieved or validated. Added functionality to automatically start the Remote Desktop Services if needed.	22 October 2024

The following prior versions of EC2Launch are no longer available for download.

Version	Details	Release date
1.3.2005008	• Updated `Set-Wallpaper` to fall back to a solid color background if the default wallpaper image is not found.	6 August 2024
1.3.2004959	• Updated installer logic to prevent unsupported installation on Windows Server 2025 or later.	2 July 2024
1.3.2004891	• Fixed an issue where `HandleUserData` was not set to `false` as expected. • Added an `Encrypted` password option to `LaunchConfig.json`. • Changed `Settings UI` behavior to encrypt the user specified password by default. • Added `SetAdminPasswordConfig.ps1` to convert the `Specify` password option to the `Encrypted` password option in the agent configuration file.	31 May 2024
1.3.2004617	• Fixed an error when setting the wallpaper.	15 January 2024

Version	Details	Release date
1.3.2004592	• Updated access permissions set by install.ps1 for `%ProgramData%\Amazon\EC2-Windows\Launch` . • Restricted EC2Launch folder/file access to read-execute only for standard user accounts. • Changed the agent to stop waiting for the Instance Metadata Service (IMDS) to initialize if IMDS is not enabled for the instance. • Added a five minute timeout when waiting for the IMDS to initialize. • Changed the agent to write telemetry to the instance console log before the `Windows is Ready` message instead of after. • Added wallpaper support to several new instance types. For more information about access permissions and user account permissions of EC2Launch directories, see the section called "EC2Launch directory structure".	2 January 2024
1.3.2004491	• Added telemetry to monitor usage of the **Specify admin password** option.	9 November 2023
1.3.2004462	• Added flush after every write to the serial console.	18 October 2023

Version	Details	Release date
1.3.2004438	• Limit domain name devolution based on registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` . • Limited `UserdataExecution.log` permissions to `Administrators` only. • Added error messages in the Windows Event Log when log initialization fails.	4 October 2023
1.3.2004256	• Added `EnableSCSIPersistentReservations` value to console log. • Added retry capability for **Get-ConsolePort**.	7 July 2023
1.3.2004052	• Fixed an error that occurred when no SSH key is specified at instance launch. • Updated to retry starting the AmazonSSMAgent Windows service on failure. • Updated to fail SysprepInstance.ps1 if BeforeSysprep.cmd fails with a non-zero exit code.	8 March 2023
1.3.2003975	• Fixed issue impacting Packer AMI builds where `SysprepInstance.ps1` returns a `$LastErrorCode` of 1.	24 December 2022

Version	Details	Release date
1.3.2003961	• Fixed issue where explicitly specified administrator passwords are overwritten with a random password on fast-launched instances. • Fixed issue where SSM Agent fails to start on smaller instance types. • Fixed an issue where the instance console log contains `RDPCERTIFICATE-THUMBPRINT: 0000000 0000000000000000` instead of a valid RDP certificate thumbprint value.	6 December 2022
1.3.2003923	• Fixes logic for finding network adapter when PnPDeviceID is empty.	9 November 2022
1.3.2003919	• Updated Get-ConsolePort to use PCI segment information. • Fixed issue where an incorrect network adapter can be selected after reboot. • Fixed start-SSM-Agent timeout logic. • Fixed backwards compatibility for Send-AdminCredentials function alias.	8 November 2022
1.3.2003857	• Prioritizes adapters with a default gateway when the primary network adapter is selected. • Extended in-memory password encryption.	3 October 2022

Version	Details	Release date
1.3.2003824	• Fixed error during `setComputerName` . • Added logic to skip Windows activation when a BYOL billing code is detected. • Added in-memory password encryption. • Fixed error during volume initialization on `m6id.4xlarge` .	30 August 2022
1.3.2003691	• Updated IMDS wait logic to make only IMDSv2 requests. • Fixed bug impacting eGPU installation.	21 June 2022
1.3.2003639	• Added network-adapter wait logic to prevent use before initialization. • Fixed minor issues.	10 May 2022
1.3.2003498	• Added telemetry. • Added shortcut to Settings UI. • Formatted PowerShell scripts. • Fixed issue with shutdown occurring before BeforeSysprep.cmd completes.	31 January 2022
1.3.2003411	Changed password generation logic to exclude passwords with low complexity.	04 August 2021
1.3.2003364	Updated Install-EgpuManager with IMDSv2 support.	07 June 2021

Version	Details	Release date
1.3.2003312	• Added log lines before and after `setMonitorAlwaysOn` setting. • Added AWS Nitro Enclaves package version to console log.	04 May 2021
1.3.2003284	Improved permission model by updating location for storing user data to `LocalAppData` .	23 March 2021
1.3.2003236	• Updated method for setting user password in `Set-AdminAccount` and `Randomize-LocalAdminPassword` . • Fixed`InitializeDisks` to check whether disk is set to read only before setting it to writable.	11 February 2021
1.3.2003210	Localization fix for `install.ps1` .	7 January 2021
1.3.2003205	Security fix for `install.ps1` to update permissions on `%ProgramData%AmazonEC2-WindowsLaunchModuleScripts` directory.	28 December 2020
1.3.2003189	Added `w32tm resync` after adding routes.	4 December 2020
1.3.2003155	Updated instance type information.	25 August 2020
1.3.2003150	Added `OsCurrentBuild` and `OsReleaseId` to console output .	22 April 2020
1.3.2003040	Fixed IMDS version 1 fallback logic.	7 April 2020
1.3.2002730	Added support for IMDS V2.	3 March 2020

Version	Details	Release date
1.3.2002240	Fixed minor issues.	31 October 2019
1.3.2001660	Fixed automatic login issue for users without password after first time executing Sysprep.	2 July 2019
1.3.2001360	Fixed minor issues.	27 March 2019
1.3.2001220	All PowerShell scripts signed.	28 February 2019
1.3.2001200	Fixed issue with InitializeDisks.ps1 where running the script on a node in a Windows Server Failover Cluster would format drives on remote nodes whose drive letter matched the local drive letter.	27 February 2019
1.3.2001160	Fixed missing wallpaper in Windows 2019.	22 February 2019
1.3.2001040	• Added plugin for setting the monitor to never turn off to fix ACPI issues. • SQL Server edition and version written to console.	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 se ttings for ENA devices. • Disabled hibernation during Sysprep.	5 December 2018

Version	Details	Release date
1.3.2000630	• 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.	9 November 2018
1.3.2000430.0	• 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.	19 September 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.	

Version	Details	Release date
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.	
1.3.521	Fixed minor issues.	
1.3.0	• Fixed a hexadecimal length issue for computer name change. • Fixed a possible reboot loop for computer name change. • Fixed an issue in wallpaper setup.	
1.2.0	• Update to display information about installed operating system (OS) in EC2 system log. • Update to display EC2Launch and SSM Agent version in EC2 system log. • Fixed minor issues.	

Version	Details	Release date
1.1.2	• Update to display ENA driver information in EC2 system log. • Update to exclude Hyper-V from primary NIC filter logic. • Added AWS KMS server and port into registry key for KMS activation. • Improved wallpaper setup for multiple users. • Update to clear routes from persistent store. • Update to remove the z from availability zone in DNS suffix list. • Update to address an issue with the <runAsLocalSystem> tag in user data.	
1.1.1	Initial release.	

Use the EC2Config service to perform tasks during EC2 legacy Windows operating system instance launch

> **Note**
>
> EC2Config has reached the end of support. The operating system versions that it runs on are no longer supported by Microsoft. We strongly recommend that you upgrade to the latest launch agent.
> The latest launch agent for Windows Server 2022 and later operating system versions is EC2Launch v2, which replaces both EC2Config and EC2Launch, and comes pre-installed on AWS Windows Server 2022 and 2025 AMIs. You can also manually install and configure the agent on Windows Server 2016 and 2019. For more information, see Install EC2Launch v2.

Windows AMIs for Windows Server versions prior to Windows Server 2016 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.

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** system dialog, 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** system dialog, 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 an Amazon EBS-backed AMI.

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 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 using Run Command in the *AWS Systems Manager User Guide*.

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 Run commands when you launch an EC2 instance with user data input.

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.

> **Note**
>
> If the Windows OS is configured to use IPv4, these IPv4 link-local addresses can be used. If the Windows OS has the IPv4 network protocol stack disabled and uses IPv6 instead, add `[fd00:ec2::250]` in place of `169.254.169.250` and `169.254.169.251`. Then add `[fd00:ec2::254]` in place of `169.254.169.254`.

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 an Amazon EC2 AMI using Windows Sysprep.

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 for instances with operating system versions prior to Windows Server 2016, 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].
- `Ec2ConfigureRDP`—Sets up a self-signed certificate on the instance, so users can securely access the instance using Remote Desktop. This setting is not used for instances with operating system versions prior to Windows Server 2016, because they can generate their own certificates.
- `Ec2OutputRDPCert`—Displays the Remote Desktop certificate information to the console so that the user can verify it against the thumbprint.
- `Ec2SetDriveLetter`—Sets the drive letters of the mounted volumes based on user-defined settings. By default, when an Amazon EBS volume is attached to an instance, it can be mounted using the drive letter on the instance. To specify your drive letter mappings, edit the `DriveLetterConfig.xml` file located in the `EC2ConfigService\Settings` directory.
- `Ec2WindowsActivate`—The plug-in handles Windows activation. It checks to see if Windows is activated. If not, it updates the 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 checkbox on the Ec2 Service Properties system dialog).

- `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.

> **Note**
>
> This adds a DNS suffix lookup for the following domain and configures other standard suffixes. For more information about how launch agents set DNS suffixes, see [Configure DNS Suffix for EC2 Windows launch agents](#).
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- `WaitForMetaDataAvailable`—Ensures that the EC2Config service will wait for metadata to be accessible and the network available before continuing with the boot. This check ensures that EC2Config can obtain information from metadata for activation and other plug-ins.
- `ShouldAddRoutes`—Adds a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.
- `RemoveCredentialsfromSyspreponStartup`—Removes the administrator password from `Sysprep.xml` the next time the service starts. To ensure that this password persists, edit this setting.

DriveLetterConfig.xml

This file contains settings that control drive letter mappings. By default, a volume can be mapped to any available drive letter. You can mount a volume to a particular drive letter as follows.

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- `VolumeName`—The volume label. For example, *`My Volume`*. To specify a mapping for an instance storage volume, use the label `Temporary Storage X`, where `X` is a number from 0 to 25.
- `DriveLetter`—The drive letter. For example, *`M:`*. The mapping fails if the drive letter is already in use.

EventLogConfig.xml

This file contains settings that control the event log information that's displayed on the console while the instance is booting. By default, we display the three most recent error entries from the System event log.

- `Category`—The event log key to monitor.
- `ErrorType`—The event type (for example, `Error`, `Warning`, `Information`.)
- `NumEntries`—The number of events stored for this category.
- `LastMessageTime`—To prevent the same message from being pushed repeatedly, the service updates this value every time it pushes a message.
- `AppName`—The event source or application that logged the event.

WallpaperSettings.xml

This file contains settings that control the information that's displayed on the desktop background. The following information is displayed by default.

- `Hostname`—Displays the computer name.

- `Instance ID`—Displays the ID of the instance.
- `Public IP Address`—Displays the public IP address of the instance.
- `Private IP Address`—Displays the private IP address of the instance.
- `Availability Zone`—Displays the Availability Zone in which the instance is running.
- `Instance Size`—Displays the type of instance.
- `Architecture`—Displays the setting of the `PROCESSOR_ARCHITECTURE` environment variable.

You can remove any of the information that's displayed by default by deleting its entry. You can add additional instance metadata to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

You can add additional System environment variables to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

This file contains settings that control how EC2Config initializes drives.

By default, EC2Config initialize drives that were not brought online with the operating system. You can customize the plugin as follows.

```
<InitializeDrivesSettings>
    <SettingsGroup>setting</SettingsGroup>
</InitializeDrivesSettings>
```

Use a settings group to specify how you want to initialize drives:

FormatWithTRIM

Enables the TRIM command when formatting drives. After a drive has been formatted and initialized, the system restores TRIM configuration.

Starting with EC2Config version 3.18, the TRIM command is disabled during the disk format operation by default. This improves formatting times. Use this setting to enable TRIM during the disk format operation for EC2Config version 3.18 and later.

FormatWithoutTRIM

Disables the TRIM command when formatting drives and improves formatting times in Windows. After a drive has been formatted and initialized, the system restores TRIM configuration.

DisableInitializeDrives

Disables formatting for new drives. Use this setting to initialize drives manually.

Install the latest version of EC2Config

> **Note**
>
> The latest launch agent for Windows Server 2022 and later operating system versions is EC2Launch v2, which replaces both EC2Config and EC2Launch. EC2Launch v2 comes pre-installed on AWS Windows Server 2022 and 2025 AMIs. You can also manually install and configure the agent on Windows Server 2016 and 2019. For more information, see Install EC2Launch v2.

For information about how to receive notifications for EC2Config updates, see Subscribe to EC2 Windows launch agent notifications. For information about the changes in each version, see the EC2Config version history.

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.

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.

To download and install the latest version of EC2Config using PowerShell

To download, unzip, and install the latest version of EC2Config using PowerShell, run the following commands from a PowerShell window:

```
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Config/EC2Install.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\" + $(Split-Path -Path $Url -Leaf)
$ExtractPath = "$env:USERPROFILE\Desktop\"
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$ExtractShell = New-Object -ComObject Shell.Application
$ExtractFiles = $ExtractShell.Namespace($DownloadZipFile).Items()
$ExtractShell.NameSpace($ExtractPath).CopyHere($ExtractFiles)
Start-Process $ExtractPath
Start-Process `
    -FilePath $env:USERPROFILE\Desktop\EC2Install.exe `
    -ArgumentList "/S"
```

> **Note**
>
> If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:
>
> ```
> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
> ```

Verify the installation by checking `C:\Program Files\Amazon\` for the `Ec2ConfigService` directory.

Configure .NET 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 sign-in credentials.

Methods

- Configure proxy settings using the AWS SDK for .NET (Preferred)
- Configure proxy settings using the system.net element
- Configure proxy settings using Microsoft Group Policy and Microsoft Internet Explorer

Configure proxy settings using the AWS SDK for .NET (Preferred)

You can configure proxy settings for the EC2Config service by specifying the `proxy` element in the `Ec2Config.exe.config` file. For more information, see Configuration Files Reference for AWS SDK for .NET.

To specify the proxy element in Ec2Config.exe.config

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory: `%ProgramFiles%\Amazon\Ec2ConfigService`.
2. Add the following `aws` element to the `configSections`. Do not add this to any existing `sectionGroups`.

   For EC2Config versions 3.17 or earlier

   ```
   <configSections>
      <section name="aws" type="Amazon.AWSSection, AWSSDK"/>
   </configSections>
   ```

   For EC2Config versions 3.18 or later

   ```
   <configSections>
        <section name="aws" type="Amazon.AWSSection, AWSSDK.Core"/>
   </configSections>
   ```
3. Add the following `aws` element to the `Ec2Config.exe.config` file.

   ```
   <aws>
      <proxy
        host="string value"
   ```

```
        port="string value"
        username="string value"
        password="string value" />
</aws>
```

4. Save your changes.

Configure proxy settings using the system.net element

You can specify proxy settings in a `system.net` element in the `Ec2Config.exe.config` file. For more information, see [defaultProxy element (network settings)](#).

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)](#).

   For example, the following configuration routes all traffic to use the proxy that is currently configured for Internet Explorer, with the exception of the metadata and licensing traffic, which will bypass the proxy.

```
<defaultProxy>
    <proxy usesystemdefault="true" />
    <bypasslist>
        <add address="169.254.169.250" />
        <add address="169.254.169.251" />
        <add address="169.254.169.254" />
        <add address="[fd00:ec2::250]" />
        <add address="[fd00:ec2::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.

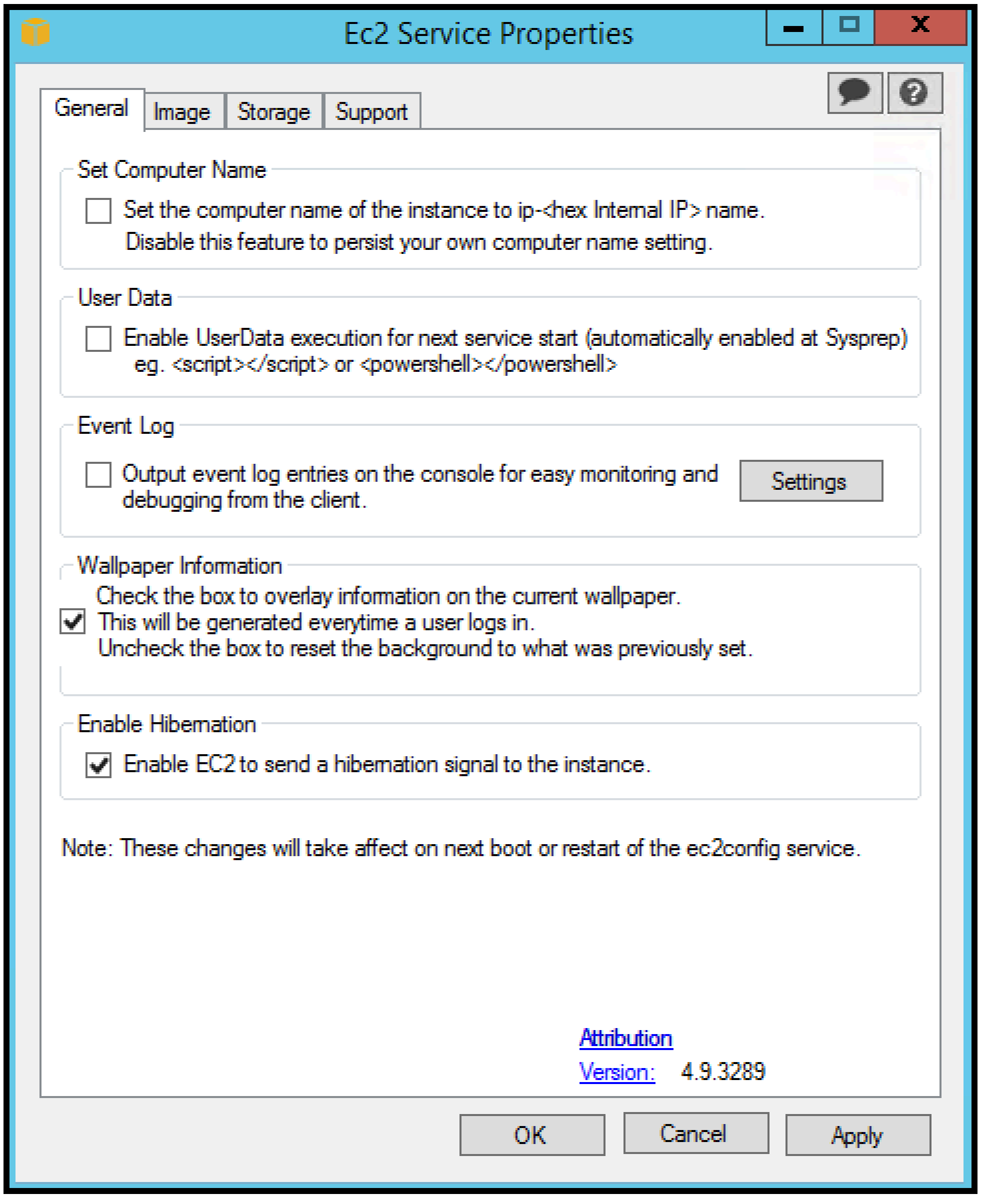
To configure proxy settings using Group Policy and Internet Explorer

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**.

Set EC2Config service properties from the system dialog on your EC2 Windows instance

The following procedure describes how to use the **EC2 Service Properties** system dialog to enable or disable settings.

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** system dialog, 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.

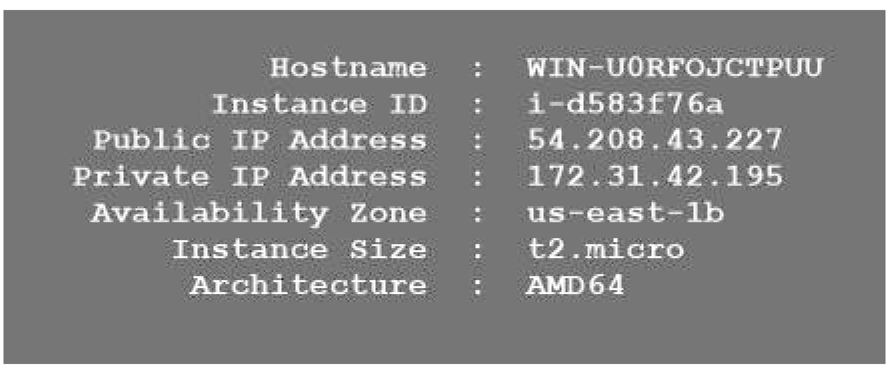
Event Log

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.

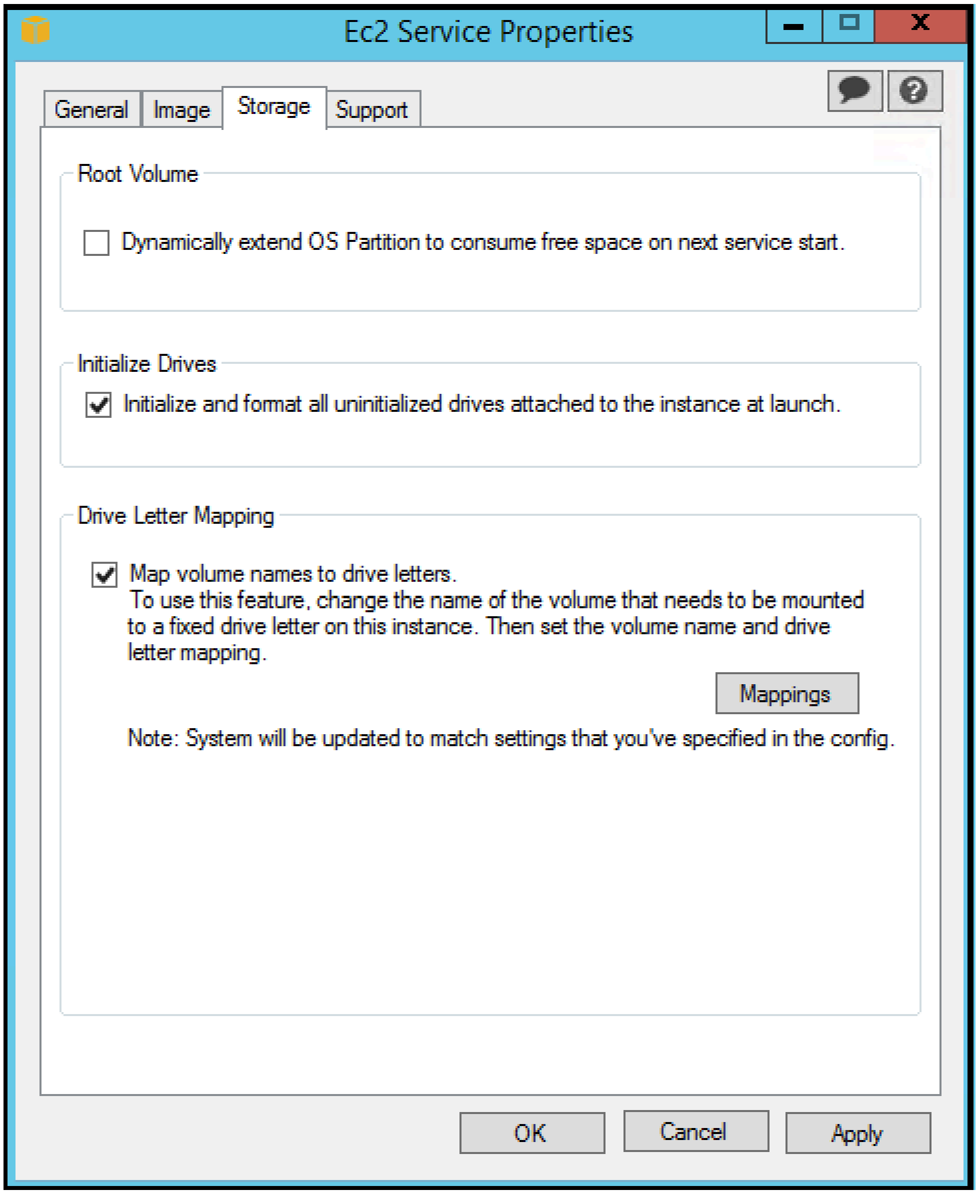


The information displayed on the desktop background is controlled by the settings file `EC2ConfigService\Settings\WallpaperSettings.xml`.

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.

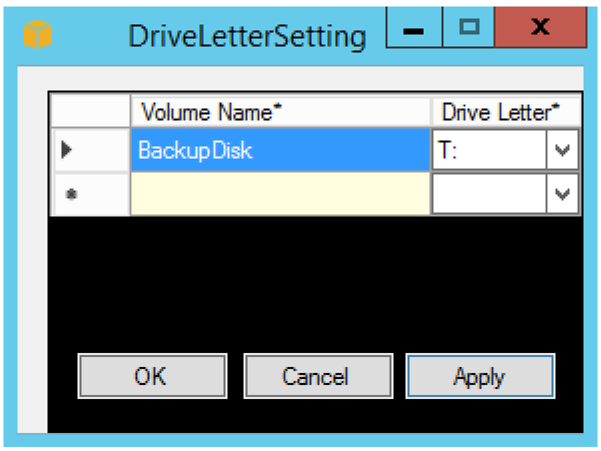
Initialize Drives

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** system dialog. If you have finished customizing your instance and want to create an AMI from that instance, see [Create an Amazon EC2 AMI using Windows Sysprep](#).

Troubleshoot issues with the EC2Config launch agent

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 and then attach the volume 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.
8. Download the latest version of the EC2Config service. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.
9. On the temporary instance, open the Run dialog box, type **`regedit`**, and press Enter.

10. Choose HKEY_LOCAL_MACHINE. From the **File** menu, choose **Load Hive**. Choose the drive and then navigate to and open the following file: Windows\System32\config\SOFTWARE. When prompted, specify a key name.
11. Select the key you just loaded and navigate to Microsoft\Windows\CurrentVersion. Choose the RunOnce key. If this key doesn't exist, choose CurrentVersion from the context (right-click) menu, choose **New** and then choose **Key**. Name the key RunOnce.
12. From the context (right-click) menu choose the RunOnce key, choose **New** and then choose **String Value**. Enter Ec2Install as the name and C:\Temp\Ec2Install.exe /quiet as the data.
13. Choose the HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT \CurrentVersion\Winlogon key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter **AutoAdminLogon** as the name and **1** as the value data.
14. Choose the HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT \CurrentVersion\Winlogon> key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter **DefaultUserName** as the name and **Administrator** as the value data.
15. Choose the HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT \CurrentVersion\Winlogon key. From the context (right-click) menu choose **New**, and then choose **String Value**. Type **DefaultPassword** as the name and enter a password in the value data.
16. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened Registry Editor.
17. From the **File** menu, choose **Unload Hive**.
18. In Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
19. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
20. Stop and start Amazon EC2 instances 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.

EC2Config version history

The following table describes the released versions of EC2Config. For information about the updates for SSM Agent, see Systems Manager SSM Agent Release Notes.

Important

EC2Config has reached the end of support. Only the latest version of the EC2Config agent is available for download. Prior versions are marked as private.

Version	Details	Release date
4.9.5777	• Fixed issue where RSS configuration was set incorrectly for some instance types. • New version of SSM Agent `3.3.484.0` .	17 June 2024
4.9.5554	• Limit domain name devolution based on registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` . • New version of SSM Agent `3.2.1630.0` .	4 October 2023
4.9.5467	• Added retry capability for discovering console port. • New version of SSM Agent `3.1.2282.0` .	1 August 2023
4.9.5288	• Updated AWS Core SDK to version `3.7.103.23` . • Fixed issue where the `AWS-UpdateEC2Config` SSM document fails to update `EC2Config` on instances enabled with only IMDSv2.	8 March 2023

Version	Details	Release date
	• New version of SSM Agent `3.1.2144.0` .	
4.9.5231	• New version of SSM Agent 3.1.1927.0.	14 February 2023
4.9.5103	• Fixed issue where ephemeral volumes are incorrectly identified on r5d and i4i instance families. • New version of SSM Agent 3.1.1856.0.	5 December 2022
4.9.5064	• Updated to use PCI segment information to select the console port. • Signed PowerShell scripts and added copyright headers. • Fixed primary network adapter selection logic. • New version of SSM Agent 3.1.1732.0.	16 November 2022
4.9.4588	• Updated IMDS wait logic to make only IMDSv2 requests. • Added libec2launch.dll launch-agent shared library. • New version of SSM Agent 3.1.1188.0.	31 May 2022
4.9.4556	• Added wait logic to ensure full initialization of NIC before use. • New version of Log4Net 2.0.14.0 picks up security patch. • New version of SSM Agent 3.1.1045.0 picks up security patch.	1 March 2022
4.9.4536	• Fixed issue where userdata crashes when the Temp folder is missing. • New version of SSM Agent 3.1.804.0.	31 January 2022

Version	Details	Release date
4.9.4508	• Fixed issue to correctly compute diskpart script path. • New version of SSM Agent 3.1.338.0.	6 October 2021
4.9.4500	• Updated `Install-EgpuManagerConfig` with IMDS v2 support. • Updated web links to use https. • New version of SSM Agent 3.1.282.0	7 September 2021
4.9.4419	• Fixed IMDS version 1 fallback logic • Updated all usage of Windows temp directory to EC2Config temp directory • New version of SSM Agent 3.0.1124.0	2 June 2021
4.9.4381	• Added support for SSM document schema version 2.2 in EC2ConfigUpdater • Added AWS Nitro Enclaves package version to console log • New version of SSM Agent 3.0.529.0	4 May 2021
4.9.4326	• Removed all links in the settings UI • This is the last EC2Config version that supports Windows Server 2008.	3 March 2021

Version	Details	Release date
4.9.4279	• Fixed security issue related to `Ec2ConfigMonitor` scheduled task • Fixed drive letter mapping issue and incorrect ephemeral disk count • Added `OsCurrentBuild` and `OsReleaseId` to console output • New version of SSM Agent 2.3.871.0	11 December 2020
4.9.4222	• Fixed IMDS version 1 fallback logic • New version of SSM Agent 2.3.842.0	7 April 2020
4.9.4122	• Added support for IMDS v2 • New version of SSM Agent 2.3.814.0	4 March 2020
4.9.3865	• Fixed issue detecting COM port for Windows Server 2008 R2 on metal instances • New version of SSM Agent 2.3.722.0	31 October 2019
4.9.3519	• New version of SSM Agent 2.3.634.0	18 June 2019
4.9.3429	• New version of SSM Agent 2.3.542.0	25 April 2019
4.9.3289	• New version of SSM Agent 2.3.444.0	11 February 2019
4.9.3270	• Added plugin for setting the monitor to never turn off to fix ACPI issues • SQL Server edition and version written to console • New version of SSM Agent 2.3.415.0	22 January 2019
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

Version	Details	Release date
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

Version	Details	Release date
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
4.9.2549	New version of SSM Agent (2.2.325.0)	8 March 2018
4.9.2461	New version of SSM Agent (2.2.257.0)	15 February 2018
4.9.2439	New version of SSM Agent (2.2.191.0)	6 February 2018
4.9.2400	New version of SSM Agent (2.2.160.0)	16 January 2018

Version	Details	Release date
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)	

Version	Details	Release date
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	
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)	

Version	Details	Release date
4.9.1876	• New version of SSM Agent (2.0.796.0) • Fixed an issue with output/error redirection for admin us erdata 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	
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)	

Version	Details	Release date
4.1.1378	New version of SSM Agent (2.0.558.0)	
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. • 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. Skip activation for BYOL users. • 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.	

Version	Details	Release date
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>FormatWithout TRIM</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.	

Version	Details	Release date
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 windo ws 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.	

Version	Details	Release date
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	

Version	Details	Release date
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	
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.	

Version	Details	Release date
3.2.97	• Performance improvements by delay loading Systems Manager SSM assemblies. • Improved exception handling for malformed sysprep 2008.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.	

Version	Details	Release date
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.	
2.2.11	Fixed a issue in CloudWatch plugin for handling empty log lines.	
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.	

Version	Details	Release date
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 d iagnostics	
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+	

Version	Details	Release date
2.2.3	• Added option to install EC2Config without service starting immediately upon install. To use, run 'Ec2Install.exe s tart=false' from the command prompt • Added parameter in wallpaper plugin to control adding/re moving wallpaper. To use, run 'Ec2WallpaperInfo.exe set' or 'Ec2WallpaperInfo.exe revert' from the command prompt • Added checking for RealTimeIsUniversal key, output i ncorrect 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 res ources are being released • Fixed issue with long execution times when joined to domai n	
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	

Version	Details	Release date
2.1.19	• Added ephemeral disk labeling support when using Intel ne twork driver (eg. C3 instance Type). For more information, see Enhanced networking on Amazon EC2 instances. • Added AMI Origin Version and AMI Origin Name support to the console output • Made changes to the Console Output for consistent format ting/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-Su bjectName:' 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	

Version	Details	Release date
2.1.16	• General Tab includes link to EC2Config download page for new Versions • Desktop Wallpaper overlay now stored in Users Local Ap pdata folder instead of My Documents to support MyDoc re direction • MSSQLServer name sync'd with system in Post-Sysprep script (2008+) • Reordered Application Folder (moved files to Plugin d irectory 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 'Temp orary Storage' for non-english Operating Systems • Fixed EC2Config Uninstall failure after running Sysprep	

Version	Details	Release date
2.1.15	• Optimized requests to the Metadata service • Metadata now bypass Proxy Settings • Ephemeral Disks labeled as 'Temporary Storage' and Impo rtant.txt placed on volume when found (Citrix PV drivers only). For more information, see [Upgrade PV drivers on EC2 Windows instances](#). • Ephemeral Disks assigned drive letters from Z to A (Citrix PV drivers only) - assignment can be overwritten using Drive Le tter 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), re attempts activation	

Version	Details	Release date
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	

Version	Details	Release date
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	

Version	Details	Release date
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)	

Version	Details	Release date
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	

Version	Details	Release date
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>	

Use EC2 Fast Launch for your Windows instances

When you configure a Windows Server AMI for EC2 Fast Launch, Amazon EC2 creates a set of pre-provisioned snapshots to use for faster launching, as follows.

1. Amazon EC2 launches a set of temporary t3 instances, based on your settings.
2. As each temporary instance completes the standard launch steps, Amazon EC2 creates a pre-provisioned snapshot of the instance. It stores the snapshot in your Amazon S3 bucket.
3. When the snapshot is ready, Amazon EC2 terminates the associated t3 instance to keep resource costs as low as possible.
4. The next time Amazon EC2 launches an instance from the EC2 Fast Launch enabled AMI, it uses one of the snapshots to significantly reduce the time it takes to launch.

Amazon EC2 automatically replenishes the snapshots you have on hand as it uses them to launch instances from the EC2 Fast Launch enabled AMI.

Any account that has access to an AMI with EC2 Fast Launch enabled can benefit from reduced launch times. When the AMI owner grants access for you to launch instances, the pre-provisioned snapshots come from the AMI owner’s account.

If an AMI that supports EC2 Fast Launch is shared with you, you can enable or disable faster launching on the shared AMI yourself. If you enable a shared AMI for EC2 Fast Launch, Amazon EC2 creates the pre-provisioned snapshots directly in your account. If you deplete the snapshots in your account, you can still use snapshots from the AMI owner's account.

> **Note**
>
> EC2 Fast Launch deletes pre-provisioned snapshots as soon as they're consumed by a launch to minimize storage costs and prevent reuse. However, if the deleted snapshots match a retention rule, Recycle Bin automatically retains them. We recommend that you review the scope of your Recycle Bin retention rules so that this doesn't happen. For more information, see Recycle Bin in the *Amazon EBS User Guide*.
>
> This feature is not the same as EBS fast snapshot restore. You must explicitly enable EBS fast snapshot restore on a per-snapshot basis, and it has its own associated costs.

The following video demonstrates how to configure your Windows AMI for faster launching with a quick overview of the related key terms and their definitions: Launching EC2 Windows instances up to 65% faster on AWS.

Resource costs

There is no service charge to configure Windows AMIs for EC2 Fast Launch. However, standard pricing applies for any underlying AWS resources that Amazon EC2 uses. To learn more about associated resource costs and how to manage them, see Manage costs for EC2 Fast Launch underlying resources.

Contents

Key terms

The EC2 Fast Launch feature uses the following key terms:

Pre-provisioned snapshot

A snapshot of an instance that was launched from a Windows AMI with EC2 Fast Launch enabled, and that has completed the following Windows launch steps, rebooting as required.

- Sysprep specialize
- Windows Out of Box Experience (OOBE)

When these steps are complete, EC2 Fast Launch stops the instance, and creates a snapshot that is later used for faster launching from the AMI, based on your configuration.

Launch frequency

Controls the number of pre-provisioned snapshots that Amazon EC2 can launch within the specified timeframe. When you enable EC2 Fast Launch for your AMI, Amazon EC2 creates the initial set of pre-provisioned snapshots in the background. For example, if the launch frequency is set to five launches per hour, which is the default, then EC2 Fast Launch creates an initial set of five pre-provisioned snapshots.

When Amazon EC2 launches an instance from an AMI with EC2 Fast Launch enabled, it uses one of the pre-provisioned snapshots to reduce the launch time. As snapshots are used, they are automatically replenished, up to the number specified by the launch frequency.

If you expect a spike in the number of instances that are launched from your AMI – during a special event, for example – you can increase the launch frequency in advance to cover the additional instances that you'll need. When your launch rate returns to normal, you can adjust the frequency back down.

When you experience a higher number of launches than anticipated, you might use up all the pre-provisioned snapshots that you have available. This doesn't cause any launches to fail. However, it can result in some instances going through the standard launch process, until snapshots can be replenished.

Target resource count

The number of pre-provisioned snapshots to keep on hand for an Amazon EC2 Windows Server AMI with EC2 Fast Launch enabled.

Max parallel launches

Controls how many instances Amazon EC2 can launch at the same time to create the pre-provisioned snapshots for EC2 Fast Launch. If your target resource count is higher than the maximum parallel launches that you've configured, then Amazon EC2 launches the number of instances specified by **Max parallel launches** to start creating the snapshots. As those instances complete the process, Amazon EC2 takes the snapshot and stops the instance. Then it continues to launch more instances until the total number of snapshots available has reached the target resource count. The value for **Max parallel launches** must be 6 or greater.

EC2 Fast Launch prerequisites for Windows

Before you set up EC2 Fast Launch, verify that you've met the following prerequisites that are required to create snapshots for the AMIs in your AWS account:

- If you provide a custom launch template when you configure EC2 Fast Launch, the service uses the VPC and other configuration settings that you've defined in the launch template. For more information, see Use a launch template when you set up EC2 Fast Launch.
- If you don't use a custom launch template to configure your settings, you must attach the EC2FastLaunchFullAccess policy to your current IAM role before you enable EC2 Fast Launch. Then the service automatically creates an AWS CloudFormation stack with the following resources in your AWS account.
  - A virtual private cloud (VPC)
  - Private subnets across multiple Availability Zones
  - A launch template configured with Instance Metadata Service Version 2 (IMDSv2)
  - A security group with no inbound or outbound rules
- Private EC2 Fast Launch AMIs must support user data script execution.
- To configure EC2 Fast Launch for an AMI, you must create the AMI using **Sysprep** with the shutdown option. The EC2 Fast Launch feature doesn't currently support AMIs that were created from a running instance.

  To create an AMI using **Sysprep**, see Create an Amazon EC2 AMI using Windows Sysprep.
- To enable EC2 Fast Launch for an encrypted AMI that uses a customer managed key for encryption, you must grant the service-linked role for EC2 Fast Launch permission to use the CMK. For more information, see the section called “Access to customer managed keys”.
- The default quota for **Max parallel launches** across all AMIs in an AWS account is 40 per Region. You can request a Service Quotas increase for your account, as follows.

  1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/.
  2. In the navigation pane, choose AWS services.

3. In the search bar, enter **`EC2 Fast Launch`**, and select the result.
4. Select the link for **Parallel instance launches** to open the service quota detail page.
5. Choose **Request increase at account level**.

For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Configure EC2 Fast Launch settings for your Amazon EC2 Windows Server AMI

You can configure EC2 Fast Launch for Windows AMIs that you own, or AMIs that are shared with you from the AWS Management Console, API, SDKs, CloudFormation, or AWS Command Line Interface (AWS CLI). Before you configure EC2 Fast Launch, verify that your AMI meets all of the prerequisites that are required to create the pre-provisioned snapshots. For more information, see EC2 Fast Launch prerequisites for Windows.

When you enable faster launching for Windows instances, Amazon EC2 checks to make sure that you have the required permissions to launch instances from the specified AMI and Launch Template (if provided), including permissions for encrypted AMIs. To prevent errors during the instance launch process, the service validates your permissions before EC2 Fast Launch is enabled. If you don't have the required permissions, the service returns an error, and does not enable EC2 Fast Launch.

EC2 Fast Launch integrates with EC2 Image Builder to help you create custom images with EC2 Fast Launch enabled. For more information, see Create distribution settings for a Windows AMI with EC2 Fast Launch enabled (AWS CLI) in the *EC2 Image Builder User Guide*.

Enable EC2 Fast Launch

Before changing these settings, make sure that your AMI, and the Region that you run in meet all EC2 Fast Launch prerequisites for Windows.

Console

To enable EC2 Fast Launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Choose the AMI to update by selecting the check box next to the **Name**.

4. From the **Actions** menu above the list of AMIs, choose **Configure fast launch**. This opens the **Configure fast launch** page, where you configure the settings for EC2 Fast Launch.
5. To start using pre-provisioned snapshots to launch instances from your Windows AMI faster, select the **Enable fast launch for Windows** checkbox.
6. From the **Set anticipated launch frequency** drop-down list, choose a value to specify the number of snapshots that are created and maintained to cover your expected instance launch volume.
7. When you're done making changes, choose **Save changes**.

> **Note**
>
> If you need to use a launch template to specify a VPC, or to configure metadata settings for IMDSv2, see Use a launch template when you set up EC2 Fast Launch.

AWS CLI

To enable EC2 Fast Launch

Use the following enable-fast-launch command to enable EC2 Fast Launch for the specified AMI, launching six parallel instances for pre-provisioning.

```
aws ec2 enable-fast-launch \
    --image-id ami-0abcdef1234567890  \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

The following is example output.

```
{
 "ImageId": "ami-0abcdef1234567890",
 "ResourceType": "snapshot",
 "SnapshotConfiguration": {
     "TargetResourceCount": 10
 },
 "LaunchTemplate": {},
 "MaxParallelLaunches": 6,
 "OwnerId": "0123456789123",
```

```
  "State": "enabling",
  "StateTransitionReason": "Client.UserInitiated",
  "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

PowerShell

To enable EC2 Fast Launch

Use the Enable-EC2FastLaunch cmdlet to enable EC2 Fast Launch for the specified AMI, launching six parallel instances for pre-provisioning.

```
Enable-EC2FastLaunch `
 -ImageId ami-0abcdef1234567890 `
 -MaxParallelLaunch 6 `
 -Region us-west-2 `
 -ResourceType snapshot
```

The following is example output.

```
ImageId                 : ami-0abcdef1234567890
 LaunchTemplate          :
 MaxParallelLaunches     : 6
 OwnerId                 : 0123456789123
 ResourceType            : snapshot
 SnapshotConfiguration : Amazon.EC2.Model.FastLaunchSnapshotConfigurationResponse
 State                   : enabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime     : 2/25/2022 12:24:11 PM
```

Disable EC2 Fast Launch

Before changing these settings, make sure that your AMI, and the Region that you run in meet all EC2 Fast Launch prerequisites for Windows.

Console

To disable EC2 Fast Launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.

3. Choose the AMI to update by selecting the check box next to the **Name**.
4. From the **Actions** menu above the list of AMIs, choose **Configure fast launch**. This opens the **Configure fast launch** page, where you configure the settings for EC2 Fast Launch.
5. Clear the **Enable fast launch for Windows** checkbox to disable EC2 Fast Launch and to remove pre-provisioned snapshots. This results in the AMI using the standard launch process for each instance, going forward.

   > **Note**
   >
   > When you disable Windows image optimization, any existing pre-provisioned snapshots are automatically deleted. This step must be completed before you can start using the feature again.

6. When you're done making changes, choose **Save changes**.

AWS CLI

To disable EC2 Fast Launch

Use the following disable-fast-launch command to disable EC2 Fast Launch on the specified AMI, and clean up existing pre-provisioned snapshots.

```
aws ec2 disable-fast-launch --image-id ami-01234567890abcedf
```

The following is example output.

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
```

```
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
}
```

PowerShell

To disable EC2 Fast Launch

Use the Disable-EC2FastLaunch cmdlet to disable EC2 Fast Launch on the specified AMI, and clean up existing pre-provisioned snapshots.

```
Disable-EC2FastLaunch -ImageId ami-0abcdef1234567890
```

The following is example output.

```
ImageId                : ami-0abcdef1234567890
 LaunchTemplate         :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
 MaxParallelLaunches    : 6
 OwnerId                : 0123456789123
 ResourceType           : snapshot
 SnapshotConfiguration :
 State                  : disabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime    : 2/25/2022 1:10:08 PM
```

Use a launch template when you set up EC2 Fast Launch

With a launch template, you can configure a set of launch parameters that Amazon EC2 uses each time it launches an instance from that template. You can specify such things as an AMI to use for your base image, instance types, storage, network settings, and more.

Launch templates are optional, except for the following specific cases, where you must use a launch template for your Windows AMI when you configure faster launching:

- You must use a launch template to specify an existing VPC for your Windows AMI. This doesn't apply if you use the default VPC for your AWS account.
- If your account includes a policy that enforces IMDSv2 for Amazon EC2 instances, you must create a launch template that specifies the metadata configuration to enforce IMDSv2.

Use the launch template that includes your metadata configuration from the EC2 console, or when you run the enable-fast-launch command in the AWS CLI, or call the EnableFastLaunch API action.

Amazon EC2 EC2 Fast Launch doesn't support the following configuration when you use a launch template. If you use a launch template for EC2 Fast Launch, you must not specify any of the following:

- User data scripts
- Termination protection
- Disabled metadata
- Spot option
- Shutdown behavior that terminates the instance
- Resource tags for network interface, elastic graphic, or spot instance requests

Specify a VPC

Step 1: Create a launch template

Create a launch template that specifies the following details for your Windows instances:

- The VPC subnet.
- An instance type of `t3.xlarge`.

For more information, see Create an Amazon EC2 launch template.

Step 2: Specify the launch template for your EC2 Fast Launch AMI

Console

To specify the launch template for EC2 Fast Launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Choose the AMI to update by selecting the check box next to the **Name**.

4. From the **Actions** menu above the list of AMIs, choose **Configure fast launch**. This opens the **Configure fast launch** page, where you configure the settings for EC2 Fast Launch.

5. The **Launch template** box performs a filtered search that finds launch templates in your account in the current Region that match the text you've entered. Specify all or part of the launch template name or ID in the box to show a list of matching launch templates. For example, if you enter `fast` in the box, Amazon EC2 finds all of the launch templates in your account in the current Region that have "fast" in the name.

   To create a new launch template, you can choose **Create launch template**.

6. When you select a launch template, Amazon EC2 shows the default version for that template in the **Source template version** box. To specify a different version, highlight the default version to replace it, and enter the version number you want in the box.

7. When you're done making changes, choose **Save changes**.

AWS CLI

To specify the launch template for EC2 Fast Launch

Use the enable-fast-launch command with the `--launch-template` option, specifying either the name or the ID of the launch template.

```
--launch-template LaunchTemplateName=my-launch-template
```

PowerShell

To specify the launch template for EC2 Fast Launch

Use the Enable-EC2FastLaunch cmdlet with the `-LaunchTemplate_LaunchTemplateId` or `-LaunchTemplate_LaunchTemplateName` parameter.

```
-LaunchTemplate_LaunchTemplateName my-launch-template
```

For more information about EC2 launch templates, see Store instance launch parameters in Amazon EC2 launch templates.

View AMIs with EC2 Fast Launch enabled

You can use the describe-fast-launch-images command in the AWS CLI, or the Get-EC2FastLaunchImage Tools for PowerShell Cmdlet to get details for AMIs that have EC2 Fast Launch enabled.

Amazon EC2 provides the following details for each Windows AMI that is returned in the results:

- The image ID for an AMI with EC2 Fast Launch enabled.
- The resource type that is used for pre-provisioning the associated Windows AMI. Supported value: `snapshot`.
- The snapshot configuration, which is a group of parameters that configure pre-provisioning for the associated Windows AMI using snapshots.
- Launch template information, including the ID, name, and version of the launch template that the associated AMI uses when it launches Window instances from pre-provisioned snapshots.
- The maximum number of instances that can be launched at the same time for creating resources.
- The owner ID for the associated AMI. This is not populated for AMIs that are shared with you.
- The current state of EC2 Fast Launch for the associated AMI. Supported values include: `enabling | enabling-failed | enabled | enabled-failed | disabling | disabling-failed`.

> **Note**
>
> You can also see the current state displayed in the **Manage image optimization** page in the EC2 console, as **Image optimization state**.

- The reason that EC2 Fast Launch for the associated AMI changed to the current state.
- The time that EC2 Fast Launch for the associated AMI changed to the current state.

AWS CLI

To find AMIs configured for EC2 Fast Launch

Use the following describe-fast-launch-images command to describe the details for each of the AMIs in the account that are configured for EC2 Fast Launch. In this example, only one AMI in the account is configured for EC2 Fast Launch.

```
aws ec2 describe-fast-launch-images
```

The following is example output.

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
    ]
}
```

PowerShell

To find AMIs configured for EC2 Fast Launch

Use the following Get-EC2FastLaunchImage cmdlet to describe the details for each of the AMIs in the account that are configured for EC2 Fast Launch. In this example, only one AMI in the account is configured for EC2 Fast Launch.

```
Get-EC2FastLaunchImage -ImageId ami-0abcdef1234567890
```

The following is example output.

```
ImageId               : ami-0abcdef1234567890
LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
MaxParallelLaunches   : 6
```

```
OwnerId                 : 012345678912
ResourceType            : snapshot
SnapshotConfiguration   :
State                   : enabled
StateTransitionReason   : Client.UserInitiated
StateTransitionTime     : 2/25/2022 12:54:43 PM
```

Manage costs for EC2 Fast Launch underlying resources

There is no service charge to configure Windows AMIs for EC2 Fast Launch. However, when you enable EC2 Fast Launch for an Amazon EC2 Windows AMI, standard pricing applies for underlying AWS resources that Amazon EC2 uses to prepare and store the pre-provisioned snapshots. You can configure cost allocation tags to help you track and manage the costs that are associated with EC2 Fast Launch resources. For more information about how to configure cost allocation tags, see Track EC2 Fast Launch costs on your bill.

The following example demonstrates how the costs associated with EC2 Fast Launch snapshots costs might be allocated.

Example scenario: The AtoZ Example company has a Windows AMI with a 50 GiB EBS root volume. They enable EC2 Fast Launch for their AMI, and set the target resource count to five. Over the course of a month, using EC2 Fast Launch for their AMI costs them around $5.00, and the cost breakdown is as follows:

1. When AtoZ Example enables EC2 Fast Launch, Amazon EC2 launches five small instances. Each instance runs through the Sysprep and OOBE Windows launch steps, rebooting as required. This takes several minutes for each instance (time can vary, based on how busy that Region or Availability Zone (AZ) is, and on the size of the AMI).

   Costs

   - Instance runtime costs (or minimum runtime, if applicable): five instances
   - Volume costs: five EBS root volumes
2. When the pre-provisioning process completes, Amazon EC2 takes a snapshot of the instance, which it stores in Amazon S3. Snapshots are typically stored for 4–8 hours before they are consumed by a launch. In this case, the cost is roughly $0.02 to $0.05 per snapshot.

   Costs

   - Snapshot storage (Amazon S3): five snapshots

3. After Amazon EC2 takes the snapshot, it stops the instance. At that point, the instance is no longer accruing costs. However EBS volume costs continue to accrue.

   Costs

   - EBS volumes: costs continue for the associated EBS root volumes.

> **Note**
>
> The costs shown here are for demonstration purposes only. Your costs will vary, depending on your AMI configuration and pricing plan.

Track EC2 Fast Launch costs on your bill

Cost allocation tags can help you organize your AWS bill to reflect the costs associated with EC2 Fast Launch. You can use the following tag that Amazon EC2 adds to the resources it creates when it prepares and stores pre-provisioned snapshots for EC2 Fast Launch:

Tag key: `CreatedBy`, **Value:** `EC2 Fast Launch`

After you activate the tag in the Billing and Cost Management console, and set up your detailed billing report, the `user:CreatedBy` column appears on the report. The column includes values from all services. However, if you download the CSV file, you can import the data into a spreadsheet, and filter for `EC2 Fast Launch` in the value. This information also appears in the AWS Cost and Usage Report when the tag is activated.

Step 1: Activate user-defined cost allocation tags

To include resource tags in your cost reports, you must first activate the tag in the Billing and Cost Management console. For more information, see Activating User-Defined Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*.

> **Note**
>
> Activation can take up to 24 hours.

Step 2: Set up a cost report

If you already have a cost report set up, a column for your tag appears the next time the report runs after activation is complete. To set up cost reports for the first time, choose one of the following.

- See [Setting up a monthly cost allocation report] in the *AWS Billing and Cost Management User Guide*.
- See [Creating Cost and Usage Reports] in the *AWS Cost and Usage Report User Guide*.

> **Note**
>
> It can take up to 24 hours for AWS to start delivering reports to your S3 bucket.

You can configure EC2 Fast Launch for Windows AMIs that you own, or AMIs that are shared with you from the Amazon EC2 console, API, SDKs, [CloudFormation], or **ec2** commands in the AWS CLI. The following sections cover configuration steps for the Amazon EC2 console and AWS CLI.

You can also create custom Windows AMIs that are configured for EC2 Fast Launch with EC2 Image Builder. For more information, see [Create distribution settings for a Windows AMI with EC2 Fast Launch enabled (AWS CLI)].

Monitor EC2 Fast Launch

This section covers how to monitor the Amazon EC2 Windows Server AMIs in your account that have EC2 Fast Launch enabled.

Monitor EC2 Fast Launch state changes with EventBridge

When the state changes for a Windows AMI with EC2 Fast Launch enabled, Amazon EC2 generates an `EC2 Fast Launch State-change Notification` event. Then Amazon EC2 sends the state change event to Amazon EventBridge (formerly known as Amazon CloudWatch Events).

You can create EventBridge rules that trigger one or more actions in response to the state change event. For example, you can create an EventBridge rule that detects when EC2 Fast Launch is enabled and performs the following actions:

- Sends a message to an Amazon SNS topic that notifies its subscribers.
- Invokes a Lambda function that performs some action.
- Sends the state change data to Amazon Data Firehose for analytics.

For more information, see [Creating Amazon EventBridge rules that react to events]() in the *Amazon EventBridge User Guide*.

State change events

The EC2 Fast Launch feature emits JSON formatted state change events on a best-effort basis. Amazon EC2 sends the events to EventBridge in near real time. This section describes the event fields and shows an example of the event format.

`EC2 Fast Launch State-change Notification`

imageId

Identifies the AMI with the EC2 Fast Launch state change.

resourceType

The type of resource to use for pre-provisioning. Supported value: `snapshot`. The default value is `snapshot`.

state

The current state of the EC2 Fast Launch feature for the specified AMI. Valid values include the following:

- **enabling** – You've enabled the EC2 Fast Launch feature for the AMI, and Amazon EC2 has started creating snapshots for the pre-provisioning process.
- **enabling-failed** – Something went wrong that caused the pre-provisioning process to fail the first time that you enabled the EC2 Fast Launch for an AMI. This can happen anytime during the pre-provisioning process.
- **enabled** – The EC2 Fast Launch feature is enabled. The state changes to `enabled` as soon as Amazon EC2 creates the first pre-provisioned snapshot for a newly enabled EC2 Fast Launch AMI. If the AMI was already enabled and goes through pre-provisioning again, the state change happens right away.
- **enabled-failed** – This state applies only if this is not the first time your EC2 Fast Launch AMI goes through the pre-provisioning process. This can happen if the EC2 Fast Launch feature is disabled and then later enabled again, or if there is a configuration change or other error after pre-provisioning is completed for the first time.
- **disabling** – The AMI owner has turned off the EC2 Fast Launch feature for the AMI, and Amazon EC2 has started the clean up process.

- **disabled** – The EC2 Fast Launch feature is disabled. The state changes to `disabled` as soon as Amazon EC2 completes the clean up process.
- **disabling-failed** – Something went wrong that caused the clean up process to fail. This means that some pre-provisioned snapshots may still remain in the account.

stateTransitionReason

The reason that the state changed for the EC2 Fast Launch AMI.

> **Note**
>
> All fields in this event message are required.

The following example shows a newly enabled EC2 Fast Launch AMI that has launched the first instance to start the pre-provisioning process. At this point, the state is `enabling`. After Amazon EC2 creates the first pre-provisioned snapshot, the state changes to `enabled`.

```
{
 "version": "0",
 "id": "01234567-0123-0123-0123-012345678901",
 "detail-type": "EC2 Fast Launch State-change Notification",
 "source": "aws.ec2",
 "account": "123456789012",
 "time": "2022-08-31T20:30:12Z",
 "region": "us-east-1",
 "resources": [
  "arn:aws:ec2:us-east-1:123456789012:image/ami-123456789012"
 ],
 "detail": {
  "imageId": "ami-123456789012",
  "resourceType": "snapshot",
  "state": "enabling",
  "stateTransitionReason": "Client.UserInitiated"
 }
}
```

Monitor EC2 Fast Launch metrics with CloudWatch

Amazon EC2 AMIs with EC2 Fast Launch enabled send metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that EC2 Fast Launch sends to CloudWatch. The `AWS/EC2` namespace includes the following EC2 Fast Launch metrics:

Metric	Description
NumberOfAvailableFastLaunchSnapshots	The number of pre-provisioned snapshots available per EC2 Fast Launch enabled AMI.
NumberOfInstancesFastLaunched	The number of instances per EC2 Fast Launch enabled AMI that were launched from pre-provisioned snapshots.
NumberOfInstancesNotFastLaunched	The number of instances per EC2 Fast Launch enabled AMI that resulted in a cold boot due to the lack of available pre-provisioned snapshots at launch time.
FastLaunchSnapshotUsedToRefillStartTime	The timestamp when Amazon EC2 launched a new image from a EC2 Fast Launch enabled AMI to create another snapshot after an existing snapshot was used.
FastLaunchSnapshotCreationTime	Measures the time it took for Amazon EC2 to launch an instance and create a snapshot for a EC2 Fast Launch enabled AMI.

Service-linked role for EC2 Fast Launch

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 about how Amazon EC2 uses IAM roles, including service-linked roles, see IAM roles for Amazon EC2.

Amazon EC2 uses the service-linked role named AWSServiceRoleForEC2FastLaunch to create and manage a set of pre-provisioned snapshots that reduce the time it takes to launch instances from your Windows AMI.

Permissions granted by AWSServiceRoleForEC2FastLaunch

The AWSServiceRoleForEC2FastLaunch service-linked role trusts the following service to assume the role:

- `ec2fastlaunch.amazonaws.com`

Amazon EC2 uses the EC2FastLaunchServiceRolePolicy managed policy to complete the following actions:

- **AWS CloudFormation** – Allow EC2 Fast Launch to get a description of associated CloudFormation stacks.
- **Amazon CloudWatch** – Post metric data associated with EC2 Fast Launch to the Amazon EC2 namespace.
- **Amazon EC2** – Access is granted for EC2 Fast Launch to perform the following actions:
  - Launch instances from an Amazon EC2 Windows Server AMI with EC2 Fast Launch enabled, in order to perform provisioning steps. Additionally specify resource pattern that allows `ec2:RunInstances` for an AMI that's associated with License Manager.
  - Stop and terminate an instance that was launched by EC2 Fast Launch after it creates the pre-provisioned snapshot.
  - Describe image and instance type resources used to launch instances from an Amazon EC2 Windows Server AMI with EC2 Fast Launch enabled and create snapshots from them.
  - Describe launch template resources and launch instances from a launch template.
  - Describe instances, instance attributes and instance status, volumes and volume attributes.
  - Describe network interfaces.
  - Delete resources that EC2 Fast Launch created, including snapshots, launch templates, volumes and network interfaces.
  - Tag resources that EC2 Fast Launch creates to launch and pre-provision, Windows instances, and create snapshots for the final launch process to consume.
- **Amazon EventBridge** – Includes access to create EventBridge event rules and retrieve details about or delete rules that it created. EC2 Fast Launch may also get a list of target services that

receive EC2 Fast Launch events that are forwarded based on event rules, and add target services to or remove them from event rules that it created.

- **IAM –** Allows EC2 Fast Launch to create the `EC2FastLaunchServiceRolePolicy` service-linked role, to get and use instance profiles whose name contains `ec2fastlaunch`, and to launch instances on your behalf using the instance profile from your launch template.
- **AWS KMS –** Includes access to create grants and list grants that were created by EC2 Fast Launch that can be retired. Also to describe or use keys for the purpose of encrypting or decrypting volumes attached to instances that EC2 Fast Launch creates, and to generate data keys that are not plaintext.

To view the permissions for this policy, see EC2FastLaunchServiceRolePolicy in the *AWS Managed Policy Reference*.

For more information about using managed policies for Amazon EC2, see AWS managed policies for Amazon EC2.

Create a service-linked role

You don't need to create this service-linked role manually. When you start using EC2 Fast Launch for your AMI, Amazon EC2 creates the service-linked role for you, if it doesn't already exist.

If the service-linked role is deleted from your account, you can enable EC2 Fast Launch for another Windows AMI to re-create the role in your account. Alternatively, you can disable EC2 Fast Launch for your current AMI, and then enable it again. However, disabling the feature results in your AMI using the standard launch process for all new instances while Amazon EC2 removes all of your pre-provisioned snapshots. After all of the pre-provisioned snapshots are gone, you can enable using EC2 Fast Launch for your AMI again.

Access to customer managed keys

To enable EC2 Fast Launch for an encrypted AMI that uses a customer managed key for encryption, you must grant the AWSServiceRoleForEC2FastLaunch role permission to use the CMK. To do this, call the create-grant command. For `--grantee-principal`, specify the ARN for the AWSServiceRoleForEC2FastLaunch role in your account. For `--operations`, specify `CreateGrant`.

```
aws kms create-grant \
    --key-id arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2FastLaunch \
    --operations CreateGrant
```

Edit a service-linked role

Amazon EC2 does not allow you to edit the AWSServiceRoleForEC2FastLaunch service-linked role. After you create a service-linked role, you can't change the name of the role, because various entities might reference the role. However, you can edit the description of the role by using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

Delete a service-linked role

You can delete a service-linked role only after first deleting all of the related resources. This protects the Amazon EC2 resources that are associated with your Amazon EC2 Windows Server AMI with EC2 Fast Launch enabled, because you can't inadvertently remove permission to access the resources.

Use the IAM console, the AWS CLI, or the AWS API to delete the **AWSServiceRoleForEC2FastLaunch** service-linked role. For more information, see Delete a service-linked role in the *IAM User Guide*.

Supported Regions

Amazon EC2 supports the EC2 Fast Launch service-linked role in all of the Regions where the Amazon EC2 service is available.

Troubleshoot Windows EC2 Fast Launch

Troubleshooting scenarios

The following scenarios can help you diagnose and fix common issues that you might encounter when you try to enable EC2 Fast Launch.

Unable to stop instance for snapshot creation

Description

When you enable EC2 Fast Launch, the service launches a set of instances that are used to create the pre-provisioned snapshots. Each instance is given 30 minutes to complete the process. If any of the instances complete successfully, then the service sets the Fast Launch status for the AMI to `Enabled`. However, if an instance fails to complete the process in the allotted time, and none of the other instances have completed the process, the service terminates all of the instances and sets the Fast Launch status for the AMI to `enabling_failed` and the Fast Launch status reason to the following:

```
Unable to stop instance ID=i-1234567890abcdef0 for snapshot creation.
```

Cause

Most often, this is caused by trying to enable EC2 Fast Launch for a Windows AMI that was created from a running instance, or an AMI that doesn't meet all of the EC2 Fast Launch prerequisites.

Solution

Ensure that the AMI you use meets all EC2 Fast Launch prerequisites.

To configure EC2 Fast Launch for an AMI, you must create the AMI using **Sysprep** with the shutdown option. For more information, see Create an Amazon EC2 AMI using Windows Sysprep.

You've reached your VPC limit

Description

If you don't use a launch template to specify an existing VPC, and don't have a default VPC defined for your account, the service automatically creates an AWS CloudFormation stack that includes a VPC and other resources, as described in EC2 Fast Launch prerequisites.

Cause

You've reached the maximum number of VPCs that are allowed in your AWS account for the Region, and you have not specified an existing VPC for EC2 Fast Launch to use. This causes the process to fail.

Solution

You can address this issue with either of the following options:

- You can request a quota increase
- You can provide a launch template that specifies an existing VPC

To request an increase to the number of VPCs that your account can define per Region, follow these steps:

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/.

2. In the **Service Console Dashboard**, choose **Amazon Virtual Private Cloud (Amazon VPC)**. This opens the VPC service quotas.
3. Filter on `VPCs per Region` to go directly to the quota.
4. Select **VPCs per Region**, and choose **Request increase at account level**.

If you have an urgent quota request, or if your quota increase request is denied, contact Support for assistance. For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Insufficient permissions to enable EC2 Fast Launch

Description

When you enable EC2 Fast Launch for the first time without specifying a Launch Template, EC2 Fast Launch creates a service-owned AWS CloudFormation stack with service default resources. However, the CloudFormation templates will fail to deploy if your IAM principal (role or user) lacks the necessary permissions.

The log message might look something like the following:

```
Can't enable EC2 Fast Launch. The IAM credentials that you are using do not have
 sufficient permissions. Attach EC2FastLaunchFullAccess in the IAM console.
```

Cause

Your IAM user or role lacks the necessary permissions to enable EC2 Fast Launch.

Solution

Verify that your IAM principal (user or role) that enables EC2 Fast Launch has the `EC2FastLaunchFullAccess` policy attached. This AWS managed policy grants full access to all EC2 Fast Launch resources. To view the permissions for this policy, see the EC2FastLaunchFullAccess policy in the *AWS Managed Policy Reference*.

Change the Windows Administrator password for your Amazon EC2 instance

If you launch your instance from an AWS Windows AMI, the launch agents that are pre-installed set the default password as follows:

- For Windows Server 2022 and later, EC2Launch v2 generates the default password.
- For Windows Server 2016 and 2019, the EC2Launch agent generates the default password.
- For Windows Server 2012 R2 and earlier, the EC2Config service generates the default password.

> **Note**
>
> For Windows Server 2016 and later AMIs, `Password never expires` is disabled for the local administrator. For AMI versions prior to Windows Server 2016, `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 the Windows administrator password for an Amazon EC2 Windows instance.

Add optional Windows Server components to Amazon EC2 Windows instances

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.

Use one of the following procedures to add Windows Server components to your instance.

Console

To add Windows components to your instance

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 Alias** filter and choose **amazon**.
5. Add the **Description** filter and enter **`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 from snapshot**.
9. For **Availability Zone**, select the Availability Zone that matches your Windows instance. Choose **Add tag** and enter **Name** for the tag key and a descriptive name for the tag value. Choose **Create volume**.

10. In the **Successfully created volume** message (green banner), choose the volume that you just created.
11. Choose **Actions**, **Attach volume**.
12. From **Instance**, select the instance ID.
13. For **Device name**, enter the name of the device for the attachment. If you need help with the device name, see [Device names for volumes on Amazon EC2 instances](#).
14. Choose **Attach volume**.
15. Connect to your instance and make the volume available. For more information, see [Make an Amazon EBS volume available for use](#) in the *Amazon EBS User Guide*.

> **Important**
>
> Do not initialize the volume.

16. 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.
17. (Optional) When you are finished with the installation media, you can detach the volume. After you detach the volume, you can delete it.

AWS CLI

To add Windows components to your instance

1. Use the [describe-snapshots](#) command with the `owner-ids` parameter and `description` filter to get a list of the available installation media snapshots.

```
aws ec2 describe-snapshots \
    --owner-ids amazon \
    --filters Name=description,Values=Windows*
```

2. In the output, note the ID of the snapshot that matches your system architecture and language preference. For example:

```
{
    "Snapshots": [
    ...
        {
```

```
                "OwnerAlias": "amazon",
                "Description": "Windows 2019 English Installation Media",
                "Encrypted": false,
                "VolumeId": "vol-be5eafcb",
                "State": "completed",
                "VolumeSize": 6,
                "Progress": "100%",
                "StartTime": "2019-10-25T20:00:47.000Z",
                "SnapshotId": "snap-22da283e",
                "OwnerId": "123456789012"
            },
        ...
      ]
}
```

3. Use the [create-volume](#) command to create a volume from the snapshot. Specify the same Availability Zone as your instance.

```
aws ec2 create-volume \
    --snapshot-id snap-0abcdef1234567890 \
    --volume-type gp2 \
    --availability-zone us-east-1a
```

4. In the output, note the volume ID.

```
{
    "AvailabilityZone": "us-east-1a",
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-01234567890abcdef",
    "State": "creating",
    "Iops": 100,
    "SnapshotId": "snap-0abcdef1234567890",
    "CreateTime": "2017-04-18T10:33:10.940Z",
    "Size": 6
}
```

5. Use the [attach-volume](#) command to attach the volume to your instance.

```
aws ec2 attach-volume \
    --volume-id vol-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 in the *Amazon EBS User Guide*.

> **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.

PowerShell

To add Windows components to your instance

1. Use the Get-EC2Snapshot cmdlet with the `Owner` and `description` filters to get a list of the available installation media snapshots.

```
Get-EC2Snapshot `
    -Owner amazon `
    -Filter @{ Name="description"; Values="Windows*" }
```

2. In the output, note the ID of the snapshot that matches your system architecture and language preference. For example:

```
...
DataEncryptionKeyId :
Description         : Windows 2019 English Installation Media
Encrypted           : False
KmsKeyId            :
OwnerAlias          : amazon
OwnerId             : 123456789012
Progress            : 100%
SnapshotId          : snap-0abcdef1234567890
StartTime           : 10/25/2019 8:00:47 PM
```

```
State                : completed
StateMessage         :
Tags                 : {}
VolumeId             : vol-01234567890abcdef
VolumeSize           : 6
...
```

3. Use the [New-EC2Volume](#) cmdlet to create a volume from the snapshot. Specify the same Availability Zone as your instance.

```
New-EC2Volume `
    -AvailabilityZone us-east-1a `
    -VolumeType gp2 `
    -SnapshotId snap-0abcdef1234567890
```

4. In the output, note the volume ID.

```
Attachments      : {}
AvailabilityZone : us-east-1a
CreateTime       : 4/18/2017 10:50:25 AM
Encrypted        : False
Iops             : 100
KmsKeyId         :
Size             : 6
SnapshotId       : snap-0abcdef1234567890
State            : creating
Tags             : {}
VolumeId         : vol-01234567890abcdef
VolumeType       : gp2
```

5. Use the [Add-EC2Volume](#) cmdlet to attach the volume to your instance.

```
Add-EC2Volume `
    -InstanceId i-1234567890abcdef0 `
    -VolumeId vol-01234567890abcdef `
    -Device xvdh
```

6. Connect to your instance and make the volume available. For more information, see [Make an Amazon EBS volume available for use](#) in the *Amazon EBS User Guide*.

> **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.

Install Windows Subsystem for Linux on your EC2 Windows instance

The Windows Subsystem for Linux (WSL) is a feature of Microsoft Windows. By installing WSL on your EC2 Windows instance, you can run native Linux command line tools directly on your Windows instance.

There are two versions of Windows Subsystem for Linux (WSL): WSL 1 and WSL 2. For more information, see Windows Subsystem for Linux Documentation on the Microsoft website.

Requirements

- The operating system must be Windows Server 2019 or later.
- You must install WSL 1 on virtualized Windows instances (the instance size is not `.metal`).
- You can install either WSL 1 or WSL 2 on bare metal instances (the instance size is `.metal`). Bare metal instances provide the required support for nested virtualization.

Install WSL on your Windows instance

To install WSL 1

1. Install WSL. The process that you'll use depends on the version of Windows Server running on the instance.
   - **Windows Server 2022 and later** - Run the following standard installation command on your EC2 instance.

```
wsl --install --enable-wsl1 --no-launch
```

- **Windows Server 2019** - Enable WSL and then install WSL as described in [Install WSL on previous versions of Windows Server] on the Microsoft website.

2. Restart your EC2 instance.

```
shutdown -r -t 20
```

3. To configure WSL to use WSL 1, run the following command on your instance. This step is required for virtualized instances (the instance size is not `.metal`).

```
wsl --set-default-version 1
```

4. Install the default distribution.

```
wsl --install
```

To install WSL 2 (bare metal instances only)

Run the following standard installation command on your EC2 instance. By default, WSL 2 is installed.

```
wsl --install
```

EC2 Windows troubleshooting utilities

The `EC2WinUtil` driver provides the following types of troubleshooting support for your Windows instance.

Crash call stacks

`EC2WinUtil` collects basic crash information from your instance and writes it out to the serial console. The following list includes some of the key details that the utility writes to the console.

- Identification of the module that generated the fault.
- The Windows error code associated with the event.
- A stack trace of the most recent calls.

With these details, you can perform initial root cause analysis and determine if further analysis is needed. Output to the serial console also enables AWS to track crash trends for Amazon EC2 drivers, and diagnose large scale crash events.

> **Note**
>
> `EC2WinUtil` doesn't collect any customer data in its crash call stacks.

Hibernate/resume stability

`EC2WinUtil` tracks the virtualization settings of the instance across hibernate/resume cycles. This helps to improve the long-term stability of instances that have enabled hibernation.

For driver release notes, see EC2 Windows Utility Driver version history

EC2 Windows Utility Driver version history

The following table shows which `EC2WinUtil` drivers run on each version of Windows Server on Amazon EC2. Earlier versions of the operating system use the driver that's preinstalled on AWS Windows Server AMIs that the instance launched from. AMIs that are shared with you or that you subscribe to through AWS Marketplace don't have the driver preinstalled.

Windows Server version	EC2WinUtil driver version
Windows Server 2025	latest version
Windows Server 2022	latest version
Windows Server 2019	latest version
Windows Server 2016	latest version

> **Note**
>
> Prior to driver version 3.0.0, the `EC2WinUtil` driver was not available to download for manual installation. Earlier versions were only available as preinstalled drivers for AWS Windows AMIs.

The following table describes the released versions of the `EC2WinUtil` driver.

Package download link	Driver version	Details	Release date
3.0.0	3.0.0	Modernized the driver for Windows 10 and added support for installation as a primitive driver.	June 13, 2024
Download not available for this version.	2.0.0	Added support for output on MMIO serial ports for metal instance types. Also improved crash parsing and updated the output format.	August 23, 2018
Download not available for this version.	1.0.1	Changed the driver name to `EC2WinUtil` due to a namespace conflict with Amazon Inspector. Several bug fixes are included.	March 1, 2018
Download not available for this version.	1.0.0	Initial release. The driver was initially called `AwsAgent`.	November 28, 2017

Upgrade an EC2 Windows instance to a newer version of Windows Server

If it's time to upgrade the Windows Server operating system on your EC2 Windows instance from an earlier version, you can use one of the following methods.

In-place upgrade

An in-place upgrade operates on an existing instance. Only the operating system files are affected during this process, while your settings, server roles, and data are left intact.

Migration (also known as a side-by-side upgrade)

A migration involves capturing settings, configurations, and data, and porting these to a newer operating system on a fresh EC2 Windows instance. You can launch your instance from a public or private Windows AMI that you subscribe to from the AWS Marketplace, or an AMI that's shared with you. You can also create a custom AMI with EC2 Image Builder. See the Image Builder User Guide for more information.

> **Note**
>
> AWS provides a set of publicly available Amazon Machine Images (AMIs) for Windows Server versions that run on EC2 instances. These AMIs are updated on a monthly basis. For information about the latest Windows AMIs, see the AWS Windows AMI Reference.

Microsoft has traditionally recommended migrating to a newer version of Windows Server instead of upgrading in place. 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 configuration settings on the new instance. An in-place upgrade can be faster, but software incompatibilities can produce errors.

Contents

Perform an in-place upgrade on your EC2 Windows instance

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. Instances use either AWS PV, Intel Network Adapter, or the Enhanced Networking drivers. For more information, see Paravirtual drivers for Windows instances.

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.
  - Upgrade Options for Windows Server 2012
  - Upgrade Options for Windows Server 2012 R2
  - Upgrade and conversion options for Windows Server 2016 and above
  - Windows Server Upgrades
- 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. For more information, see Amazon EC2 instance type changes.
- 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 in the *Amazon EBS User Guide*.
- 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 an EC2 Windows instance to a Nitro-based instance type.
- 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 first.

Upgrade an instance in-place with AWS PV, Intel Network Adapter, or the Enhanced Networking drivers

Use the following procedure to upgrade a Windows Server instance using the AWS PV, Intel Network Adapter, or the Enhanced Networking network drivers.

To perform the in-place upgrade

1. Create an AMI of the system you plan to upgrade for either backup or testing purposes. You can then perform the upgrade on the copy to simulate a test environment. If the upgrade completes, you can switch traffic to this instance with little downtime. If the upgrade fails, you can revert to the backup. For more information, see Create an Amazon EBS-backed AMI.
2. Ensure that your Windows Server instance is using the latest network drivers.
   a. To update your AWS PV driver, see Upgrade PV drivers on EC2 Windows instances.
   b. To update your ENA driver, see Install the ENA driver on EC2 Windows instances.
   c. To update your Intel drivers, see Enhanced networking with the Intel 82599 VF interface
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 later, perform the following on your instance before you proceed.
   a. Uninstall the EC2Config service. For more information, see Windows Service administration for EC2Launch v2 and EC2Config agents.
   b. Install EC2Launch v1 or the EC2Launch v2 agent. For more information, see Use the EC2Launch v1 agent to perform tasks during EC2 Windows instance launch and Use the EC2Launch v2 agent to perform tasks during EC2 Windows instance launch.
   c. Install the AWS Systems Manager SSM Agent. For more information, see Manually install SSM Agent on Amazon EC2 for Windows Server 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**.
   b. From the filter bar, choose **Public snapshots**.
   c. In the search bar, specify the following filters:
      - Choose **Owner Alias**, then **=**, then **amazon**.
      - Choose **Description**, and then start typing **`Windows`**. Select the Windows filter that matches the system architecture and language preference you're upgrading to. For example, choose **Windows 2019 English Installation Media** to upgrade to Windows Server 2019.

   d. Select the checkbox next to the snapshot that matches the system architecture and language preference you're upgrading to, and then choose **Actions**, **Create volume from snapshot**.

   e. On the **Create volume** page, choose the Availability Zone that matches your Windows instance, and then choose **Create volume**.

7. In the **Successfully created volume vol-*1234567890example*** banner at the top of the page, choose the ID of the volume that you just created.

8. Choose **Actions**, **Attach volume**.

9. On the **Attach volume** page, for **Instance**, select the instance ID of your Windows instance, and then choose **Attach volume**.

10. Make the new volume available for use by following the steps at [Make an Amazon EBS volume available for use](#).

> **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 /dynamicupdate disable
   ```

   > **Note**
   >
   > Running the setup.exe with the `/dynamicupdate` option set to disabled prevents Windows from installing updates during the Windows Server upgrade process, as installing updates during the upgrade can cause failures. You can install updates with Windows Update after the upgrade completes.

   If you are upgrading to an earlier version of Windows Server, run the following:

   ```
   Sources\setup.exe
   ```

b. For **Select the operating system you want to install**, select the full installation option 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 may fail one or more status checks during the upgrade process. When the upgrade completes, all 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 all status checks after several hours, see Troubleshoot an operating system upgrade on an EC2 Windows instance.

Post upgrade tasks

1. Log in to the instance to initiate an upgrade for the .NET Framework and reboot the system when prompted.
2. If you haven't already done so in a prior step, install the EC2Launch v1 or EC2Launch v2 agent. For more information, see Use the EC2Launch v1 agent to perform tasks during EC2 Windows instance launch and Use the EC2Launch v2 agent to perform tasks during EC2 Windows instance launch.
3. 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 AWS NVMe drivers or Enable enhanced networking on Windows.
4. Re-enable antivirus and anti-spyware software and firewalls.

Use Automation runbooks to upgrade an EC2 Windows instance

You can perform an automated upgrade of your Windows and SQL Server instances on AWS with AWS Systems Manager Automation runbooks.

Contents

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*.
- 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 in the *AWS Systems Manager User Guide*.
- **AWS Systems Manager SSM runbooks**. An SSM runbook defines the actions that Systems Manager performs on your managed instances. SSM runbooks use JavaScript Object Notation (JSON) or YAML, and they include steps and parameters that you specify. This topic uses two Systems Manager SSM runbooks for automation. For more information, see AWS Systems Manager Automation runbook reference in the *AWS Systems Manager User Guide*.

Execution options

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 Windows Server

The `AWSEC2-CloneInstanceAndUpgradeWindows` runbook creates an Amazon Machine Image (AMI) from a Windows Server instance in your account and upgrades this AMI to a supported version of your choice. This multi-step process can take up to two hours to complete.

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 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 Amazon VPC. After testing, and before you perform another upgrade, schedule application downtime before completely switching to the upgraded instance.

Prerequisites

In order to automate your Windows Server upgrade with the AWS Systems Manager Automation document, you must perform the following tasks:

- Create an IAM role with the specified IAM policies to allow Systems Manager to perform automation tasks on your Amazon EC2 instances and verify that you meet the prerequisites to use Systems Manager. For more information, see Creating a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.
- Select the option for how you want the automation to be run. The options for execution are **Simple execution**, **Rate control**, **Multi-account and Region**, and **Manual execution**. For more information about these options, see Execution options.
- Verify that SSM Agent is installed on your instance. For more information see Installing and configuring SSM Agent on Amazon EC2 instances for Windows Server.
- Windows PowerShell 3.0 or later must be installed on your instance.
- For instances that are joined to a Microsoft Active Directory domain, we recommend specifying a `SubnetId` that does not have connectivity to your domain controllers to help avoid hostname conflicts.
- The instance subnet must have outbound connectivity to the internet, which provides access to AWS services such as Amazon S3 and access to download patches from Microsoft. This requirement is met if either the subnet is a public subnet and the instance has a public IP address, or if the subnet is a private subnet with a route that sends internet traffic to a public NAT device.
- This Automation works with instances running Windows Server 2008 R2, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019.
- Verify that the instance has 20 GB of free disk space in the boot disk.
- If the instance does not use a Windows license provided by AWS, then specify an Amazon EBS snapshot ID that includes Windows Server 2012 R2 installation media. To do this:
  1. Verify that the Amazon EC2 instance is running Windows Server 2012 or later.
  2. Create a 6 GB Amazon EBS volume in the same Availability Zone where the instance is running. Attach the volume to the instance. Mount it, for example, as drive D.

3. Right-click the ISO and mount it to an instance as, for example, drive E.
4. Copy the content of the ISO from drive E:\ to drive D:\
5. Create an Amazon EBS snapshot of the 6 GB volume created in step 2 above.

Windows Server upgrade limitations

This automation doesn't support upgrading Windows domain controllers, clusters, or Windows desktop operating systems. In addition, this automation doesn't support Amazon EC2 instances for Windows Server with the following roles installed:

- Remote Desktop Session Host (RDSH)
- Remote Desktop Connection Broker (RDCB)
- Remote Desktop Virtualization Host (RDVH)
- Remote Desktop Web Access (RDWA)

Steps to perform an automated upgrade of Windows Server

Follow these steps to upgrade your Windows Server instance using the AWSEC2-CloneInstanceAndUpgradeWindows automation runbook.

1. Open Systems Manager from the **AWS Management Console**.
2. From the left navigation pane, under **Change Management**, 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. Choose **Execute automation** 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, 2016, or 2019 with the SSM agent installed.

- `InstanceProfile`.

  Type: String

  (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 Configure EC2 instance permissions 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, choose **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 AMI?

Upgrade SQL Server

The AWSEC2-CloneInstanceAndUpgradeSQLServer script creates an AMI from an Amazon EC2 instance running SQL Server in your account, and then upgrades the AMI to a later version of SQL Server. 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. After the upgrade is complete, the automation creates a new AMI before terminating the upgraded instance.

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 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 Amazon VPC. After testing, and before you perform another upgrade, schedule application downtime before completely switching to the upgraded instance.

Prerequisites

In order to automate your SQL Server upgrade with the AWS Systems Manager Automation document, you must perform the following tasks:

- Create an IAM role with the specified IAM policies to allow Systems Manager to perform automation tasks on your Amazon EC2 instances and verify that you meet the prerequisites to use Systems Manager. For more information, see Creating a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.
- Select the option for how you want the automation to be run. The options for execution are **Simple execution**, **Rate control**, **Multi-account and Region**, and **Manual execution**. For more information about these options, see Execution options.
- The Amazon EC2 instance must use Windows Server 2008 R2 or later and SQL Server 2008 or later.
- Verify that SSM Agent is installed on your instance. For more information, see Working with SSM Agent on Amazon EC2 instances for Windows Server.
- Verify that the instance has enough free disk space:
  - If you are upgrading from Windows Server 2008 R2 to 2012 R2, or from Windows Server 2012 R2 to a later operating system, verify that you have 20 GB of free disk space in the instance boot disk.
  - If you are upgrading from Windows Server 2008 R2 to 2016 or later, verify that the instance has 40 GB of free disk space in the instance boot disk.
- For instances that use a Bring Your Own License (BYOL) SQL Server version, the following additional prerequisites apply:
  - Provide an Amazon EBS snapshot ID that includes the target SQL Server installation media. To do this:
    1. Verify that the Amazon EC2 instance is running Windows Server 2008 R2 or later.
    2. Create a 6 GB Amazon EBS volume in the same Availability Zone where the instance is running. Attach the volume to the instance. Mount it, for example, as drive D.
    3. Right-click the ISO and mount it to an instance as, for example, drive E.
    4. Copy the content of the ISO from drive E:\ to drive D:\

5. Create an Amazon EBS snapshot of the 6 GB volume created in step 2.

SQL Server automated upgrade limitations

The following limitations apply when using the AWSEC2-CloneInstanceAndUpgradeSQLServer runbook to perform an automated upgrade:

- The upgrade can be performed on only a SQL Server using Windows authentication.
- Verify that no security patch updates are pending on the instances. Open **Control Panel**, then choose **Check for updates**.
- SQL Server deployments in HA and mirroring mode are not supported.

Steps to perform an automated upgrade of SQL Server

Follow these steps to upgrade your SQL Server using the AWSEC2-CloneInstanceAndUpgradeSQLServer automation runbook.

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 Amazon EBS snapshot of the volume and copy the snapshot ID onto a clipboard for later use. For more information, see Create Amazon EBS snapshots in the **Amazon EBS User Guide**.
4. Attach the instance profile to the Amazon 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.
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, under **Change Management** choose **Automation**.
7. Choose **Execute automation**.
8. Search for the automation document called `AWSEC2-CloneInstanceAndUpgradeSQLServer`.

9. Choose the `AWSEC2-CloneInstanceAndUpgradeSQLServer` SSM document, and then choose **Next**.
10. Ensure that the **Simple execution** option is selected.
11. 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.
    - `SQLServerSnapshotId`

      Type: String

      (Required) The Snapshot ID for the target SQL Server installation media. This parameter is not required for SQL Server license-included instances.
    - `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.

- `TargetSQLVersion`

  Type: String

  (Optional) The target SQL Server version. The default is `2016`.

12. After you have entered the parameters, choose **Execute**. When the automation begins, you can monitor the execution progress.
13. When **Execution status** shows **Success**, expand **Outputs** to view the AMI information. You can use the AMI ID to launch your SQL Server instance for the VPC of your choice.
14. Open the Amazon EC2 console. In the left navigation pane, choose **AMIs**. You should see the new AMI.
15. To verify that the new SQL Server version has been successfully installed, choose the new AMI and choose **Launch**.
16. 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.
17. Add a tag to help you identify your instance.
18. Add the security group or groups to the instance.
19. Choose **Launch Instance**.
20. Choose the tag name for the instance and select **Connect** under the **Actions** dropdown.
21. Verify that the new SQL Server version is the database engine on the new instance.

Migrate an EC2 Windows instance to a Nitro-based instance type

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 Nitro-based instances, such as M5 and C5.

When migrating from Xen-based instances to Nitro-based 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

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, and then continue with Part 4 and Part 5.

For more information, see Amazon EC2 Update — Additional Instance Types, Nitro System, and CPU Options.

Note

The following migration procedures can be performed on Windows Server version 2016 and later. Earlier operating system versions that have reached end of life are not tested, and might not be compatible with the latest instance types.
To migrate Linux instances, see the section called "Instance type changes".

Contents

Before you begin

This procedure assumes that you have a Xen-based instance, such as an M4 or C4, and you are migrating to a Nitro-based instance.

You must use PowerShell version 3.0 or later to successfully perform the upgrade.

> **Note**
>
> When migrating, the static IP or custom DNS network settings on the existing network interface card 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. However, we recommend that you complete them before you migrate, especially if you are migrating to a 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 EC2 Windows instances.

To upgrade a Domain Controller, see Upgrade a domain controller (AWS PV upgrade).

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.

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 ENA on your EC2 instances.

1. Download the latest driver to the instance. If you need a previous version of the driver, see ENA Windows driver version history.
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 ENA on your EC2 instances.

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 Xen-based instance with the intention to migrate the instance to a Nitro-based instance.

1. Download the latest driver package to the instance.

   If you need a previous version of the driver, see NVMe Windows driver releases for supported versions.
2. Extract the zip archive.
3. Install the driver as described in `Readme.txt`.
4. Open a **PowerShell** session and run the following command:

   ```
   PS C:\> 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 and start the instance, then proceed to Part 4. If you start the instance again on a Xen-based instance type before migrating to a Nitro-based 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.

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 Use the EC2Launch v1 agent to perform tasks during EC2 Windows instance launch.

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:

```
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f71556f20b4a 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 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. Chipset INF 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).

To automatically upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager

1. Open the Systems Manager console at https://console.aws.amazon.com/systems-manager.
2. Choose **Automation**, **Execute Automation**.

3. Search for and then select the **AWSSupport-UpgradeWindowsAWSDrivers** automation document, and then choose **Execute automation**.
4. In the **Input Parameters** section, configure the following options:

    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 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.

5. Choose **Execute**.
6. 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.

Troubleshoot an operating system upgrade on an EC2 Windows instance

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 all status checks after several hours, check the following.

- If you upgraded to Windows Server 2008 and all 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 all 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".
- 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.

Tutorial: Connect an Amazon EC2 instance to an Amazon RDS database

Tutorial objective

The objective of this tutorial is to learn how to configure a secure connection between an Amazon EC2 instance and an Amazon RDS database by using the AWS Management Console.

There are different options for configuring the connection. In this tutorial, we explore the following three options:

- Option 1: Automatically connect an instance to an RDS database using the EC2 console

  Use the automatic connection feature in the EC2 console to automatically configure the connection between your EC2 instance and your RDS database to allow traffic between the EC2 instance and the RDS database.
- Option 2: Automatically connect an instance to an RDS database using the RDS console

  Use the automatic connection feature in the RDS console to automatically configure the connection between your EC2 instance and your RDS database to allow traffic between the EC2 instance and the RDS database.
- Option 3: Manually connect an instance to an RDS database by creating security groups

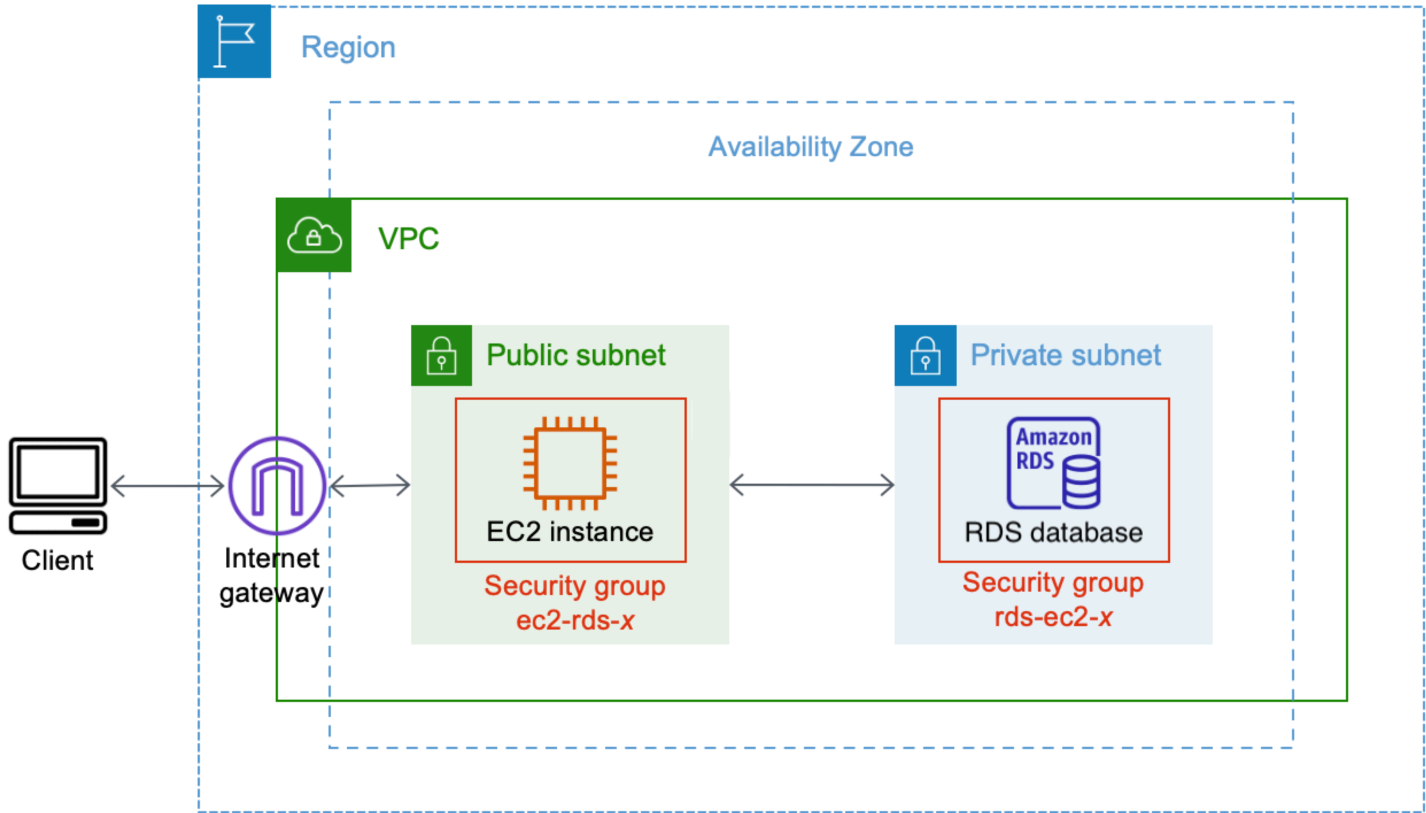
  Configure the connection between your EC2 instance to your RDS database by manually configuring and assigning the security groups to reproduce the configuration that is automatically created by the automatic connection feature in Option 1 and Option 2.

Context

As context for why you'd want to configure a connection between your EC2 instance and an RDS database, let's consider the following scenario: Your website presents a form to your users to fill in. You need to capture the form data in a database. You can host your website on an EC2 instance that's been configured as a web server, and you can capture the form data in an RDS database. The EC2 instance and the RDS database need to be connected to each other so that the form data can go from the EC2 instance to the RDS database. This tutorial explains how to configure that connection. Note that this is just one example of a use case for connecting an EC2 instance and an RDS database.

Architecture

The following diagram shows the resources that are created and the architectural configuration that results from completing all the steps in this tutorial.



The diagram illustrates the following resources that you'll create:

- You'll create an EC2 instance and an RDS database in the same AWS Region, VPC, and Availability Zone.
- You'll create the EC2 instance in a public subnet.
- You'll create the RDS database in a private subnet.

  When you use the RDS console to create the RDS database and automatically connect the EC2 instance, the VPC, DB subnet group, and public access settings for the database are automatically selected. The RDS database is automatically created in a private subnet within the same VPC as the EC2 instance.
- Internet users can connect to the EC2 instance by using SSH or HTTP/HTTPS via an Internet gateway.

- Internet users cannot connect directly to the RDS database; only the EC2 instance is connected to the RDS database.
- When you use the automatic connection feature to allow traffic between the EC2 instance and the RDS database, the following security groups are automatically created and added:
  - Security group **ec2-rds-*x*** is created and added to the EC2 instance. It has one outbound rule that references the **rds-ec2-*x*** security group as its destination. This allows traffic from the EC2 instance to reach the RDS database with the **rds-ec2-*x*** security group.
  - Security group **rds-ec2-*x*** is created and added to the RDS database. It has one inbound rule that references the **ec2-rds-*x*** security group as its source. This allows traffic from the EC2 instance with the **ec2-rds-*x*** security group to reach the RDS database.

  By using separate security groups (one for the EC2 instance, and one for the RDS database), you have better control over the security of the instance and the database. If you were to use the same security group on both the instance and the database, and then modified the security group to suit, say, only the database, the modification would affect both the instance and the database. In other words, if you were to use one security group, you could unintentionally modify the security of a resource (either the instance or the database) because you'd forgotten that the security group was attached to it.

  The security groups that are automatically created also respect least privilege as they only allow the mutual connection for this workload on the database port by creating a workload-specific security group pair.

Considerations

Consider the following when you complete the tasks in this tutorial:

- **Two consoles** – You will use the following two consoles for this tutorial:
  - Amazon EC2 console – You will use the EC2 console to launch instances, to automatically connect an EC2 instance to an RDS database, and for the manual option to configure the connection by creating the security groups.
  - Amazon RDS console – You will use the RDS console to create an RDS database and to automatically connect an EC2 instance to an RDS database.
- **One VPC –** To use the automatic connection feature, your EC2 instance and your RDS database must be in the same VPC.

If you were to manually configure the connection between your EC2 instance and your RDS database, you could launch your EC2 instance in one VPC and your RDS database in another VPC; however, you'd need to set up additional routing and VPC configuration. This scenario is not covered in this tutorial.

- **One AWS Region –** The EC2 instance and RDS database must be located in the same Region.
- **Two security groups –** The connectivity between the EC2 instance and the RDS database is configured by two security groups—a security group for your EC2 instance, and a security group for your RDS database.

  When you use the automatic connection feature in the EC2 console or RDS console to configure the connectivity (Option 1 and Option 2 of this tutorial), the security groups are automatically created and assigned to the EC2 instance and RDS database.

  If you do not use the automatic connection feature, you'll need to manually create and assign the security groups. You do this in Option 3 of this tutorial.

Time to complete the tutorial

30 minutes

You can complete the entire tutorial in one sitting, or you can complete it one task at a time.

Costs

By completing this tutorial, you might incur costs for the AWS resources that you create.

You can use Amazon EC2 under the Free Tier provided your AWS account qualifies for the Free Tier and you configure your resources according to the Free Tier requirements. For more information, Free Tier benefits before and after July 15, 2025.

If your EC2 instance and your RDS database are in different Availability Zones, you will incur data transfer fees. To avoid incurring these fees, the EC2 instance and the RDS database must be in the same Availability Zone. For information about data transfer fees, see Data Transfer on the Amazon EC2 On-Demand Pricing page.

To prevent incurring costs after you've completed the tutorial, make sure to delete the resources if they are no longer needed. For the steps to delete the resources, see Task 4 (*Optional*): Clean up.

Option 1: Automatically connect an instance to an RDS database using the EC2 console

The objective of Option 1 is to explore the automatic connection feature in the EC2 console that automatically configures the connection between your EC2 instance and RDS database to allow traffic from the EC2 instance to the RDS database. In Option 3, you'll learn how to manually configure the connection.

Tasks

Before you begin

You'll need the following to complete this tutorial:

- An RDS database that is in the same VPC as the EC2 instance. You can either use an existing RDS database or follow the steps in Task 1 to create a new RDS database.
- An EC2 instance that is in the same VPC as the RDS database. You can either use an existing EC2 instance or follow the steps in Task 2 to create a new EC2 instance.
- Permissions to call the following operations:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`

- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Task 1 (*Optional*): Create an RDS database

> **Note**
>
> Creating a Amazon RDS database is not the focus of this tutorial. If you already have an RDS database and would like to use it in this tutorial, you can skip this task.
> If you use an existing RDS database, make sure that it is in the same VPC as your EC2 instance so that you can use the automatic connection feature.

The objective of this task is to create an RDS database so that you can complete Task 3 where you configure the connection between your EC2 instance and your RDS database. The steps in this task configure the RDS database as follows:

- Engine type: MySQL
- Template: Free tier
- DB instance identifier: **`tutorial-database-1`**
- DB instance class: `db.t3.micro`

> **Important**
>
> In a production environment, you should configure your database to meet your specific needs.

To create a MySQL RDS database

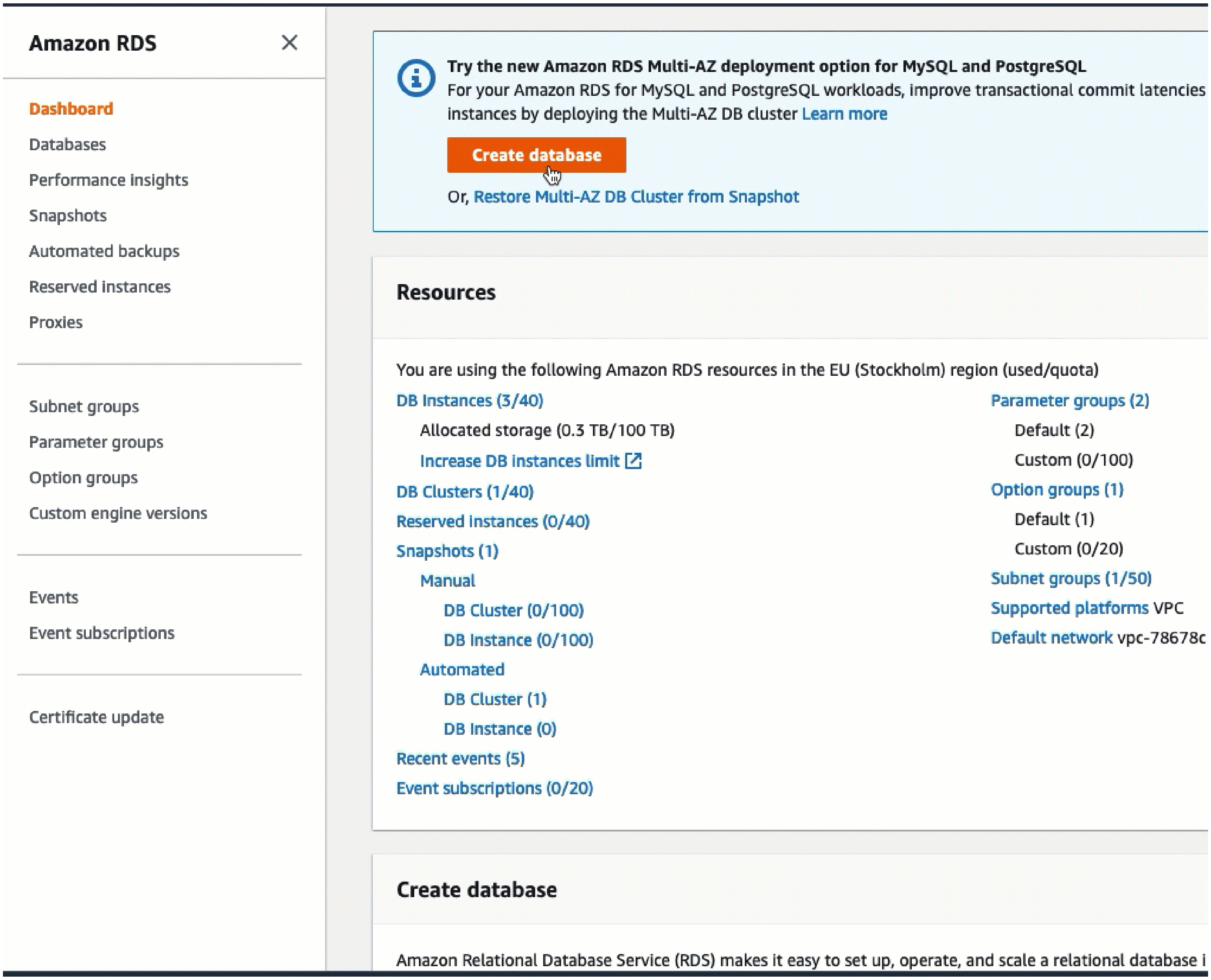
1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.

2. From the Region selector (at top right), choose an AWS Region. The database and the EC2 instance must be in the same Region in order to use the automatic connection feature in the EC2 console.
3. On the dashboard, choose **Create database**.
4. Under **Choose a database creation method**, check that **Standard create** is selected. If you choose **Easy create**, the VPC selector is not available. You must ensure that your database is in the same VPC as your EC2 instance in order to use the automatic connection feature in the EC2 console.
5. Under **Engine options**, for **Engine type**, choose **MySQL**.
6. Under **Templates**, choose a sample template to meet your needs. For this tutorial, choose **Free tier** to create an RDS database at no cost. However, note that the Free Tier is only available for accounts that qualify for the Free Tier. You can read more by choosing the **Info** link in the **Free tier** box.
7. Under **Settings**, do the following:
    a. For **DB instance identifier**, enter a name for the database. For this tutorial, enter **`tutorial-database-1`**.
    b. For **Master username**, leave the default name, which is **`admin`**.
    c. For **Master password**, enter a password that you can remember for this tutorial, and then, for **Confirm password**, enter the password again.
8. Under **Instance configuration**, for **DB instance class**, leave the default, which is **db.t3.micro**.If your account qualifies for the Free Tier, you can use this database class for free. For more information, see [AWS Free Tier].
9. Under **Connectivity**, for **Compute resource**, choose **Don't connect to an EC2 compute resource** because you'll connect the EC2 instance and the RDS database later in Task 3.

    (Later, in Option 2 of this tutorial, you'll try out the automatic connection feature in the RDS console by choosing **Connect to an EC2 compute resource**.)
10. For **Virtual private cloud (VPC)**, choose a VPC. The VPC must have a DB subnet group. To use the automatic connection feature, your EC2 instance and RDS database must be in the same VPC.
11. Keep all the default values for the other fields on this page.
12. Choose **Create database**.

On the **Databases** screen, the **Status** of the new database is **Creating** until the database is ready to use. When the status changes to **Available**, you can connect to the database. Depending on the database class and the amount of storage, it can take up to 20 minutes before the new database is available.

View an animation: Create an RDS database



Task 2 (*Optional*): Launch an EC2 instance

> **Note**
>
> Launching an instance is not the focus of this tutorial. If you already have an Amazon EC2 instance and would like to use it in this tutorial, you can skip this task.
> If you use an existing EC2 instance, make sure that it is in the same VPC as your RDS database so that you can use the automatic connection feature.

The objective of this task is to launch an EC2 instance so that you can complete Task 3 where you configure the connection between your EC2 instance and your Amazon RDS database. The steps in this task configure the EC2 instance as follows:

- Instance name: **`tutorial-instance-1`**
- AMI: Amazon Linux 2
- Instance type: `t2.micro`
- Auto-assign public IP: Enabled
- Security group with the following three rules:
  - Allow SSH from your IP address
  - Allow HTTPS traffic from anywhere
  - Allow HTTP traffic from anywhere

> **Important**
>
> In a production environment, you should configure your instance to meet your specific needs.

To launch an EC2 instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Region selector (at top right), choose an AWS Region. The instance and the RDS database must be in the same Region in order to use the automatic connection feature in the EC2 console.

3. On the **EC2 Dashboard**, choose **Launch instance**.
4. Under **Name and tags**, for **Name**, enter a name to identify your instance. For this tutorial, name the instance **`tutorial-instance-1`**. While the instance name is not mandatory, when you select your instance in the EC2 console, the name will help you easily identify it.
5. Under **Application and OS Images**, choose an AMI that meets your web server needs. This tutorial uses **Amazon Linux 2**.
6. Under **Instance type**, for **Instance type**, select an instance type that meets your web server needs. This tutorial uses `t2.micro`.

> **Note**
>
> Depending on when you created your account, you might be eligible to use Amazon EC2 under the Free Tier.
> If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type, or the **t3.micro** instance type in Regions where **t2.micro** is unavailable. Be aware that when you launch a **t3.micro** instance, it defaults to **Unlimited** mode, which might incur additional charges based on CPU usage. If an instance type can be used under the Free Tier, it is labeled **Free tier eligible**.
> If you created your AWS account on or after July 15, 2025, you can use **t3.micro**, **t3.small**, **t4g.micro**, **t4g.small**, **c7i-flex.large**, and **m7i-flex.large** instance types for 6 months or until your credits are used up.
> For more information, see Free Tier benefits before and after July 15, 2025.

7. Under **Key pair (login)**, for **Key pair name**, choose your key pair.
8. Under **Network settings**, do the following:

   a. For **Network** and **Subnet**, if you haven't made changes to your default VPC or subnets, you can keep the default settings.

      If you have made changes to your default VPC or subnets, check the following:

      i. The instance must be in the same VPC as the RDS database to use the automatic connection feature. By default you have only one VPC.
      ii. The VPC that you're launching your instance into must have an internet gateway attached to it so that you can access your web server from the internet. Your default VPC is automatically set up with an internet gateway.

iii. To ensure that your instance receives a public IP address, for **Auto-assign public IP**, check that **Enable** is selected. If **Disable** is selected, choose **Edit** (to the right of **Network Settings**), and then, for **Auto-assign public IP**, choose **Enable**.

b. To connect to your instance by using SSH, you need a security group rule that authorizes SSH (Linux) or RDP (Windows) traffic from your computer's public IPv4 address. By default, when you launch an instance, a new security group is created with a rule that allows inbound SSH traffic from anywhere.

   To make sure that only your IP address can connect to your instance, under **Firewall (security groups)**, from the drop-down list next to the **Allow SSH traffic from** checkbox, choose **My IP**.

c. To allow traffic from the internet to your instance, select the following checkboxes:

   - **Allow HTTPs traffic from the internet**
   - **Allow HTTP traffic from the internet**

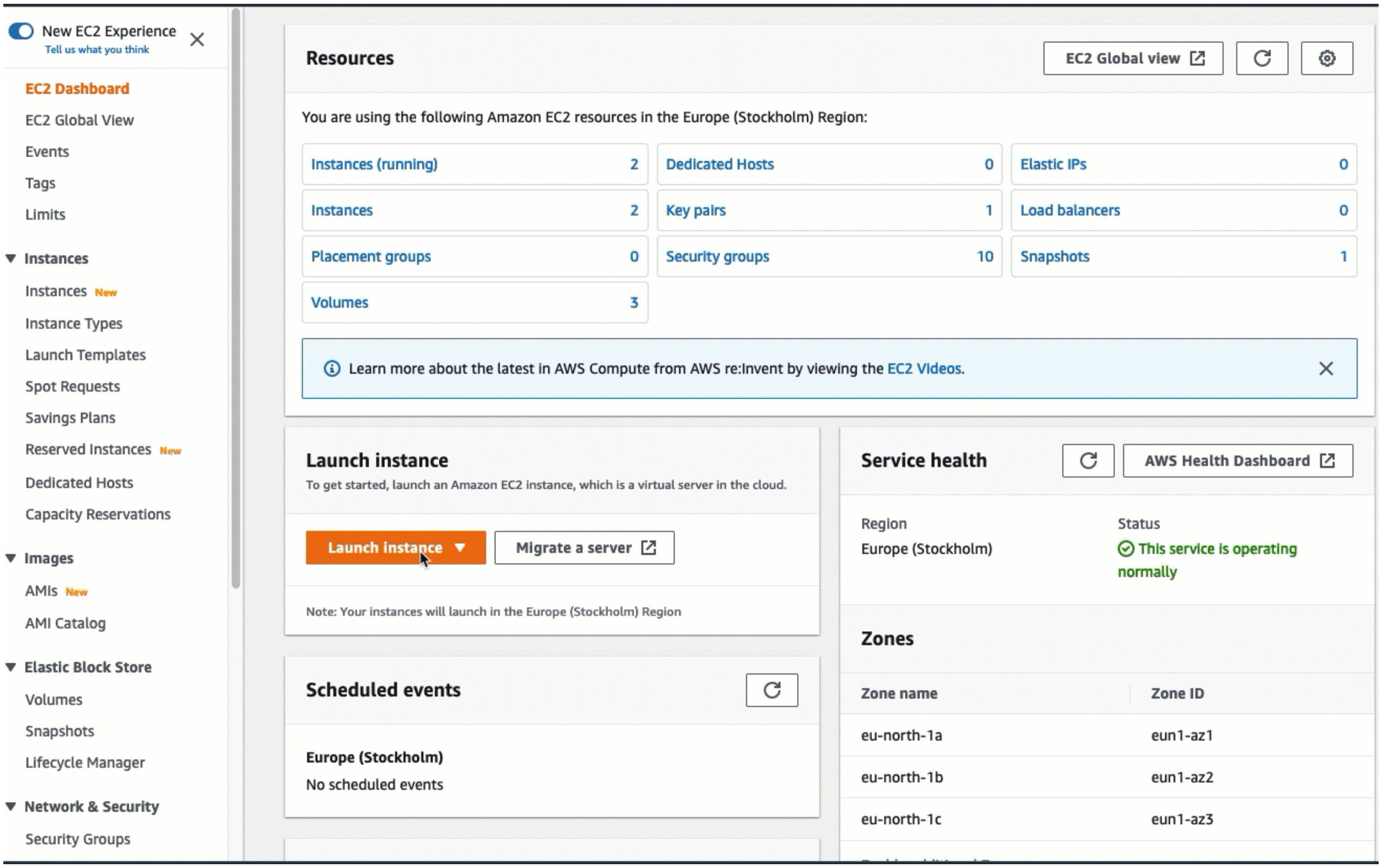
9. In the **Summary** panel, review your instance configuration and then choose **Launch instance**.

10. Keep the confirmation page open. You'll need it for the next task when you automatically connect your instance to your database.

    If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

For more information about launching an instance, see Launch an EC2 instance using the launch instance wizard in the console.

View an animation: Launch an EC2 instance



Task 3: Automatically connect your EC2 instance to your RDS database

The objective of this task is to use the automatic connection feature in the EC2 console to automatically configure the connection between your EC2 instance and your RDS database.

To automatically connect an EC2 instance to an RDS database using the EC2 console

1. On the instance launch confirmation page (it should be open from the previous task), choose **Connect an RDS database**.

   If you closed the confirmation page, follow these steps:

   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

   b. In the navigation pane, choose **Instances**.

   c. Select the EC2 instance that you just created, and then choose **Actions**, **Networking**, **Connect RDS database**.

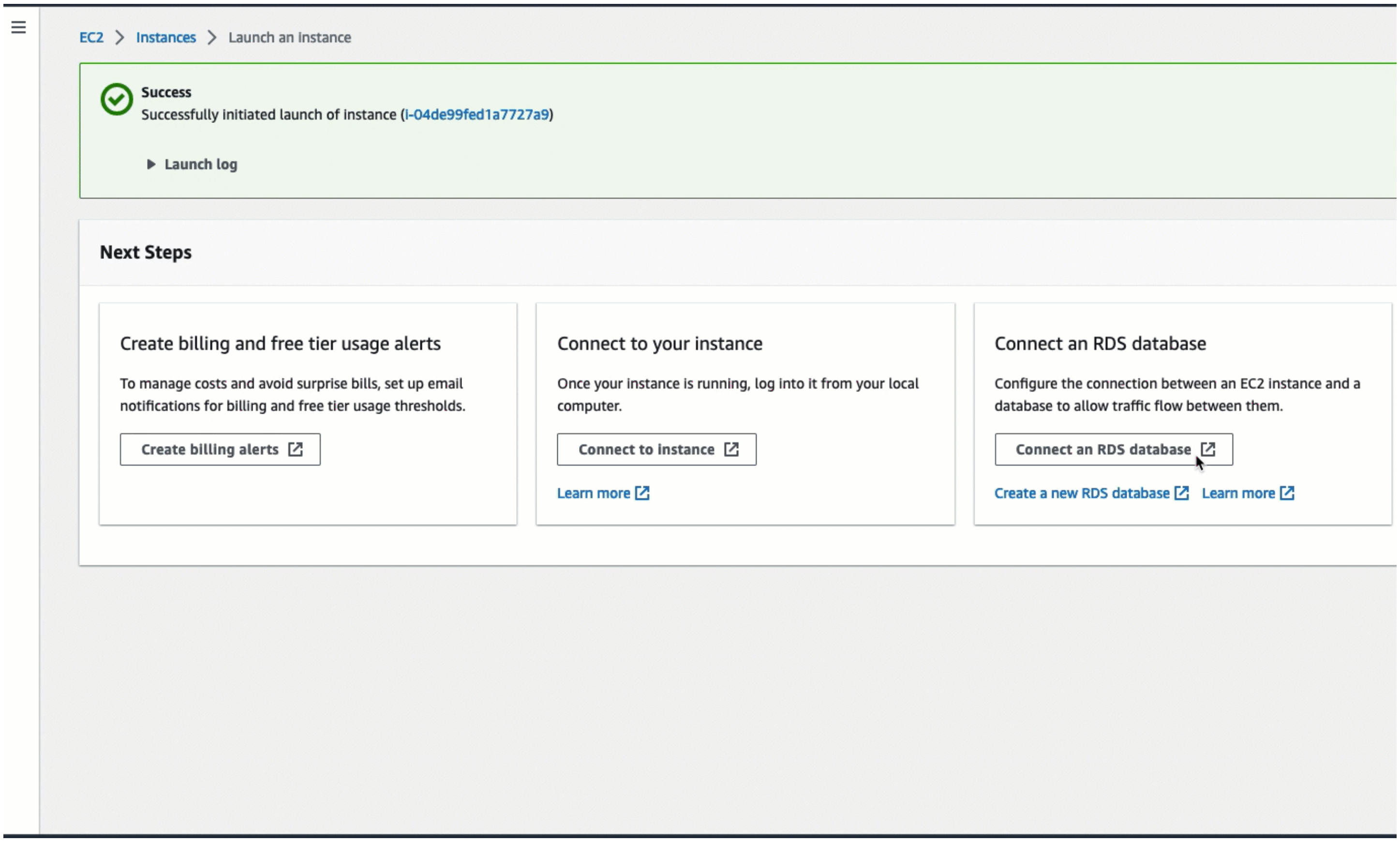
If **Connect RDS database** is not available, check that the EC2 instance is in the **Running** state.

2. For **Database role**, choose **Instance**. *Instance* in this case refers to the database instance.
3. For **RDS database**, choose the RDS database that you created in Task 1.

> **Note**
>
> The EC2 instance and the RDS database must be in the same VPC in order to connect to each other.

4. Choose **Connect**.

View an animation: Automatically connect a newly-launched EC2 instance to an RDS database



Task 4: Verify the connection configuration

The objective of this task is to verify that the two security groups were created and assigned to the instance and database.

When you use the automatic connection feature in the console to configure the connectivity, the security groups are automatically created and assigned to the instance and database, as follows:

- Security group **rds-ec2-*x*** is created and added to the RDS database. It has one inbound rule that references the **ec2-rds-*x*** security group as its source. This allows traffic from the EC2 instance with the **ec2-rds-*x*** security group to reach the RDS database.
- Security group **ec2-rds-*x*** is created and added to the EC2 instance. It has one outbound rule that references the **rds-ec2-*x*** security group as its destination. This allows traffic from the EC2 instance to reach the RDS database with the **rds-ec2-*x*** security group.

To verify the connection configuration using the console

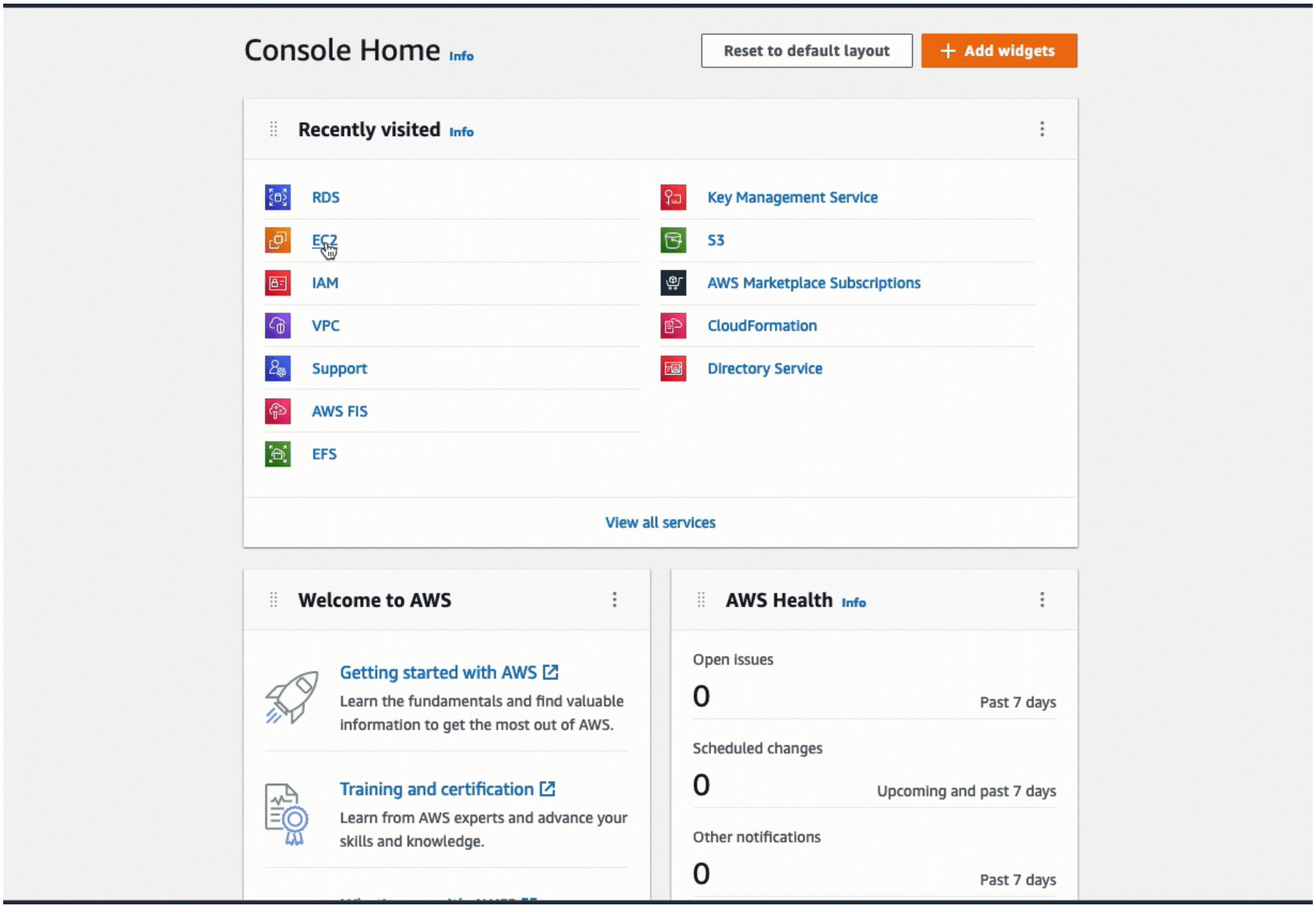
1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation page, choose **Databases**.
3. Choose the RDS database that you created for this tutorial.
4. On the **Connectivity & security** tab, under **Security**, **VPC security groups**, verify that a security group called **rds-ec2-*x*** is displayed.
5. Choose the **rds-ec2-*x*** security group. The **Security Groups** screen in the EC2 console opens.
6. Choose the **rds-ec2-*x*** security group to open it.
7. Choose the **Inbound rules** tab.
8. Verify that the following security group rule exists, as follows:
   - Type: **MYSQL/Aurora**
   - Port range: **3306**
   - Source: ***sg-0987654321example*** **/ ec2-rds-*x*** – This is the security group that is assigned to the EC2 instance that you verified in the preceding steps.
   - Description: **Rule to allow connections from EC2 instances with *sg-1234567890example* attached**
9. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
10. In the navigation pane, choose **Instances**.
11. Choose the EC2 instance that you selected to connect to the RDS database in the previous task, and choose the **Security** tab.
12. Under **Security details**, **Security groups**, verify that a security group called **ec2-rds-*x*** is in the list. *x* is a number.

13. Choose the **ec2-rds-*x*** security group to open it.
14. Choose the **Outbound rules** tab.
15. Verify that the following security group rule exists, as follows:

    - Type: **MYSQL/Aurora**
    - Port range: **3306**
    - Destination: ***sg-1234567890example* / rds-ec2-*x***
    - Description: **Rule to allow connections to `database-tutorial` from any instances this security group is attached to**

By verifying that these security groups and security group rules exist and that they are assigned to the RDS database and EC2 instance as described in this procedure, you can verify that the connection was automatically configured by using the automatic connection feature.

View an animation: Verify the connection configuration



You have completed Option 1 of this tutorial. You can now either complete Option 2, which teaches you how to use the RDS console to automatically connect an EC2 instance to an RDS database, or you can complete Option 3, which teaches you how to manually configure the security groups that were automatically created in Option 1.

Task 5 (*Optional*): Clean up

Now that you have completed the tutorial, it is good practice to clean up (delete) any resources you no longer want to use. Cleaning up AWS resources prevents your account from incurring any further charges.

If you launched an EC2 instance specifically for this tutorial, you can terminate it to stop incurring any charges associated with it.

To terminate 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 that you created for this tutorial, and choose **Instance state**, **Terminate instance**.
4. Choose **Terminate** when prompted for confirmation.

If you created an RDS database specifically for this tutorial, you can delete it to stop incurring any charges associated with it.

To delete an RDS database using the console

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, choose **Databases**.
3. Select the RDS database that you created for this tutorial, and choose **Actions**, **Delete**.
4. Enter **`delete me`** in the box, and then choose **Delete**.

Option 2: Automatically connect an instance to an RDS database using the RDS console

The objective of Option 2 is to explore the automatic connect feature in the RDS console that automatically configures the connection between your EC2 instance and RDS database to allow traffic from the EC2 instance to the RDS database. In Option 3, you'll learn how to manually configure the connection.

Tasks

- Before you begin
- Task 1 (Optional): Launch an EC2 instance
- Task 2: Create an RDS database and automatically connect it to your EC2 instance
- Task 3: Verify the connection configuration
- Task 4 (Optional): Clean up

Before you begin

You'll need the following to complete this tutorial:

- An EC2 instance that is in the same VPC as the RDS database. You can either use an existing EC2 instance or follow the steps in Task 1 to create a new instance.
- Permissions to call the following operations:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Task 1 (*Optional*): Launch an EC2 instance

> **Note**
>
> Launching an instance is not the focus of this tutorial. If you already have an Amazon EC2 instance and would like to use it in this tutorial, you can skip this task.

The objective of this task is to launch an EC2 instance so that you can complete Task 2 where you configure the connection between your EC2 instance and your Amazon RDS database. The steps in this task configure the EC2 instance as follows:

- Instance name: **`tutorial-instance-2`**
- AMI: Amazon Linux 2
- Instance type: `t2.micro`
- Auto-assign public IP: Enabled
- Security group with the following three rules:
  - Allow SSH from your IP address
  - Allow HTTPS traffic from anywhere
  - Allow HTTP traffic from anywhere

> **Important**
>
> In a production environment, you should configure your instance to meet your specific needs.

To launch an EC2 instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the **EC2 Dashboard**, choose **Launch instance**.
3. Under **Name and tags**, for **Name**, enter a name to identify your instance. For this tutorial, name the instance **`tutorial-instance-2`**. While the instance name is not mandatory, when you select your instance in the RDS console, the name will help you easily identify it.

4. Under **Application and OS Images**, choose an AMI that meets your web server needs. This tutorial uses **Amazon Linux**.

5. Under **Instance type**, for **Instance type**, select an instance type that meets your web server needs. This tutorial uses `t2.micro`.

> **Note**
>
> Depending on when you created your account, you might be eligible to use Amazon EC2 under the Free Tier.
> If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type, or the **t3.micro** instance type in Regions where **t2.micro** is unavailable. Be aware that when you launch a **t3.micro** instance, it defaults to **Unlimited** mode, which might incur additional charges based on CPU usage. If an instance type can be used under the Free Tier, it is labeled **Free tier eligible**.
> If you created your AWS account on or after July 15, 2025, you can use **t3.micro**, **t3.small**, **t4g.micro**, **t4g.small**, **c7i-flex.large**, and **m7i-flex.large** instance types for 6 months or until your credits are used up.
> For more information, see Free Tier benefits before and after July 15, 2025.

6. Under **Key pair (login)**, for **Key pair name**, choose your key pair.

7. Under **Network settings**, do the following:

   a. For **Network** and **Subnet**, if you haven't made changes to your default VPC or subnets, you can keep the default settings.

      If you have made changes to your default VPC or subnets, check the following:

      i. The instance must be in the same VPC as the RDS database to use the automatic connection configuration. By default you have only one VPC.

      ii. The VPC that you're launching your instance into must have an internet gateway attached to it so that you can access your web server from the internet. Your default VPC is automatically set up with an internet gateway.

      iii. To ensure that your instance receives a public IP address, for **Auto-assign public IP**, check that **Enable** is selected. If **Disable** is selected, choose **Edit** (to the right of **Network Settings**), and then, for **Auto-assign public IP**, choose **Enable**.

b. To connect to your instance by using SSH, you need a security group rule that authorizes SSH (Linux) or RDP (Windows) traffic from your computer's public IPv4 address. By default, when you launch an instance, a new security group is created with a rule that allows inbound SSH traffic from anywhere.

   To make sure that only your IP address can connect to your instance, under **Firewall (security groups)**, from the drop-down list next to the **Allow SSH traffic from** checkbox, choose **My IP**.

c. To allow traffic from the internet to your instance, select the following checkboxes:

   - **Allow HTTPs traffic from the internet**
   - **Allow HTTP traffic from the internet**

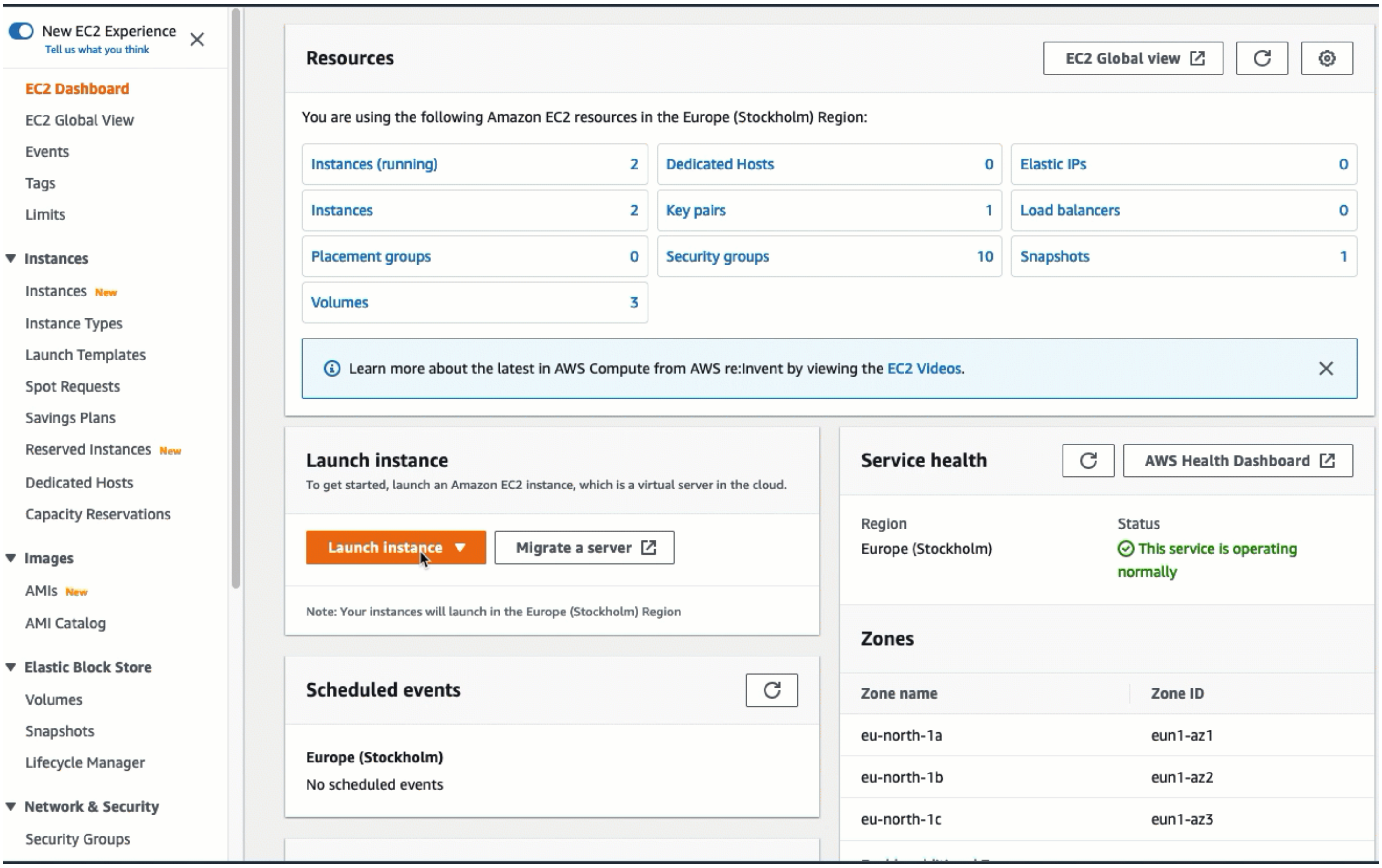
8. In the **Summary** panel, review your instance configuration and then choose **Launch instance**.

9. Choose **View all instances** to close the confirmation page and return to the console. Your instance will first be in a `pending` state, and will then go into the `running` state.

   If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

For more information about launching an instance, see Launch an EC2 instance using the launch instance wizard in the console.

View an animation: Launch an EC2 instance



Task 2: Create an RDS database and automatically connect it to your EC2 instance

The objective of this task is to create an RDS database and use the automatic connection feature in the RDS console to automatically configure the connection between your EC2 instance and your RDS database. The steps in this task configure the DB instance as follows:

- Engine type: MySQL
- Template: Free tier
- DB instance identifier: **`tutorial-database`**
- DB instance class: `db.t3.micro`

> **Important**
>
> In a production environment, you should configure your instance to meet your specific needs.

To create an RDS database and automatically connect it to an EC2 instance

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. From the Region selector (at top right), choose the AWS Region in which you created the EC2 instance. The EC2 instance and the RDS database must be in the same Region.
3. On the dashboard, choose **Create database**.
4. Under **Choose a database creation method**, check that **Standard create** is selected. If you choose **Easy create**, the automatic connection feature is not available.
5. Under **Engine options**, for **Engine type**, choose **MySQL**.
6. Under **Templates**, choose a sample template to meet your needs. For this tutorial, choose **Free tier** to create an RDS database at no cost. However, note that the Free Tier is only available if your account qualifies for the Free Tier. You can read more by choosing the **Info** link in the **Free tier** box.
7. Under **Settings**, do the following:

    a. For **DB instance identifier**, enter a name for the database. For this tutorial, enter **`tutorial-database`**.

    b. For **Master username**, leave the default name, which is **`admin`**.

    c. For **Master password**, enter a password that you can remember for this tutorial, and then, for **Confirm password**, enter the password again.

8. Under **Instance configuration**, for **DB instance class**, leave the default, which is **db.t3.micro**. If your account qualifies for the Free Tier, you can use this instance for free. For more information, see [AWS Free Tier](#).
9. Under **Connectivity**, for **Compute resource**, choose **Connect to an EC2 compute resource**. This is the automatic connection feature in the RDS console .
10. For **EC2 instance**, choose the EC2 instance that you want to connect to. For the purposes of this tutorial, you can either choose the instance that you created in the previous task, which you named **`tutorial-instance`**, or choose another existing instance. If you don't see your instance in the list, choose the refresh icon to the right of **Connectivity**.

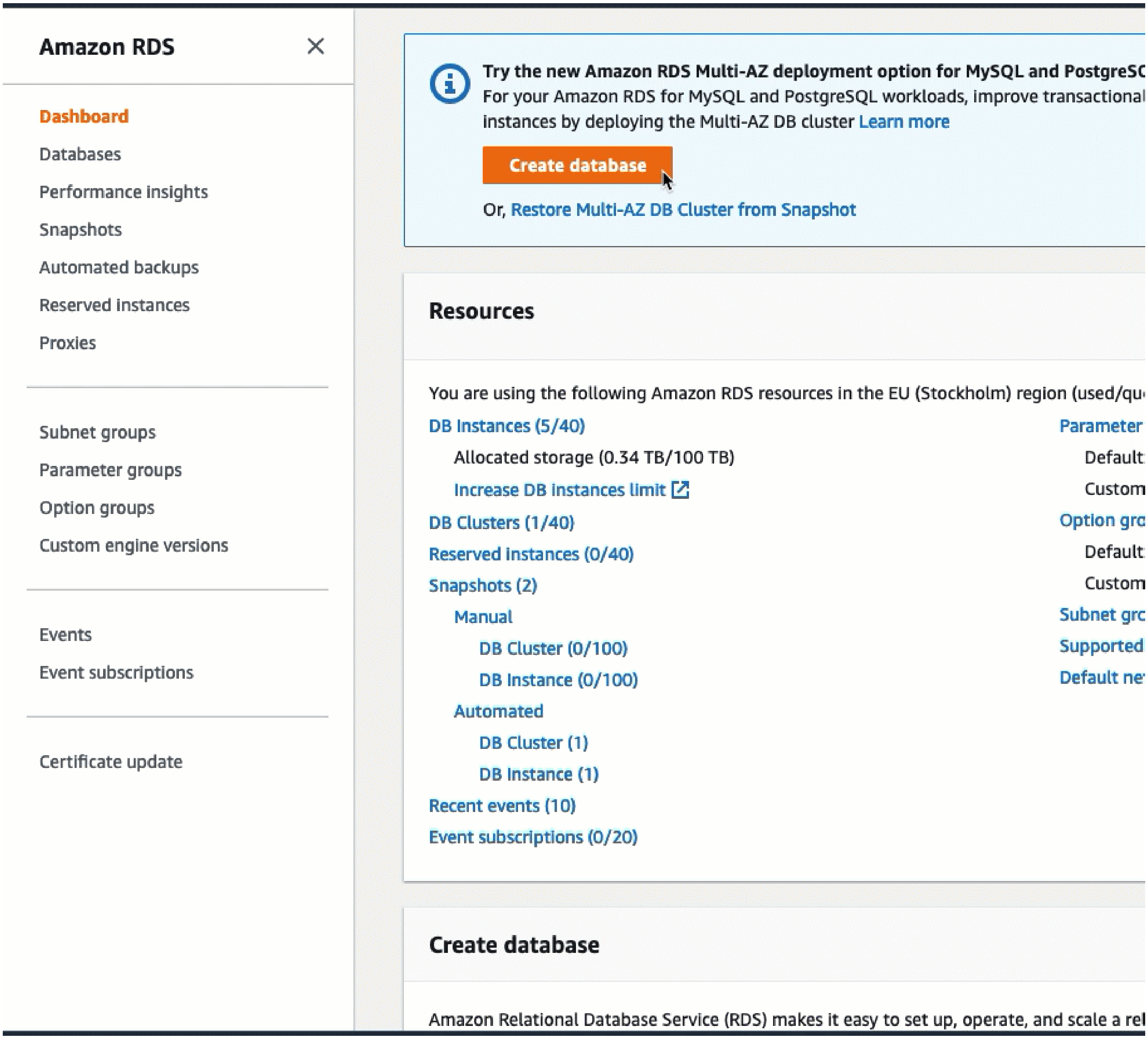
When you use the automatic connection feature, a security group is added to this EC2 instance, and another security group is added to the RDS database. The security groups are automatically configure to allow traffic between the EC2 instance and the RDS database. In the next task, you'll verify that the security groups were created and assigned to the EC2 instance and RDS database.

11. Choose **Create database**.

    On the **Databases** screen, the **Status** of the new database is **Creating** until the database is ready to use. When the status changes to **Available**, you can connect to the database. Depending on the database class and the amount of storage, it can take up to 20 minutes before the new database is available.

To learn more, see Configure automatic network connectivity with an EC2 instance in the *Amazon RDS User Guide*.

View an animation: Create an RDS database and automatically connect it to an EC2 instance



Task 3: Verify the connection configuration

The objective of this task is to verify that the two security groups were created and assigned to the instance and database.

When you use the automatic connection feature in the console to configure the connectivity, the security groups are automatically created and assigned to the instance and database, as follows:

- Security group **rds-ec2-*x*** is created and added to the RDS database. It has one inbound rule that references the **ec2-rds-*x*** security group as its source. This allows traffic from the EC2 instance with the **ec2-rds-*x*** security group to reach the RDS database.
- Security group **ec2-rds-*x*** is created and added to the EC2 instance. It has one outbound rule that references the **rds-ec2-*x*** security group as its destination. This allows traffic from the EC2 instance to reach the RDS database with the **rds-ec2-*x*** security group.

To verify the connection configuration using the console

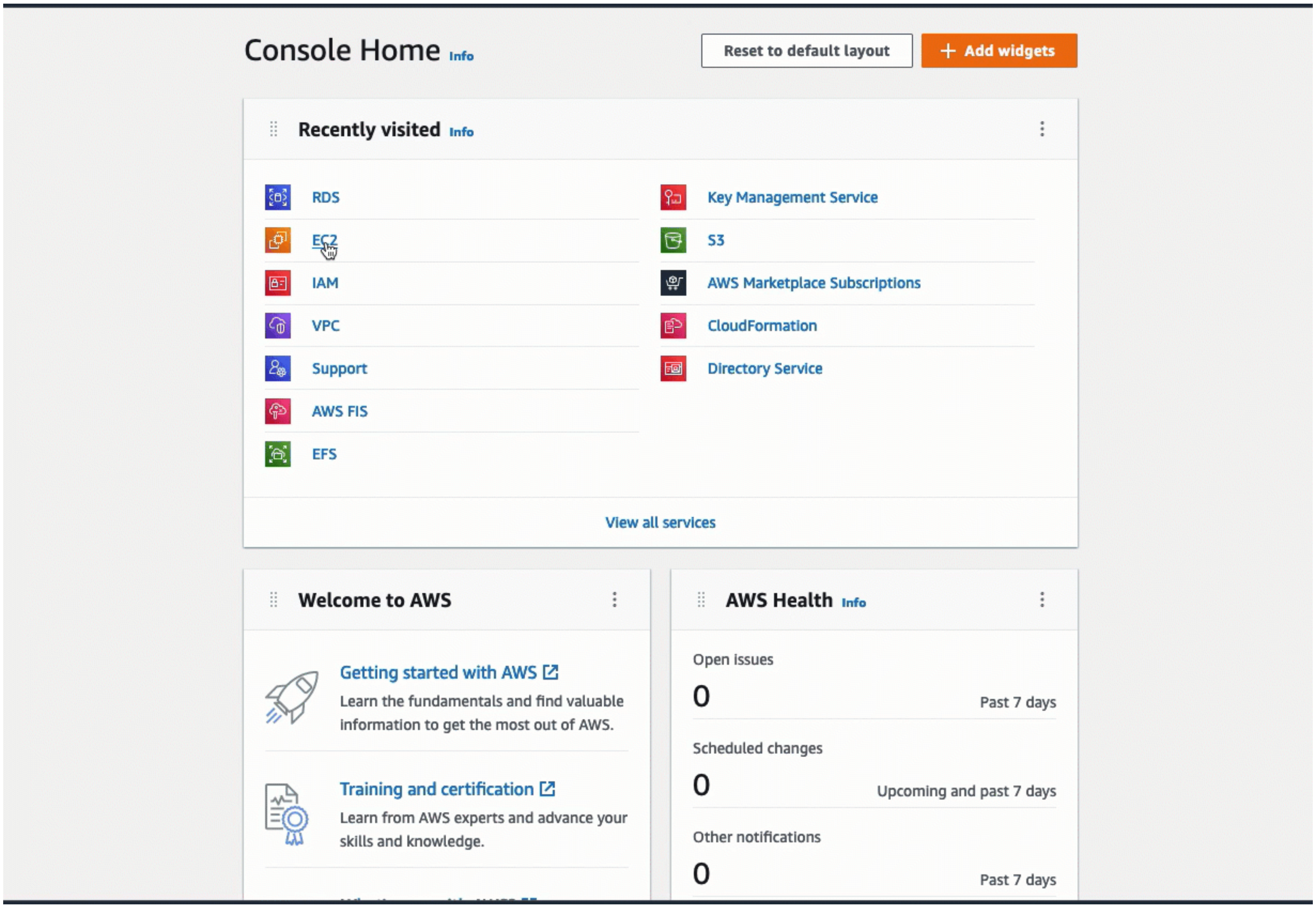
1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation page, choose **Databases**.
3. Choose the RDS database that you created for this tutorial.
4. On the **Connectivity & security** tab, under **Security**, **VPC security groups**, verify that a security group called **rds-ec2-*x*** is displayed.
5. Choose the **rds-ec2-*x*** security group. The **Security Groups** screen in the EC2 console opens.
6. Choose the **rds-ec2-*x*** security group to open it.
7. Choose the **Inbound rules** tab.
8. Verify that the following security group rule exists, as follows:
   - Type: **MYSQL/Aurora**
   - Port range: **3306**
   - Source: ***`sg-0987654321example`*** **/ ec2-rds-*x*** – This is the security group that is assigned to the EC2 instance that you verified in the preceding steps.
   - Description: **Rule to allow connections from EC2 instances with *`sg-1234567890example`* attached**
9. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
10. In the navigation pane, choose **Instances**.
11. Choose the EC2 instance that you selected to connect to the RDS database in the previous task, and choose the **Security** tab.
12. Under **Security details**, **Security groups**, verify that a security group called **ec2-rds-*x*** is in the list. *x* is a number.
13. Choose the **ec2-rds-*x*** security group to open it.
14. Choose the **Outbound rules** tab.

15. Verify that the following security group rule exists, as follows:

    * Type: **MYSQL/Aurora**
    * Port range: **3306**
    * Destination: ***sg-1234567890example*** **/ rds-ec2-*x***
    * Description: **Rule to allow connections to `database-tutorial` from any instances this security group is attached to**

By verifying that these security groups and security group rules exist and that they are assigned to the RDS database and EC2 instance as described in this procedure, you can verify that the connection was automatically configured by using the automatic connection feature.

View an animation: Verify the connection configuration



You have completed Option 2 of this tutorial. You can now either complete Option 3, which teaches you how to manually configure the security groups that were automatically created in Option 2.

Task 4 (*Optional*): Clean up

Now that you have completed the tutorial, it is good practice to clean up (delete) any resources you no longer want to use. Cleaning up AWS resources prevents your account from incurring any further charges.

If you launched an EC2 instance specifically for this tutorial, you can terminate it to stop incurring any charges associated with it.

To terminate 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 that you created for this tutorial, and choose **Instance state**, **Terminate instance**.
4. Choose **Terminate** when prompted for confirmation.

If you created an RDS database specifically for this tutorial, you can delete it to stop incurring any charges associated with it.

To delete an RDS database using the console

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, choose **Databases**.
3. Select the RDS database that you created for this tutorial, and choose **Actions**, **Delete**.
4. Enter **`delete me`** in the box, and then choose **Delete**.

Option 3: Manually connect an instance to an RDS database by creating security groups

The objective of Option 3 is to learn how to manually configure the connection between an EC2 instance and an RDS database by manually reproducing the configuration of the automatic connection feature.

Tasks

Before you begin

You'll need the following to complete this tutorial:

- An EC2 instance that is in the same VPC as the RDS database. You can either use an existing EC2 instance or follow the steps in Task 1 to create a new instance.
- An RDS database that is in the same VPC as the EC2 instance. You can either use an existing RDS database or follow the steps in Task 2 to create a new database.
- Permissions to call the following operations:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Task 1 (*Optional*): Launch an EC2 instance

> **Note**
>
> Launching an instance is not the focus of this tutorial. If you already have an Amazon EC2 instance and would like to use it in this tutorial, you can skip this task.

The objective of this task is to launch an EC2 instance so that you can complete Task 3 where you configure the connection between your EC2 instance and your Amazon RDS database. The steps in this task configure the EC2 instance as follows:

- Instance name: **`tutorial-instance`**
- AMI: Amazon Linux 2
- Instance type: `t2.micro`
- Auto-assign public IP: Enabled
- Security group with the following three rules:
  - Allow SSH from your IP address
  - Allow HTTPS traffic from anywhere
  - Allow HTTP traffic from anywhere

> **Important**
>
> In a production environment, you should configure your instance to meet your specific needs.

To launch an EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the **EC2 Dashboard**, choose **Launch instance**.
3. Under **Name and tags**, for **Name**, enter a name to identify your instance. For this tutorial, name the instance **`tutorial-instance-manual-1`**. While the instance name is not mandatory, the name will help you easily identify it.

4. Under **Application and OS Images**, choose an AMI that meets your web server needs. This tutorial uses **Amazon Linux**.

5. Under **Instance type**, for **Instance type**, select an instance type that meets your web server needs. This tutorial uses `t2.micro`.

> **Note**
>
> Depending on when you created your account, you might be eligible to use Amazon EC2 under the Free Tier.
> If your created your AWS account before July 15, 2025 and it's less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type, or the **t3.micro** instance type in Regions where **t2.micro** is unavailable. Be aware that when you launch a **t3.micro** instance, it defaults to **Unlimited** mode, which might incur additional charges based on CPU usage. If an instance type can be used under the Free Tier, it is labeled **Free tier eligible**.
> If you created your AWS account on or after July 15, 2025, you can use **t3.micro**, **t3.small**, **t4g.micro**, **t4g.small**, **c7i-flex.large**, and **m7i-flex.large** instance types for 6 months or until your credits are used up.
> For more information, see Free Tier benefits before and after July 15, 2025.

6. Under **Key pair (login)**, for **Key pair name**, choose your key pair.

7. Under **Network settings**, do the following:

   a. For **Network** and **Subnet**, if you haven't made changes to your default VPC or subnets, you can keep the default settings.

      If you have made changes to your default VPC or subnets, check the following:

      i. The instance must be in the same VPC as the RDS database. By default you have only one VPC.

      ii. The VPC that you're launching your instance into must have an internet gateway attached to it so that you can access your web server from the internet. Your default VPC is automatically set up with an internet gateway.

      iii. To ensure that your instance receives a public IP address, for **Auto-assign public IP**, check that **Enable** is selected. If **Disable** is selected, choose **Edit** (to the right of **Network Settings**), and then, for **Auto-assign public IP**, choose **Enable**.

b. To connect to your instance by using SSH, you need a security group rule that authorizes SSH (Linux) or RDP (Windows) traffic from your computer's public IPv4 address. By default, when you launch an instance, a new security group is created with a rule that allows inbound SSH traffic from anywhere.

   To make sure that only your IP address can connect to your instance, under **Firewall (security groups)**, from the drop-down list next to the **Allow SSH traffic from** checkbox, choose **My IP**.

c. To allow traffic from the internet to your instance, select the following checkboxes:

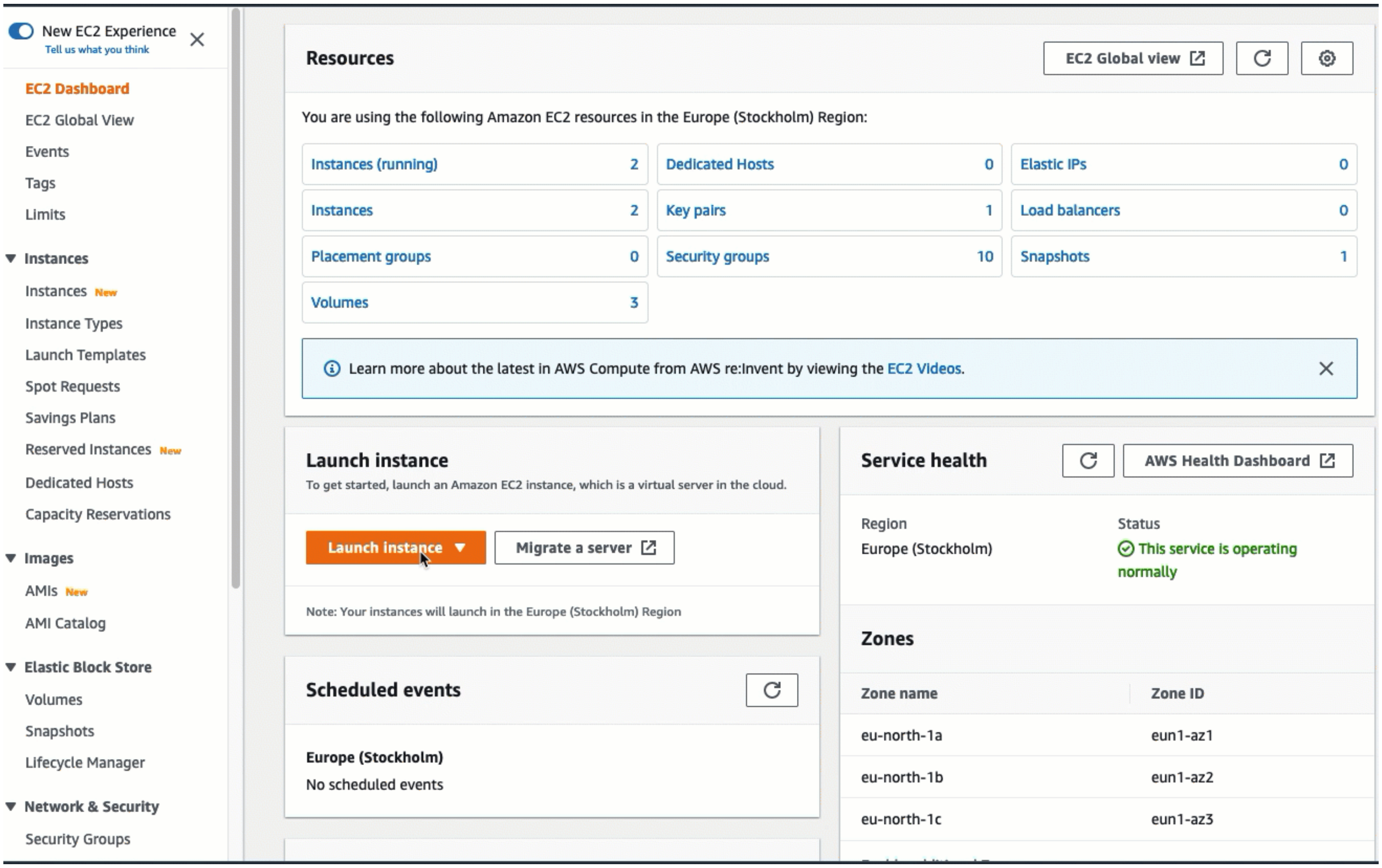
   - **Allow HTTPs traffic from the internet**
   - **Allow HTTP traffic from the internet**

8. In the **Summary** panel, review your instance configuration and then choose **Launch instance**.
9. Choose **View all instances** to close the confirmation page and return to the console. Your instance will first be in a `pending` state, and will then go into the `running` state.

   If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot Amazon EC2 instance launch issues.

For more information about launching an instance, see Launch an EC2 instance using the launch instance wizard in the console.

View an animation: Launch an EC2 instance



Task 2 (*Optional*): Create an RDS database

> **Note**
>
> Creating an RDS database is not the focus of this part of the tutorial. If you already have an RDS database and would like to use it for this tutorial, you can skip this task.

The objective of this task is to create an RDS database. You'll use this instance in Task 3 when you connect it to your EC2 instance. The steps in this task configure the RDS database as follows:

- Engine type: MySQL
- Template: Free tier
- DB instance identifier: **`tutorial-database-manual`**
- DB instance class: `db.t3.micro`

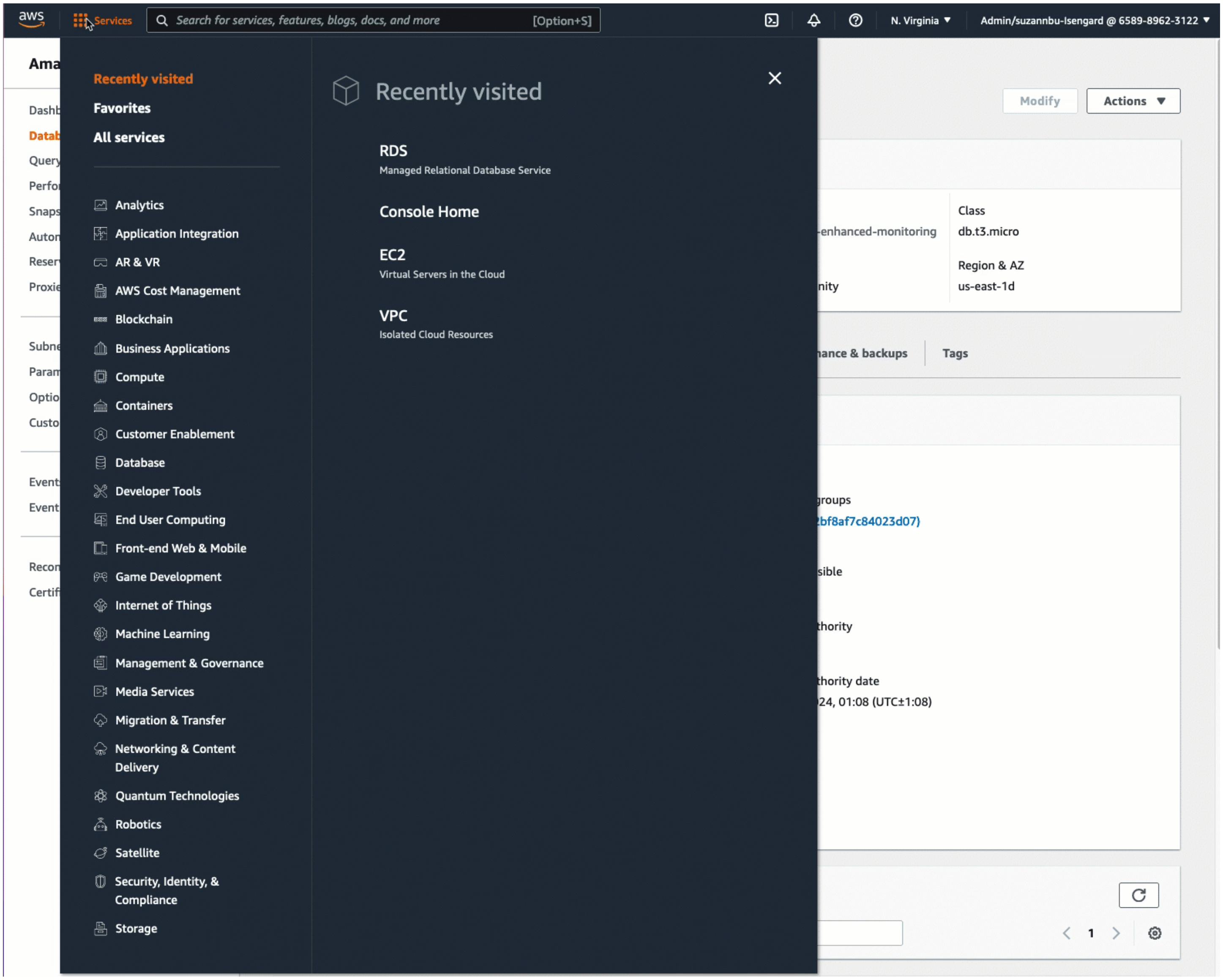
> **Important**
>
> In a production environment, you should configure your instance to meet your specific needs.

To create a MySQL DB instance

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. From the Region selector (at top right), choose the AWS Region in which you created the EC2 instance. The EC2 instance and the DB instance must be in the same Region.
3. On the dashboard, choose **Create database**.
4. Under **Choose a database creation method**, choose **Easy create**. When you choose this option, the automatic connection feature to automatically configure the connection is not available.
5. Under **Engine options**, for **Engine type**, choose **MySQL**.
6. For **DB instance size**, choose **Free tier**.
7. For **DB instance identifier** enter a name for the RDS database. For this tutorial, enter **`tutorial-database-manual`**.
8. For **Master username**, leave the default name, which is **`admin`**.
9. For **Master password**, enter a password that you can remember for this tutorial, and then, for **Confirm password**, enter the password again.
10. Choose **Create database**.

    On the **Databases** screen, the **Status** of the new DB instance is **Creating** until the DB instance is ready to use. When the status changes to **Available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.

View an animation: Create a DB instance



Task 3: Manually connect your EC2 instance to your RDS database by creating security groups and assigning them to the instances

The objective of this task is to reproduce the connection configuration of the automatic connection feature by performing the following manually: You create two new security groups, and then add a security group each to the EC2 instance and the RDS database.

To create two new security groups and assign one each to the EC2 instance and RDS database

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. First create the security group to add to the EC2 instance, as follows:

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

b. Choose **Create security group**.

c. For **Security group name**, enter a descriptive name for the security group. For this tutorial, enter **`ec2-rds-manual-configuration`**.

d. For **Description**, enter a brief description. For this tutorial, enter **`EC2 instance security group to allow EC2 instance to securely connect to RDS database`**.

e. Choose **Create security group**. You'll come back to this security group to add an outbound rule after you've created the RDS database security group.

3. Now, create the security group to add to the RDS database, as follows:

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

   b. Choose **Create security group**.

   c. For **Security group name**, enter a descriptive name for the security group. For this tutorial, enter **`rds-ec2-manual-configuration`**.

   d. For **Description**, enter a brief description. For this tutorial, enter **`RDS database security group to allow EC2 instance to securely connect to RDS database`**.

   e. Under **Inbound rules**, choose **Add rule**, and do the following:

      i. For **Type**, choose **MYSQL/Aurora**.

      ii. For **Source**, choose the EC2 instance security group **ec2-rds-manual-configuration** that you created in Step 2 of this procedure.

   f. Choose **Create security group**.

4. Edit the EC2 instance security group to add an outbound rule, as follows:

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

   b. Select the EC2 instance security group (you named it **`ec2-rds-manual-configuration`**), and choose the **Outbound rules** tab.

   c. Choose **Edit outbound rules**.

   d. Choose **Add rule**, and do the following:

      i. For **Type**, choose **MYSQL/Aurora**.

    ii. For **Destination**, choose the RDS database security group **rds-ec2-manual-configuration** that you created in Step 3 of this procedure.

    iii. Choose **Save rules**.

5. Add the EC2 instance security group to the EC2 instance as follows:

    a. In the navigation pane, choose **Instances**.

    b. Select your EC2 instance, and choose **Actions**, **Security**, **Change security groups**.

    c. Under **Associated security groups**, choose the **Select security groups** field, choose **ec2-rds-manual-configuration** that you created earlier, and then choose **Add security group**.

    d. Choose **Save**.

6. Add the RDS database security group to the RDS database as follows:

    a. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.

    b. In the navigation pane, choose **Databases** and select your database.

    c. Choose **Modify**.

    d. Under **Connectivity**, for **Security group**, choose **rds-ec2-manual-configuration** that you created earlier, and then choose **Continue**.

    e. Under **Scheduling of modifications**, choose **Apply immediately**.

    f. Choose **Modify DB instance**.

    You have now completed the manual steps that mimic the automatic steps that occur when you use the automatic connection feature.

You have completed Option 3 of this tutorial. If you've completed Options 1, 2, and 3, and you no longer need the resources that were created in this tutorial, you should delete them to prevent incurring unnecessary costs. For more information, see Task 4 (*Optional*): Clean up.

Task 4 (*Optional*): Clean up

Now that you have completed the tutorial, it is good practice to clean up (delete) any resources you no longer want to use. Cleaning up AWS resources prevents your account from incurring any further charges.

If you launched an EC2 instance specifically for this tutorial, you can terminate it to stop incurring any charges associated with it.

To terminate 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 that you created for this tutorial, and choose **Instance state**, **Terminate instance**.
4. Choose **Terminate** when prompted for confirmation.

If you created an RDS database specifically for this tutorial, you can delete it to stop incurring any charges associated with it.

To delete an RDS database using the console

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, choose **Databases**.
3. Select the RDS database that you created for this tutorial, and choose **Actions**, **Delete**.
4. Enter **`delete me`** in the box, and then choose **Delete**.

EC2 Fleet and Spot Fleet

EC2 Fleet and Spot Fleet are designed to be a useful way to launch a fleet of tens, hundreds, or thousands of Amazon EC2 instances in a single operation. Each instance in a fleet is either configured by a launch template or a set of launch parameters that you configure manually at launch.

Topics

Features and benefits

Fleets provide the following features and benefits, enabling you to maximize cost savings and optimize availability and performance when running applications on multiple EC2 instances.

Multiple instance types

A fleet can launch multiple instance types, ensuring it isn't dependent on the availability of any single instance type. This increases the overall availability of instances in your fleet.

Distributing instances across Availability Zones

A fleet can launch into multiple Availability Zones, enabling you to reduce costs and improve availability. If your fleet includes Spot Instances, the fleet automatically selects Availability Zones based on your preferences regarding price and interruptions.

Multiple purchasing options

A fleet can launch multiple purchase options (Spot and On-Demand Instances), allowing you to optimize costs through Spot Instance usage. You can also take advantage of Reserved Instance and Savings Plans discounts by using them in conjunction with On-Demand Instances in the fleet.

Automated replacement of Spot Instances

If your fleet includes Spot Instances, it can automatically request replacement Spot capacity if your Spot Instances are interrupted. Through Capacity Rebalancing, a fleet can also monitor and proactively replace your Spot Instances that are at an elevated risk of interruption.

Reserve On-Demand capacity

A fleet can use an On-Demand Capacity Reservation to reserve On-Demand capacity. A fleet can also include Capacity Blocks for ML, allowing you to reserve GPU instances on a future date to support short duration machine learning (ML) workloads.

Which is the best fleet method to use?

As a general best practice, we recommend launching fleets of Spot and On-Demand Instances with Amazon EC2 Auto Scaling because it provides additional features you can use to manage your fleet. The list of additional features includes automatic health check replacements for both Spot and On-Demand Instances, application-based health checks, and an integration with Elastic Load Balancing to ensure an even distribution of application traffic to your healthy instances. You can also use Auto Scaling groups when you use AWS services such as Amazon ECS, Amazon EKS (self-managed node groups), and Amazon VPC Lattice. For more information, see the Amazon EC2 Auto Scaling User Guide.

If you can't use Amazon EC2 Auto Scaling, then you might consider using EC2 Fleet or Spot Fleet. EC2 Fleet and Spot Fleet offer the same core functionality. However, EC2 Fleet is only available using a command line and does not provide console support. Spot Fleet provides console support, but is based on a legacy API with no planned investment.

Use the following table to determine which fleet method to use.

Fleet method	When to use?	Use case
[Amazon EC2 Auto Scaling]()	• You need multiple instances with either a single configuration or a mixed configuration. • You want to automate the lifecycle management of your instances.	Create an Auto Scaling group that manages the lifecycle of your instances while maintaini ng the desired number of instances. Supports horizontal scaling (adding more instances) between specified minimum and maximum limits.
[EC2 Fleet]()	• You need multiple instances with either a single configuration or a mixed configuration. • You want to self-manage your instance lifecycle. • If you don't need auto scaling, we recommend that you use an `instant` type EC2 Fleet.	Create an `instant` fleet of both On-Demand Instances and Spot Instances in a single operation, with multiple launch specifications that vary by instance type, AMI, Availability Zone, or subnet. The Spot Instance allocation strategy defaults to `lowest-price` per unit, but we recommend changing it to `price-capacity-optimized`.
[Spot Fleet]()	• We strongly discourage using Spot Fleet because it is based on a legacy API with no planned investmen t. • If you want to manage your instance lifecycle, rather use EC2 Fleet. • If you don't want to manage your instance	Use Spot Fleet only if you need console support for a use case for when you would use EC2 Fleet.

Fleet method	When to use?	Use case
	lifecycle, rather use an Auto Scaling group.	

Configuration options for your EC2 Fleet or Spot Fleet

When planning your EC2 Fleet or Spot Fleet, we recommend that you consider the following options when deciding how to configure your fleet.

Configuration option	Question	Documentation
Fleet request type	Do you want a fleet that submits a one-time request for the desired target capacity, or a fleet that maintains target capacity over time?	EC2 Fleet and Spot Fleet request types
Spot Instances	Do you plan to include Spot Instances in your fleet? Review the Spot best practices and use them when you plan your fleet so that you can provision the instances at the lowest possible price.	Best practices for Amazon EC2 Spot
Spending limit for your fleet	Do you want to limit how much you'll pay for your fleet per hour?	Set a spending limit for your EC2 Fleet or Spot Fleet
Instance types and attribute-based instance type selection	Do you want to specify the instance types in your fleet, or let Amazon EC2 select the instance types that meet your application requirements?	Specify attributes for instance type selection for EC2 Fleet or Spot Fleet

Configuration option	Question	Documentation
Instance weighting	Do you want to assign weights to each instance type to represent their compute capacity and performance, so that Amazon EC2 can select any combination of available instance types to fulfil your desired target capacity?	Use instance weighting to manage cost and performance of your EC2 Fleet or Spot Fleet
Allocation strategies	Do you want to decide whether to optimize for available capacity, price, or instance types to use for the Spot Instances and On-Demand Instances in your fleet?	Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity
Capacity Rebalancing	Do you want your fleet to automatically replace at-risk Spot Instances?	Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances
On-Demand Capacity Reservation	Do you want to reserve capacity for the On-Demand Instances in your fleet?	Use Capacity Reservations to reserve On-Demand capacity in EC2 Fleet

EC2 Fleet and Spot Fleet request types

The request type for an EC2 Fleet or Spot Fleet determines whether the request is synchronous or asynchronous, and whether it is a one-time request for the desired target capacity or an ongoing effort to maintain the capacity over time. When configuring your fleet, you must specify the request type.

Both EC2 Fleet and Spot Fleet offer two request types: `request` and `maintain`. In addition, EC2 Fleet offers a third request type called `instant`.

Fleet request types

`instant` (EC2 Fleet only)

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 and provides errors for those instances that could not be launched. For more information, see Configure an EC2 Fleet of type instant.

`request`

If you configure the request type as `request`, the fleet places an asynchronous one-time request for your desired capacity. If capacity diminishes due to 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. When creating a Spot Fleet of type `request` using the console, clear the **Maintain target capacity** checkbox.

`maintain` (default)

If you configure the request type as `maintain`, the fleet places an asynchronous request for your desired capacity, and maintains it by automatically replenishing any interrupted Spot Instances. When creating a Spot Fleet of type `maintain` using the console, select the **Maintain target capacity** checkbox

Configure 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 price-capacity-optimized allocation strategy.**

    Benefit: The `price-capacity-optimized` allocation strategy identifies instances from the most-available Spot capacity pools, and then automatically provisions instances from the lowest priced of these 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.

You can use EC2 Fleet of type *instant* to launch instances into Capacity Blocks. For more information, see [Tutorial: Configure your EC2 Fleet to launch instances into Capacity Blocks]().

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.

How instant EC2 Fleet works

When working with an EC2 Fleet of type `instant`, the sequence of events is as follows:

1. **Configure:** Configure the CreateFleet request type as `instant`. For more information, see Create an EC2 Fleet. Note that after you make the API call, you can't modify it.
2. **Request:** When you make the API call, Amazon EC2 places a synchronous one-time request for your desired capacity.
3. **Response:** The API response lists the instances that launched, along with errors for those instances that could not be launched.
4. **Describe:** You can describe your EC2 Fleet, list the instances associated with your EC2 Fleet, and view the history of your EC2 Fleet.
5. **Terminate instances:** You can terminate the instances at any time.
6. **Delete fleet request:** The fleet request can be deleted either manually or automatically:
   - Manual: You can delete the fleet request after your instances launch.

     Note that a deleted `instant` fleet with running instances is not supported. When you delete an `instant` fleet, Amazon EC2 automatically terminates all its instances. For fleets with more than 1000 instances, the deletion request might fail. If your fleet has more than 1000 instances, first terminate most of the instances manually, leaving 1000 or fewer. Then delete the fleet, and the remaining instances will be terminated automatically.
   - Automatic: Amazon EC2 deletes the fleet request some time after either:
     - All the instances are terminated.
     - The fleet fails to launch any instances.

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 in the *Amazon EC2 API Reference*.

Examples

- Example 1: Launch Spot Instances with the capacity-optimized allocation strategy

Example 1: Launch Spot Instances with the capacity-optimized 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 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.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
```

```
            "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"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
```

```
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

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.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "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"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
```

```
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

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 Use instance weighting to manage cost and performance of your EC2 Fleet or Spot Fleet.

```
{
   "SpotOptions":{
      "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":"subnet-49e41922",
               "WeightedCapacity":8
            }
         ]
      }
   ],
   "TargetCapacitySpecification":{
      "TotalTargetCapacity":40,
      "DefaultTargetCapacityType":"spot"
   },
   "Type":"instant"
}
```

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.

```
{
    "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"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "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 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.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "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"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "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 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.

When specifying MinTargetCapacity, you must specify at least one of these parameters: SingleInstanceType or SingleAvailabilityZone. In this example, SingleInstanceType is specified, so that all 20 instances must use the same instance type.

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, and 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.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": 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":"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.

```
{
    "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":"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 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 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-4xl or 4 c5n.9xlarge (144 divided by 36) using the launch template ec2-fleet-9xl, or 2 c5n.18xlarge (144 divided by 72) using the launch template ec2-fleet-18xl, or a mix of the instance types with weights adding up to the desired capacity based on the capacity-optimized allocation strategy.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
```

```
            "LaunchTemplateSpecification":{
               "LaunchTemplateName":"ec2-fleet-lt-18xl",
               "Version":"$Latest"
            },
            "Overrides":[
               {
                  "InstanceType":"c5n.18xlarge",
                  "SubnetId":"subnet-fae8c380",
                  "WeightedCapacity":72
               },
               {
                  "InstanceType":"c5n.18xlarge",
                  "SubnetId":"subnet-e7188bab",
                  "WeightedCapacity":72
               },
               {
                  "InstanceType":"c5n.18xlarge",
                  "SubnetId":"subnet-49e41922",
                  "WeightedCapacity":72
               }
            ]
         },
         {
            "LaunchTemplateSpecification":{
               "LaunchTemplateName":"ec2-fleet-lt-9xl",
               "Version":"$Latest"
            },
            "Overrides":[
               {
                  "InstanceType":"c5n.9xlarge",
                  "SubnetId":"subnet-fae8c380",
                  "WeightedCapacity":36
               },
               {
                  "InstanceType":"c5n.9xlarge",
                  "SubnetId":"subnet-e7188bab",
                  "WeightedCapacity":36
               },
               {
                  "InstanceType":"c5n.9xlarge",
                  "SubnetId":"subnet-49e41922",
                  "WeightedCapacity":36
               }
            ]
```

```
      },
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-4xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.4xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":16
            },
            {
               "InstanceType":"c5n.4xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":16
            },
            {
               "InstanceType":"c5n.4xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":16
            }
         ]
      }
    ],
    "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 TotalTargetCapacity – OnDemandTargetCapacity = 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.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "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"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

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"
    },
    "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": 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
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 12.0
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 10,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

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 c3.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](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-capacity-reservations.html) 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.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 0
}
```

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.

```
{
    "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
             }
          ]
       }
     ],
     "TargetCapacitySpecification": {
         "TotalTargetCapacity": 20,
         "DefaultTargetCapacityType": "spot"
     },
     "Type": "instant"
}
```

Example 12: Specify a Systems Manager parameter instead of an AMI ID

The following example uses a launch template to specify the configuration for the instances in the fleet. In this example, for `ImageId`, instead of specifying an AMI ID, the AMI is referenced with a System Manager parameter. On instance launch, the Systems Manager parameter will resolve to an AMI ID.

In this example, the Systems Manager parameter is specified in a valid format: `resolve:ssm:golden-ami`. There are other valid formats for the Systems Manager parameter. For more information, see [Use a Systems Manager parameter instead of an AMI ID](#).

> **Note**
>
> The fleet type must be of type `instant`. Other fleet types do not support specifying a System Manager parameter instead of an AMI ID.

```
{
```

```
    "LaunchTemplateData": {
        "ImageId": "resolve:ssm:golden-ami",
        "InstanceType": "m5.4xlarge",
        "TagSpecifications": [{
            "ResourceType": "instance",
            "Tags": [{
                "Key": "Name",
                "Value": "webserver"
            }]
        }]
    }
}
```

Set a spending limit for your EC2 Fleet or Spot Fleet

You can set a limit on how much you're willing to spend per hour on your EC2 Fleet or Spot Fleet. When your spending limit is reached, the fleet stops launching instances, even if the target capacity hasn't been reached.

There are separate spending limits for On-Demand Instances and Spot Instances.

To configure a spending limit for On-Demand Instances and Spot Instances in your EC2 Fleet

Use the create-fleet command and the following parameters:

- For On-Demand Instances: In the `OnDemandOptions` structure, specify your spending limit in the `MaxTotalPrice` field.
- For Spot Instances: In the `SpotOptions` structure, specify your spending limit in the `MaxTotalPrice` field.

To configure a spending limit for On-Demand Instances and Spot Instances in your Spot Fleet

You can use the Amazon EC2 console or the AWS CLI to configure your spending limit.

(Console) When creating the Spot Fleet, select the **Set maximum cost for Spot Instances** checkbox, and then enter a value for **Set your max cost (per hour)**. For more information, see step 6.e. in Create a Spot Fleet request using defined parameters.

(AWS CLI) Use the request-spot-fleet command and the following parameters:

- For On-Demand Instances: Specify your spending limit in the `OnDemandMaxTotalPrice` field.

- For Spot Instances: Specify your spending limit in the `SpotMaxTotalPrice` field.

Examples

The following examples show two different scenarios. In the first example, the fleet stops launching On-Demand Instances when it has met the target capacity set for On-Demand Instances (`OnDemandTargetCapacity`). In the second example, the fleet stops launching On-Demand Instances when it has reached the maximum amount you're willing to pay per hour for On-Demand Instances (`MaxTotalPrice`).

Example: Stop launching On-Demand 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

The 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 On-Demand 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 the 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, the fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the `MaxTotalPrice` for On-Demand Instances.

Burstable performance instances

If you launch your Spot Instances using a burstable performance instance type, 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 to avoid paying higher costs. If you launch burstable performance Spot Instances in Unlimited mode 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.

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 for T2 Spot Instances, or use an instance type with dedicated CPU.

Specify attributes for instance type selection for EC2 Fleet or Spot Fleet

When you create an EC2 Fleet or Spot Fleet, you must specify one or more instance types for configuring the On-Demand Instances and Spot Instances in the fleet. As an alternative to manually specifying the instance types, you can specify the attributes that an instance must have, and Amazon EC2 will identify all the instance types with those attributes. This is known as *attribute-based instance type selection*. For example, you can specify the minimum and maximum number of vCPUs required for your instances, and the fleet will launch the instances using any available instance types that meet those vCPU requirements.

Attribute-based instance type selection is ideal for workloads and frameworks that can be flexible about what instance types they use, such as when running containers or web fleets, processing big data, and implementing continuous integration and deployment (CI/CD) tooling.

Benefits

Attribute-based instance type selection has the following benefits:

- **Easily use the right instance types** – With so many instance types available, finding the right instance types for your workload can be time consuming. When you specify instance attributes, the instance types will automatically have the required attributes for your workload.
- **Simplified configuration** – To manually specify multiple instance types for a fleet, you must create a separate launch template override for each instance type. But with attribute-based instance type selection, to provide multiple instance types, you need only specify the instance attributes in the launch template or in a launch template override.
- **Automatic use of new instance types** – When you specify instance attributes rather than instance types, your fleet can use newer generation instance types as they're released, "future proofing" the fleet's configuration.
- **Instance type flexibility** – When you specify instance attributes rather than instance types, the fleet can select from a wide range of instance types for launching Spot Instances, which adheres to the Spot best practice of instance type flexibility.

Topics

How attribute-based instance type selection works

To use attribute-based instance type selection in your fleet configuration, you replace the list of instance types with a list of instance attributes that your instances require. EC2 Fleet or Spot Fleet will launch instances on any available instance types that have the specified instance attributes.

Topics

- How EC2 Fleet or Spot Fleet uses attribute-based instance type selection when provisioning a fleet

Types of instance attributes

There are several instance attributes that you can specify to express your compute requirements, such as:

- **vCPU count** – The minimum and maximum number of vCPUs per instance.
- **Memory** – The minimum and maximum GiBs of memory per instance.
- **Local storage** – Whether to use EBS or instance store volumes for local storage.
- **Burstable performance** – Whether to use the T instance family, including T4g, T3a, T3, and T2 types.

For a description of each attribute and the default values, see InstanceRequirements in the *Amazon EC2 API Reference*.

Where to configure attribute-based instance type selection

Depending on whether you use the console or the AWS CLI, you can specify the instance attributes for attribute-based instance type selection as follows:

In the console, you can specify the instance attributes in the following fleet configuration components:

- In a launch template, and then reference the launch template in the fleet request
- (Spot Fleet only) In the fleet request

In the AWS CLI, you can specify the instance attributes in one or all of the following fleet configuration components:

- In a launch template, and then reference the launch template in the fleet request
- In a launch template override

  If you want a mix of instances that use different AMIs, you can specify instance attributes in multiple launch template overrides. For example, different instance types can use x86 and Arm-based processors.

- (Spot Fleet only) In a launch specification

How EC2 Fleet or Spot Fleet uses attribute-based instance type selection when provisioning a fleet

EC2 Fleet or Spot Fleet provisions a fleet in the following way:

- It identifies the instance types that have the specified attributes.
- It uses price protection to determine which instance types to exclude.
- It determines the capacity pools from which it will consider launching the instances based on the AWS Regions or Availability Zones that have matching instance types.
- It applies the specified allocation strategy to determine from which capacity pools to launch the instances.

  Note that attribute-based instance type selection does not pick the capacity pools from which to provision the fleet; that's the job of the allocation strategies.

  If you specify an allocation strategy, the fleet will launch instances according to the specified allocation strategy.

  - For Spot Instances, attribute-based instance type selection supports the **price capacity optimized**, **capacity optimized**, and **lowest price** allocation strategies. Note that we don't recommend the **lowest price** Spot allocation strategy because it has the highest risk of interruption for your Spot Instances.
  - For On-Demand Instances, attribute-based instance type selection supports the **lowest price** allocation strategy.
- If there is no capacity for the instance types with the specified instance attributes, no instances can be launched, and the fleet returns an error.

Price protection

Price protection is a feature that prevents your EC2 Fleet or Spot Fleet from using instance types that you would consider too expensive even if they happen to fit the attributes that you specified. To use price protection, you set a price threshold. Then, when Amazon EC2 selects instance types with your attributes, it excludes instance types priced above your threshold.

The way that Amazon EC2 calculates the price threshold is as follows:

- Amazon EC2 first identifies the lowest priced instance type from those that match your attributes.
- Amazon EC2 then takes the value (expressed as a percentage) that you specified for the price protection parameter and multiplies it with the price of the identified instance type. The result is the price that is used as the price threshold.

There are separate price thresholds for On-Demand Instances and Spot Instances.

When you create a fleet with attribute-based instance type selection, price protection is enabled by default. You can keep the default values, or you can specify your own.

You can also turn off price protection. To indicate no price protection threshold, specify a high percentage value, such as 999999.

Topics

How the lowest priced instance type is identified

Amazon EC2 determines the price to base the price threshold on by identifying the instance type with the lowest price from those that match your specified attributes. It does this in the following way:

- It first looks at the current generation C, M, or R instance types that match your attributes. If it finds any matches, it identifies the lowest priced instance type.
- If there is no match, it then looks at any current generation instance types that match your attributes. If it finds any matches, it identifies the lowest priced instance type.
- If there is no match, it then looks at any previous generation instance types that match your attributes, and identifies the lowest priced instance type.

On-Demand Instance price protection

The price protection threshold for On-Demand instance types is calculated *as a percentage higher* than the identified lowest priced On-Demand instance type

(`OnDemandMaxPricePercentageOverLowestPrice`). You specify the percentage higher that you're willing to pay. If you don't specify this parameter, then a default value of `20` is used to calculate a price protection threshold of 20% higher than the identified price.

For example, if the identified On-Demand instance price is `0.4271`, and you specify 25, then the price threshold is 25% more than `0.4271`. It is calculated as follows: `0.4271 * 1.25 = 0.533875`. The calculated price is the maximum you're willing to pay for On-Demand Instances, and, in this example, Amazon EC2 will exclude any On-Demand instance types that cost more than `0.533875`.

Spot Instance price protection

By default, Amazon EC2 will automatically apply optimal Spot Instance price protection to consistently select from a wide range of instance types. You can also manually set the price protection yourself. However, letting Amazon EC2 do it for you can improve the likelihood that your Spot capacity is fulfilled.

You can manually specify the price protection using one of the following options. If you manually set the price protection, we recommend using the first option.

- **A *percentage of* the identified lowest priced *On-Demand* instance type** [`MaxSpotPriceAsPercentageOfOptimalOnDemandPrice`]

  For example, if the identified On-Demand instance type price is `0.4271`, and you specify `60`, then the price threshold is 60% of `0.4271`. It is calculated as follows: `0.4271 * 0.60 = 0.25626`. The calculated price is the maximum you're willing to pay for Spot Instances, and, in this example, Amazon EC2 will exclude any Spot instance types that cost more than `0.25626`.
- **A *percentage higher than* the identified lowest priced *Spot* instance type** [`SpotMaxPricePercentageOverLowestPrice`]

  For example, if the identified Spot instance type price is `0.1808`, and you specify 25, then the price threshold is 25% more than `0.1808`. It is calculated as follows: `0.1808 * 1.25 = 0.226`. The calculated price is the maximum you're willing to pay for Spot Instances, and, in this example, Amazon EC2 will exclude any Spot instance types that cost more than `0.266`. We do not recommend using this parameter because Spot prices can fluctuate, and therefore your price protection threshold might also fluctuate.

Specify the price protection threshold

To specify the price protection threshold using the AWS CLI

While creating an EC2 Fleet or Spot Fleet using the AWS CLI, configure the fleet for attribute-based instance type selection, and then do the following:

- To specify the On-Demand Instance price protection threshold, in the JSON configuration file, in the `InstanceRequirements` structure, for `OnDemandMaxPricePercentageOverLowestPrice`, enter the price protection threshold as a percentage.
- To specify the Spot Instance price protection threshold, in the JSON configuration file, in the `InstanceRequirements` structure, specify *one* of the following parameters:
  - For `MaxSpotPriceAsPercentageOfOptimalOnDemandPrice`, enter the price protection threshold as a percentage.
  - For `SpotMaxPricePercentageOverLowestPrice`, enter the price protection threshold as a percentage.

For more information, see Create an EC2 Fleet with attribute-based instance type selection or Create a Spot Fleet with attribute-based instance type selection.

(Spot Fleet only) To specify the price protection threshold using the console

While creating a Spot Fleet in the console, configure the fleet for attribute-based instance type selection, and then do the following:

- To specify the On-Demand Instance price protection threshold, under **Additional instance attribute**, choose **On-demand price protection**, choose **Add attribute**, and then enter the price protection threshold as a percentage.
- To specify the Spot Instance price protection threshold, **Additional instance attribute**, choose **Spot price protection**, choose **Add attribute**, choose a base value on which to base your price, and then enter the price protection threshold as a percentage.

> **Note**
>
> When creating the fleet, if you set `TargetCapacityUnitType` to `vcpu` or `memory-mib`, the price protection threshold is applied based on the per-vCPU or per-memory price instead of the per-instance price.

Performance protection

Performance protection is a feature that ensures your EC2 Fleet or Spot Fleet uses instance types that are similar to or exceed a specified performance baseline. To use performance protection, you specify an instance family as a baseline reference. The capabilities of the specified instance family establish the lowest acceptable level of performance. When Amazon EC2 selects instance types for your fleet, it considers your specified attributes and the performance baseline. Instance types that fall below the performance baseline are automatically excluded from selection, even if they match your other specified attributes. This ensures that all selected instance types offer performance similar to or better than the baseline established by the specified instance family. Amazon EC2 uses this baseline to guide instance type selection, but there is no guarantee that the selected instance types will always exceed the baseline for every application.

Currently, this feature only supports CPU performance as a baseline performance factor. The CPU performance of the specified instance family's CPU processor serves as the performance baseline, ensuring that selected instance types are similar to or exceed this baseline. Instance families with the same CPU processors lead to the same filtering results, even if their network or disk performance differs. For example, specifying either `c6in` or `c6i` as the baseline reference would produce identical performance-based filtering results because both instance families use the same CPU processor.

Unsupported instance families

The following instance families are not supported for performance protection:

- `c1`
- `g3` | `g3s`
- `hpc7g`
- `m1` | `m2`
- `mac1` | `mac2` | `mac2-m1ultra` | `mac2-m2` | `mac2-m2pro`

- p3dn | p4d | p5
- t1
- u-12tb1 | u-18tb1 | u-24tb1 | u-3tb1 | u-6tb1 | u-9tb1 | u7i-12tb | u7in-16tb | u7in-24tb | u7in-32tb

If you enable performance protection by specifying a supported instance family, the returned instance types will exclude the above unsupported instance families.

If you specify an unsupported instance family as a value for baseline performance, the API returns an empty response for GetInstanceTypesFromInstanceRequirements and an exception for CreateFleet, RequestSpotFleet, ModifyFleet, and ModifySpotFleetRequest.

Example: Set a CPU performance baseline

In the following example, the instance requirement is to launch with instance types that have CPU cores that are as performant as the `c6i` instance family. This will filter out instance types with less performant CPU processors, even if they meet your other specified instance requirements such as the number of vCPUs. For example, if your specified instance attributes include 4 vCPUs and 16 GB of memory, an instance type with these attributes but with lower CPU performance than `c6i` will be excluded from selection.

```
"BaselinePerformanceFactors": {
        "Cpu": {
            "References": [
                {
                    "InstanceFamily": "c6i"
                }
            ]
        }
```

Considerations

- You can specify either instance types or instance attributes in an EC2 Fleet or Spot Fleet, but not both at the same time.

  When using the CLI, the launch template overrides will override the launch template. For example, if the launch template contains an instance type and the launch template override contains instance attributes, the instances that are identified by the instance attributes will override the instance type in the launch template.

- When using the CLI, when you specify instance attributes as overrides, you can't also specify weights or priorities.
- You can specify a maximum of four `InstanceRequirements` structures in a request configuration.

Create an EC2 Fleet with attribute-based instance type selection

You can configure an EC2 Fleet to use attribute-based instance type selection. It is not possible to create an EC2 Fleet using the Amazon EC2 console.

The attributes for attribute-based instance type selection are specified in the `InstanceRequirements` structure. When `InstanceRequirements` is included in the fleet configuration, `InstanceType` and `WeightedCapacity` must be excluded; they can't determine the fleet configuration at the same time as instance attributes.

In the AWS CLI and PowerShell examples, the following attributes are specified:

- `VCpuCount` – A minimum of 2 vCPUs and a maximum of 4 vCPUs. If you don't need a maximum limit, you can omit the maximum value.
- `MemoryMiB` – A minimum of 8 GiB of memory and a maximum of 16 GiB. If you don't need a maximum limit, you can omit the maximum value.

This configuration identifies any instance types with 2 to 4 vCPUs and 8 to 16 GiB of memory. However, price protection and the allocation strategy might exclude some instance types when [EC2 Fleet provisions the fleet](#).

AWS CLI

To create an EC2 Fleet with attribute-based instance type selection

Use the [create-fleet](#) command to create an EC2 Fleet. Specify the fleet configuration in a JSON file.

```
aws ec2 create-fleet \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

The following example *`file_name`*`.json` file contains the parameters that configure an EC2 Fleet to use attribute-based instance type selection.

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized"
    },
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

PowerShell

To create an EC2 Fleet with attribute-based instance type selection

Use the New-EC2Fleet cmdlet.

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRangeRequest
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiBRequest
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirementsRequest
$instanceRequirements.VCpuCount = $vcpuCount
```

```
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object
 Amazon.EC2.Model.FleetLaunchTemplateSpecificationRequest
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.FleetLaunchTemplateOverridesRequest
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.FleetLaunchTemplateConfigRequest
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

New-EC2Fleet `
    -SpotOptions_AllocationStrategy "price-capacity-optimized" `
    -LaunchTemplateConfig @($launchTemplateConfig) `
    -TargetCapacitySpecification_DefaultTargetCapacityType "spot" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -Type "instant"
```

Create a Spot Fleet with attribute-based instance type selection

You can configure a fleet to use attribute-based instance type selection.

The attributes for attribute-based instance type selection are specified in the `InstanceRequirements` structure. When `InstanceRequirements` is included in the fleet configuration, `InstanceType` and `WeightedCapacity` must be excluded; they can't determine the fleet configuration at the same time as instance attributes.

In the AWS CLI and PowerShell examples, the following attributes are specified:

- `VCpuCount` – A minimum of 2 vCPUs and a maximum of 4 vCPUs. If you don't need a maximum limit, you can omit the maximum value.
- `MemoryMiB` – A minimum of 8 GiB of memory and a maximum of 16 GiB. If you don't need a maximum limit, you can omit the maximum value.

This configuration identifies any instance types that have 2 to 4 vCPUs and 8 to 16 GiB of memory. However, price protection and the allocation strategy might exclude some instance types when Spot Fleet provisions the fleet.

Console

To configure a Spot Fleet for attribute-based instance type selection

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**, and then choose **Request Spot Instances**.
3. Follow the steps to create a Spot Fleet. For more information, see Create a Spot Fleet request using defined parameters.

   While creating the Spot Fleet, configure the fleet for attribute-based instance type selection as follows:

   a. For **Instance type requirements**, choose **Specify instance attributes that match your compute requirements**.
   b. For **vCPUs**, enter the desired minimum and maximum number of vCPUs. To specify no limit, select **No minimum**, **No maximum**, or both.
   c. For **Memory (GiB)**, enter the desired minimum and maximum amount of memory. To specify no limit, select **No minimum**, **No maximum**, or both.
   d. (Optional) For **Additional instance attributes**, you can optionally specify one or more attributes to express your compute requirements in more detail. Each additional attribute adds further constraints to your request.
   e. (Optional) Expand **Preview matching instance types** to view the instance types that have your specified attributes.

AWS CLI

To configure a Spot Fleet for attribute-based instance type selection

Use the request-spot-fleet command to create a Spot Fleet. Specify the fleet configuration in a JSON file.

```
aws ec2 request-spot-fleet \
    --region us-east-1 \
    --spot-fleet-request-config file://file_name.json
```

The following example *file_name*.json file contains the parameters that configure a Spot Fleet to use attribute-based instance type selection.

```
{
    "AllocationStrategy": "priceCapacityOptimized",
    "TargetCapacity": 20,
    "Type": "request",
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }]
}
```

PowerShell

To configure a Spot Fleet for attribute-based instance type selection

Use the Request-EC2SpotFleet cmdlet.

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRange
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiB
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirements
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object Amazon.EC2.Model.FleetLaunchTemplateSpecification
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
```

```
$override = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

Request-EC2SpotFleet `
    -SpotFleetRequestConfig_AllocationStrategy "PriceCapacityOptimized" `
    -SpotFleetRequestConfig_TargetCapacity 20 `
    -SpotFleetRequestConfig_Type "Request" `
    -SpotFleetRequestConfig_LaunchTemplateConfig $launchTemplateConfig
```

Examples of EC2 Fleet configurations that are valid and not valid

If you use the AWS CLI to create an EC2 Fleet, you must make sure that your fleet configuration is valid. The following examples show configurations that are valid and not valid.

Configurations are considered not valid when they contain the following:

- A single `Overrides` structure with both `InstanceRequirements` and `InstanceType`
- Two `Overrides` structures, one with `InstanceRequirements` and the other with `InstanceType`
- Two `InstanceRequirements` structures with overlapping attribute values within the same `LaunchTemplateSpecification`

Example configurations

- Valid configuration: Single launch template with overrides
- Valid configuration: Single launch template with multiple InstanceRequirements
- Valid configuration: Two launch templates, each with overrides
- Valid configuration: Only InstanceRequirements specified, no overlapping attribute values
- Configuration not valid: Overrides contain InstanceRequirements and InstanceType
- Configuration not valid: Two Overrides contain InstanceRequirements and InstanceType
- Configuration not valid: Overlapping attribute values

Valid configuration: Single launch template with overrides

The following configuration is valid. It contains one launch template and one `Overrides` structure containing one `InstanceRequirements` structure. A text explanation of the example configuration follows.

```
{
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5000,
        "DefaultTargetCapacityType": "spot",
        "TargetCapacityUnitType": "vcpu"
        }
    }
}
```

`InstanceRequirements`

To use attribute-based instance selection, you must include the `InstanceRequirements` structure in your fleet configuration, and specify the desired attributes for the instances in the fleet.

In the preceding example, the following instance attributes are specified:

- `VCpuCount` – The instance types must have a minimum of 2 and a maximum of 8 vCPUs.
- `MemoryMiB` – The instance types must have a maximum of 10240 MiB of memory. A minimum of 0 indicates no minimum limit.
- `MemoryGiBPerVCpu` – The instance types must have a maximum of 10,000 GiB of memory per vCPU. The `Min` parameter is optional. By omitting it, you indicate no minimum limit.

`TargetCapacityUnitType`

The `TargetCapacityUnitType` parameter specifies the unit for the target capacity. In the example, the target capacity is `5000` and the target capacity unit type is `vcpu`, which together specify a desired target capacity of 5,000 vCPUs. EC2 Fleet will launch enough instances so that the total number of vCPUs in the fleet is 5,000 vCPUs.

Valid configuration: Single launch template with multiple InstanceRequirements

The following configuration is valid. It contains one launch template and one `Overrides` structure containing two `InstanceRequirements` structures. The attributes specified in `InstanceRequirements` are valid because the values do not overlap—the first `InstanceRequirements` structure specifies a `VCpuCount` of 0-2 vCPUs, while the second `InstanceRequirements` structure specifies 4-8 vCPUs.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
```

```
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
  }
}
```

Valid configuration: Two launch templates, each with overrides

The following configuration is valid. It contains two launch templates, each with one `Overrides` structure containing one `InstanceRequirements` structure. This configuration is useful for `arm` and `x86` architecture support in the same fleet.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
```

```
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

Valid configuration: Only `InstanceRequirements` specified, no overlapping attribute values

The following configuration is valid. It contains two `LaunchTemplateSpecification` structures, each with a launch template and an `Overrides` structure containing an `InstanceRequirements` structure. The attributes specified in `InstanceRequirements` are valid because the values do not overlap—the first `InstanceRequirements` structure specifies a `VCpuCount` of 0-2 vCPUs, while the second `InstanceRequirements` structure specifies 4-8 vCPUs.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
```

```
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

Configuration not valid: `Overrides` contain `InstanceRequirements` and `InstanceType`

The following configuration is not valid. The `Overrides` structure contains both `InstanceRequirements` and `InstanceType`. For the `Overrides`, you can specify either `InstanceRequirements` or `InstanceType`, but not both.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

Configuration not valid: Two Overrides contain InstanceRequirements and InstanceType

The following configuration is not valid. The `Overrides` structures contain both `InstanceRequirements` and `InstanceType`. You can specify either `InstanceRequirements` or `InstanceType`, but not both, even if they're in different `Overrides` structures.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
```

```
    }
}
```

Configuration not valid: Overlapping attribute values

The following configuration is not valid. The two `InstanceRequirements` structures each contain `"VCpuCount": {"Min": 0, "Max": 2}`. The values for these attributes overlap, which will result in duplicate capacity pools.

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
```

```
                "TotalTargetCapacity": 1,
                "DefaultTargetCapacityType": "spot"
            }
        }
}
```

Examples of Spot Fleet configurations that are valid and not valid

If you use the AWS CLI to create a Spot Fleet, you must make sure that your fleet configuration is valid. The following examples show configurations that are valid and not valid.

Configurations are considered not valid when they contain the following:

- A single `Overrides` structure with both `InstanceRequirements` and `InstanceType`
- Two `Overrides` structures, one with `InstanceRequirements` and the other with `InstanceType`
- Two `InstanceRequirements` structures with overlapping attribute values within the same `LaunchTemplateSpecification`

Example configurations

- Valid configuration: Single launch template with overrides
- Valid configuration: Single launch template with multiple InstanceRequirements
- Valid configuration: Two launch templates, each with overrides
- Valid configuration: Only InstanceRequirements specified, no overlapping attribute values
- Configuration not valid: Overrides contain InstanceRequirements and InstanceType
- Configuration not valid: Two Overrides contain InstanceRequirements and InstanceType
- Configuration not valid: Overlapping attribute values

Valid configuration: Single launch template with overrides

The following configuration is valid. It contains one launch template and one `Overrides` structure containing one `InstanceRequirements` structure. A text explanation of the example configuration follows.

```
{
    "SpotFleetRequestConfig": {
```

```
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
        "TargetCapacity": 5000,
            "OnDemandTargetCapacity": 0,
            "TargetCapacityUnitType": "vcpu"
    }
}
```

`InstanceRequirements`

To use attribute-based instance selection, you must include the `InstanceRequirements` structure in your fleet configuration, and specify the desired attributes for the instances in the fleet.

In the preceding example, the following instance attributes are specified:

- `VCpuCount` – The instance types must have a minimum of 2 and a maximum of 8 vCPUs.
- `MemoryMiB` – The instance types must have a maximum of 10240 MiB of memory. A minimum of 0 indicates no minimum limit.
- `MemoryGiBPerVCpu` – The instance types must have a maximum of 10,000 GiB of memory per vCPU. The `Min` parameter is optional. By omitting it, you indicate no minimum limit.

`TargetCapacityUnitType`

The `TargetCapacityUnitType` parameter specifies the unit for the target capacity. In the example, the target capacity is `5000` and the target capacity unit type is `vcpu`, which together specify a desired target capacity of 5,000 vCPUs. Spot Fleet will launch enough instances so that the total number of vCPUs in the fleet is 5,000 vCPUs.

Valid configuration: Single launch template with multiple InstanceRequirements

The following configuration is valid. It contains one launch template and one `Overrides` structure containing two `InstanceRequirements` structures. The attributes specified in `InstanceRequirements` are valid because the values do not overlap—the first `InstanceRequirements` structure specifies a `VCpuCount` of 0-2 vCPUs, while the second `InstanceRequirements` structure specifies 4-8 vCPUs.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
```

```
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
            ]
        }
    ],
    "TargetCapacity": 1,
    "OnDemandTargetCapacity": 0,
    "Type": "maintain"
  }
}
```

Valid configuration: Two launch templates, each with overrides

The following configuration is valid. It contains two launch templates, each with one `Overrides` structure containing one `InstanceRequirements` structure. This configuration is useful for `arm` and `x86` architecture support in the same fleet.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
```

```
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

Valid configuration: Only `InstanceRequirements` specified, no overlapping attribute values

The following configuration is valid. It contains two `LaunchTemplateSpecification` structures, each with a launch template and an `Overrides` structure containing an `InstanceRequirements` structure. The attributes specified in `InstanceRequirements` are valid because the values do not overlap—the first `InstanceRequirements` structure specifies a `VCpuCount` of 0-2 vCPUs, while the second `InstanceRequirements` structure specifies 4-8 vCPUs.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
```

```
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

Configuration not valid: `Overrides` contain `InstanceRequirements` and `InstanceType`

The following configuration is not valid. The `Overrides` structure contains both `InstanceRequirements` and `InstanceType`. For the `Overrides`, you can specify either `InstanceRequirements` or `InstanceType`, but not both.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
```

```
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

Configuration not valid: Two `Overrides` contain `InstanceRequirements` and `InstanceType`

The following configuration is not valid. The `Overrides` structures contain both `InstanceRequirements` and `InstanceType`. You can specify either `InstanceRequirements` or `InstanceType`, but not both, even if they're in different `Overrides` structures.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
```

```
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

Configuration not valid: Overlapping attribute values

The following configuration is not valid. The two `InstanceRequirements` structures each contain `"VCpuCount": {"Min": 0, "Max": 2}`. The values for these attributes overlap, which will result in duplicate capacity pools.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
```

```
                        "MemoryMiB": {
                            "Min": 0
                        }
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
              }
          ]
        }
    ],
    "TargetCapacity": 1,
    "OnDemandTargetCapacity": 0,
    "Type": "maintain"
  }
}
```

Preview instance types with specified attributes

You can use the get-instance-types-from-instance-requirements command to preview the instance types that match the attributes that you specify. This is especially useful for working out what attributes to specify in your request configuration without launching any instances. Note that the command does not consider available capacity.

To preview a list of instance types by specifying attributes using the AWS CLI

1. (Optional) To generate all of the possible attributes that can be specified, use the get-instance-types-from-instance-requirements command and the `--generate-cli-skeleton` parameter. You can optionally direct the output to a file to save it by using `input > attributes.json`.

```
aws ec2 get-instance-types-from-instance-requirements \
    --region us-east-1 \
```

```
    --generate-cli-skeleton input > attributes.json
```

Expected output

```
{
    "DryRun": true,
    "ArchitectureTypes": [
        "i386"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 0,
            "Max": 0
        },
        "MemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "CpuManufacturers": [
            "intel"
        ],
        "MemoryGiBPerVCpu": {
            "Min": 0.0,
            "Max": 0.0
        },
        "ExcludedInstanceTypes": [
            ""
        ],
        "InstanceGenerations": [
            "current"
        ],
        "SpotMaxPricePercentageOverLowestPrice": 0,
        "OnDemandMaxPricePercentageOverLowestPrice": 0,
        "BareMetal": "included",
        "BurstablePerformance": "included",
        "RequireHibernateSupport": true,
        "NetworkInterfaceCount": {
            "Min": 0,
            "Max": 0
        },
```

```
        "LocalStorage": "included",
        "LocalStorageTypes": [
            "hdd"
        ],
        "TotalLocalStorageGB": {
            "Min": 0.0,
            "Max": 0.0
        },
        "BaselineEbsBandwidthMbps": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorTypes": [
            "gpu"
        ],
        "AcceleratorCount": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorManufacturers": [
            "nvidia"
        ],
        "AcceleratorNames": [
            "a100"
        ],
        "AcceleratorTotalMemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "NetworkBandwidthGbps": {
            "Min": 0.0,
            "Max": 0.0
        },
        "AllowedInstanceTypes": [
            ""
        ]
    },
    "MaxResults": 0,
    "NextToken": ""
}
```

2. Create a JSON configuration file using the output from the previous step, and configure it as follows:

Note

You must provide values for `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount`, and `MemoryMiB`. You can omit the other attributes; when omitted, the default values are used.
For a description of each attribute and their default values, see get-instance-types-from-instance-requirements.

a. For `ArchitectureTypes`, specify one or more types of processor architecture.

b. For `VirtualizationTypes`, specify one or more types of virtualization.

c. For `VCpuCount`, specify the minimum and maximum number of vCPUs. To specify no minimum limit, for `Min`, specify `0`. To specify no maximum limit, omit the `Max` parameter.

d. For `MemoryMiB`, specify the minimum and maximum amount of memory in MiB. To specify no minimum limit, for `Min`, specify `0`. To specify no maximum limit, omit the `Max` parameter.

e. You can optionally specify one or more of the other attributes to further constrain the list of instance types that are returned.

3. To preview the instance types that have the attributes that you specified in the JSON file, use the get-instance-types-from-instance-requirements command, and specify the name and path to your JSON file by using the `--cli-input-json` parameter. You can optionally format the output to appear in a table format.

```
aws ec2 get-instance-types-from-instance-requirements \
    --cli-input-json file://attributes.json \
    --output table
```

Example *`attributes.json`* file

In this example, the required attributes are included in the JSON file. They are `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount`, and `MemoryMiB`. In addition, the optional `InstanceGenerations` attribute is also included. Note that for `MemoryMiB`, the `Max` value can be omitted to indicate that there is no limit.

```
{
```

```
    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

Example output

```
------------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+----------------------------------------+
||             InstanceTypes            ||
|+--------------------------------------+|
||             InstanceType             ||
|+--------------------------------------+|
||  c4.xlarge                           ||
||  c5.xlarge                           ||
||  c5a.xlarge                          ||
||  c5ad.xlarge                         ||
||  c5d.xlarge                          ||
||  c5n.xlarge                          ||
||  d2.xlarge                           ||
...
```

4. After identifying instance types that meet your needs, make note of the instance attributes that you used so that you can use them when configuring your fleet request.

Use instance weighting to manage cost and performance of your EC2 Fleet or Spot Fleet

With instance weighting, you assign a weight to each instance type in your EC2 Fleet or Spot Fleet to represent their compute capacity and performance relative to each other. Based on the weights, the fleet can use any combination of the specified instance types, as long as it can fulfil the desired target capacity. This can help you manage the cost and performance of your fleet.

The weight represents the capacity units that an instance type contributes to the total target capacity.

Example: Use instance weighting for performance management

Suppose your fleet has two instance types, and you assign a different weight to each instance type to reflect how many you need of each to achieve the same performance, as follows:

- `m5.large` – weight: 1
- `m5.2xlarge` – weight: 4

By assigning these weights, you're saying that you'd need 4 `m5.large` instances to achieve the same performance as 1 `m5.2xlarge`.

To calculate how many instances of each instance type are needed for a given target capacity, use the following formula:

`target capacity / weight = number of instances`

If your target capacity is 8 units, the fleet could fulfill the target capacity with either `m5.large` or `m5.2xlarge`, or a mix of both, as follows:

- 8 `m5.large` instances (capacity of 8 / weight of 1 = 8 instances)
- 2 `m5.2xlarge` instances (capacity of 8 / weight of 4 = 2 instances)
- 4 `m5.large` and 1 `m5.2xlarge`

Example: Use instance weighting for cost management

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. The 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 a fleet with a target capacity of 10.

Instance type	Instance weight	Target capacity	Number of instances launched	Price per instance hour	Price per unit hour
`r3.xlarge`	2	10	5 (10 divided by 2)	$0.05	$0.025 (.05 divided by 2)
`r3.8xlarge`	8	10	2 (10 divided by 8, result rounded up)	$0.10	$0.0125 (.10 divided by 8)

Use the 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 fleet either in instances (the default) or in the units of your choice, such as vCPU, 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 a 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 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 fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider a 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 fleet would launch four instances (30 divided by 8, result rounded up). 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.

Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity

When you use multiple capacity pools (each comprising an instance type and Availability Zone) in an EC2 Fleet or Spot Fleet, you can use an *allocation strategy* to manage how Amazon EC2 fulfills your Spot and On-Demand capacities from these pools. The allocation strategies can optimize for available capacity, price, and the instance types to use. There are different allocation strategies for Spot Instances and On-Demand Instances.

Topics

- [Allocation strategies for Spot Instances]
- [Allocation strategies for On-Demand Instances]

- Choose the appropriate Spot allocation strategy
- Maintain target capacity for Spot Instances
- Prioritize instance types for On-Demand capacity

Allocation strategies for Spot Instances

Your launch configuration determines all the possible Spot capacity pools (instance types and Availability Zones) from which EC2 Fleet or Spot Fleet can launch Spot Instances. However, when launching instances, the fleet uses the allocation strategy that you specify to pick the specific pools from all your possible pools.

> **Note**
>
> (Linux instances only) If you configure your Spot Instance to launch with AMD SEV-SNP turned on, you are charged an additional hourly usage fee that is equivalent to 10% of the On-Demand hourly rate for the selected instance type. If the allocation strategy uses price as an input, the fleet does not include this additional fee; only the Spot price is used.

You can specify one of the following allocation strategies for Spot Instances:

Price capacity optimized (recommended)

The fleet identifies the pools with the highest capacity availability for the number of instances that are launching. This means that we will request Spot Instances from the pools that we believe have the lowest chance of interruption in the near term. The fleet then requests Spot Instances from the lowest priced of these pools.

The **price capacity optimized** allocation strategy is the best choice for most Spot workloads, such as stateless containerized applications, microservices, web applications, data and analytics jobs, and batch processing.

If you're using the AWS CLI, the parameter name is `price-capacity-optimized` for EC2 Fleet and `priceCapacityOptimized` for Spot Fleet.

Capacity optimized

The fleet identifies the pools with the highest capacity availability for the number of instances that are launching. This means that we will request Spot Instances from the pools that we believe have the lowest chance of interruption in the near term. You can optionally set a priority for each instance type in your fleet, where the 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 that may have a higher cost of interruption associated with restarting work, such as long Continuous Integration (CI), image and media rendering, Deep Learning, and High Performance Compute (HPC) workloads that may have a higher cost of interruption associated with restarting work. 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. The 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 the 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. Note that when you set the priority for instance types for your Spot capacity, the same priority is also applied to your On-Demand Instances if the On-Demand allocation strategy is set to **prioritized**. For Spot Fleet, using priorities is supported only if your fleet uses a launch template.

If you're using the AWS CLI, the parameter names are `capacity-optimized` and `capacity-optimized-prioritized` for EC2 Fleet and `capacityOptimized` and `capacityOptimizedPrioritized` for Spot Fleet.

Diversified

The Spot Instances are distributed across all Spot capacity pools. If you're using the AWS CLI, the parameter name is `diversified` for both EC2 Fleet and Spot Fleet.

Lowest price (not recommended)

> **Warning**
>
> We don't recommend the **lowest price** allocation strategy because it has the highest risk of interruption for your Spot Instances.

The Spot Instances come from the lowest priced pool that has available capacity. When using the AWS CLI, this is the default strategy. However, we recommend that you override the default by specifying the **price capacity optimized** allocation strategy.

With the lowest price strategy, if the lowest priced pool doesn't have available capacity, the Spot Instances come from the next lowest priced pool that has available capacity. If a pool runs out of capacity before fulfilling your desired capacity, the fleet will continue to fulfill your request by drawing from the next lowest priced pool. To ensure that your desired capacity is met, you might receive Spot Instances from several pools.

Because this strategy only considers instance price and not capacity availability, it might lead to high interruption rates.

The lowest price allocation strategy is only available when using the AWS CLI. The parameter name is `lowest-price` for EC2 Fleet and `lowestPrice` for Spot Fleet.

Number of pools to use

The number of Spot pools across which to allocate your target Spot capacity. Valid only when the allocation strategy is set to **lowest price**. The fleet selects the lowest priced Spot pools and evenly allocates your target Spot capacity across the number of Spot pools that you specify.

Note that the 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, the fleet will continue to fulfill your request by drawing from the next lowest priced 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.

This parameter is only available when specifying the **lowest price** allocation strategy and only when using the AWS CLI. The parameter name is `InstancePoolsToUseCount` for both EC2 Fleet and Spot Fleet.

Allocation strategies for On-Demand Instances

Your launch configuration determines all the possible capacity pools (instance types and Availability Zones) from which EC2 Fleet or Spot Fleet can launch On-Demand Instances. However, when launching instances, the fleet uses the allocation strategy that you specify to pick the specific pools from all your possible pools.

You can specify one of the following allocation strategies for On-Demand Instances:

Lowest price

The On-Demand Instances come from the lowest priced pool that has available capacity. This is the default strategy.

If the lowest priced pool doesn't have available capacity, the On-Demand Instances come from the next lowest priced pool that has available capacity.

If a pool runs out of capacity before fulfilling your desired capacity, the fleet will continue to fulfill your request by drawing from the next lowest priced pool. To ensure that your desired capacity is met, you might receive On-Demand Instances from several pools.

Prioritized

The fleet uses the priority that you assigned to each launch template override, launching instance types in order of the highest priority first. This strategy can't be used with attribute-based instance type selection. For an example of how to use this allocation strategy, see Prioritize instance types for On-Demand capacity.

Choose the appropriate Spot allocation strategy

You can optimize your fleet for your use case by choosing the appropriate Spot allocation strategy.

Balance lowest price and capacity availability

To balance the trade-offs between the lowest priced Spot capacity pools and the Spot capacity pools with the highest capacity availability, we recommend that you use the **price capacity optimized** allocation strategy. This strategy makes decisions about which pools to request Spot Instances from based on both the price of the pools and the capacity availability of Spot Instances in those pools. This means that we will request Spot Instances from the pools that we believe have the lowest chance of interruption in the near term, while still taking price into consideration.

If your fleet runs resilient and stateless workloads, including containerized applications, microservices, web applications, data and analytics jobs, and batch processing, then use the **price capacity optimized** allocation strategy for optimal cost savings and capacity availability.

If your fleet runs workloads that might have a higher cost of interruption associated with restarting work, then you should implement checkpointing so that applications can restart from that point if they're interrupted. By using checkpointing, you make the **price capacity optimized** allocation strategy a good fit for these workloads because it allocates capacity from the lowest priced pools that also offer a low Spot Instance interruption rate.

For example JSON configurations that use the **price capacity optimized** allocation strategy, see the following:

- EC2 Fleet – Example 10: Launch Spot Instances in a price-capacity-optimized fleet
- Spot Fleet – Example 11: Launch Spot Instances in a priceCapacityOptimized fleet

When workloads have a high cost of interruption

You can optionally use the **capacity optimized** strategy if you run workloads that either use similarly priced instance types, or where the cost of interruption is so significant that any cost saving is inadequate in comparison to a marginal increase in interruptions. This strategy allocates capacity from the most available Spot capacity pools that offer the possibility of fewer interruptions, which can lower the overall cost of your workload.

When the possibility of interruptions must be minimized but the preference for certain instance types matters, you can 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.

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 allocation strategy is set to **prioritized**. Also note that, for Spot Fleet, using priorities is supported only if your fleet uses a launch template.

For example JSON configurations that use the **capacity optimized** allocation strategy, see the following:

- EC2 Fleet – Example 8: Launch Spot Instances in a capacity-optimized fleet
- Spot Fleet – Example 9: Launch Spot Instances in a capacity-optimized fleet

For example JSON configurations that use the **capacity optimized prioritized** allocation strategy, see the following:

- EC2 Fleet – Example 9: Launch Spot Instances in a capacity-optimized fleet with priorities
- Spot Fleet – Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities

When your workload is time flexible and capacity availability is not a factor

If your fleet is small or runs for a short time, you can use **price capacity optimized** to maximize cost savings while still considering capacity availability.

When your fleet is large or runs for a long time

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 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 fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the On-Demand price.

Maintain target capacity for Spot Instances

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot capacity pool, a fleet of type `maintain` launches replacement Spot Instances. The allocation strategy determines the pools from which the replacement instances are launched, as follows:

- If the allocation strategy is **price capacity optimized**, the fleet launches replacement instances in the pools that have the most Spot Instance capacity availability while also taking price into consideration and identifying lowest priced pools with high capacity availability.
- If the allocation strategy is **capacity optimized**, the fleet launches replacement instances in the pools that have the most Spot Instance capacity availability.
- If the allocation strategy is **diversified**, the fleet distributes the replacement Spot Instances across the remaining pools.

Prioritize instance types for On-Demand capacity

When an EC2 Fleet or Spot Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest priced instance type first. If the On-Demand allocation strategy is set to **prioritized**, the fleet uses priority to determine which instance type to use first when fulfilling On-Demand capacity. The priority is assigned to the launch template override, and the highest priority is launched first.

Example: Prioritize instance types

In this example, you configure three launch template overrides, each with a different instance type.

The On-Demand price for the instance types range in price. The following are the instance types used in this example, listed in order of price, starting with the least expensive instance type:

- `m4.large` – least expensive
- `m5.large`
- `m5a.large`

If you do not use priority to determine the order, the fleet fulfills the On-Demand capacity by starting with the least expensive instance type.

However, say you have unused `m5.large` Reserved Instances that you want to use first. You can set the launch template override priority so that the instance types are used in the order of priority, as follows:

- `m5.large` – priority 1
- `m4.large` – priority 2
- `m5a.large` – priority 3

Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances

With Capacity Rebalancing, your EC2 Fleet or Spot Fleet can maintain the desired Spot capacity by proactively replacing Spot Instances at risk of interruption. When a Spot Instance is at an elevated risk of interruption, Amazon EC2 sends a rebalance recommendation. If Capacity Rebalancing is enabled, the rebalance recommendation triggers the launch of a new Spot Instance before the at-risk instance is interrupted.

Capacity Rebalancing helps you maintain workload availability by proactively augmenting your fleet with new Spot Instances before running instances are interrupted by Amazon EC2.

To configure EC2 Fleet to use Capacity Rebalancing to launch a replacement Spot Instance

Use the create-fleet command and the relevant parameters in the `MaintenanceStrategies` structure. For an example JSON configuration, see Example 7: Configure Capacity Rebalancing to launch replacement Spot Instances.

To configure Spot Fleet to use Capacity Rebalancing to launch a replacement Spot Instance

You can use the Amazon EC2 console or the AWS CLI to configure Capacity Rebalancing.

(Console) When creating the Spot Fleet, select the **Capacity rebalance** checkbox. For more information, see step 6.d. in Create a Spot Fleet request using defined parameters.

(AWS CLI) Use the request-spot-fleet command and the relevant parameters in the `SpotMaintenanceStrategies` structure. For an example JSON configuration, see Example 8: Configure Capacity Rebalancing to launch replacement Spot Instances.

Topics

- Limitations
- Configuration options
- Considerations

Limitations

- Capacity Rebalancing is available only 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.

Configuration options

The `ReplacementStrategy` for EC2 Fleet and Spot Fleet supports the following two values:

`launch-before-terminate`

Amazon EC2 terminates the Spot Instances that receive a rebalance notification after new replacement Spot Instances are launched. When you specify `launch-before-terminate`, you must also specify a value for `termination-delay`. After the new replacement instances are launched, Amazon EC2 waits for the duration of the `termination-delay`, and then terminates the old instances. For `termination-delay`, the minimum is 120 seconds (2 minutes), and the maximum is 7200 seconds (2 hours).

We recommend that you use `launch-before-terminate` only if you can predict how long your instance shutdown procedures will take to complete. This will ensure that the old instances are terminated only after the shutdown procedures are completed. Note that Amazon EC2 can interrupt the old instances with a two-minute warning before the `termination-delay`.

We strongly recommend against using the `lowest-price` (EC2 Fleet) or `lowestPrice` (Spot Fleet) allocation strategy in combination with `launch-before-terminate` to avoid having replacement Spot Instances that are also at an elevated risk of interruption.

`launch`

Amazon EC2 launches replacement Spot Instances when a rebalance notification is emitted for existing Spot Instances. Amazon EC2 does not terminate the instances that receive a rebalance notification. You can terminate the old instances, or you can leave them running. You are charged for all instances while they are running.

Considerations

If you configure an EC2 Fleet or Spot Fleet for Capacity Rebalancing, consider the following:

Provide as many Spot capacity pools in the request as possible

Configure your 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.

Avoid an elevated risk of interruption of replacement Spot Instances

To avoid an elevated risk of interruption, we recommend the `capacity-optimized` or `capacity-optimized-prioritized` allocation strategy. These strategies ensure that replacement Spot Instances are launched in the most optimal Spot capacity pools, and are therefore less likely to be interrupted in the near future. For more information, see Use the price and capacity optimized allocation strategy.

If you use the `lowest-price` allocation strategy, your replacement Spot Instances might be at an elevated risk of interruption. This is because Amazon EC2 will always launch instances in the lowest priced pool that has available capacity at that moment, even if your replacement Spot Instances are likely to be interrupted soon after being launched.

Amazon EC2 will only launch a new instance if availability is the same or better

One of the goals of Capacity Rebalancing is to improve a Spot Instance's availability. If an existing Spot Instance receives a rebalance recommendation, Amazon EC2 will only launch a new instance if the new instance provides the same or better availability than the existing instance. If the risk of interruption of a new instance will be worse than the existing instance, then Amazon EC2 will not launch a new instance. Amazon EC2 will, however, continue to assess the Spot capacity pools, and will launch a new instance if availability improves.

There is a chance that your existing instance will be interrupted without Amazon EC2 proactively launching a new instance. When this happens, Amazon EC2 will attempt to launch a new instance regardless of whether the new instance has a high risk of interruption.

Capacity Rebalancing does not increase your Spot Instance interruption rate

When you enable Capacity Rebalancing, it does not increase your Spot Instance interruption rate (the number of Spot Instances that are reclaimed when Amazon EC2 needs the capacity back). However, if Capacity Rebalancing detects an instance is at risk of interruption, Amazon EC2 will immediately attempt to launch a new instance. The result is that more instances might be replaced than if you'd waited for Amazon EC2 to launch a new instance after the at-risk instance was interrupted.

While you might replace more instances with Capacity Rebalancing enabled, you benefit from being proactive rather than reactive by having more time to take action before your instances are interrupted. With a Spot Instance interruption notice, you typically only have up to two minutes to gracefully shut down your instance. With Capacity Rebalancing launching a new instance in advance, you give existing processes a better chance of completing on your at-risk instance, you can start your instance shutdown procedures, and prevent new work from being scheduled on your at-risk instance. You can also start preparing the newly-launched instance to take over the application. With Capacity Rebalancing's proactive replacement, you benefit from graceful continuity.

As a theoretical example to demonstrate the risks and benefits of using Capacity Rebalancing, consider the following scenario:

- 2:00 PM – A rebalance recommendation is received for instance-A, and Amazon EC2 immediately starts attempting to launch a replacement instance-B, giving you time to start your shutdown procedures.*
- 2:30 PM – A rebalance recommendation is received for instance-B, replaced with instance-C, giving you time to start your shutdown procedures.*
- 2:32 PM – If Capacity Rebalancing wasn't enabled, and if a Spot Instance interruption notice would've been received at 2:32 PM for instance-A, you would only have had up to two minutes to take action, but Instance-A would have been running up till this time.

* If `launch-before-terminate` is specified, Amazon EC2 will terminate the at-risk instance after the replacement instance comes online.

Amazon EC2 can launch new replacement Spot Instances until fulfilled capacity is double target capacity

When a 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. Depending on the replacement strategy, Amazon EC2 either terminates the instance after a preconfigured termination delay, or leaves it running. This gives you the opportunity to perform rebalancing actions on the instance.

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 fleet with a target capacity of 100 Spot Instances. All of the Spot Instances receive a rebalance recommendation, which causes Amazon EC2 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 can't exceed double its target capacity.

Note that you are charged for all of the instances while they are running.

We recommend that you configure your fleet to terminate Spot Instances that receive a rebalance recommendation

If you configure your fleet for Capacity Rebalancing, we recommend that you choose `launch-before-terminate` with an appropriate termination delay only if you can predict how long your instance shutdown procedures will take to complete. This will ensure that the old instances are terminated only after the shutdown procedures are completed.

If you choose to terminate the instances that are recommended for rebalance yourself, 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 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. Amazon EC2 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.

The fleet does not count instances that receive a rebalance recommendation when calculating fulfilled capacity during scale in or out

If your 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, Amazon EC2 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 fleet with a target capacity of 100 Spot Instances. 10 instances receive a rebalance recommendation, so Amazon EC2 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 Amazon EC2. You need to manually terminate these instances, or you can leave them running.
- Scale out – If you increase your desired target capacity, Amazon EC2 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 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.

Use Capacity Reservations to reserve On-Demand capacity in EC2 Fleet

With On-Demand Capacity Reservations, you can reserve compute capacity for your On-Demand Instances in a specified Availability Zone for any duration. You can configure an EC2 Fleet to use the Capacity Reservations first when launching On-Demand Instances.

On-Demand Capacity Reservations are available only for EC2 Fleet with the request type set to `instant`.

Capacity Reservations are configured as either `open` or `targeted`. EC2 Fleet can launch On-Demand Instances into either `open` or `targeted` Capacity Reservations, as follows:

- If a Capacity Reservation is `open`, On-Demand Instances that have matching attributes automatically run in the reserved capacity.
- If a Capacity Reservation is `targeted`, On-Demand Instances must specifically target it to run in the reserved capacity. This is useful for using up specific Capacity Reservations or for controlling when to use specific Capacity Reservations.

If you use `targeted` Capacity Reservations in your EC2 Fleet, there must be enough Capacity Reservations to fulfil the target On-Demand capacity, otherwise the launch fails. To avoid a launch fail, rather add the `targeted` Capacity Reservations to a resource group, and then target the resource group. The resource group doesn't need to have enough Capacity Reservations; if it runs out of Capacity Reservations before the target On-Demand capacity is fulfilled, the fleet can launch the remaining target capacity into regular On-Demand capacity.

To use Capacity Reservations with EC2 Fleet

1. Configure the fleet as type `instant`. You can't use Capacity Reservations for fleets of other types.
2. Configure the usage strategy for Capacity Reservations as `use-capacity-reservations-first`.
3. In the launch template, for **Capacity reservation**, choose either **Open** or **Target by group**. If you choose **Target by group**, specify the Capacity Reservations resource group ID.

When the fleet attempts to fulfil the On-Demand capacity, if it finds that multiple instance pools have unused matching Capacity Reservations, it determines the pools in which to launch the On-Demand Instances based on the On-Demand allocation strategy (`lowest-price` or `prioritized`).

Related resources

- For CLI examples of how to configure a fleet to use Capacity Reservations to fulfil On-Demand capacity, see [Example CLI configurations for EC2 Fleet](), specifically Examples 5 through 7.
- For a tutorial that takes you through the steps for creating Capacity Reservations, using them in your fleet, and viewing how many Capacity Reservations are remaining, see [Tutorial: Configure EC2 Fleet to launch On-Demand Instances using targeted Capacity Reservations]()

- For information about configuring Capacity Reservations, see Reserve compute capacity with EC2 On-Demand Capacity Reservations and the On-Demand Capacity Reservation FAQs.

Work with EC2 Fleet

To start using an EC2 Fleet, create a request that includes the total target capacity, On-Demand capacity, Spot capacity, and a launch template specifying the configuration for the instances in the fleet. You can optionally specify additional parameters, or let the fleet use default values. You can also tag the fleet request, and its instances and volumes, when you create the fleet.

The 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.

Once your fleet is launched, you can describe the fleet request, the instances in the fleet, and any fleet events. You can also assign additional tags as needed.

If you need to change any fleet parameters, such as the total target capacity, you can modify the fleet, provided it was configured to maintain capacity. You can't modify the capacity of a one-time request after it's been submitted.

The fleet request remains active until it expires or you delete it. When you delete the fleet request, you can either terminate the instances or leave them running. If you choose to leave them running, the On-Demand Instances run until you terminate them, and the Spot Instances run until they're interrupted or you terminate them.

Topics

- EC2 Fleet request states
- EC2 Fleet prerequisites
- Create an EC2 Fleet
- Tag a new or existing EC2 Fleet request and the instances and volumes it launches
- Describe an EC2 Fleet, its instances, and its events
- Modify an EC2 Fleet
- Delete an EC2 Fleet request and the instances in the fleet

EC2 Fleet request states

An EC2 Fleet request can be one of various states, with each state indicating a different stage of the request's lifecycle and its management of instances.

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. If a request would exceed your fleet limits, it is deleted immediately.

`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 Spot 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.

`deleted_terminating`

The EC2 Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.

`deleted`

The EC2 Fleet request is deleted and has no running instances. The request is deleted two days after its instances are terminated.

EC2 Fleet prerequisites

To create an EC2 Fleet, the following prerequisites must be in place:

- Launch template
- Service-linked role for EC2 Fleet
- Grant access to customer managed keys for use with encrypted AMIs and EBS snapshots
- Permissions for EC2 Fleet users

Launch template

A launch template specifies the configuration information about the instances to launch, such as the instance type and Availability Zone. For more information about launch templates, see Store instance launch parameters in Amazon EC2 launch templates.

Service-linked role for EC2 Fleet

The `AWSServiceRoleForEC2Fleet` role grants the EC2 Fleet permission to request, launch, terminate, and tag instances on your behalf. Amazon EC2 uses this service-linked role to complete the following actions:

- `ec2:RunInstances` – Launch instances.
- `ec2:RequestSpotInstances` – Request Spot Instances.
- `ec2:TerminateInstances` – Terminate instances.
- `ec2:DescribeImages` – Describe Amazon Machine Images (AMIs) for the instances.
- `ec2:DescribeInstanceStatus` – Describe the status of the instances.
- `ec2:DescribeSubnets` – Describe the subnets for instances.
- `ec2:CreateTags` – Add tags to the EC2 Fleet, instances, and volumes.

Ensure that this role exists before you use the AWS CLI or an API to create an EC2 Fleet.

> **Note**
>
> An `instant` EC2 Fleet does not require this role.

To create the role, use the IAM console as follows.

To create the AWSServiceRoleForEC2Fleet role for EC2 Fleet

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. On the **Select trusted entity** page, do the following:

    a. For **Trusted entity type**, choose **AWS service**.

    b. Under **Use case**, for **Service or use case**, choose **EC2 - Fleet**.

    > **Tip**
    >
    > Be sure to choose **EC2 - Fleet**. If you choose **EC2**, the **EC2 - Fleet** use case does not appear in the **Use case** list. The **EC2 - Fleet** use case will automatically create a policy with the required IAM permissions and will suggest **AWSServiceRoleForEC2Fleet** as the role name.

    c. Choose **Next**.
5. On the **Add permissions page**, choose **Next**.
6. On the **Name, review, and create** 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 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 or an encrypted Amazon EBS snapshot 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.

```
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/AWSServiceRoleForEC2Fleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

Permissions for EC2 Fleet users

If your users will create or manage an EC2 Fleet, be sure to grant them the required permissions.

To create a policy 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":"arn:aws:iam::123456789012:role/DevTeam*"
        }
    ]
}
```

The `ec2:*` grants a user permission to call all Amazon EC2 API actions. To limit the user to specific Amazon EC2 API actions, specify those actions instead.

The 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 a 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. To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

Create an EC2 Fleet

To create an EC2 Fleet, define the fleet configuration in a JSON file and reference the file with the create-fleet command. In the JSON file, you must specify the total target capacity for the fleet, separate target capacities for Spot Instances and On-Demand Instances, and a launch template that defines the configuration for the instances in the fleet, such as an AMI, instance type, subnet or Availability Zone, and one or more security groups. You can optionally specify additional configurations, such as parameters to override the launch template configuration, allocation strategies for selecting Spot Instances and On-Demand Instances from the EC2 capacity pools, and the maximum amount you're willing to pay for the fleet. For more information, see Configuration options for your EC2 Fleet or Spot Fleet.

The EC2 Fleet launches On-Demand Instances when capacity is available, and launches Spot Instances when your maximum price exceeds the Spot price and capacity is available.

If your fleet includes Spot Instances and is of type `maintain`, Amazon EC2 will attempt to maintain your fleet target capacity when your Spot Instances are interrupted.

EC2 Fleet limitations

The following limitations apply to EC2 Fleet:

- Creating an EC2 Fleet is available only through the Amazon EC2 API, AWS CLI, AWS SDKs, and AWS CloudFormation.
- 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.

Create an EC2 Fleet

To launch a fleet of instances using EC2 Fleet, you need only specify the following parameters in your fleet request, and the fleet will use the default values for the other parameters:

- `LaunchTemplateId` or `LaunchTemplateName` – Specifies the launch template to use (which contains the parameters for the instances to launch, such as the instance type and Availability Zone)
- `TotalTargetCapacity` – Specifies the total target capacity for the fleet
- `DefaultTargetCapacityType` – Specifies whether the default purchasing option is On-Demand or Spot

To override the parameters specified in the launch template, you can specify one or more overrides. Each override can vary by instance type, Availability Zone, subnet, and maximum price, and can include a different weighted capacity. As an alternative to specifying an instance type, you can specify the attributes that an instance must have, and Amazon EC2 will identify all the instance types with those attributes. For more information, see Specify attributes for instance type selection for EC2 Fleet or Spot Fleet.

For EC2 Fleets of type `instant`, you can specify a Systems Manager parameter instead of the AMI ID. You can specify the Systems Manager parameter in the override or in the launch template. For more information, see Use a Systems Manager parameter instead of an AMI ID.

You can specify the fleet parameters in a JSON file. For information about all the possible parameters you can specify, see View all the EC2 Fleet configuration options.

For fleet configuration examples, see Example CLI configurations for EC2 Fleet.

There is currently no console support for creating an EC2 Fleet.

To create an EC2 Fleet

Use the create-fleet command to create the fleet and specify the JSON file that contains the fleet configuration parameters.

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

The following is example output for a fleet of type `request` or `maintain`.

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

The following is example output for a fleet of type `instant` that launched the target capacity.

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ],
      "InstanceType": "c5.large",
      "Platform": null
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
```

```
          "Overrides": {
            "InstanceType": "c4.large",
            "AvailabilityZone": "us-east-1a"
          }
        },
        "Lifecycle": "on-demand",
        "InstanceIds": [
          "i-5678901234abcdef0",
          "i-5432109876abcdef9"
        ]
    ]
}
```

The following is example output for a fleet of type `instant` that launched part of the target capacity with errors for instances that were not launched.

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientInstanceCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
```

```
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ]
  ]
}
```

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": ""
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a",
        }
      },
```

```
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": []
}
```

Create an EC2 Fleet that replaces unhealthy Spot Instances

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 Amazon EC2 instances.

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.
- 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. Use the information for creating an EC2 Fleet in Create an EC2 Fleet.
2. To configure the fleet to replace unhealthy Spot Instances, in the JSON file, for `ReplaceUnhealthyInstances`, specify `true`.

View all the EC2 Fleet configuration options

To view the full list of EC2 Fleet configuration parameters, you can generate a JSON file. For a description of each parameter, see create-fleet.

To generate a JSON file with all possible EC2 Fleet parameters

Use the create-fleet (AWS CLI) command and the `--generate-cli-skeleton` parameter to generate an EC2 Fleet JSON file, and direct the output to a file to save it.

```
aws ec2 create-fleet \
    --generate-cli-skeleton input > ec2createfleet.json
```

The following is example output.


```
{
    "DryRun": true,
    "ClientToken": "",
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch"
            }
        },
        "InstanceInterruptionBehavior": "hibernate",
        "InstancePoolsToUseCount": 0,
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        },
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "ExcessCapacityTerminationPolicy": "termination",
```

```
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "",
                "LaunchTemplateName": "",
                "Version": ""
            },
            "Overrides": [
                {
                    "InstanceType": "r5.metal",
                    "MaxPrice": "",
                    "SubnetId": "",
                    "AvailabilityZone": "",
                    "WeightedCapacity": 0.0,
                    "Priority": 0.0,
                    "Placement": {
                        "AvailabilityZone": "",
                        "Affinity": "",
                        "GroupName": "",
                        "PartitionNumber": 0,
                        "HostId": "",
                        "Tenancy": "dedicated",
                        "SpreadDomain": "",
                        "HostResourceGroupArn": ""
                    },
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 0
                        },
                        "MemoryMiB": {
                            "Min": 0,
                            "Max": 0
                        },
                        "CpuManufacturers": [
                            "amd"
                        ],
                        "MemoryGiBPerVCpu": {
                            "Min": 0.0,
                            "Max": 0.0
                        },
                        "ExcludedInstanceTypes": [
                            ""
                        ],
```

```
                        "InstanceGenerations": [
                            "previous"
                        ],
                        "SpotMaxPricePercentageOverLowestPrice": 0,
                        "OnDemandMaxPricePercentageOverLowestPrice": 0,
                        "BareMetal": "included",
                        "BurstablePerformance": "required",
                        "RequireHibernateSupport": true,
                        "NetworkInterfaceCount": {
                            "Min": 0,
                            "Max": 0
                        },
                        "LocalStorage": "excluded",
                        "LocalStorageTypes": [
                            "ssd"
                        ],
                        "TotalLocalStorageGB": {
                            "Min": 0.0,
                            "Max": 0.0
                        },
                        "BaselineEbsBandwidthMbps": {
                            "Min": 0,
                            "Max": 0
                        },
                        "AcceleratorTypes": [
                            "inference"
                        ],
                        "AcceleratorCount": {
                            "Min": 0,
                            "Max": 0
                        },
                        "AcceleratorManufacturers": [
                            "amd"
                        ],
                        "AcceleratorNames": [
                            "a100"
                        ],
                        "AcceleratorTotalMemoryMiB": {
                            "Min": 0,
                            "Max": 0
                        }
                    }
                }
            ]
```

```
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 0,
        "OnDemandTargetCapacity": 0,
        "SpotTargetCapacity": 0,
        "DefaultTargetCapacityType": "on-demand",
        "TargetCapacityUnitType": "memory-mib"
    },
    "TerminateInstancesWithExpiration": true,
    "Type": "instant",
    "ValidFrom": "1970-01-01T00:00:00",
    "ValidUntil": "1970-01-01T00:00:00",
    "ReplaceUnhealthyInstances": true,
    "TagSpecifications": [
        {
            "ResourceType": "fleet",
            "Tags": [
                {
                    "Key": "",
                    "Value": ""
                }
            ]
        }
    ],
    "Context": ""
}
```

Tag a new or existing EC2 Fleet request and the instances and volumes it launches

To help categorize and manage your EC2 Fleet requests and the instances and volumes that it launches, you can tag them with custom metadata. You can assign a tag to an EC2 Fleet request when you create it, or afterward. Similarly, you can assign a tag to the instances and volumes when they're launched by the fleet, 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 of them.

> **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.

Prerequisite

Grant the user the permission to tag resources. For more information, see Example: Tag resources.

To grant a user the permission to tag resources

Create a IAM policy that includes the following:

- The `ec2:CreateTags` action. This grants the user permission to create tags.
- The `ec2:CreateFleet` action. This grants the user permission to create an EC2 Fleet request.
- For `Resource`, we recommend that you specify `"*"`. This allows users to tag all resource types.

> **Important**
>
> We currently do not support resource-level permissions for the `create-fleet` resource. If you specify `create-fleet` as a resource, you will get an unauthorized exception when you try to tag the fleet. The following example illustrates how *not* to set the policy.
>
> ```
> {
>     "Effect": "Allow",
>     "Action": [
>         "ec2:CreateTags",
>         "ec2:CreateFleet"
>     ],
>     "Resource": "arn:aws:ec2:us-east-1:111122223333:create-fleet/*"
> }
> ```

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

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 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 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

Use the create-tags command to tag existing resources.

```
aws ec2 create-tags \
    --resources fleet-12a34b55-67cd-8ef9-
ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE \
    --tags Key=purpose,Value=test
```

Describe an EC2 Fleet, its instances, and its events

You can describe your EC2 Fleet configuration, the instances in your EC2 Fleet, and the event history of your EC2 Fleet.

Tasks

- Describe your EC2 Fleet
- Describe all instances in your EC2 Fleet
- Describe the event history for your EC2 Fleet

Describe your EC2 Fleet

AWS CLI

To describe your EC2 Fleet

Use the describe-fleets command.

```
aws ec2 describe-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

The following is example output.

```
{
    "Fleets": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2022-02-09T03:35:52+00:00",
            "FleetId": "fleet-364457cd-3a7a-4ed9-83d0-7b63e51bb1b7",
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "FulfilledCapacity": 2.0,
            "FulfilledOnDemandCapacity": 0.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "$Latest"
                    }
                }
```

```
            ],
            "TargetCapacitySpecification": {
                "TotalTargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "TerminateInstancesWithExpiration": false,
            "Type": "maintain",
            "ReplaceUnhealthyInstances": false,
            "SpotOptions": {
                "AllocationStrategy": "capacity-optimized",
                "InstanceInterruptionBehavior": "terminate"
            },
            "OnDemandOptions": {
                "AllocationStrategy": "lowestPrice"
            }
        }
    ]
}
```

PowerShell

To describe your EC2 Fleet

Use the Get-EC2FleetList cmdlet.

```
Get-EC2FleetList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Describe all instances in your EC2 Fleet

The returned list of running instances is refreshed periodically and might be out of date.

AWS CLI

To describe the instances for the specified EC2 Fleet

Use the describe-fleet-instances command.

```
aws ec2 describe-fleet-instances \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

The following is example output.

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-09cd595998cb3765e",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-86k84j6p"
        },
        {
            "InstanceId": "i-09cf95167ca219f17",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-dvxi7fsm"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

PowerShell

To describe the instances for the specified EC2 Fleet

Use the Get-EC2FleetInstanceList cmdlet.

```
Get-EC2FleetInstanceList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Describe the event history for your EC2 Fleet

For more information about the events in the event history, see EC2 Fleet event types.

AWS CLI

To describe the events for the specified EC2 Fleet

Use the describe-fleet-history command.

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2020-06-01T00:00:00Z
```

The following is example output.


```
{
    "HistoryRecords": [
        {
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:05.000Z"
        },
        {
            "EventInformation": {
                "EventSubType": "active"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:15.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
                "EventSubType": "progress"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-083a1c446e66085d2"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-090db02406cc3c2d6"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        }
```

```
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2020-06-01T00:00:00.000Z"
}
```

PowerShell

To describe the events for the specified EC2 Fleet

Use the Get-EC2FleetHistory cmdlet.

```
Get-EC2FleetHistory `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE `
    -UtcStartTime 2020-06-01T00:00:00Z
```

Modify an EC2 Fleet

You can modify the total target capacity, Spot capacity, and On-Demand capacity of an EC2 Fleet. You can also modify whether running instances should be terminated if the new total target capacity is reduced below the current size of the fleet.

Considerations

Consider the following when modifying an EC2 Fleet:

- **Fleet type** – You can only modify an EC2 Fleet of type `maintain`. You can't modify an EC2 Fleet of type `request` or `instant`.
- **Fleet parameters** – You can modify the following parameters of an EC2 Fleet:
  - `target-capacity-specification` – Increase or decrease the target capacity for:
    - `TotalTargetCapacity`
    - `OnDemandTargetCapacity`
    - `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`
    - `termination`

- **Fleet behavior when increasing total target capacity** – When you increase the total target capacity, the EC2 Fleet launches the additional instances according to the instance purchasing option specified for `DefaultTargetCapacityType`, which is either On-Demand Instances or Spot Instances, and according to the specified allocation strategy.
- **Fleet behavior when decreasing Spot target capacity** – When you decrease the Spot target capacity, the EC2 Fleet deletes any open requests that exceed the new target capacity. You can request that the fleet terminate Spot Instances until the size of the fleet reaches the new target capacity. When an EC2 Fleet terminates a Spot Instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

  Instances are selected for termination based on the allocation strategy:

  - `capacity-optimized` – Terminates instances from pools with the least available capacity.
  - `price-capacity-optimized` – Uses a combination of price and available capacity: terminates instances from pools with the least available capacity and which are the highest-priced among these pools.
  - `diversified` – Terminates instances across all pools.
  - `lowest-price` – Terminates instances from highest-priced 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 that you terminate manually.
- **Fleet state** – You can modify an EC2 Fleet that is in the `submitted` or `active` state. When you modify a fleet, it enters the `modifying` state.

Commands for modifying an EC2 Fleet

AWS CLI

To modify the total target capacity of an EC2 Fleet

Use the modify-fleet command.

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=20
```

If you are decreasing the target capacity but want to keep the fleet at its current size, you can modify the previous example as follows.

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=10 \
    --excess-capacity-termination-policy no-termination
```

PowerShell

To modify the total target capacity of an EC2 Fleet

Use the Edit-EC2Fleet cmdlet.

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20
```

If you are decreasing the target capacity but want to keep the fleet at its current size, you can modify the previous example as follows.

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

Delete an EC2 Fleet request and the instances in the fleet

If you no longer require an EC2 Fleet request, you can delete it. After you delete a fleet request, all Spot requests associated with the fleet are canceled, so that no new Spot Instances are launched.

When you delete an EC2 Fleet request, you must also specify if you want to terminate all of its instances. These include both On-Demand Instances and Spot Instances. For `instant` fleets, EC2 Fleet must terminate the instances when the fleet is deleted. A deleted `instant` fleet with running instances is not supported.

If you specify that the instances must be terminated when the fleet request is deleted, the fleet request 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 fleets of type `instant` in a single operation.

- You can delete up to 100 fleets of type `maintain` or `request` in a single operation.
- You can delete up to 125 fleets in a single operation, provided you do not exceed the quota for each fleet type, as specified above.
- If you exceed the specified number of fleets to delete, no fleets are deleted.
- A deleted `instant` fleet with running instances is not supported. When you delete an `instant` fleet, Amazon EC2 automatically terminates all its instances. For `instant` fleets with more than 1000 instances, the deletion request might fail. If your fleet has more than 1000 instances, first terminate most of the instances manually, leaving 1000 or fewer. Then delete the fleet, and the remaining instances will be terminated automatically.

AWS CLI

To delete an EC2 Fleet request and terminate its instances

Use the delete-fleets command with the `--terminate-instances` option.

```
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 request without terminating its instances

Modify the previous example by using the `--no-terminate-instances` option instead. Note that `--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"
        }
    ]
}
```

PowerShell

To delete an EC2 Fleet request and terminate its instances

Use the Remove-EC2Fleet cmdlet with the `-TerminateInstance` parameter.

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

To delete an EC2 Fleet request without terminating its instances

Modify the previous example by changing the value of the `-TerminateInstance` parameter.

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

Troubleshoot when a fleet fails to delete

If an EC2 Fleet request fails to delete, `UnsuccessfulFleetDeletions` in the output returns the ID of the EC2 Fleet request, an error code, and an error message.

The error codes are:

- `ExceededInstantFleetNumForDeletion`

- `fleetIdDoesNotExist`
- `fleetIdMalformed`
- `fleetNotInDeletableState`
- `NoTerminateInstancesNotSupported`
- `UnauthorizedOperation`
- `unexpectedError`

Troubleshoot `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: VvuncIxj7Z_CPGNYXWqnuFV-
YjByeAU66Q9752NtQ-I3-qnDLWs6JLFd
KnSMMiq5s6cGqjjPtEDpsnGHzzyHasFHOaRYJpaDVravoW25azn6KNkUQQlFwhJyujt2dtNCdduJfrqcFYAjlEiRMkfDHt7
BHturzDK6A560Y2nDSUiMmAB1y9UNtqaZJ9SNe5sNxKMqZaqKtjRbk02RZu5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtJ76
VPiU5v2s-
UgZ7h0p2yth6ysUdhlONg6dBYu8_y_HtEI54invCj4CoK0qawqzMNe6rcmCQHvtCxtXsbkgyaEbcwmrm2m01-
EMhekLFZeJLr
DtYOpYcEl4_nWFX1wtQDCnNNCmxnJZAoJvb3VMDYpDTsxjQv1PxODZuqWHs23YXWVywzgnLtHeRf2o4lUhGBw17mXsS07k7
PT9vrHtQiILor5VVTsjSPWg7edj__1rsnXhwPSu8gI48ZLRGrPQqFq0RmKO_QIE8N8s6NWzCK4yoX-9gDcheurOGpkprPIC
</Message></Error></Errors><RequestID>89b1215c-7814-40ae-a8db-41761f43f2b0</
RequestID></Response>
```

To resolve the error, you must add the `ec2:TerminateInstances` action to the IAM policy, as shown in the following example.

Work with Spot Fleet

To start using a Spot Fleet, create a request that includes the total target capacity for Spot Instances, an optional On-Demand portion, and either manually specify an AMI and a key pair, or specify a launch template that includes the configuration for the instances in the fleet. You can optionally specify additional parameters, or let the fleet use default values. You can also tag the fleet request, and its instances and volumes, when you create the fleet.

The 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.

Once your fleet is launched, you can describe the fleet request, the instances in the fleet, and any fleet events. You can also assign additional tags as needed.

If you need to change any fleet parameters, such as the total target capacity, you can modify the fleet, provided it was configured to maintain capacity. You can't modify the capacity of a one-time request after it's been submitted.

The fleet request remains active until it expires or you cancel (delete) it. When you cancel the fleet request, you can either terminate the instances or leave them running. If you choose to leave them running, the On-Demand Instances run until you terminate them, and the Spot Instances run until they're interrupted or you terminate them.

Topics

Spot Fleet request states

A Spot Fleet request can be one of various states, with each state indicating a different stage of the request's lifecycle and its management of instances.

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. If a request would exceed your Spot Fleet quotas, it is canceled immediately.

`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 request is canceled. Only a `maintain` fleet type can be modified. This state does not apply to a one-time `request` fleet type.

`cancelled_running`

The Spot Fleet is canceled (deleted) and does not launch additional Spot 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.

`cancelled_terminating`

The Spot Fleet is canceled (deleted) and its instances are terminating. The request remains in this state until all instances are terminated.

`cancelled`

The Spot Fleet is canceled (deleted) and has no running instances. The request is deleted two days after its instances are terminated.

Spot Fleet permissions

If your 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 users for Spot Fleet
- Service-linked role for Spot Fleet
- Service-linked role for Spot Instances
- IAM role for tagging a Spot Fleet

Grant permission to users for Spot Fleet

If your users will create or manage a Spot Fleet, be sure to grant them the required permissions.

To create a policy 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"
            ],
            "Resource": "*"
        }
    ]
}
```

The preceding example policy grants a 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 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 a 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. To provide access, add permissions to your users, groups, or roles:

   - Users and groups in AWS IAM Identity Center:

     Create a permission set. Follow the instructions in [Create a permission set]() in the *AWS IAM Identity Center User Guide*.

   - Users managed in IAM through an identity provider:

Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.

- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

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 Service-linked roles in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2SpotFleet** to launch and manage instances on your behalf.

> **Important**
>
> If you specify an encrypted AMI or an encrypted Amazon EBS snapshot 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.

Permissions granted by AWSServiceRoleForEC2SpotFleet

The **AWSServiceRoleForEC2SpotFleet** role grants the Spot Fleet permission to request, launch, terminate, and tag instances on your behalf. Amazon EC2 uses this service-linked role to complete the following actions:

- `ec2:RequestSpotInstances` - Request Spot Instances
- `ec2: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*.

If you use the AWS CLI or an API to create a Spot Fleet, you must first ensure that this role exists.

To create the AWSServiceRoleForEC2SpotFleet role for Spot Fleet 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 trusted entity** page, do the following:

   a. For **Trusted entity type**, choose **AWS service**.

   b. Under **Use case**, for **Service or use case**, choose **EC2**.

   c. For **Use case**, choose **EC2 - Spot Fleet**.

> **Note**
>
> The **EC2 - Spot Fleet** use case will automatically create a policy with the required IAM permissions and will suggest **AWSEC2SpotFleetServiceRolePolicy** as the role name.

   d. Choose **Next**.

5. On the **Add permissions** page, choose **Next**.
6. On the **Name, review, and create** page, choose **Create role**.

To create the AWSServiceRoleForEC2SpotFleet role for Spot Fleet 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 or an encrypted Amazon EBS snapshot in your Spot Fleet request and you use a customer managed key for encryption, you must grant the **AWSServiceRoleForEC2SpotFleet** role permission to use the CMK so that Amazon EC2 can launch instances on your behalf. To do this, you must add a grant to the CMK, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see Using Grants and Using Key Policies in AWS KMS in the *AWS Key Management Service Developer Guide*.

To grant the AWSServiceRoleForEC2SpotFleet role permissions to use the CMK

- Use the create-grant command to add a grant to the CMK and to specify the principal (the **AWSServiceRoleForEC2SpotFleet** service-linked role) that is given permission to perform the operations that the grant permits. The CMK is specified by the `key-id` parameter and the ARN of the CMK. The principal is specified by the `grantee-principal` parameter and the ARN of the **AWSServiceRoleForEC2SpotFleet** service-linked role.

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
spotfleet.amazonaws.com/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.

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 new or existing Spot Fleet request and the instances and volumes it launches.

Important

If you choose to tag instances in the fleet and you also choose to maintain target capacity (the Spot Fleet request is of type `maintain`), the differences in the permissions that are set for the 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 role**.
4. On the **Select trusted entity** page, under **Trusted entity type**, choose **AWS service**.
5. Under **Use case**, from **Use cases for other AWS services**, choose **EC2**, and then choose **EC2 - Spot Fleet Tagging**.
6. Choose **Next**.
7. On the **Add permissions** page, choose **Next**.
8. On the **Name, review, and create** page, for **Role name**, enter a name for the role (for example, **`aws-ec2-spot-fleet-tagging-role`**).
9. Review the information on the page, and then choose **Create role**.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. We recommend that you use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in the `aws-ec2-spot-fleet-tagging-role` trust policy to limit the permissions that Spot Fleet gives another service to the resource.

To add the aws:SourceArn and aws:SourceAccount condition keys to the `aws-ec2-spot-fleet-tagging-role` trust policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. Find the `aws-ec2-spot-fleet-tagging-role` that you created previously and choose the link (not the checkbox).
4. Under **Summary**, choose the **Trust relationships** tab, and then choose **Edit trust policy**.
5. In the JSON statement, add a `Condition` element containing your `aws:SourceAccount` and `aws:SourceArn` global condition context keys to prevent the confused deputy problem, as follows:

   ```
   "Condition": {
   ```

```
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/
sfr-*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
```

> **Note**
>
> If the aws:SourceArn value contains the account ID and you use both global condition context keys, the aws:SourceAccount value and the account in the aws:SourceArn value must use the same account ID when used in the same policy statement.

The final trust policy will be as follows:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "spotfleet.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-
request/sfr-*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
}
```

6. Choose **Update policy**.

The following table provides potential values for `aws:SourceArn` to limit the scope of the your `aws-ec2-spot-fleet-tagging-role` in varying degrees of specificity.

API operation	Called service	Scope	`aws:SourceArn`
RequestSpotFleet	AWS STS (`AssumeRole`)	Limit the `AssumeRole` capability on `aws-ec2-spot-fleet-tagging-role` to spot-fleet-requests in the specified account.	`arn:aws:ec2:*:` *`123456789012`* `:spot-fleet-request/sfr-*`
RequestSpotFleet	AWS STS (`AssumeRole`)	Limit the `AssumeRole` capability on `aws-ec2-spot-fleet-tagging-role` to spot-fleet-requests in the specified account and specified Region. Note that this role will not be usable in other Regions.	`arn:aws:ec2:` *`us-east-1`* `:`*`123456789012`* `:spot-fleet-request/sfr-*`
RequestSpotFleet	AWS STS (`AssumeRole`)	Limit the `AssumeRole` capability on `aws-ec2-spot-fleet-tagging-role` to only actions affecting the fleet sfr-11111111-1111-1111-1111-111111111111.	`arn:aws:ec2:` *`us-east-1`* `:`*`123456789012`* `:spot-fleet-request/sfr-`*`11111111-1111-1111-1111-111111111111`*

API operation	Called service	Scope	aws:SourceArn
		Note that this role may not be usable for other Spot Fleets. A lso, this role cannot be used to launch any new Spot Fleets th rough request-spot-fleet.	

Create a Spot Fleet

Using the AWS Management Console, you can quickly create a Spot Fleet request by choosing only an AMI and your desired total target capacity. Amazon EC2 will configure a fleet that best meets your needs and follows Spot best practices. Alternatively, you can modify any of the default settings.

If you want to include On-Demand Instances in your fleet, you must specify a launch template in your request and specify you desired On-Demand capacity.

The fleet launches On-Demand Instances when capacity is available, and launches Spot Instances when your maximum price exceeds the Spot price and capacity is available.

If your fleet includes Spot Instances and is of type `maintain`, Amazon EC2 will attempt to maintain your fleet target capacity when your Spot Instances are interrupted.

Required permissions

For more information, see the section called “Spot Fleet permissions”.

Tasks

- Quickly create a Spot Fleet request
- Create a Spot Fleet request using defined parameters
- Create a Spot Fleet that replaces unhealthy Spot Instances

Quickly create a Spot Fleet request

Follow these steps to quickly create a Spot Fleet request using the Amazon EC2 console.

To create a Spot Fleet request using the recommended settings

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Create Spot Fleet Request**.
4. Under **Launch parameters**, choose **Manually configure launch parameters**.
5. For **AMI**, choose an AMI.
6. Under **Target capacity**, for **Total target capacity**, specify the number of units to request. For the type of unit, you can choose **Instances**, **vCPUs**, or **Memory (GiB)**.
7. Under **Your fleet request at a glance**, review your fleet configuration, and choose **Launch**.

Create a Spot Fleet request using defined parameters

You can create a Spot Fleet by using parameters that you define.

Console

To create a Spot Fleet request using defined parameters

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Create Spot Fleet Request**.
4. For **Launch parameters**, you can either manually configure the launch parameters or you can use a launch template, as follows:
   a. [Manually configure] To define the launch parameters in the Amazon EC2 console, choose **Manually configure launch parameters**, and then do the following:
      i. 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.

> **Note**
>
> If an AMI specified in the launch parameters is deregistered or disabled, no new instances can be launched from the AMI. For fleets that are set to maintain target capacity, the target capacity will not be maintained.

ii. (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 **Key pairs** page. When you are done, return to the **Spot Requests** page and refresh the list.

iii. (Optional) Expand **Additional launch parameters**, and do the following:

    A. (Optional) To enable Amazon EBS optimization, for **EBS-optimized**, select **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, choose **Add new volume**, and 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**, select **Enable CloudWatch detailed monitoring**.

    E. (Optional) To run a Dedicated Spot Instance, for **Tenancy**, choose **Dedicated - run a dedicated instance**.

    F. (Optional) For **Security groups**, choose one or more security groups or create a new one.

       [Existing security group] Choose one or more security groups.

       [New security group] Choose **Create new security group** to go the **Security Groups** page. When you are done, return to the **Spot Requests** and refresh the list.

    G. (Optional) To make your instances reachable from the internet, for **Auto-assign IPv4 Public IP**, choose **Enable**.

H. (Optional) To launch your Spot Instances with an IAM role, for **IAM instance profile**, choose the role.

I. (Optional) To run a start-up script, copy it to **User data**.

J. (Optional) To add a tag, choose **Create tag** and enter the key and value for the tag, and choose **Create**. Repeat for each tag.

   For each tag, to tag the instances and the Spot Fleet request with the same tag, ensure that both **Instances** and **Fleet** are selected. To tag only the instances launched by the fleet, clear **Fleet**. To tag only the Spot Fleet request, clear **Instances**.

b. [Launch template] To use a configuration you created in a launch template, choose **Use a launch template**, and for **Launch template**, choose a launch template.

> **Note**
>
> If you want On-Demand capacity in your Spot Fleet, you must specify a launch template.

5. 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) To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid until**.

   d. (Optional) By default, Amazon EC2 terminates your Spot Instances when the Spot Fleet request expires. To keep them running after your request expires, clear **Terminate the instances when the request expires**.

   e. (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.

6. For **Target capacity**, do the following:

   a. For **Total target capacity**, specify the number of units to request. For the type of unit, you can choose **Instances**, **vCPUs**, or **Memory (MiB)**. To specify a target capacity of 0 so that you can add capacity later, you must first select **Maintain target capacity**.

b. (Optional) For **Include On-Demand base capacity**, 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 optional On-Demand capacity, you must first choose a launch template.

c. (Optional) By default, Amazon EC2 terminates Spot Instances when they are interrupted. To maintain the target capacity, select **Maintain target capacity**. You can then specify that Amazon EC2 terminates, stops, or hibernates Spot Instances when they are interrupted. To do so, choose the corresponding option from **Interruption behavior**.

> **Note**
>
> If an AMI specified in the launch parameters is deregistered or disabled, no new instances can be launched from the AMI. In this case, for fleets that are set to maintain target capacity, the target capacity will not be maintained.

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**, and then choose an instance replacement strategy. If you choose **Launch before terminate**, specify the delay (in seconds) before Amazon EC2 terminates the old instances. For more information, see Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances.

e. (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 Set a spending limit for your EC2 Fleet or Spot Fleet.

7. For **Network**, do the following:

a. For **Network**, choose an existing VPC or create a new one.

[Existing VPC] Choose the VPC.

[New VPC] Choose **Create new VPC** to go the Amazon VPC console. When you're done, return to this screen and refresh the list.

b. (Optional) For **Availability Zone**, let Amazon EC2 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 this screen and refresh the list.

8. For **Instance type requirements**, you can either specify instance attributes and let Amazon EC2 identify the optimal instance types with these attributes, or you can specify a list of instances. For more information, see Specify attributes for instance type selection for EC2 Fleet or Spot Fleet.

   a. If you choose **Specify instance attributes that match your compute requirements**, specify your instance attributes as follows:

      i. For **vCPUs**, enter the desired minimum and maximum number of vCPUs. To specify no limit, select **No minimum** or **No maximum**, or both.

      ii. For **Memory (GiB)**, enter the desired minimum and maximum amount of memory. To specify no limit, select **No minimum** or **No maximum**, or both.

      iii. (Optional) For **Additional instance attribute**, you can optionally specify one or more attributes to express your compute requirements in more detail. Each additional attribute adds a further constraint to your request. You can omit the additional attributes; when omitted, the default values are used. For a description of each attribute and their default values, see get-spot-placement-scores.

      iv. (Optional) To view the instance types with your specified attributes, expand **Preview matching instance types**. To exclude instance types from being used in your request, select the instances and then choose **Exclude selected instance types**.

   b. If you choose **Manually select instance types**, Spot Fleet provides a default list of instance types. To select more instance types, choose **Add instance types**, select the instance types to use in your request, and choose **Select**. To delete instance types, select the instance types and choose **Delete**.

9. For **Allocation strategy**, choose a Spot allocation strategy and an On-Demand allocation strategy that meets your needs. For more information, see Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity.
10. For **Your fleet request at a glance**, review your fleet configuration, and make any adjustments if necessary.
11. (Optional) To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**.
12. When you're ready to launch your Spot Fleet, 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`.

AWS CLI

To create a Spot Fleet request

Use the request-spot-fleet command.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For example configuration files, see Example CLI configurations Spot Fleet.

PowerShell

To create a Spot Fleet request

Use the Request-EC2SpotFleet cmdlet. The following example launches Spot Instances in a capacity-optimized fleet.

```
Request-EC2SpotFleet `
    -SpotFleetRequestConfig_TargetCapacity 50 `
    -SpotFleetRequestConfig_AllocationStrategy "CapacityOptimized" `
    -SpotFleetRequestConfig_IamFleetRole "arn:aws:iam::123456789012:role/my-spot-fleet-role" `
    -SpotFleetRequestConfig_LaunchTemplateConfig @($launchConfig)
```

Define the launch configuration as follows, setting the launch template and override properties that you need. For example configurations, see Example CLI configurations Spot Fleet.

```
$lcSpec = Amazon.EC2.Model.FleetLaunchTemplateSpecification
```

```
# To do - Set FleetLaunchTemplateSpecification properties
$lcOverrides = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
# To do - Set LaunchTemplateOverrides properties
$launchConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchConfig.LaunchTemplateSpecification $lcSpec
$launchConfig.Overrides @($lcOverrides)
```

Create a Spot Fleet that replaces unhealthy Spot Instances

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 Amazon EC2 instances.

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.
- 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

1. Follow the steps for creating a Spot Fleet in Create a Spot Fleet request using defined parameters.

2. To configure the fleet to replace unhealthy Spot Instances, expand **Additional launch parameters**, and under **Health check**, select **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

Use the request-spot-fleet command with the `ReplaceUnhealthyInstances` property of `SpotFleetRequestConfig`.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
        "TargetCapacity": 10,
        "ReplaceUnhealthyInstances": true
    }
}
```

PowerShell

To configure a Spot Fleet request to replace unhealthy Spot Instances

Use the Request-EC2SpotFleet cmdlet with the `-SpotFleetRequestConfig_ReplaceUnhealthyInstance` parameter.

```
-SpotFleetRequestConfig_ReplaceUnhealthyInstance $true
```

Tag a new or existing Spot Fleet request and the instances and volumes it launches

To help categorize and manage your Spot Fleet requests and the instances and volumes that it launches, you can tag them with custom metadata. You can assign a tag to a Spot Fleet request when you create it, or afterward. Similarly, you can assign a tag to the instances and volumes when they're launched by the fleet, 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 of them.

Note

You can only tag volumes that are attached to On-Demand Instances. You can't tag volumes that are attached to Spot Instances.

You can assign tags using the Amazon EC2 console or a command line tool.

For more information about how tags work, see Tag your Amazon EC2 resources.

Contents

Prerequisite

Grant the user the permission to tag resources. For more information, see Example: Tag resources.

To grant a user the permission to tag resources

Create an IAM policy that includes the following:

- The `ec2:CreateTags` action. This grants the user permission to create tags.
- The `ec2:RequestSpotFleet` action. This grants the 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": "*"
        }
    ]
}
```

> **Important**
>
> We currently do not support resource-level permissions for the `spot-fleet-request` resource. If you specify `spot-fleet-request` as a resource, you will get an unauthorized exception when you try to tag the fleet. The following example illustrates how *not* to set the policy.
>
> ```
> {
>     "Effect": "Allow",
>     "Action": [
>         "ec2:CreateTags",
>         "ec2:RequestSpotFleet"
>     ],
>     "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/*"
> }
> ```

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in [Create a permission set](#) in the *AWS IAM Identity Center User Guide*.

- Users managed in IAM through an identity provider:

Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.

- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

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 console

1. Follow the Create a Spot Fleet request using defined parameters procedure.
2. The way you add a tag depends on whether you manually configured the fleet or used a launch template.
   - If you manually configured the fleet, do the following:

     To add a tag, expand **Additional launch parameters**, choose **Create 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 **Instances** and **Fleet** are selected. To tag only the Spot Fleet request, clear **Instances**. To tag only the instances launched by the fleet, clear **Fleet**.

     > **Note**
     > When you manually configure a fleet, there is no option to tag volumes. Volume tags are only supported for volumes that are attached to On-Demand Instances. When you manually configure a fleet, you can't specify On-Demand Instances.

   - If you used a launch template, do the following:

     To add a tag to the fleet request, under **Tags**, choose **Create Tag**, and enter the key and value for the tag. Repeat for each tag.

To tag the resources in your fleet, you must specify tags in the launch template.

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 that is referenced in the Spot Fleet request.

Volume tags:

- Specify the tags for the volumes in the launch template that is referenced in the Spot Fleet request. Volume tagging in `LaunchSpecifications` is not supported.

In the following example, the Spot Fleet request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123. The instances that are launched by the fleet are tagged with one tag (which is the same as one of the tags for the Spot Fleet request): Key=Cost-Center and Value=123.

```
{
    "SpotFleetRequestConfig": {
```

```
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1,
        "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](#) that is referenced in the Spot Fleet request, or you can specify the tags in the Spot Fleet request configuration as follows:

- Specify the tags for the instances in `LaunchSpecifications`.
- For `ResourceType`, specify `instance`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the instances that are launched by the fleet are tagged with one tag: Key=Cost-Center and Value=123.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
```

```
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1
    }
}
```

To tag volumes attached to On-Demand Instances launched by a Spot Fleet using the AWS CLI

To tag volumes when they are created by the fleet, you must specify the tags in the [launch template](#) 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 create a Spot Fleet request, you can add tags to the fleet request using the console.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request.
4. 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

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 **Tags** tab.

To describe Spot Fleet request tags

Use the describe-tags command to view the tags for the specified resource. In the following example, you describe the tags for the specified Spot Fleet request.

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
    "Tags": [
        {
            "Key": "Environment",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Production"
        },
        {
            "Key": "Another key",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Another value"
        }
    ]
}
```

You can also view the tags of a Spot Fleet request by describing the Spot Fleet request.

Use the describe-spot-fleet-requests command to view the configuration of the specified Spot Fleet request, which includes any tags that were specified for the fleet request.

```
aws ec2 describe-spot-fleet-requests \
```

```
    --spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```


```
{
    "SpotFleetRequestConfigs": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2020-02-13T02:49:19.709Z",
            "SpotFleetRequestConfig": {
                "AllocationStrategy": "capacityOptimized",
                "OnDemandAllocationStrategy": "lowestPrice",
                "ExcessCapacityTerminationPolicy": "Default",
                "FulfilledCapacity": 2.0,
                "OnDemandFulfilledCapacity": 0.0,
                "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-
tagging-role",
                "LaunchSpecifications": [
                    {
                        "ImageId": "ami-0123456789EXAMPLE",
                        "InstanceType": "c4.large"
                    }
                ],
                "TargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "Type": "maintain",
                "ReplaceUnhealthyInstances": false,
                "InstanceInterruptionBehavior": "terminate"
            },
            "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "SpotFleetRequestState": "active",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                },
                {
                    "Key": "Another key",
                    "Value": "Another value"
                }
            ]
        }
    ]
}
```

Describe a Spot Fleet request, its instances, and event history

You can describe your Spot Fleet configuration, the instances in your Spot Fleet, and the event history of your Spot Fleet.

Console

To describe your Spot Fleet

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request. The ID begins with **sfr-**. 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**.

AWS CLI

To describe your Spot Fleet request

Use the describe-spot-fleet-requests command.

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

To describe the running instances for the specified Spot Fleet request

Use the describe-spot-fleet-instances command.

```
aws ec2 describe-spot-fleet-instances \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

To describe the event history for the specified Spot Fleet request

Use the describe-spot-fleet-request-history command.

```
aws ec2 describe-spot-fleet-request-history \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
```

```
    --start-time 2024-05-18T00:00:00Z
```

PowerShell

To describe your Spot Fleet request

Use the Get-EC2SpotFleetRequest cmdlet.

```
Get-EC2SpotFleetRequest
```

To describe the running instances for the specified Spot Fleet request

Use the Get-EC2SpotFleetInstance cmdlet.

```
Get-EC2SpotFleetInstance `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
```

To describe the event history for the specified Spot Fleet request

Use the Get-EC2SpotFleetRequestHistory cmdlet.

```
Get-EC2SpotFleetRequestHistory `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -UtcStartTime 2024-05-18T00:00:00Z
```

Modify a Spot Fleet request

You can modify an active Spot Fleet request to complete the following tasks:

- Increase the total target capacity and On-Demand portion
- Decrease the total target capacity and On-Demand portion

When you increase the total target capacity, the Spot Fleet launches additional Spot Instances according to the allocation strategy for its Spot Fleet request. When you increase the On-Demand portion, the Spot Fleet launches additional On-Demand Instances.

When you decrease the total 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 `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.

Considerations

- 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 a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Console

To modify a Spot Fleet request

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.
4. Choose **Actions**, **Modify target capacity**.
5. 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**.

AWS CLI

To modify a Spot Fleet request

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
```

PowerShell

To modify a Spot Fleet request

Use the Edit-EC2SpotFleetRequest cmdlet to update the target capacity of the specified Spot Fleet request.

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 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.

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

Cancel (delete) a Spot Fleet request

If you no longer require a Spot Fleet, you can cancel the Spot Fleet request, which deletes the request. After you cancel a fleet request, all Spot requests associated with the fleet are also canceled, so that no new Spot Instances are launched.

When you cancel a Spot Fleet request, you must also specify if you want to terminate all of its instances. These include both On-Demand Instances and Spot Instances.

If you specify that the instances must be terminated when the fleet request is canceled, the fleet request enters the `cancelled_terminating` state. Otherwise, it enters the `cancelled_running` state, and the instances continue to run until they are interrupted or you terminate them manually.

Restrictions

- You can cancel up to 100 fleets in a single request. If you exceed the specified number, no fleets are canceled.

Console

To cancel (delete) a Spot Fleet request

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.
4. Choose **Actions**, **Cancel request**.
5. In the **Cancel Spot request** dialog box, do the following:

   a. To terminate the associated instances at the same time as canceling the Spot Fleet request, leave the **Terminate instances** checkbox selected. To cancel the Spot Fleet request without terminating the associated instances, clear the **Terminate instances** checkbox.

   b. Choose **Confirm**.

AWS CLI

To cancel (delete) a Spot Fleet request and terminate its instances

Use the [cancel-spot-fleet-requests](#) command with the `--terminate-instances` option.

```
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": []
}
```

To cancel (delete) a Spot Fleet request without terminating its instances

Modify the previous example by using the `--no-terminate-instances` option instead.

```
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": []
}
```

PowerShell

To cancel (delete) a Spot Fleet request and terminate its instances

Use the Stop-EC2SpotFleetRequest cmdlet with the `-TerminateInstance` parameter.

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

To cancel (delete) a Spot Fleet request without terminating its instances

Modify the previous example by changing the value of the `-TerminateInstance` parameter.

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

Understand automatic scaling for Spot Fleet

Automatic scaling enables your Spot Fleet to increase or decrease its target capacity based on demand. With automatic scaling, a Spot Fleet can either launch instances (scale out) or terminate instances (scale in) within a specified range, in response to one or more scaling policies.

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.

Types of automatic scaling

Spot Fleet supports the following types of automatic scaling:

- Target tracking scaling – Increase or decrease the current capacity of the fleet by targeting a value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home—you select the desired temperature and the thermostat does the rest.
- Step scaling – 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 – Increase or decrease the current capacity of the fleet based on the date and time.

Considerations

When using automatic scaling for your Spot Fleet, consider the following:

- **Instance weighting** – If you're using instance weighting, keep in mind that Spot Fleet can exceed the target capacity as needed. Fulfilled capacity can be a floating-point number but target capacity must be an integer, so Spot Fleet rounds up to the next integer. You must take these behaviors into account when you look at the outcome of a scaling policy when an alarm is triggered. For example, suppose that the target capacity is 30, the fulfilled capacity is 30.1, and the scaling policy subtracts 1. When the alarm is triggered, the automatic scaling process

subtracts 1 from 30.1 to get 29.1 and then rounds it up to 30, so no scaling action is taken.
As another example, suppose that you selected instance weights of 2, 4, and 8, and a target capacity of 10, but no weight 2 instances were available so Spot Fleet provisioned instances of weights 4 and 8 for a fulfilled capacity of 12. If the scaling policy decreases target capacity by 20% and an alarm is triggered, the automatic scaling process subtracts 12*0.2 from 12 to get 9.6 and then rounds it up to 10, so no scaling action is taken.

- **Cooldown period** – 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.
- **Use detailed monitoring** – 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 Manage detailed monitoring for your EC2 instances and Create a Spot Fleet request using defined parameters.
- **AWS CLI** – If you use the AWS CLI for configuring scaling for Spot Fleet, you'll use the application-autoscaling commands.

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 required for using Spot Fleet and Amazon EC2, the user that accesses the fleet scaling settings must have the appropriate permissions for the services that support automatic scaling.

To use automatic scaling for Spot Fleet, 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 Identity and Access Management for Application Auto Scaling in the *Application Auto Scaling User Guide*.

The Application Auto Scaling service also needs permission to describe your Spot Fleet and CloudWatch alarms, and permissions to modify your Spot Fleet target capacity on your behalf. If you enable automatic scaling for your Spot Fleet, it creates a service-linked role named `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`. This service-linked role grants Application Auto Scaling permission to describe the alarms for your policies, to monitor the current capacity of the fleet, and to modify the capacity of the fleet. The original managed Spot Fleet role for Application Auto Scaling was `aws-ec2-spot-fleet-autoscale-role`, but it is no longer required. The service-linked role is the default role for Application Auto Scaling. For more information, see [Service-linked roles for Application Auto Scaling]() in the *Application Auto Scaling User Guide*.

Target tracking scaling: Scale Spot Fleet by targeting a value for a specific metric

With target tracking scaling, you create a target tracking scaling policy by selecting a metric and setting a target value. Spot Fleet then creates and manages the CloudWatch alarms that trigger the scaling policy, and calculates the scaling adjustment based on the chosen metric and target value. The scaling policy adjusts capacity by adding or removing instances as needed to keep the metric at, or close to, the specified target value. A target tracking policy not only keeps the metric close to the target value, but also adjusts to the fluctuations in the metric due to a fluctuating load pattern and minimizes rapid capacity fluctuations.

You can create multiple target tracking scaling policies for a Spot Fleet, provided each policy uses a different metric. The fleet scales based on the policy that specifies the largest fleet capacity. This allows you to cover multiple scenarios to ensure sufficient capacity for 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 a Spot Instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

> **Note**
>
> 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.

Prerequisites

- The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for requests of type `request`.
- Configure the [IAM permissions required for Spot Fleet automatic scaling]().
- Review the [Considerations]().

To configure a target tracking policy

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.
4. Choose the **Auto Scaling** tab near the bottom of the screen. If you selected the link for your Spot Fleet, there is no tab; instead, scroll down to the **Auto Scaling** section.
5. If automatic scaling is not configured, choose **Configure**.
6. 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.
7. For **Policy name**, enter a name for the policy.
8. Choose a **Target metric**.
9. Enter a **Target value** for the metric.
10. For **Cooldown period**, specify a new value (in seconds) or keep the default.
11. (Optional) To omit creating a scale-in policy based on the current configuration, select **Disable scale-in**. You can create a scale-in policy using a different configuration.
12. 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.

Step scaling: 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 a Spot Instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Prerequisites

- The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for requests of type `request`.
- Configure the IAM permissions required for Spot Fleet automatic scaling.
- 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.
- Review the Considerations.

To create a CloudWatch alarm

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, expand **Alarms** and choose **All alarms**.
3. Choose **Create alarm**.
4. On the **Specify metric and conditions** page, choose **Select metric**.
5. Choose **EC2 Spot**, then **Fleet Request Metrics**, and then select a metric (for example, **TargetCapacity**), 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 name and 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

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.

4. Choose the **Auto Scaling** tab near the bottom of the screen. If you selected the link for your Spot Fleet, there is no tab; instead, scroll down to the **Auto Scaling** section.
5. If automatic scaling is not configured, choose **Configure**.
6. Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Scaling policies do not scale your fleet below the minimum capacity or above the maximum capacity.
7. Under **Scaling policies**, for **Policy type**, choose **Step scaling policy**.
8. Initially, **Scaling policies** contain step scaling policies named **ScaleUp** and **ScaleDown**. You can complete these policies, or choose **Remove policy** to delete them. You can also choose **Add policy**.
9. 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 alarm** to open the Amazon CloudWatch console and create an alarm.

   c. For **Modify capacity**, define the amount by which to scale and the lower and upper bound of the step adjustment. You can add or remove a specific number of instances or a percentage of the existing fleet size, or set the fleet to an exact size.

      For example, to create a step scaling policy that increases the capacity of the fleet by 30 percent, choose **Add**, enter **30** in the next field, and then choose **percent**. By default, the lower bound for an add policy is the alarm threshold and the upper bound is positive (+) infinity. By default, the upper bound for a remove policy is the alarm threshold and the lower bound is negative (-) infinity.

   d. (Optional) To add another step, choose **Add step**.

   e. For **Cooldown period**, specify a new value (in seconds) or keep the default.
10. 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.

Scheduled scaling: Scale Spot Fleet on a schedule

Scaling your fleet on a schedule enables you to scale your application in response to predictable changes in demand. By creating *scheduled actions*, you can instruct Spot Fleet to perform scaling activities at specific times. To create a scheduled action, you must specify an existing Spot Fleet, the time when the scaling activity must occur, and the desired minimum and maximum capacity. Scheduled actions can be configured to scale once or on a recurring schedule. If you needs change, you can edit or delete scheduled actions.

Prerequisites

- Sheduled actions can only be created for existing Spot Fleets. You can't create a scheduled action when you create a Spot Fleet.
- The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for requests of type `request`.
- Configure the IAM permissions required for Spot Fleet automatic scaling.
- Review the Considerations.

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.
4. Choose the **Scheduled Scaling** tab near the bottom of the screen. If you selected the link for your Spot Fleet, there is no tab; instead, scroll down to the **Scheduled Scaling** section.
5. Choose **Create scheduled action**.
6. For **Name**, specify a name for the scheduled action.
7. Enter a value for **Minimum capacity**, **Maximum capacity**, or both.
8. For **Recurrence**, choose **Once**.
9. (Optional) Choose a date and time for **Start time**, **End time**, or both.
10. Choose **Create**.

To create a recurring 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.
4. Choose the **Scheduled Scaling** tab near the bottom of the screen. If you selected the link for your Spot Fleet, there is no tab; instead, scroll down to the **Scheduled Scaling** section.
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 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 EventBridge User Guide*.
8. (Optional) Choose a date and time for **Start time**, **End time**, or both.
9. 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.
4. Choose the **Scheduled Scaling** tab near the bottom of the screen. If you selected the link for your Spot Fleet, there is no tab; instead, scroll down to the **Scheduled Scaling** section.
5. Select the scheduled action and choose **Actions**, **Edit**.
6. 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.
4. Choose the **Scheduled Scaling** tab near the bottom of the screen. If you selected the link for your Spot Fleet, there is no tab; instead, scroll down to the **Scheduled Scaling** section.
5. Select the scheduled action and choose **Actions**, **Delete**.
6. 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 your EC2 Fleet or Spot Fleet

Effective monitoring of your EC2 Fleet or Spot Fleet is essential for maintaining optimal performance and ensuring reliability. There are various tools to help you achieve this, including Amazon CloudWatch and Amazon EventBridge, which are covered in this topic.

With CloudWatch, you can collect and track metrics, set alarms, and automatically react to changes in your fleet's status.

With EventBridge, you can monitor and respond programmatically to events emitted by your fleet. By defining rules in EventBridge, you can automate responses to specific fleet events, such as instance termination or fleet state changes, improving your operational efficiency.

Topics

Monitor your EC2 Fleet or Spot Fleet using CloudWatch

You can monitor your EC2 Fleet or Spot Fleet using the Amazon CloudWatch metrics described in this section.

> **Important**
>
> To ensure accuracy, we recommend that you enable detailed monitoring when using these metrics. For more information, see Manage detailed monitoring for your EC2 instances.

For more information about using CloudWatch, see Monitor your instances using CloudWatch.

EC2 Fleet and Spot Fleet metrics

The `AWS/EC2Spot` namespace includes the following metrics for your fleet, plus the CloudWatch metrics for the Spot Instances in your fleet. For more information, see Instance metrics.

Metric	Description
`AvailableInstancePoolsCount`	The Spot capacity pools specified in the fleet request. Units: Count
`BidsSubmittedForCapacity`	The capacity for which Amazon EC2 has submitted fleet requests. Units: Count
`EligibleInstancePoolCount`	The Spot capacity pools specified in the fleet request where Amazon EC2 can fulfill requests. Amazon EC2 does not fulfill requests in pools where the maximum price you're willing to pay for Spot Instances is less than the Spot price or the Spot price is greater than the price for On-Demand Instances. Units: Count
`FulfilledCapacity`	The capacity that Amazon EC2 has fulfilled. Units: Count
`MaxPercentCapacityAllocation`	The maximum value of `PercentCapacityAllocation` across all fleet pools specified in the fleet request. Units: Percent
`PendingCapacity`	The difference between `TargetCapacity` and `FulfilledCapacity` . Units: Count
`PercentCapacityAllocation`	The capacity allocated for the Spot capacity pool for the specified dimensions. To get the maximum value recorded

Metric	Description
	across all Spot capacity pools, use `MaxPercentCapacityAllocation` . Units: Percent
`TargetCapacity`	The target capacity of the fleet request. Units: Count
`TerminatingCapacity`	The capacity that is being terminated because the provisioned capacity is greater than the target capacity. Units: Count

If the unit of measure for a metric is `Count`, the most useful statistic is `Average`.

EC2 Fleet and Spot Fleet dimensions

To filter the data for your fleet, use the following dimensions.

Dimensions	Description
`AvailabilityZone`	Filter the data by Availability Zone.
`FleetRequestId`	Filter the data by fleet request.
`InstanceType`	Filter the data by instance type.

View the CloudWatch metrics for your EC2 Fleet or Spot Fleet

You can view the CloudWatch metrics for your fleet using the Amazon CloudWatch console. These metrics are displayed as monitoring graphs. These graphs show data points if the 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 fleet metrics or fleet metrics groups by fleet request ID, instance type, or Availability Zone.

To view fleet metrics

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, expand **Metrics**, and choose **All metrics**.
3. Choose the **EC2 Spot** namespace.

> **Note**
>
> If the **EC2 Spot** namespace is not displayed, there are two reasons for this. Either you have never used EC2 Fleet or Spot Fleet in the Region—only the AWS services that you're using send metrics to Amazon CloudWatch. Or, if you have used EC2 Fleet or Spot Fleet in the Region, but not for the past two weeks, the namespace does not appear.

4. To filter the metrics by dimension, choose one of the following:
   - **Fleet Request Metrics** – Group by fleet request
   - **By Availability Zone** – Group by fleet request and Availability Zone
   - **By Instance Type** – Group by fleet request and instance type
   - **By Availability Zone/Instance Type** – Group by fleet request, Availability Zone, and instance type
5. To view the data for a metric, select the checkbox next to the metric.

Monitor and programmatically respond to the events emitted by your EC2 Fleet or Spot Fleet using Amazon EventBridge

When the state of an EC2 Fleet or Spot Fleet changes, it 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.

You can use Amazon EventBridge to create rules that trigger programmatic actions in response to an event. For example, you can create two EventBridge rules: one triggered when a fleet state changes, and another triggered when an instance in the fleet is terminated. In this example, you can configure the first rule so that, if the fleet state changes, the rule invokes an SNS topic, sending an email notification to you. You can configure the second rule so that, if an instance in the fleet is terminated, the rule invokes a Lambda function to launch a new instance.

> **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. To use Amazon EventBridge to monitor fleet events, the request type must be `maintain` or `request`.

For instructions on how to describe a fleet's event history, see Describe the event history for your EC2 Fleet.

Topics

- Create Amazon EventBridge rules to monitor EC2 Fleet or Spot Fleet events
- EC2 Fleet event types
- Spot Fleet event types

Create Amazon EventBridge rules to monitor EC2 Fleet or Spot Fleet events

When a state change notification is emitted for an EC2 Fleet or Spot Fleet, it is sent as an event to Amazon EventBridge as a JSON file. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes the target (or targets) specified in the rule.

You can write EventBridge rules to automate actions based on matching event patterns.

The following fields in the event 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 and Spot Fleet events and example event data, see EC2 Fleet event types and Spot Fleet event types.

Examples

- Create an EventBridge rule to send a notification
- Create an EventBridge rule to trigger a Lambda function

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.

Prerequisite

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. For **Define rule detail**, do the following:

   a. 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.

   b. For **Event bus**, choose **default**. When an AWS service in your account generates an event, it always goes to your account's default event bus.
   c. For **Rule type**, choose **Rule with an event pattern**.
   d. Choose **Next**.
4. For **Build event pattern**, do the following:

   a. For **Event source**, choose **AWS events or EventBridge partner events**.
   b. For **Event pattern**, for this example you'll specify the following event pattern to match the `EC2 Fleet Instance Change` event.

```
{
 "source": ["aws.ec2fleet"],
```

```
  "detail-type": ["EC2 Fleet Instance Change"]
}
```

To add the event pattern, you can either use a template by choosing **Event pattern form**, or specify your own pattern by choosing **Custom pattern (JSON editor)**, as follows:

i. To use a template to create the event pattern, do the following:

   A. Choose **Event pattern form**.
   B. For **Event source**, choose **AWS services**.
   C. For **AWS Service**, choose **EC2 Fleet**.
   D. For **Event type**, choose **EC2 Fleet Instance Change**.
   E. To customize the template, choose **Edit pattern** and make your changes to match the example event pattern.

ii. (Alternative) To specify a custom event pattern, do the following:

   A. Choose **Custom pattern (JSON editor)**.
   B. In the **Event pattern** box, add the event pattern for this example.

c. Choose **Next**.

5. For **Select target(s)**, do the following:

   a. For **Target types**, choose **AWS service**.
   b. For **Select a target**, choose **SNS topic** to send an email, text message, or mobile push notification when the event occurs.
   c. 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*.
   d. (Optional) Under **Additional settings**, you can optionally configure additional settings. For more information, see [Creating Amazon EventBridge rules that react to events] (step 16) in the *Amazon EventBridge User Guide*.
   e. Choose **Next**.

6. (Optional) For **Tags**, you can optionally assign one or more tags to your rule, and then choose **Next**.
7. For **Review and create**, do the following:

   a. Review the details of the rule and modify them as necessary.

b. Choose **Create rule**.

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 the Lambda function to use in the EventBridge rule

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, see Create your first Lambda function in the *AWS Lambda Developer Guide*.

To create an EventBridge rule to trigger a Lambda function when an instance in an EC2 Fleet changes state

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Choose **Create rule**.
3. For **Define rule detail**, do the following:

   a. 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.

   b. For **Event bus**, choose **default**. When an AWS service in your account generates an event, it always goes to your account's default event bus.

   c. For **Rule type**, choose **Rule with an event pattern**.

   d. Choose **Next**.

4. For **Build event pattern**, do the following:

   a. For **Event source**, choose **AWS events or EventBridge partner events**.

   b. For **Event pattern**, for this example you'll specify the following event pattern to match the `EC2 Fleet Instance Change` event and `launched` sub-type.

   ```
   {
    "source": ["aws.ec2fleet"],
    "detail-type": ["EC2 Fleet Instance Change"],
    "detail": {
      "sub-type": ["launched"]
   }
   ```

   To add the event pattern, you can either use a template by choosing **Event pattern form**, or specify your own pattern by choosing **Custom pattern (JSON editor)**, as follows:

   i. To use a template to create the event pattern, do the following:

      A. Choose **Event pattern form**.

      B. For **Event source**, choose **AWS services**.

      C. For **AWS Service**, choose **EC2 Fleet**.

      D. For **Event type**, choose **EC2 Fleet Instance Change**.

      E. Choose **Edit pattern**, and add `"detail": {"sub-type": ["launched"]` to match the example event pattern. For proper JSON format, insert a comma (`,`) after the preceding square bracket (`]`).

   ii. (Alternative) To specify a custom event pattern, do the following:

      A. Choose **Custom pattern (JSON editor)**.

      B. In the **Event pattern** box, add the event pattern for this example.

   c. Choose **Next**.

5. For **Select target(s)**, do the following:

   a. For **Target types**, choose **AWS service**.

   b. For **Select a target**, choose **SNS topic** to send an email, text message, or mobile push notification when the event occurs.

   c. For **Topic**, choose **Lambda function**, and for **Function**, choose the function that you created to respond when the event occurs.

   d. (Optional) Under **Additional settings**, you can optionally configure additional settings. For more information, see Creating Amazon EventBridge rules that react to events (step 16) in the *Amazon EventBridge User Guide*.

   e. Choose **Next**.

6. (Optional) For **Tags**, you can optionally assign one or more tags to your rule, and then choose **Next**.

7. For **Review and create**, do the following:

   a. Review the details of the rule and modify them as necessary.

   b. Choose **Create rule**.

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*.

EC2 Fleet event types

There are five EC2 Fleet event types. For each event type, there are several sub-types.

Event types

- EC2 Fleet State Change
- EC2 Fleet Spot Instance Request Change
- EC2 Fleet Instance Change
- EC2 Fleet Information
- EC2 Fleet Error

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.

```
{
    "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:

`active`

The EC2 Fleet request has been validated and Amazon EC2 is attempting to maintain the target number of running instances.

`deleted`

The EC2 Fleet request is deleted and has no running instances. The EC2 Fleet will be deleted two days after its instances are terminated.

`deleted_running`

The EC2 Fleet request is deleted and does not launch additional instances. Its existing instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.

`deleted_terminating`

The EC2 Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.

`expired`

The EC2 Fleet request has expired. If the request was created with `TerminateInstancesWithExpiration` set, a subsequent `terminated` event indicates that the 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.

`modify_succeeded`

The EC2 Fleet request was modified.

`submitted`

The EC2 Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances.

`progress`

The EC2 Fleet request is in the process of being fulfilled.

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:

`active`

The Spot Instance request is fulfilled and has an associated Spot Instance.

`cancelled`

You cancelled the Spot Instance request, or the Spot Instance request expired.

`disabled`

You stopped the Spot Instance.

`submitted`

The Spot Instance request is submitted.

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-6b0809bfff0a"
    ],
    "detail": {
        "instance-id": "i-0c594155dd5ff1829",
        "description": "{\"instanceType\":\"c5.large\",\"image\":\"ami-6057e21a\",
\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":\"us-east-1d\"}",
        "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 when a Spot Instance was terminated by Amazon EC2 during scale-down, when the target capacity of the fleet was modified down, for example, from a target capacity of 4 to a target capacity of 3.

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 fulfil 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": "c4.xlarge, ami-0947d2ba12ee1ff75, Linux/UNIX, us-east-1a,
 Spot price in either SpotFleetRequestConfigData or SpotFleetLaunchSpecification or
 LaunchTemplate or LaunchTemplateOverrides is less than Spot market price $0.0619",
        "sub-type": "launchSpecUnusable"
    }
}
```

The possible values for `sub-type` are:

`fleetProgressHalted`

The price in every launch specification is not valid because it is below the Spot price (all the launch specifications have produced `launchSpecUnusable` events). A launch specification might become valid if the Spot price changes.

`launchSpecTemporarilyBlacklisted`

The configuration is not valid and several attempts to launch instances have failed. For more information, see the description of the event.

`launchSpecUnusable`

The price in a launch specification is not valid because it is below the Spot price.

`registerWithLoadBalancersFailed`

An attempt to register instances with load balancers 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 fulfil 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:

`iamFleetRoleInvalid`

The EC2 Fleet does not have the required permissions to either launch or terminate an instance.

`allLaunchSpecsTemporarilyBlacklisted`

None of the configurations are valid, and several attempts to launch instances have failed. 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.

`spotFleetRequestConfigurationInvalid`

The configuration is not valid. For more information, see the description of the event.

Spot Fleet event types

There are five Spot Fleet event types. For each event type, there are several sub-types.

Event types

- EC2 Spot Fleet State Change
- EC2 Spot Fleet Spot Instance Request Change
- EC2 Spot Fleet Instance Change
- EC2 Spot Fleet Information
- EC2 Spot Fleet Error

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:

`active`

The Spot Fleet request has been validated and Amazon EC2 is attempting to maintain the target number of running instances.

`cancelled`

The Spot Fleet request is canceled and has no running instances. The Spot Fleet will be deleted two days after its instances are terminated.

`cancelled_running`

The Spot Fleet request is canceled 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_terminating`

The Spot Fleet request is canceled and its instances are terminating. The request remains in this state until all instances are terminated.

`expired`

The Spot Fleet request has expired. If the request was created with `TerminateInstancesWithExpiration` set, a subsequent `terminated` event indicates that the 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.

`modify_succeeded`

The Spot Fleet request was modified.

`submitted`

The Spot Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances.

`progress`

The Spot Fleet request is in the process of being fulfilled.

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",
        "sub-type": "cancelled"
    }
}
```

The possible values for `sub-type` are:

`active`

The Spot Instance request is fulfilled and has an associated Spot Instance.

`cancelled`

You cancelled the Spot Instance request, or the Spot Instance request expired.

`disabled`

You stopped the Spot Instance.

`submitted`

The Spot Instance request is submitted.

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.


```
{
    "version": "0",
    "id": "11591686-5bd7-bbaa-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-032930428bf1abbff\",\"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 when a Spot Instance was terminated by Amazon EC2 during scale-down, when the target capacity of the fleet was modified down, for example, from a target capacity of 4 to a target capacity of 3.

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": "73a60f70-3409-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:

`fleetProgressHalted`

The price in every launch specification is not valid because it is below the Spot price (all the launch specifications have produced `launchSpecUnusable` events). A launch specification might become valid if the Spot price changes.

`launchSpecTemporarilyBlacklisted`

The configuration is not valid and several attempts to launch instances have failed. For more information, see the description of the event.

`launchSpecUnusable`

The price in a launch specification is not valid because it is below the Spot price.

`registerWithLoadBalancersFailed`

An attempt to register instances with load balancers 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": "2020-11-09T06:56:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/
sfr-38725d30-25f1-4f30-83ce-2907c56dba17"
```

```
    ],
    "detail": {
        "description": "r4.2xlarge, ami-032930428bf1abbff, Linux/UNIX: The
 associatePublicIPAddress parameter can only be specified for the network interface
 with DeviceIndex 0. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

The possible values for `sub-type` are:

`iamFleetRoleInvalid`

The Spot Fleet does not have the required permissions to either launch or terminate an instance.

`allLaunchSpecsTemporarilyBlacklisted`

None of the configurations are valid, and several attempts to launch instances have failed. 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.

`spotFleetRequestConfigurationInvalid`

The configuration is not valid. For more information, see the description of the event.

Tutorials for EC2 Fleet

There are different ways to configure an EC2 Fleet. The configuration you choose depends on your specific use case.

The following tutorials cover some of the possible use cases and provide the tasks required to implement them.

Use case	Link to tutorial
Use instance weighting to manage the availability and performance of your EC2 Fleet.	[Tutorial: Configure EC2 Fleet to use instance weighting]()

Use case	Link to tutorial
With instance weighting, you assign a weight to each instance type in your EC2 Fleet to represent their compute capacity and performance relative to each other. Based on the weights, the fleet can use any combination of the specified instance types, as long as it can fulfil the desired target capacity.	
Use On-Demand capacity to ensure availability during peak periods, but benefit from additional Spot capacity at a lower cost. Configure your EC2 Fleet to use On-Demand Instances as the primary capacity to ensure available capacity during peak periods. In addition, allocate some capacity to Spot Instances to benefit from discounted pricing, while keeping in mind that Spot Instances can be interrupted if Amazon EC2 needs the capacity back.	Tutorial: Configure EC2 Fleet to use On-Demand Instances as the primary capacity
Use Capacity Reservations to reserve compute capacity for your On-Demand Instances. Configure your EC2 Fleet to use `targeted` Capacity Reservations first when launching On-Demand Instances. If you have strict capacity requirements, and are running business-critical workloads that require a certain level of long or short-term capacity assurance, we recommend that you create a Capacity Reservation to ensure that you always have access to Amazon EC2 capacity when you need it, for as long as you need it.	Tutorial: Configure EC2 Fleet to launch On-Demand Instances using targeted Capacity Reservations

Use case	Link to tutorial
Use Capacity Blocks to reserve highly sought-after GPU instances for your ML workloads. Configure your EC2 Fleet to launch instances into Capacity Blocks.	Tutorial: Configure your EC2 Fleet to launch instances into Capacity Blocks

Tutorial: Configure EC2 Fleet to use instance weighting

This tutorial uses a fictitious company called Example Corp to illustrate the process of requesting an EC2 Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to use the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews Spot Best Practices. Next, Example Corp determines the requirements for their EC2 Fleet.

Instance types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16

Instance type	Memory (GiB)	vCPUs
r3.8xlarge	244	32

Target capacity in units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as one unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their EC2 Fleet request to 20 units.

Instance weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their EC2 Fleet request.

Price per unit hour

Example Corp uses the On-Demand price per instance hour as a starting point for their price. They could also use recent Spot prices, or a combination of the two. To calculate the price per unit hour, they divide their starting price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	$0.7	1	$0.7
r3.4xLarge	$1.4	2	$0.7

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.8xLarge	$2.8	4	$0.7

Example Corp could use a global price per unit hour of $0.7 and be competitive for all three instance types. They could also use a global price per unit hour of $0.7 and a specific price per unit hour of $0.9 in the `r3.8xlarge` launch specification.

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.

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 an Amazon EC2 launch template.

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.

```
{
    "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.

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).

If Example Corp used the `diversified` strategy, the Spot Instances would come from all three pools. The EC2 Fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 units.

Tutorial: Configure EC2 Fleet to use On-Demand Instances 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, aims 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 handle peak periods, but would like to benefit from additional capacity at a lower cost. The company 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 – To enhance performance, but at a lower price, ABC Online plans to provision 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.

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 an Amazon EC2 launch template.

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.

```
{
    "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.

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 available capacity.

Tutorial: Configure EC2 Fleet to launch On-Demand Instances using targeted Capacity Reservations

This tutorial walks you through all the steps that you must perform so that your EC2 Fleet launches On-Demand Instances into `targeted` Capacity Reservations.

You will learn how to configure a fleet to use `targeted` On-Demand Capacity Reservations first when launching On-Demand Instances. You will also learn how to configure the fleet so that, when the total On-Demand target capacity exceeds the number of available unused Capacity Reservations, the fleet uses the specified allocation strategy for selecting the instance pools in which to launch the remaining target capacity.

EC2 Fleet configuration

In this tutorial, the fleet is configured as follows:

- Target capacity: 10 On-Demand Instances
- Total unused `targeted` Capacity Reservations: 6 (less than the fleet's On-Demand target capacity of 10 On-Demand Instances)

- Number of Capacity Reservation pools: 2 (`us-east-1a` and `us-east-1b`)
- Number of Capacity Reservations per pool: 3
- On-Demand allocation strategy: `lowest-price` (When the number of unused Capacity Reservations is less than the On-Demand target capacity, the fleet determines the pools in which to launch the remaining On-Demand capacity based on the On-Demand allocation strategy.)

  Note that you can also use the `prioritized` allocation strategy instead of the `lowest-price` allocation strategy.

To launch On-Demand Instances into targeted Capacity Reservations, you must perform a number of steps, as follows:

- Step 1: Create Capacity Reservations
- Step 2: Create a Capacity Reservation resource group
- Step 3: Add the Capacity Reservations to the Capacity Reservation resource group
- (Optional) Step 4: View the Capacity Reservations in the resource group
- Step 5: Create a launch template that specifies that the Capacity Reservation targets a specific resource group
- (Optional) Step 6: Describe the launch template
- Step 7: Create an EC2 Fleet
- (Optional) Step 8: View the number of remaining unused Capacity Reservations

Step 1: Create Capacity Reservations

Use the create-capacity-reservation command to create the Capacity Reservations, three for `us-east-1a` and another three for `us-east-1b`. Except for the Availability Zone, the other attributes of the Capacity Reservations are identical.

3 Capacity Reservations in `us-east-1a`

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1a \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

Example of resulting Capacity Reservation ID

```
cr-1234567890abcdef1
```

3 Capacity Reservations in `us-east-1b`

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1b \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

Example of resulting Capacity Reservation ID

```
cr-54321abcdef567890
```

Step 2: Create a Capacity Reservation resource group

Use the `resource-groups` service and the create-group command to create a Capacity Reservation resource group. In this example, the resource group is named `my-cr-group`. For information about why you must create a resource group, see Use Capacity Reservations to reserve On-Demand capacity in EC2 Fleet.

```
aws resource-groups create-group \
    --name my-cr-group \
    --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-
types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

Step 3: Add the Capacity Reservations to the Capacity Reservation resource group

Use the `resource-groups` service and the group-resources command to add the Capacity Reservations that you created in Step 1 to the Capacity Reservations resource group. Note that you must reference the On-Demand Capacity Reservations by their ARNs.

```
aws resource-groups group-resources \
    --group my-cr-group \
    --resource-arns \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1 \
```

```
    arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

Example output

```
{
   "Failed": [],
   "Succeeded": [
   "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
   "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
   ]
}
```

(Optional) Step 4: View the Capacity Reservations in the resource group

Use the `resource-groups` service and the list-group-resources command to optionally describe the resource group to view its Capacity Reservations.

```
aws resource-groups list-group-resources --group my-cr-group
```

Example output

```
{
    "ResourceIdentifiers": [
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1"
        },
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-54321abcdef567890"
        }
    ]
}
```

Step 5: Create a launch template that specifies that the Capacity Reservation targets a specific resource group

Use the create-launch-template command to create a launch template in which to specify the Capacity Reservations to use. In this example, the fleet will use `targeted` Capacity Reservations, which have been added to a resource group. Therefore, the launch template data specifies that the Capacity Reservation targets a specific resource group. In this example, the launch template is named `my-launch-template`.

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --launch-template-data \
        '{"ImageId": "ami-0123456789example",
          "CapacityReservationSpecification":
            {"CapacityReservationTarget":
                { "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-east-1:123456789012:group/my-cr-group" }
            }
        }'
```

(Optional) Step 6: Describe the launch template

Use the describe-launch-template-versions command to optionally describe the launch template to view its configuration.

```
aws ec2 describe-launch-template-versions --launch-template-name my-launch-template
```

Example output

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-01234567890example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2021-01-19T20:50:19.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Admin",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0947d2ba12ee1ff75",
                "CapacityReservationSpecification": {
                    "CapacityReservationTarget": {
                        "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-east-1:123456789012:group/my-cr-group"
                    }
                }
```

```
            }
        }
    ]
}
```

Step 7: Create an EC2 Fleet

Create an EC2 Fleet that specifies the configuration information for the instances that it will launch. The following EC2 Fleet configuration shows only the pertinent configurations for this example. The launch template `my-launch-template` is the launch template you created in Step 5. There are two instance pools, each with the same instance type (`c5.xlarge`), but with different Availability Zones (`us-east-1a` and `us-east-1b`). The price of the instance pools is the same because pricing is defined for the Region, not per Availability Zone. The total target capacity is 10, and the default target capacity type is `on-demand`. The On-Demand allocation strategy is `lowest-price`. The usage strategy for Capacity Reservations is `use-capacity-reservations-first`.

> **Note**
>
> The fleet type must be `instant`. Other fleet types do not support `use-capacity-reservations-first`.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
```

```
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
             "UsageStrategy": "use-capacity-reservations-first"
         }
    },
    "Type": "instant"
}
```

After you create the `instant` fleet using the preceding configuration, the following 10 instances are launched to meet the target capacity:

- The Capacity Reservations are used first to launch 6 On-Demand Instances as follows:
  - 3 On-Demand Instances are launched into the 3 `c5.xlarge` `targeted` Capacity Reservations in `us-east-1a`
  - 3 On-Demand Instances are launched into the 3 `c5.xlarge` `targeted` Capacity Reservations in `us-east-1b`
- To meet the target capacity, 4 additional On-Demand Instances are launched into regular On-Demand capacity according to the On-Demand allocation strategy, which is `lowest-price` in this example. However, because the pools are the same price (because price is per Region and not per Availability Zone), the fleet launches the remaining 4 On-Demand Instances into either of the pools.

(Optional) Step 8: View the number of remaining unused Capacity Reservations

After the fleet is launched, you can optionally run [describe-capacity-reservations]() to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```
{ "CapacityReservationId":  "cr-111",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}
```

```
 { "CapacityReservationId":  "cr-222",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}
```

Tutorial: Configure your EC2 Fleet to launch instances into Capacity Blocks

This tutorial walks you through the steps that you must perform so that your EC2 Fleet launches instances into Capacity Blocks.

In most cases, the target capacity of the EC2 Fleet request should be less than or equal to the available capacity of the Capacity Block reservation that you are targeting. Target capacity requests that exceed the limits of the Capacity Block reservation will not be fulfilled. If the target capacity request exceeds the limits of your Capacity Block reservation, you will receive an `Insufficient Capacity Exception` for the capacity that exceeds the limits of your Capacity Block reservation.

> **Note**
>
> For Capacity Blocks, EC2 Fleet will not fallback to launching On-Demand Instances for the remainder of the desired target capacity.

If EC2 Fleet is unable to fulfill the requested target capacity in an available Capacity Block reservation, EC2 Fleet will fulfill as much capacity as it can and return the instances that it was able to launch. You can repeat the call to EC2 Fleet again until all the instances are provisioned.

After configuring the EC2 Fleet request, you must wait until the start date of your Capacity Block reservation. If you make requests to EC2 Fleet to launch into a Capacity Block that hasn't started yet, you will receive an `Insufficient Capacity Error`.

After your Capacity Block reservation becomes active, you can make EC2 Fleet API calls and provision the instances into your Capacity Block based on the parameters you selected. Instances running in the Capacity Block continue to run until you manually stop or terminate them or until Amazon EC2 terminates the instances when the Capacity Block reservation ends.

For more information about Capacity Blocks, see Capacity Blocks for ML.

Considerations

- Only EC2 Fleet requests of type `instant` are supported for launching instances into Capacity Blocks. For more information, see [Configure an EC2 Fleet of type instant](#).
- Multiple Capacity Blocks in the same EC2 Fleet request aren't supported.
- Using `OnDemandTargetCapacity` or `SpotTargetCapacity` while also setting `capacity-block` as the `DefaultTargetCapacity` isn't supported.
- If `DefaultTargetCapacityType` is set to `capacity-block`, you can't provide `OnDemandOptions::CapacityReservationOptions`. An exception will occur.

To configure an EC2 Fleet to launch instances into Capacity Blocks

1. **Create a launch template.**

   In the launch template, do the following:

   - For `InstanceMarketOptionsRequest`, set `MarketType` to `capacity-block`.
   - To target the Capacity Block reservation, for `CapacityReservationID`, specify the Capacity Block reservation ID.

   Make note of launch template name and version. You'll use this information in the next step.

   For more information about creating a launch template, see [Create an Amazon EC2 launch template](#).

2. **Configure the EC2 Fleet.**

   Create a file, `config.json`, with the following configuration for your EC2 Fleet. In the following example, replace the resource identifiers with your own resource identifiers.

   For more information about configuring an EC2 Fleet, see [Create an EC2 Fleet](#).

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "CBR-launch-template",
                "Version": "1"
            },
```

```
                "Overrides": [
                    {
                        "InstanceType": "p5.48xlarge",
                        "AvailabilityZone": "us-east-1a"
                    },
                ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 10,
            "DefaultTargetCapacityType": "capacity-block"
        },
        "Type": "instant"
}
```

3. **Launch the fleet.**

   Use the following create-fleet command.

   ```
   aws ec2 create-fleet --cli-input-json file://config.json
   ```

   For more information, see Create an EC2 Fleet.

Example CLI configurations for EC2 Fleet

You can define your EC2 Fleet configuration in a JSON file, and then reference that file with the create-fleet command to create your fleet, as follows:

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

The following examples illustrate launch configurations for various EC2 Fleet use cases. For more information about the configuration parameters, see create-fleet.

Examples

- Example 1: Launch Spot Instances as the default purchasing option
- Example 2: Launch On-Demand Instances as the default purchasing option
- Example 3: Launch On-Demand Instances as the primary capacity
- Example 4: Launch On-Demand Instances using multiple Capacity Reservations

For more CLI examples for fleets of type `instant`, see Configure an EC2 Fleet of type instant.

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 On-Demand Instances using multiple 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`. This example demonstrates how the fleet selects the Capacity Reservations to use when there are more Capacity Reservations than are needed to fulfil the target capacity.

In this example, the fleet configuration is as follows:

- Target capacity: 12 On-Demand Instances
- Total unused Capacity Reservations: 15 (more than the fleet's target capacity of 12 On-Demand Instances)
- Number of Capacity Reservation pools: 3 (`m5.large`, `m4.xlarge`, and `m4.2xlarge`)
- Number of Capacity Reservations per pool: 5
- On-Demand allocation strategy: `lowest-price` (When there are multiple unused Capacity Reservations in multiple instance pools, the fleet determines the pools in which to launch the On-Demand Instances based on the On-Demand allocation strategy.)

  Note that you can also use the `prioritized` allocation strategy instead of the `lowest-price` allocation strategy.

Capacity Reservations

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

Fleet configuration

The following fleet configuration shows only the pertinent configurations for this example. The total target capacity is 12, and the default target capacity type is `on-demand`. The On-Demand allocation strategy is `lowest-price`. The usage strategy for Capacity Reservations is `use-capacity-reservations-first`.

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 of type `instant`. Other fleet types do not support `use-capacity-reservations-first`.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-abc1234567example",
                "Version": "1"
            },
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
```

```
    },
    "Type": "instant"
}
```

After you create the `instant` fleet using the preceding configuration, the following 12 instances are launched to meet the target capacity:

- 5 `m5.large` On-Demand Instances in `us-east-1a` – `m5.large` in `us-east-1a` is the lowest price, and there are 5 available unused `m5.large` Capacity Reservations
- 5 `m4.xlarge` On-Demand Instances in us-east-1a – `m4.xlarge` in `us-east-1a` is the next lowest price, and there are 5 available unused `m4.xlarge` Capacity Reservations
- 2 `m4.2xlarge` On-Demand Instances in us-east-1a – `m4.2xlarge` in `us-east-1a` is the third lowest price, and there are 5 available unused `m4.2xlarge` Capacity Reservations of which only 2 are needed to meet the target capacity

After the fleet is launched, you can run [describe-capacity-reservations](#) to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the `m5.large` and `m4.xlarge` Capacity Reservations were used, with 3 `m4.2xlarge` Capacity Reservations remaining unused.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 3
}
```

Example 5: Launch On-Demand Instances using Capacity Reservations when the total target capacity exceeds 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`. This example demonstrates how the fleet selects the instance pools in which to launch On-Demand Instances when the total target capacity exceeds the number of available unused Capacity Reservations.

In this example, the fleet configuration is as follows:

- Target capacity: 16 On-Demand Instances
- Total unused Capacity Reservations: 15 (less than the fleet's target capacity of 16 On-Demand Instances)
- Number of Capacity Reservation pools: 3 (`m5.large`, `m4.xlarge`, and `m4.2xlarge`)
- Number of Capacity Reservations per pool: 5
- On-Demand allocation strategy: `lowest-price` (When the number of unused Capacity Reservations is less than the On-Demand target capacity, the fleet determines the pools in which to launch the remaining On-Demand capacity based on the On-Demand allocation strategy.)

  Note that you can also use the `prioritized` allocation strategy instead of the `lowest-price` allocation strategy.

Capacity Reservations

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
```

```
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

Fleet configuration

The following fleet configuration shows only the pertinent configurations for this example. The total target capacity is 16, and the default target capacity type is `on-demand`. The On-Demand allocation strategy is `lowest-price`. The usage strategy for Capacity Reservations is `use-capacity-reservations-first`.

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`. Other fleet types do not support `use-capacity-reservations-first`.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            },
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

After you create the `instant` fleet using the preceding configuration, the following 16 instances are launched to meet the target capacity:

- 6 m5.large On-Demand Instances in us-east-1a – m5.large in us-east-1a is the lowest price, and there are 5 available unused m5.large Capacity Reservations. The Capacity Reservations are used first to launch 5 On-Demand Instances. After the remaining m4.xlarge and m4.2xlarge Capacity Reservations are used, to meet the target capacity an additional On-Demand Instance is launched according to the On-Demand allocation strategy, which is lowest-price in this example.
- 5 m4.xlarge On-Demand Instances in us-east-1a – m4.xlarge in us-east-1a is the next lowest price, and there are 5 available unused m4.xlarge Capacity Reservations
- 5 m4.2xlarge On-Demand Instances in us-east-1a – m4.2xlarge in us-east-1a is the third lowest price, and there are 5 available unused m4.2xlarge Capacity Reservations

After the fleet is launched, you can run describe-capacity-reservations to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 0
}
```

Example 6: Launch On-Demand Instances using targeted Capacity Reservations

You can configure a fleet to use targeted On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to use-capacity-

`reservations-first`. This example demonstrates how to launch On-Demand Instances into `targeted` Capacity Reservations, where the attributes of the Capacity Reservations are the same except for their Availability Zones (`us-east-1a` and `us-east-1b`). It also demonstrates how the fleet selects the instance pools in which to launch On-Demand Instances when the total target capacity exceeds the number of available unused Capacity Reservations.

In this example, the fleet configuration is as follows:

- Target capacity: 10 On-Demand Instances
- Total unused `targeted` Capacity Reservations: 6 (less than the fleet's On-Demand target capacity of 10 On-Demand Instances)
- Number of Capacity Reservation pools: 2 (`us-east-1a` and `us-east-1b`)
- Number of Capacity Reservations per pool: 3
- On-Demand allocation strategy: `lowest-price` (When the number of unused Capacity Reservations is less than the On-Demand target capacity, the fleet determines the pools in which to launch the remaining On-Demand capacity based on the On-Demand allocation strategy.)

  Note that you can also use the `prioritized` allocation strategy instead of the `lowest-price` allocation strategy.

For a walkthrough of the procedures that you must perform to accomplish this example, see [Tutorial: Configure EC2 Fleet to launch On-Demand Instances using targeted Capacity Reservations](#).

Capacity Reservations

The account has the following 6 unused Capacity Reservations in 2 different pools. In this example, the pools differ by their Availability Zones. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

```
{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1b",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

Fleet configuration

The following fleet configuration shows only the pertinent configurations for this example. The total target capacity is 10, and the default target capacity type is `on-demand`. The On-Demand allocation strategy is `lowest-price`. The usage strategy for Capacity Reservations is `use-capacity-reservations-first`.

In this example, the On-Demand Instance price for `c5.xlarge` in `us-east-1` is $0.17 per hour.

> **Note**
>
> The fleet type must be `instant`. Other fleet types do not support `use-capacity-reservations-first`.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
```

```
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
             "UsageStrategy": "use-capacity-reservations-first"
         }
    },
    "Type": "instant"
}
```

After you create the `instant` fleet using the preceding configuration, the following 10 instances are launched to meet the target capacity:

- The Capacity Reservations are used first to launch 6 On-Demand Instances as follows:
  - 3 On-Demand Instances are launched into the 3 `c5.xlarge` `targeted` Capacity Reservations in `us-east-1a`
  - 3 On-Demand Instances are launched into the 3 `c5.xlarge` `targeted` Capacity Reservations in `us-east-1b`
- To meet the target capacity, 4 additional On-Demand Instances are launched into regular On-Demand capacity according to the On-Demand allocation strategy, which is `lowest-price` in this example. However, because the pools are the same price (because price is per Region and not per Availability Zone), the fleet launches the remaining 4 On-Demand Instances into either of the pools.

After the fleet is launched, you can run describe-capacity-reservations to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}
```

```
{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}
```

Example 7: 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-before-terminate`. To configure the time delay from when the new replacement Spot Instances are launched to when the old Spot Instances are automatically deleted, for `termination-delay`, specify a value in seconds. For more information, see Configuration options.

Note

We recommend using `launch-before-terminate` only if you can predict how long your instance shutdown procedures will take to complete so that the old instances are only terminated after these procedures are completed. You are charged for all 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 Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances.

```
{
    "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-before-terminate",
                "TerminationDelay": "720"
            }
        }
    }
}
```

Example 8: 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.

```
{
    "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 9: 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.

```
{
    "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"
}
```

Example 10: Launch Spot Instances in a price-capacity-optimized fleet

The following example demonstrates how to configure an EC2 Fleet with a Spot allocation strategy that optimizes for both capacity and lowest price. To optimize for capacity while taking price into consideration, you must set the Spot `AllocationStrategy` to `price-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 while also choosing the pool that is the lowest priced.

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
```

```
        "MinTargetCapacity": 2,
        "SingleInstanceType": true
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
    },
    "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,
        "OnDemandTargetCapacity":0,
        "SpotTargetCapacity":50,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 11: Configure attribute-based instance type selection

The following example demonstrates how to configure an EC2 Fleet to use attribute-based instance type selection for identifying instance types. To specify the required instance attributes, you specify the attributes in the `InstanceRequirements` structure.

In the following example, two instance attributes are specified:

- `VCpuCount` – A minimum of 2 vCPUs is specified. Because no maximum is specified, there is no maximum limit.
- `MemoryMiB` – A minimum of 4 MiB of memory is specified. Because no maximum is specified, there is no maximum limit.

Any instance types that have 2 or more vCPUs and 4 MiB or more of memory will be identified. However, price protection and the allocation strategy might exclude some instance types when EC2 Fleet provisions the fleet.

For a list and descriptions of all the possible attributes that you can specify, see InstanceRequirements in the *Amazon EC2 API Reference*.

```
{
 "SpotOptions": {
  "AllocationStrategy": "price-capacity-optimized"
 },
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }],
 "TargetCapacitySpecification": {
```

```
    "TotalTargetCapacity": 20,
    "DefaultTargetCapacityType": "spot"
  },
  "Type": "instant"
}
```

Example CLI configurations Spot Fleet

You can define your Spot Fleet configuration in a JSON file, and then reference that file using the request-spot-fleet AWS CLI command to create your fleet, as follows:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://file_name.json
```

The following examples illustrate launch configurations for various Spot Fleet use cases. For more information about the configuration parameters, see request-spot-fleet. For more information about creating a Spot Fleet, see Create a Spot Fleet.

Note

For Spot Fleet, you can't specify a network interface ID in a launch template or launch specification. Make sure you omit the `NetworkInterfaceID` parameter in your launch template or launch specification.

Examples

- Example 1: Launch Spot Instances using the lowest-priced Availability Zone in the Region
- Example 2: Launch Spot Instances using the lowest-priced Availability Zone from a list
- Example 3: Launch Spot Instances using the lowest-priced instance type from a list
- Example 4. Override the price for the request
- Example 5: Launch a Spot Fleet using the diversified allocation strategy
- Example 6: Launch a Spot Fleet using instance weighting
- Example 7: Launch a Spot Fleet using On-Demand capacity
- Example 8: Configure Capacity Rebalancing to launch replacement Spot Instances
- Example 9: Launch Spot Instances in a capacity-optimized fleet
- Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities

Example 1: Launch Spot Instances using the lowest-priced Availability Zone in the Region

The following example specifies a single launch specification without an Availability Zone or subnet. The Spot Fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. The price you pay does not exceed the On-Demand price.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

Example 2: Launch Spot Instances using the lowest-priced Availability Zone from a list

The following examples specify two launch specifications with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

The Spot Fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "Placement": {
              "AvailabilityZone": "us-west-2a, us-west-2b"
          },
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

Subnets

You can specify default subnets or nondefault subnets, and the nondefault subnets can be from a default VPC or a nondefault VPC. The Spot service launches the instances in whichever subnet is in the lowest-priced Availability Zone.

You can't specify different subnets from the same Availability Zone in a Spot Fleet request.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
```

```
            "InstanceType": "m3.medium",
            "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            }
        }
    ]
}
```

If the instances are launched in a default VPC, they receive a public IPv4 address by default. If the instances are launched in a nondefault VPC, they do not receive a public IPv4 address by default. Use a network interface in the launch specification to assign a public IPv4 address to instances launched in a nondefault VPC. When you specify a network interface, you must include the subnet ID and security group ID using the network interface.

```
    ...
        {
            "ImageId": "ami-1a2b3c4d",
            "KeyName": "my-key-pair",
            "InstanceType": "m3.medium",
            "NetworkInterfaces": [
                {
                    "DeviceIndex": 0,
                    "SubnetId": "subnet-1a2b3c4d",
                    "Groups": [ "sg-1a2b3c4d" ],
                    "AssociatePublicIpAddress": true
                }
            ],
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            }
        }
    ...
```

Example 3: Launch Spot Instances using the lowest-priced instance type from a list

The following examples specify two launch configurations with different instance types, but the same AMI and Availability Zone or subnet. The Spot Fleet launches the instances using the specified instance type with the lowest price.

Availability Zone

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "c5.4xlarge",
          "Placement": {
            "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "r3.8xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      }
  ]
}
```

Subnet

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
```

```
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "c5.4xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}
```

Example 4. Override the price for the request

We 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

```
{
  "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 [Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity](#).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "m3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      }
```

```
  ]
}
```

Subnet

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}
```

A best practice to increase the chance that a spot request can be fulfilled by EC2 capacity in the event of an outage in one of the Availability Zones is to diversify across zones. For this scenario, include each Availability Zone available to you in the launch specification. And, instead of using the same subnet each time, use three unique subnets (each mapping to a different zone).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
```

```
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2a"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2c"
            }
        }
    ]
}
```

Subnet

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-2a2b3c4d"
        },
        {
```

```
                "ImageId": "ami-1a2b3c4d",
                "InstanceType": "r3.2xlarge",
                "SubnetId": "subnet-3a2b3c4d"
            }
        ]
}
```

Example 6: Launch a Spot Fleet using instance weighting

The following examples use instance weighting, which means that the price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight. The Spot Fleet selects the instance type with the lowest price per unit hour. The Spot Fleet calculates the number of Spot Instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot Fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

If the `r3.2xlarge` request is successful, Spot provisions 4 of these instances. Divide 20 by 6 for a total of 3.33 instances, then round up to 4 instances.

If the `c3.xlarge` request is successful, Spot provisions 7 of these instances. Divide 20 by 3 for a total of 6.66 instances, then round up to 7 instances.

For more information, see Use instance weighting to manage cost and performance of your EC2 Fleet or Spot Fleet.

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
          "WeightedCapacity": 6
      },
      {
```

```
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 3
        }
      ]
}
```

Subnet

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 6
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 3
      }
  ]
}
```

Example 7: Launch a Spot Fleet using 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.

```
{
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "AllocationStrategy": "lowestPrice",
  "TargetCapacity": 10,
  "SpotPrice": null,
  "ValidFrom": "2018-04-04T15:58:13Z",
  "ValidUntil": "2019-04-04T15:58:13Z",
  "TerminateInstancesWithExpiration": true,
  "LaunchSpecifications": [],
  "Type": "maintain",
  "OnDemandTargetCapacity": 5,
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",
        "Version": "2"
      },
      "Overrides": [
        {
          "InstanceType": "t2.medium",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-d0dc51fb"
        }
      ]
    }
  ]
}
```

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-before-terminate`. To configure the time delay from the launch of the new replacement Spot Instances to the automatic deletion of the old Spot Instances, for `termination-delay`, specify a value in seconds. For more information, see Configuration options.

> **Note**
>
> We recommend using `launch-before-terminate` only if you can predict how long your instance shutdown procedures will take to complete. This ensures that the old instances are terminated only after the shutdown procedures are completed. You are charged for all 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 [Use Capacity Rebalancing in EC2 Fleet and Spot Fleet to replace at-risk Spot Instances](#).

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "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"
                            }
                        }
                 ]
          }
    ],
        "TargetCapacity": 5,
        "SpotMaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
            }
        }
    }
}
```

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.

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.

```
{
```

```
    "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"
                }
            ]
        }
    ]
}
```

Example 11: Launch Spot Instances in a priceCapacityOptimized fleet

The following example demonstrates how to configure a Spot Fleet with a Spot allocation strategy that optimizes for both capacity and lowest price. To optimize for capacity while taking price into consideration, you must set the Spot `AllocationStrategy` to `priceCapacityOptimized`.

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 while also choosing the pool that is the lowest priced.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "OnDemandAllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111111111111:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-0123456789example",
                    "Version": "1"
                },
                "Overrides": [
                     {
                           "InstanceType": "r4.2xlarge",
                           "AvailabilityZone": "us-west-2a"
                      },
                      {
                           "InstanceType": "m4.2xlarge",
                           "AvailabilityZone": "us-west-2b"
                      },
                      {
                           "InstanceType": "c5.2xlarge",
                           "AvailabilityZone": "us-west-2b"
                      }
                ]
            }
        ],
        "TargetCapacity": 50,
        "Type": "request"
    }
}
```

Example 12: Configure attribute-based instance type selection

The following example demonstrates how to configure a Spot Fleet to use attribute-based instance type selection for identifying instance types. To specify the required instance attributes, you specify the attributes in the `InstanceRequirements` structure.

In the following example, two instance attributes are specified:

- `VCpuCount` – A minimum of 2 vCPUs is specified. Because no maximum is specified, there is no maximum limit.
- `MemoryMiB` – A minimum of 4 MiB of memory is specified. Because no maximum is specified, there is no maximum limit.

Any instance types that have 2 or more vCPUs and 4 MiB or more of memory will be identified. However, price protection and the allocation strategy might exclude some instance types when Spot Fleet provisions the fleet.

For a list and descriptions of all the possible attributes that you can specify, see InstanceRequirements in the *Amazon EC2 API Reference*.

```
{
 "AllocationStrategy": "priceCapacityOptimized",
 "TargetCapacity": 20,
 "Type": "request",
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }]
}
```

Quotas for EC2 Fleet and Spot Fleet

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific.

The usual Amazon EC2 quotas apply to instances launched by an EC2 Fleet or a Spot Fleet, such as Spot Instance limits and volume limits.

In addition, your AWS account has the following quotas related to EC2 Fleet and Spot Fleet:

Quota description	Quota
The number of EC2 Fleets and Spot Fleets per Region of type `maintain` and `request` in the `active`, `deleted_running` , and `cancelled_running` states	1,000 [1] [2] [3]
The number of EC2 Fleets of type `instant`	Unlimited
The number of Spot capacity pools (unique combination of instance type and subnet) for EC2 Fleets and Spot Fleets of type `maintain` and `request`	300 [1]
The number of Spot capacity pools (unique combination of instance type and subnet) for EC2 Fleets of type `instant`	Unlimited
The size of the user data in a launch specification	16 KB [2]
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 [1]
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.	

[1] These quotas apply to both your EC2 Fleets and your Spot Fleets.

[2] These are hard quotas. You can't request an increase for these quotas.

[3] After you delete an EC2 Fleet or cancel a Spot Fleet request, and if you specified that the fleet should *not* terminate its Spot Instances when you deleted or canceled the request, the fleet request enters the `deleted_running` (EC2 Fleet) or `cancelled_running` (Spot Fleet) state and the instances continue to run until they are interrupted or you terminate them manually. If you terminate the instances, the fleet request enters the `deleted_terminating` (EC2 Fleet) or `cancelled_terminating` (Spot Fleet) state and does not count towards this quota. For more information, see Delete an EC2 Fleet request and the instances in the fleet and Cancel (delete) a Spot Fleet request.

Request a quota increase for target capacity

If you need more than the default quota for target capacity, you can request a quota increase.

To request a quota increase for target capacity

1. Open the Support Center Create case form.
2. Choose **Service limit increase**.
3. For **Limit type**, choose **EC2 Fleet**.
4. For **Region**, choose the AWS Region in which to request the quota increase.
5. For **Limit**, choose **Target Fleet Capacity per Fleet (in units)** or **Target Fleet Capacity per Region (in units)**, depending on which quota you want to increase.
6. For **New limit value**, enter the new quota value.
7. To request an increase for another quota, choose **Add another request**, and repeat Steps 4–6.
8. For **Use case description**, enter your reason for requesting a quota increase.
9. Under **Contact options**, specify your preferred contact language and contact method.
10. Choose **Submit**.

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

- Network maximum transmission unit (MTU) for your EC2 instance
- Virtual private clouds for your EC2 instances

Regions and Zones

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of AWS Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones.

- **Regions** are separate geographic areas.
- **Availability Zones** are multiple, isolated locations within each Region.
- **Local Zones** provide you with the ability to place resources, such as compute and storage, in multiple locations closer to your end users.
- **Wavelength Zones** provide you with the ability 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 Outposts** brings native AWS services, infrastructure, and operating models to virtually any data center, colocation space, or on-premises facility.

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.

For more information, see AWS Global Infrastructure.

Contents

Regions

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

When you launch an instance, select a Region that puts your instances close to specific customers, or that meets the legal or other requirements that you have. You can launch instances in multiple Regions.

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

Available Regions

For the list of available Regions, see AWS Regions.

Regional endpoints for Amazon EC2

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 service endpoints in the *Amazon EC2 Developer Guide*.

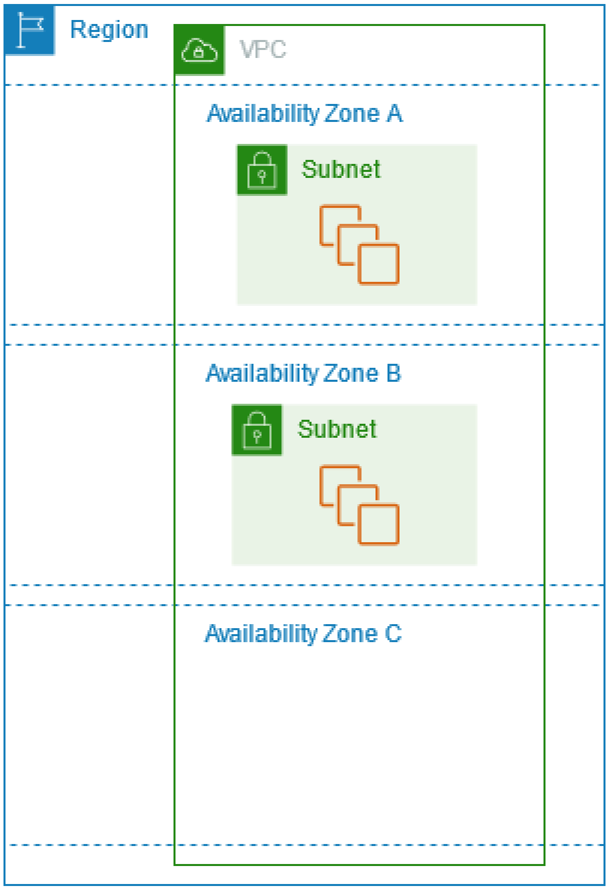
For more information, see AWS Regions in the *AWS Regions and Availability Zones User Guide*.

Availability Zones

Each Region has multiple, isolated locations known as *Availability Zones*. The code for an Availability Zone is its Region code followed by a letter identifier. For example, `us-east-1a`.

By launching EC2 instances in multiple Availability Zones, you can protect your applications from the failure of a single location in the Region.

The following diagram illustrates multiple Availability Zones in an AWS Region. Availability Zone A and Availability Zone B each have one subnet, and each subnet has EC2 instances. Availability Zone C has no subnets, therefore you can't launch instances into this Availability Zone.



For more information, see Virtual private clouds for your EC2 instances.

Availability Zones by Region

For the list of Availability Zones by Region, see AWS Availability Zones.

Instances in Availability Zones

When you launch an instance, you select a Region and a virtual private cloud (VPC). Then, you can either select a subnet from one of the Availability Zones or let us choose a subnet for you. When you launch your initial instances, we recommend that you let us select an Availability Zone for you based on system health and available capacity. If you launch additional instances, specify an Availability Zone only if your new instances must be close to, or separated from, your existing instances.

If you distribute instances across multiple Availability Zones and an instance fails, you can design your application so that an instance in another Availability Zone handles requests instead.

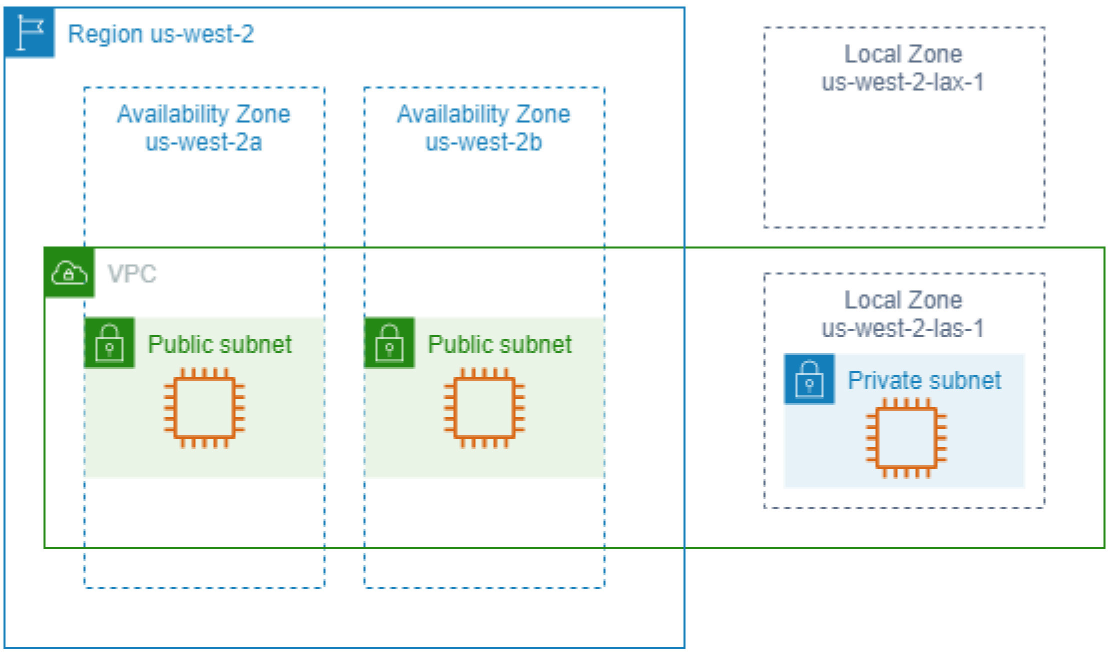
For more information, see AWS Availability Zones in the *AWS Regions and Availability Zones User Guide*.

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 [What is AWS Local Zones?] in the *AWS Local Zones User Guide*.

The code for a Local Zone is its Region code followed by an identifier that indicates its physical location. For example, `us-west-2-lax-1` in Los Angeles.

The following diagram illustrates the AWS Region `us-west-2`, two of its Availability Zones, and two of its Local Zones. The VPC spans the Availability Zones and one of the Local Zones. Each zone in the VPC has one subnet, and each subnet has an instance.



Available Local Zones

For the list of available Local Zones, see [Available Local Zones] in the *AWS Local Zones User Guide*. For the list of announced Local Zones, see [AWS Local Zones locations].

Instances in Local Zones

To use a Local Zone, you must first enable it. Then, create a subnet in the Local Zone. You can specify the Local Zone subnet when you launch instances, which places it in the Local Zone subnet in the Local Zone.

When you launch an instance 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-provided Elastic IPv4 addresses
- Amazon-provided IPv6 VPC addresses (available only in the Los Angeles zones)

For more information about how to launch an instance in a Local Zone, see [Getting started with AWS Local Zones](#) in the *AWS Local Zones User Guide*.

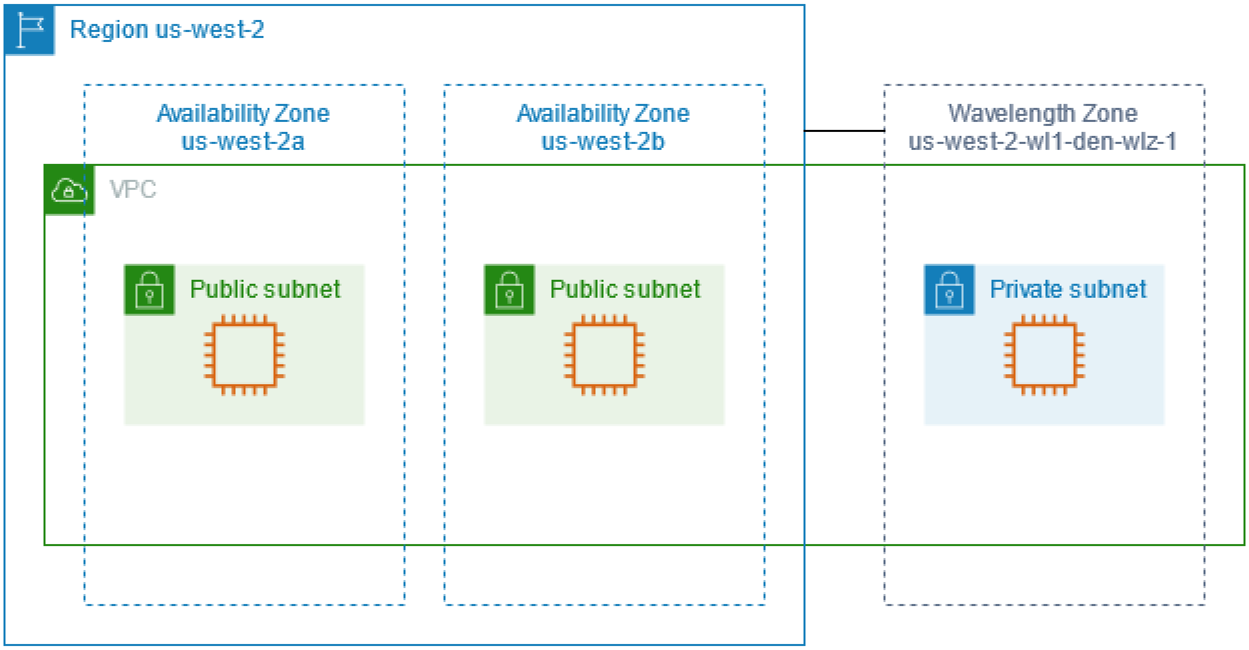
Wavelength Zones

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.

The code for a Wavelength Zone is its Region code followed by an identifier that indicates the physical location. For example, `us-east-1-wl1-bos-wlz-1` in Boston.

The following diagram illustrates the AWS Region `us-west-2`, two of its Availability Zones, and a Wavelength Zone. The VPC spans the Availability Zones and the Wavelength Zone. Each zone in the VPC has one subnet, and each subnet has an instance.



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*.

Available Wavelength Zones

For the list of available Wavelength Zones, see [Available Wavelength Zones](#) in the *AWS Wavelength Guide*.

Instances in Wavelength Zones

To use a Wavelength Zone, you must first opt in to the Zone. Then, create a subnet in the Wavelength Zone. You can specify the Wavelength subnet when you launch instances. You also allocate a carrier IP address from a network border group, which is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, `us-east-1-wl1-bos-wlz-1`.

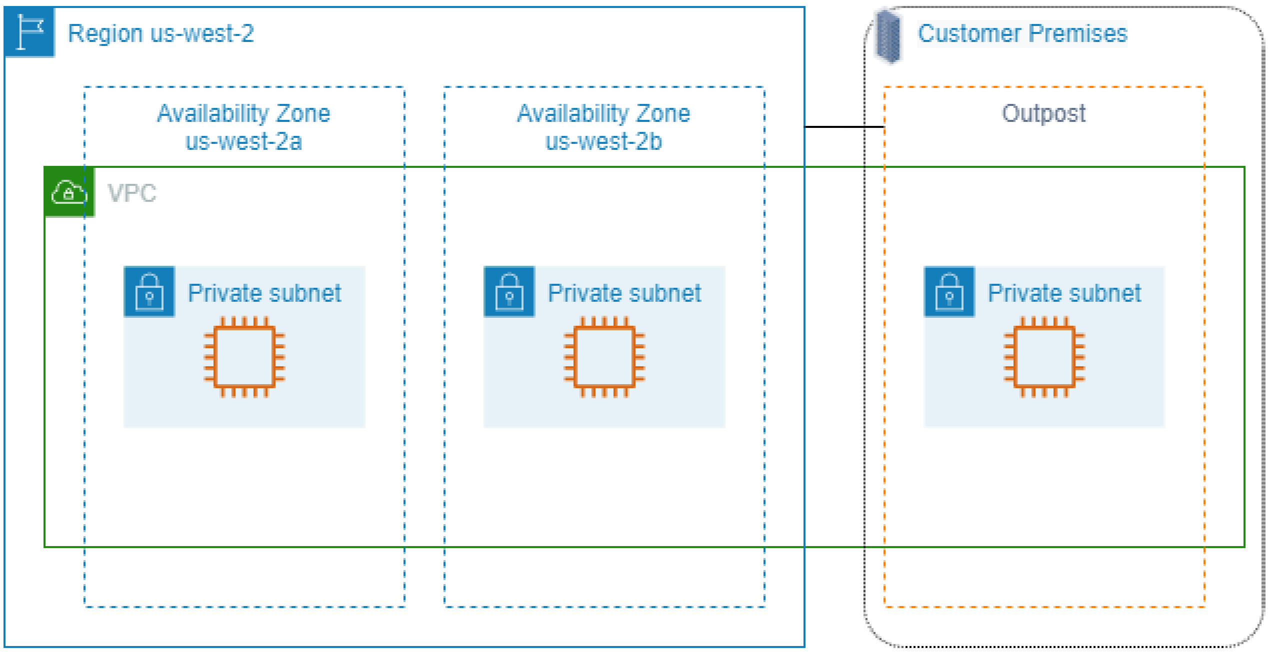
For step-by-step directions to launch an instance in a Wavelength Zone, see [Get started with AWS Wavelength](#) in the *AWS Wavelength Developer Guide*.

AWS Outposts

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

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

The following diagram illustrates the AWS Region `us-west-2`, two of its Availability Zones, and an Outpost. The VPC spans the Availability Zones and the Outpost. The Outpost is in an on-premises customer data center. Each zone in the VPC has one subnet, and each subnet has an instance.



Instances on an Outpost

To begin using AWS Outposts, you must create an Outpost and order Outpost capacity. AWS Outposts offers two form factors, Outposts racks and Outposts servers. For more information about Outposts configurations, see [AWS Outposts Family](). After your Outpost equipment is installed, the compute and storage capacity is available for you when you launch EC2 instances on your Outpost.

To launch EC2 instances, you must create an Outpost subnet. Security groups control inbound and outbound traffic for instances in an Outpost subnet, just as they do for instances in an Availability Zone subnet. To connect to instances in Outpost subnets using SSH, specify a key pair when you launch them, just as you do for instances in Availability Zone subnets.

For more information, see Get started with Outposts racks or Get started with Outposts servers.

Volumes on an Outposts rack

If your Outposts compute capacity is on an Outpost rack, 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 in the *Amazon EBS User Guide*.

We recommend that you limit the root volume for an instance on an Outpost rack to 30 GiB or smaller. You can specify data volumes in the block device mapping of the AMI or the instance to provide additional storage. To trim unused blocks from the boot volume, see How to Build Sparse EBS Volumes in the *AWS Partner Network Blog*.

We recommend that you increase the NVMe timeout for the root volume. For more information, see I/O operation timeout in the *Amazon EBS User Guide*.

Volumes on an Outposts server

Instances on Outposts servers provide instance store volumes but do not support EBS volumes. Choose an Amazon EBS-backed AMI with only a single EBS snapshot. Choose an instance size with enough instance storage to meet the needs of your application. For more information, see Instance store volume limits.

Amazon EC2 instance IP addressing

Amazon EC2 and Amazon VPC support both the IPv4 and IPv6 addressing protocols. By default, Amazon VPC uses 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 assign IPv6 addresses from that block to instances in your subnets.

When you launch an EC2 instance, you specify a VPC and a subnet. The instance receives a private IPv4 address from the CIDR range of the subnet. You can optionally configure your instances with public IPv4 addresses and IPv6 addresses. If EC2 instances in different VPCs communicate using public IP addresses, the traffic stays in the AWS private global network and does not traverse the public internet.

Contents

Private IPv4 addresses

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.

VPC subnets can be one of the following types:

- IPv4-only subnets – You can only create resources in these subnets with IPv4 addresses assigned to them.
- IPv6-only subnets – You can only create resources in these subnets with IPv6 addresses assigned to them.
- IPv4 and IPv6 subnets – You can create resources in these subnets with either IPv4 or IPv6 addresses assigned to them.

When you launch an EC2 instance into an IPv4-only or dual stack (IPv4 and IPv6) subnet, the instance receives a primary private IP address from the IPv4 address range of the subnet. For more information, see IP addressing 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 (index 0) 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.

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

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.

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:

- **Modify the public IP addressing attribute of your subnet.** For more information, see Modify the public IPv4 addressing attribute for your subnet in the *Amazon VPC User Guide*.
- **Enable or disable the public IP addressing feature during launch.** This overrides the subnet's public IP addressing attribute. For more information, see Assign a public IPv4 address at launch.
- **Unassign a public IP address from your instance after launch.** For more information, see the section called “Manage IP addresses”.

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.

We release the public IP address from your instance and assign a new one in the following cases:

- We release the public IP address when the instance is stopped, hibernated, or terminated. We assign a new public IP address when you start your stopped or hibernated instance.
- We release the public IP address when you associate an Elastic IP address with the instance. We assign a new public IP address when you disassociate the Elastic IP address from your instance.
- If we release the public IP address of your instance and it has a secondary network interface, we do not assign a new public IP address.
- If we release the public IP address of your instance and it has a secondary private IP address that is associated with an Elastic IP address, we do not assign 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.

If you are using Amazon VPC IP Address Manager (IPAM), you can get a contiguous block of public IPv4 addresses from AWS and use it to allocate Elastic IP addresses to AWS resources. Using contiguous IPv4 address blocks can significantly reduce management overhead for security access control lists and simplify IP address allocation and tracking for enterprises scaling on AWS. For more information, see Allocate sequential Elastic IP addresses from an IPAM pool in the *Amazon VPC IPAM User Guide*.

Considerations

- AWS charges for all public IPv4 addresses, including public IPv4 addresses associated with running instances and Elastic IP addresses. For more information, see the **Public IPv4 Address** tab on the Amazon VPC pricing page.
- 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.

Public IPv4 address optimization

AWS charges for all public IPv4 addresses, including public IPv4 addresses associated with running instances and Elastic IP addresses. For more information, see the **Public IPv4 Address** tab on the Amazon VPC pricing page.

The following list contains actions you can take to optimize the number of public IPv4 addresses you use:

- Use an elastic load balancer to load balance traffic to your EC2 instances and disable **Auto-assign public IP** on the primary ENI assigned to the instances. Load balancers use a single public IPv4 address, so this reduces your public IPv4 address count. You may also want consolidate existing load balancers to further reduce the public IPv4 address count.
- If the only reason for using a NAT gateway is to SSH into an EC2 instance in a private subnet for maintenance or emergencies, consider using EC2 Instance Connect Endpoint instead. With EC2 Instance Connect Endpoint, you can connect to an instance from the internet without requiring the instance to have a public IPv4 address.
- If your EC2 instances are in a public subnet with public IP addresses allocated to them, consider moving the instances to a private subnet, removing the public IP addresses, and using a public NAT gateway to allow access to and from your EC2 instances. There are cost considerations for using NAT gateways. Use this calculation method to decide if NAT gateways are cost effective. You can get the `Number of public IPv4 addresses` required for this calculation by creating an AWS Billing Cost and Usage Report.

  ```
  NAT gateway per hour + NAT gateway public IPs + NAT gateway transfer / Existing
   public IP cost
  ```

  Where:

  - `NAT gateway per hour = $0.045 * 730 hours in a month * Number of Availability Zones the NAT gateways are in`
  - `NAT gateway public IPs = $0.005 * 730 hours in a month * Number of IPs associated with your NAT gateways`
  - `NAT gateway transfer = $0.045 * Number of GBs that will go through the NAT gateway in a month`
  - `Existing public IP cost = $0.005 * 730 hours in a month * Number of public IPv4 addresses`

If the total is less than 1, NAT gateways are cheaper than public IPv4 addresses.

- Use AWS PrivateLink to connect privately to AWS services or services hosted by other AWS accounts rather than using public IPv4 addresses and internet gateways.
- Bring your own IP address range (BYOIP) to AWS and use the range for public IPv4 addresses rather than using Amazon-owned public IPv4 addresses.
- Turn off auto-assign public IPv4 address for instances launched into subnets. This option is generally disabled by default for VPCs when you create a subnet, but you should check your existing subnets to ensure it's disabled.
- If you have EC2 instances that do not need public IPv4 addresses, check that the network interfaces attached to your instances have **Auto-assign public IP** disabled.
- Configure accelerator endpoints in AWS Global Accelerator for EC2 instances in private subnets to enable internet traffic to flow directly to the endpoints in your VPCs without requiring public IP addresses. You can also bring your own addresses to AWS Global Accelerator and use your own IPv4 addresses for your accelerator's static IP addresses.

IPv6 addresses

IPv6 addresses are globally unique and can be configured to remain private or reachable over the Internet. Both public and private IPv6 addressing is available in AWS:

- **Private IPv6**: AWS considers private IPv6 addresses those that are not advertised and cannot be advertised on the Internet from AWS.
- **Public IPv6**: AWS considers public IPv6 addresses those that are advertised on the Internet from AWS.

For more information about public and private IPv6 addresses, see IPv6 addresses in the *Amazon VPC User Guide*.

All instance types support IPv6 addresses except for the following: C1, M1, M2, M3, and T1.

Your EC2 instances receive 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 Modify the IP addressing attributes of 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 (index 0) of the instance. You can manage the IPv6 addresses for your instances primary network interface as follows:

- Assign and unassign IPv6 addresses from the network interface. 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 Maximum IP addresses per network interface.
- Enable a primary IPv6 address. A primary IPv6 address enables you to avoid disrupting traffic to instances or ENIs. For more information, see Create a network interface for your EC2 instance or Manage the IP addresses for your network interface.

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 control whether instances are reachable via their IPv6 addresses by controlling the routing for your subnet or by using security group and network ACL rules. For more information, see Internetwork traffic privacy in the *Amazon VPC User Guide*.

For more information about reserved IPv6 address ranges, see IANA IPv6 Special-Purpose Address Registry and RFC4291.

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 Maximum IP addresses per network interface.

Use cases

- 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.

How multiple IP addresses work

- You can assign a secondary private IPv4 address to any network interface.
- 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.

For more information, see the section called “Secondary IP addresses”.

EC2 instance hostnames

When you create an EC2 instance, AWS creates a hostname for that instance. For more information on the types of hostnames and how they're provisioned by AWS, see EC2 instance hostnames and domains. Amazon provides a DNS server that resolves Amazon-provided hostnames to IPv4 and IPv6 addresses. The Amazon DNS server is located at the base of your VPC network range plus two. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*.

Link-local addresses

Link-local addresses are well-known, non-routable IP addresses. Amazon EC2 uses addresses from the link-local address space to provide services that are accessible only from an EC2 instance. These services do not run on the instance, they run on the underlying host. When you access the link-local addresses for these services, you're communicating with either the Xen hypervisor or the Nitro controller.

Link-local address ranges

- IPv4 – 169.254.0.0/16 (169.254.0.0 to 169.254.255.255)
- IPv6 – fe80::/10

Services that you access using link-local addresses

- Instance Metadata Service
- Amazon Route 53 Resolver (also known as the Amazon DNS server)
- Amazon Time Sync Service
- AWS KMS servers

Manage the IPv4 addresses for your EC2 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.

Tasks

- Assign a public IPv4 address at launch

- Assign a private IPv4 address at launch
- View the primary IPv4 address
- View the IPv4 addresses using instance metadata

Assign a public IPv4 address at 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 0. This feature depends on certain conditions at the time you launch your instance.

Considerations

- You can unassign the public IP address from your instance after launch by managing the IP addresses associated with a network interface. For more information about public IPv4 addresses, see Public IPv4 addresses.
- You can't 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 device index 0.
- 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. You can also modify your subnet's public IPv4 addressing behavior. For more information, see Modify the public IPv4 addressing attribute for your subnet.

Console

To assign a public IPv4 address at launch

Follow the procedure to launch an instance, and when you configure Network Settings, choose the option to **Auto-assign Public IP**.

AWS CLI

To assign a public IPv4 address at launch

Use the run-instances command with the `--associate-public-ip-address` option.

```
--associate-public-ip-address
```

PowerShell

To assign a public IPv4 address at launch

Use the New-EC2Instance cmdlet with the `-AssociatePublicIp` parameter.

```
-AssociatePublicIp $true
```

Assign a private IPv4 address at launch

You can specify a private IPv4 address from the IPv4 address range of the subnet, or let Amazon EC2 chose one for you. This address is assigned to the primary network interface.

To assign IPv4 addresses after launch, see the section called “Assign secondary IP addresses to an instance”.

Console

To assign a private IPv4 address at launch

Follow the procedure to launch an instance. When you configure Network Settings, expand **Advanced network configuration** and enter a value for **Primary IP**.

AWS CLI

To assign a private IPv4 address at launch

Use the run-instances command with the `--private-ip-address` option.

```
--private-ip-addresses 10.251.50.12
```

To let Amazon EC2 choose the IP address, omit this option.

PowerShell

To assign a private IPv4 address at launch

Use the New-EC2Instance cmdlet with the `-PrivateIpAddress` parameter.

```
-PrivateIpAddress 10.251.50.12
```

To let Amazon EC2 choose the IP address, omit this parameter.

View the primary IPv4 address

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.

Console

To view the IPv4 addresses 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.
4. On the **Networking** tab, find **Public IPv4 address** and **Private IPv4 addresses**.
5. (Optional) The **Networking** tab also contains the network interfaces and Elastic IP addresses for the instance.

AWS CLI

To view the primary IPv4 address for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].PrivateIpAddress" \
    --output text
```

The following is example output.

```
10.251.50.12
```

PowerShell

To view the primary IPv4 address for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.PrivateIpAddress
```

The following is example output.

```
10.251.50.12
```

View the IPv4 addresses using instance metadata

You can get the IPv4 addresses for your instance by retrieving instance metadata. For more information, see Use instance metadata to manage your EC2 instance.

To view the IPv4 addresses using instance metadata

1. Connect to your instance. For more information, see Connect to your EC2 instance.
2. Run one of the following commands.

   IMDSv2

   Linux

   Run the following command from your Linux instance.

   ```
   TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
   metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
   meta-data/local-ipv4
   ```

   Windows

   Run the following command from your Windows instance.

   ```
   [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
   seconds" = "21600"} `
   ```

```
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

Linux

Run the following command from your Linux instance.

```
curl http://169.254.169.254/latest/meta-data/local-ipv4
```

Windows

Run the following command from your Windows instance.

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use one of the following commands to access the public IP address. If there is an Elastic IP address associated with the instance, the command returns the Elastic IP address.

IMDSv2

Linux

Run the following command from your Linux instance.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
 "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/public-ipv4
```

Windows

Run the following command from your Windows instance.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/public-ipv4
```

IMDSv1

Linux

Run the following command from your Linux instance.

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

Windows

Run the following command from your Windows instance.

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/public-ipv4
```

Manage the IPv6 addresses for your EC2 instances

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. You can access the IPv6 addresses for your instances in the console on either the **Instances** page or the **Network Interfaces** page. The following tasks configure IP addresses for your instances. To configure IP addresses for your network interfaces instead, see the section called "Manage IP addresses".

Tasks

Assign an IPv6 address to an instance

You can specify an IPv6 address from the IPv6 address range of the subnet, or let Amazon EC2 choose one for you. This address is assigned to the primary network interface. Note that the following instance types do not support IPv6 addresses: C1, M1, M2, M3, and T1.

Console

To assign an IPv6 address at launch

Follow the procedure to launch an instance. When you configure Network Settings, choose the option to **Auto-assign IPv6 IP**. If you don't see this option, the selected subnet does not have an associated IPv6 CIDR block.

To assign an IPv6 address 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**.
5. Enter an IPv6 address from the range of the subnet, or leave the field blank to let Amazon EC2 choose the IPv6 address for you. If you don't see this option, the instance subnet does not have an associated IPv6 CIDR block.
6. Choose **Save**.

AWS CLI

To assign an IPv6 address at launch

Use the run-instances command with the `--ipv6-addresses` option. The following example assigns two IPv6 addresses.

```
--ipv6-addresses Ipv6Address=2001:db8::1234:5678:1.2.3.4
 Ipv6Address=2001:db8::1234:5678:5.6.7.8
```

To let Amazon EC2 choose the IPv6 addresses, use the `--ipv6-address-count` option instead. The following example assigns two IPv6 addresses.

```
--ipv6-address-count 2
```

To assign an IPv6 address after launch

Use the assign-ipv6-addresses command. The following example assigns two IPv6 addresses.

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 2001:db8::1234:5678:5.6.7.8
```

To let Amazon EC2 choose the IPv6 addresses, use the `--ipv6-address-count` option instead. The following example assigns two IPv6 addresses.

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-address-count 2
```

PowerShell

To assign an IPv6 address at launch

Use the New-EC2Instance cmdlet with the `-Ipv6Address` parameter. The following example assigns two IPv6 addresses.

```
-Ipv6Address $ipv6addr1,$ipv6addr2
```

Define the IPv6 addresses as follows.

```
$ipv6addr1 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
$ipv6addr2 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr2.Ipv6Address = "2001:db8::1234:5678:5.6.7.8"
```

To let Amazon EC2 choose the IPv6 addresses, use the `-Ipv6AddressCount` parameter instead. The following example assigns two IPv6 addresses.

```
-Ipv6AddressCount 2
```

To assign an IPv6 address after launch

Use the Register-EC2Ipv6AddressList cmdlet. The following example assigns two IPv6 addresses.

```
Register-EC2Ipv6AddressList `
```

```
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address "2001:db8::1234:5678:1.2.3.4","2001:db8::1234:5678:5.6.7.8"
```

To let Amazon EC2 choose the IPv6 addresses, use the `-Ipv6AddressCount` parameter instead. The following example assigns two IPv6 addresses.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6AddressCount 2
```

View the IPv6 addresses for an instance

You can view the IPv6 addresses for your instances.

Console

To view the IPv6 addresses 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.
4. On the **Networking** tab, locate **IPv6 addresses**.

AWS CLI

To view the IPv6 address for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].Ipv6Address" \
    --output text
```

The following is example output.

```
2001:db8::1234:5678:1.2.3.4
```

PowerShell

To view the IPv6 address for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.Ipv6Address
```

The following is example output.

```
2001:db8::1234:5678:1.2.3.4
```

View IPv6 addresses using instance metadata

After you connect to your instance, you can retrieve the IPv6 addresses using instance metadata. First, you must get the MAC address of the instance from `http://169.254.169.254/latest/meta-data/network/interfaces/macs/`.

IMDSv2

Linux

Run the following command from your Linux instance.

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Windows

Run the following cmdlets from your Windows instance.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
```

```
    -Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/mac-address/ipv6s
```

IMDSv1

Linux

Run the following command from your Linux instance.

```
curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/
ipv6s
```

Windows

Run the following cmdlet from your Windows instance.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/mac-address/ipv6s
```

Unassign an IPv6 address from an instance

You can unassign an IPv6 address from an instance at any time.

Console

To unassign an IPv6 address from an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance, 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**.

AWS CLI

To unassign an IPv6 address from an instance

Use the unassign-ipv6-addresses command.

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4
```

PowerShell

To unassign an IPv6 address from an instance

Use the Unregister-EC2Ipv6AddressList cmdlet.

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8::1234:5678:1.2.3.4
```

Secondary IP addresses for your EC2 instances

The first IPv4 address assigned to a network interface is known as the primary IP address. Secondary IP addresses are additional IPv4 address assigned to a network interface. For more information, see the section called "Multiple IP addresses".

You can also assign multiple IPv6 addresses to an instance. For more information, see the section called "IPv6 addresses".

Tasks

Assign secondary IP addresses to an instance

You can assign secondary IP addresses to the network interface for an instance as you launch the instance, or after the instance is running.

Console

To assign a secondary IP address at launch

1. Follow the procedure to launch an instance. When you configure Network Settings, expand **Advanced network configuration**.

2. For **Secondary IP**, choose **Automatically assign** and enter the number of IP addresses for Amazon EC2 to assign. Alternatively, choose **Manually assign** and enter the IPv4 addresses.
3. Complete the remaining steps to launch the instance.

To assign a secondary IP address 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.
5. To add an IPv4 address, under **IPv4 addresses**, choose **Assign new IP address**. Enter an IPv4 address from the range of the subnet, or leave the field blank to let Amazon EC2 choose one for you.
6. Choose **Save**.

AWS CLI

To assign a secondary IP address at launch

Use the run-instances command with the `--secondary-private-ip-addresses` option.

```
--secondary-private-ip-addresses 10.251.50.12
```

To let Amazon EC2 choose the IP address, use the `--secondary-private-ip-address-count` option instead. The following example assigns one secondary IP address.

```
--secondary-private-ip-address-count 1
```

Alternatively, you can create a network interface. For more information, see the section called "Create a network interface".

To assign a secondary IP address after launch

Use the assign-private-ip-addresses command with the `--private-ip-addresses` option.

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
```

```
    --private-ip-addresses 10.251.50.12
```

To let Amazon EC2 choose the IPv4 address, use the `--secondary-private-ip-address-count` parameter instead. The following example assigns one IPv4 address.

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --secondary-private-ip-address-count 1
```

PowerShell

To assign a secondary IP address at launch

You must create a network interface. For more information, see the section called "Create a network interface".

To assign a secondary IP address after launch

Use the Register-EC2PrivateIpAddress cmdlet with the `-PrivateIpAddress` parameter.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

To let Amazon EC2 choose the IPv4 addresses, use the `-SecondaryPrivateIpAddressCount` parameter instead. The following example assigns one IPv4 address.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SecondaryPrivateIpAddressCount 1
```

Configure the operating system to use secondary IP addresses

After you assign a secondary IP address to your instance, you must configure the operating system on your instance to recognize the additional private IPv4 address.

Linux instances

- If you are using Amazon Linux, the ec2-net-utils package can take care of this step for you. It configures additional network interfaces that you attach while the instance is running, refreshes secondary IPv4 addresses during DHCP lease renewal, and updates the related routing rules. You can immediately refresh the list of interfaces by using one of the following commands, depending on your system: `sudo systemctl restart systemd-networkd` (AL2023) or `sudo service network restart` (Amazon Linux 2). You can view the up-to-date list using the following command: `ip addr li`. If you require manual control over your network configuration, you can remove the ec2-net-utils package. For more information, see Configure your network interface using ec2-net-utils.

- If you are using another Linux distribution, see the documentation for your Linux distribution. Search for information about configuring additional network interfaces and secondary IPv4 addresses. If the instance has two or more interfaces on the same subnet, search for information about using routing rules to work around asymmetric routing.

Windows instances

For more information, see Configure secondary private IPv4 addresses for Windows instances.

Unassign a secondary IP address from an instance

If you no longer require a secondary IP 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.

Console

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. Expand the network interface. For **IPv4 addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Save**.

AWS CLI

To unassign a secondary private IP address

Use the unassign-private-ip-addresses command.

```
aws ec2 unassign-private-ip-addresses \
    --network-interface eni-1234567890abcdef0\
    --private-ip-addresses 10.251.50.12
```

PowerShell

To unassign a secondary private IP address

Use the Unregister-EC2PrivateIpAddress cmdlet.

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterface eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

Configure secondary private IPv4 addresses for Windows instances

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.

> **Note**
>
> These instructions are based on Windows Server 2022. The implementation of these steps might vary based on the operating system of the Windows instance.

Tasks

- Prerequisites
- Step 1: Configure static IP addressing in your instance
- Step 2: Configure a secondary private IP address for your instance
- Step 3: Configure applications to Use the secondary private IP address

Prerequisites

- 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 secondary IP addresses to an instance.

Step 1: Configure static IP addressing in 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 in 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:
   - Run the following command in PowerShell:

     ```
     ipconfig /all
     ```

     Review the output and note the **IPv4 Address**, **Subnet Mask**, **Default Gateway**, and **DNS Servers** values for the network interface. Your output should resemble the following example:

     ```
     ...

     Ethernet adapter Ethernet 4:

        Connection-specific DNS Suffix  . : us-west-2.compute.internal
        Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
        Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
        DHCP Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
        IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Lease Obtained. . . . . . . . . . : Monday, April 8, 2024 12:19:29 PM
        Lease Expires . . . . . . . . . . : Monday, April 8, 2024 4:49:30 PM
     ```

```
   Default Gateway . . . . . . . . . : 10.200.0.1
   DHCP Server . . . . . . . . . . . : 10.200.0.1
   DHCPv6 IAID . . . . . . . . . . . : 151166011
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-
E7
   DNS Servers . . . . . . . . . . . : 10.200.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

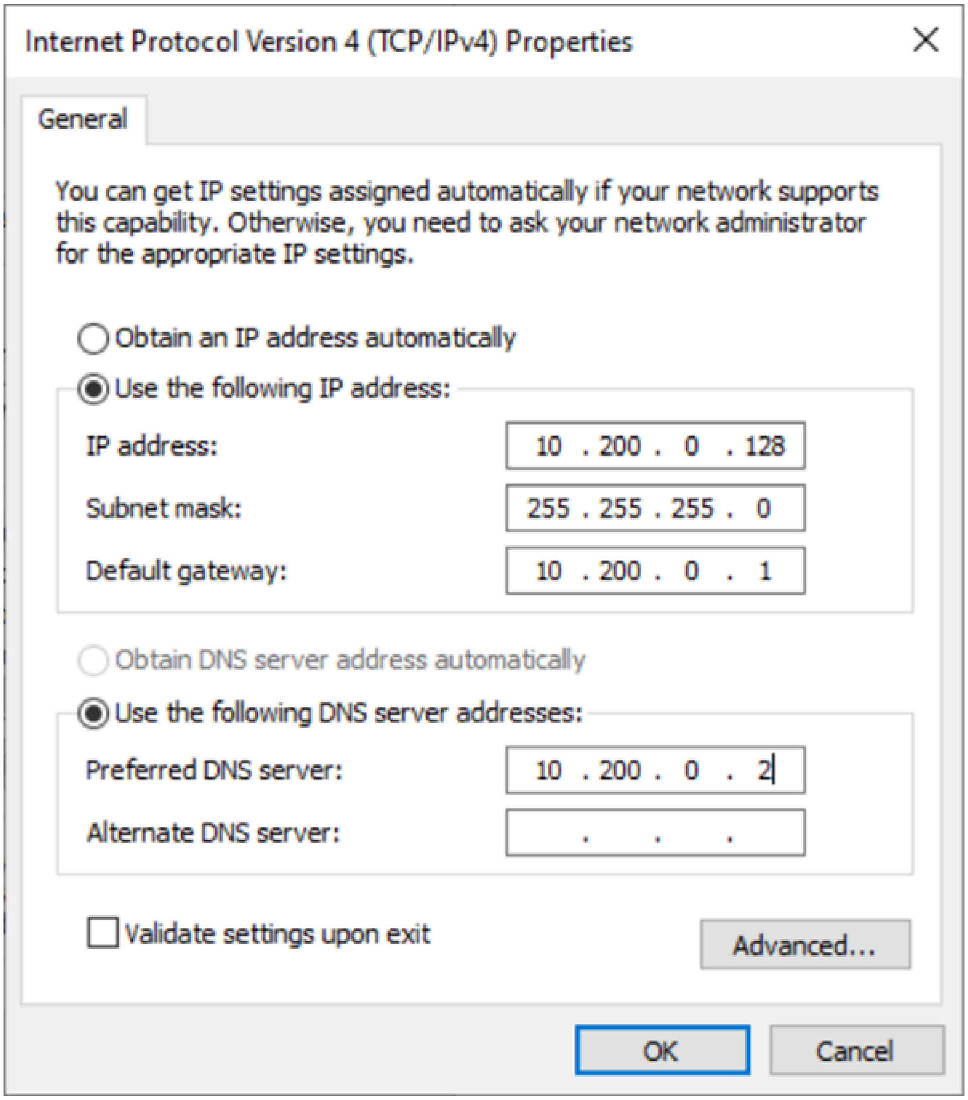
3. Open the **Network and Sharing Center** by running the following command in PowerShell:

```
& $env:SystemRoot\system32\control.exe ncpa.cpl
```

4. Open the context (right-click) menu for the network interface (Local Area Connection or Ethernet) and choose **Properties**.
5. Choose **Internet Protocol Version 4 (TCP/IPv4)**, **Properties**.
6. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, choose **Use the following IP address**, enter the following values, and then choose **OK**.

Field	Value
IP address	The IPv4 address obtained in step 2 above.
Subnet mask	The subnet mask obtained in step 2 above.
Default gateway	The default gateway address obtained in step 2 above.
Preferred DNS server	The DNS server obtained in step 2 above.
Alternate DNS server	The alternate DNS server obtained in step 2 above. If an alternate DNS server was not listed, leave this field blank.

> **Important**
>
> If you set the IP address to any value other than the current IP address, you will lose connectivity to the instance.



You will lose RDP connectivity to the Windows instance for a few seconds while the instance converts from using DHCP to static addressing. The instance retains the same IP address information as before, but now this information is static and not managed by DHCP.

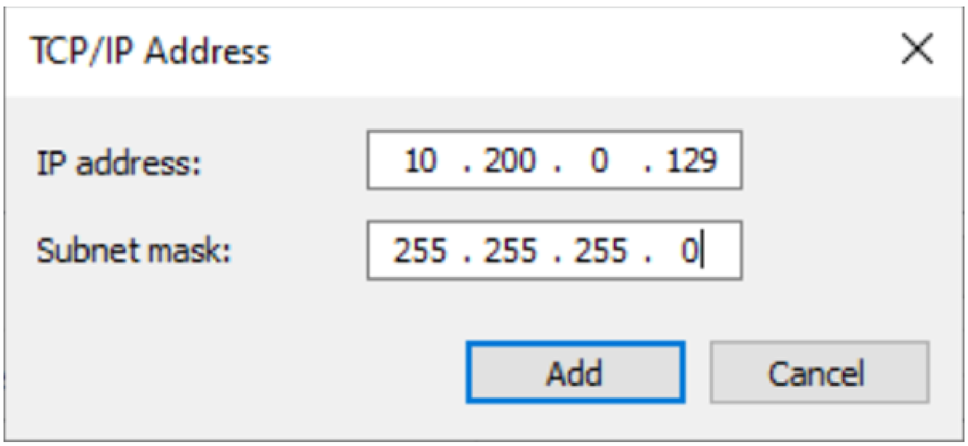
Step 2: Configure a secondary private IP address for your instance

After you have set up static IP addressing on your Windows instance, you are ready to prepare a second private IP address.

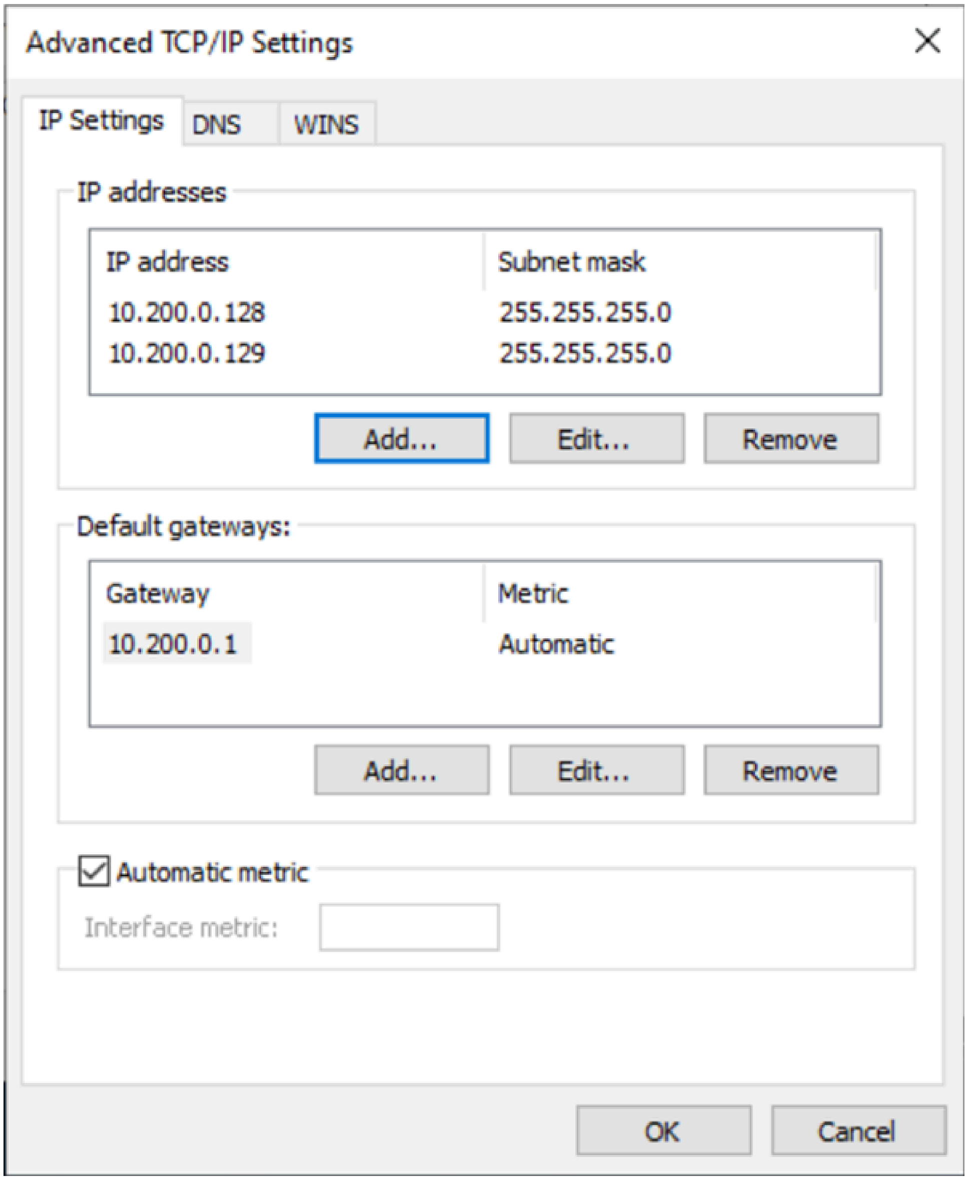
To configure a secondary IP address

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

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 or Ethernet) 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 in your instance](), and then choose **Add**.



11. Verify the IP address settings and choose **OK**.



12. Choose **OK**, **Close**.
13. To confirm that the secondary IP address has been added to the operating system, run the `ipconfig /all` command in PowerShell. Your output should resemble the following:

```
Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
   Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
```

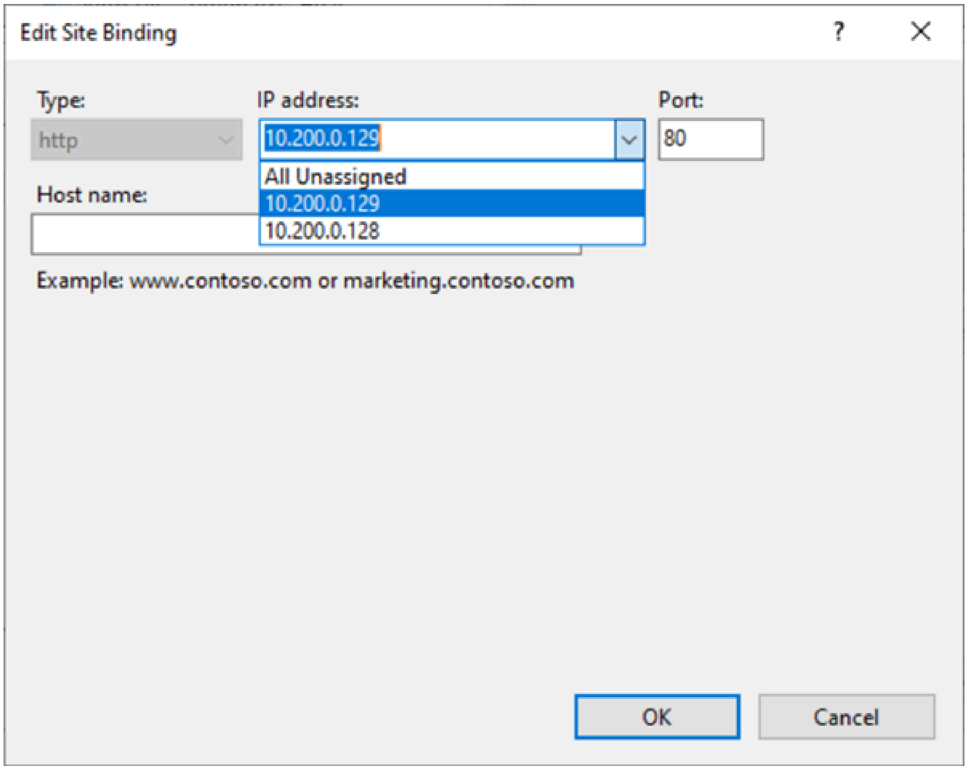
```
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.200.0.129(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.0.1
   DHCPv6 IAID . . . . . . . . . . . : 151166011
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-E7
   DNS Servers . . . . . . . . . . . : 10.200.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Step 3: Configure applications to Use the secondary private IP address

You can configure any applications to use the secondary private IP address. For example, if your instance is running a website on IIS, you can configure IIS to use the secondary private IP address.

To configure IIS to use the secondary private IP address

1. Connect to your instance.
2. Open Internet Information Services (IIS) Manager.
3. In the **Connections** pane, expand **Sites**.
4. Open the context (right-click) menu for your website and choose **Edit Bindings**.
5. In the **Site Bindings** dialog box, for **Type**, choose **http**, **Edit**.
6. In the **Edit Site Binding** dialog box, for **IP address**, select the secondary private IP address. (By default, each website accepts HTTP requests from all IP addresses.)



7. Choose **OK**, **Close**.

EC2 instance hostnames and domains

Understanding EC2 instance hostnames and domains is important for effectively managing and accessing your Amazon EC2 instances. Each EC2 instance can have different types of hostnames - private and public - that serve different purposes and follow specific naming conventions.

This topic explains the structure of EC2 instance hostnames, how they're constructed, and the different hostname types available. You'll learn how to view and modify hostname settings, understand when to use each type, and learn best practices for hostname management in your AWS environment.

Contents

Understanding EC2 instance hostnames and domains

A EC2 instance address is made up of different components. The following is an example of an EC2 instance address that uses the private IPv4 address of the instance:

```
   IP address         Domain name
   #--------# #-----------------------#
ip-10-24-34-0.us-west-2.compute.internal
#-----------#
  Hostname
#---------------------------------------#
    Fully qualified domain name (FQDN)
```

Where:

- **IP address**: The primary IPv4 address of the primary network interface associated with an instance.
- **Hostname**: The local name of a specific EC2 instance (used by the operating system and for local network identification)
- **Domain name**: The part of the FQDN that AWS provides
- **Fully qualified domain name (FQDN)**: The complete address that includes both the hostname and the domain name. This is the full, globally unique identifier used to reach your instance across networks.

Depending on the hostname type you choose for the instance or primary network interface attached to the instance, the hostname and domain name formats will be different from the example above. This section explains the hostname type options.

Hostname types

AWS provides two types of hostnames: **private** and **public**. The following table compares the key differences between private and public hostnames, including how they resolve, how they're configured, and when to use each type.

	Private hostnames	Public hostnames
DNS resolution	Private hostnames enable private FQDNs that are not accessible	Public hostnames enable public FQDNs that are accessible from the

	Private hostnames	Public hostnames
	from the public internet. Private hostnames only allow requests to resolve to private IPv4 and IPv6 GUA addresses within the VPC.	public internet. Public hostnames enable requests to resolve to private IPv4 and IPv6 GUA within the VPC and public IPs from the internet (split-horizon DNS).
Configuration	Private hostnames are configured at the instance level.	Public hostnames are configured at the network interface level.
When to use	• For hostnames that resolve only within your VPC, for internal communication • To keep your internal naming scheme private • To simplify internal resource management without external DNS considerations	• To access instances over both IPv4 and IPv6 from the internet • To enable smooth migration from IPv4 to IPv6 environments

Contents

Public hostnames

You can use *public hostnames* to access EC2 instances using hostnames that resolve to the public IPv4 or IPv6 address of the instance and ease your transition to IPv6.

Public hostnames allow you to:

- Access your EC2 instances over both IPv4 and IPv6, giving you more flexibility in how you connect to your resources.
- Migrate from IPv4 to IPv6 environments at your own pace. You can, for example, decouple your database and application migrations, reducing complexity and risk.

- Use multiple hostname options (IPv4-only, IPv6-only, and dual-stack) that automatically resolve to the appropriate IP addresses.
- Benefit from improved security with split-horizon DNS, which resolves hostnames to private IP addresses when queried from within your VPC.

Contents

Public hostname types and when to use them

To use public hostnames, you have to modify an existing network interface. This section describes the three public hostname type options and helps you decide which to use:

- **Dual-stack - IP based name**
  - This is the best option if you are migrating or planning to migrate from IPv4 to IPv6. This option allows for connectivity over both IPv4 and IPv6, giving flexibility to clients that may be using either protocol and enables clients to keep the same hostname throughout the migration to IPv6.
  - Requests from within the VPC resolve to both the private IPv4 address and the IPv6 Global Unicast Address (GUA) of the network interface. Requests from the internet resolve to both the public IPv4 and the IPv6 GUA address of the network interface.
  - **Example**
    - When you choose this option, a dual-stack FQDN will be generated for this network interface. This is an example of the FQDN that will be generated:
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO.ap-southeast-2.ip.aws
    - Where:
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO is the hostname that is a base36 representation of the primary public IPv6 address (f5lnz-0khrm-nt2u3-gyqqt-nbdl5) on the network interface along with a base36 representation of the primary public IPv4 address (q3cdpO) on the network interface.

        - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 is resolved by the Amazon DNS resolver to the IPv6 GUA address FFFF:1407:4:f000:81d:2689:1066:4489. This is the first IPv6 GUA assigned to the network interface.
        - q3cdpO is resolved to the IPv4 address of 52.54.55.56. This is the public IPv4 address attached to the primary network interface.
        - ap-southeast-2 is the Region of the subnet that the network interface is in.
        - ip.aws is the domain provided by AWS.
- **IPv6 - IP based name**
    - This is a good option if you have already migrated to IPv6 and require connectivity only over IPv6.
    - Requests from within the VPC or from the internet resolve to the IPv6 GUA of the network interface.
    - **Example**
        - When you choose this option, a FQDN will be generated for this network interface. This is an example of the FQDN that will be generated:
            - f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
        - Where:
            - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 is a hostname that is a base36 representation of the primary public IPv6 address on the network interface.
            - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 is resolved by the Amazon DNS resolver to the IPv6 GUA address FFFF:1407:4:f000:81d:2689:1066:4489. This is the first IPv6 GUA assigned to the network interface.
            - ap-southeast-2 is the Region of the subnet that the network interface is in.
            - ip.aws is the domain provided by AWS.
- **IPv4 - IP based name**
    - This is a good option if the instance using this network interface need to maintain IPv4 access during the transition to IPv6 or if applications or systems running on the instance only support IPv4. This is the best option if you only need to maintain IPv4 connectivity and your workloads don't require IPv6 support. For example, if you are migrating to IPv6, you may decide to keep some applications on IPv4 while others move to IPv6.
    - Requests from within the VPC resolve to the private primary IPv4 address of the network interface. Requests from the internet resolve to the public IPv4 address of the network interface.

- **Example**
  - If you choose this option, an IPv4-enabled public hostname will be generated for this network interface. This is an example of the DNS name that will be generated:
    - ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
  - Where:
    - ec2-52-54-55-66 is a hostname that is a base36 representation of the primary public IPv4 address of a network interface.
    - ec2-52-54-55-66 resolves to the IPv4 address of 52.54.55.56. This is the public IPv4 address attached to the primary network interface.
    - ap-southeast-2 is the Region of the subnet that the network interface is in.
    - ip.aws is the domain provided by AWS.

> **Important**
>
> In the examples above, you can see that IP addresses are used to generate the hostname. If you change the primary private IPv4 address or the first IPv6 GUA assigned to the network interface, the portion of the hostname that translates to the IP address will change and **the previously-generated public hostname will no longer be valid**. In addition, changing the primary IPv4 public address forces a downstream refresh of Instance Metadata Service (IMDS) in the *Amazon EC2 User Guide*, so the EC2 instance IMDS metadata is automatically updated.

View public hostnames

If the VPC that the network interface is in does not have both EnableDnsHostnames and EnableDnsSupport enabled, there is no hostname type defined or generated.

Console

You can view the public hostnames for an instance or primary network interface.

To view the hostname type and DNS names of 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 checkbox for the instance.
4. On the **Network** tab, under **Hostname and DNS**, find the following:
   - **Public hostname type**
   - **Public DNS**
   - **IPv4-only IP based name**
   - **IPv6-only - IP based name**
   - **Dualstack - IP based name**

To view the hostname type and DNS names of 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. In the search field, enter the ID of the instance. Select the ID of the network interface to open its details page.
4. Under **Hostname and DNS**, find the following:
   - **Public hostname type**
   - **Public DNS name**
   - **Public IPv4 DNS name**
   - **Public IPv6 DNS name**
   - **Public Dualstack DNS name**

AWS CLI

To view the hostname type and DNS names of a network interface

Use the describe-network-interfaces command.

```
aws ec2 describe-network-interfaces \
    --network-interface-id eni-1234567890abcdef0 \
    --query NetworkInterfaces[].PublicIpDnsNameOptions
```

The following is example output. Because the hostname type is `public-dual-stack-dns-name`, the DNS hostname is the same as `PublicDualStackDnsName`.

```
[
     {
          "DnsHostnameType": "public-dual-stack-dns-name",
          "PublicIpv4DnsName": "ec2-52-54-55-66.ap-
southeast-2.compute.amazonaws.com",
          "PublicIpv6DnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-
southeast-2.ip.aws",
          "PublicDualStackDnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-
southeast-2.ip.aws"
     }
]
```

PowerShell

To view the hostname type and DNS names of a network interface

Use the Get-EC2NetworkInterface cmdlet.

```
(Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0).PublicIpDnsNameOptions
```

The following is example output. Because the hostname type is `public-dual-stack-dns-name`, the DNS hostname is the same as `PublicDualStackDnsName`.

```
DnsHostnameType        : public-dual-stack-dns-name
PublicDualStackDnsName : f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-southeast-2.ip.aws
PublicIpv4DnsName      : ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
PublicIpv6DnsName      : f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
```

Modify public hostnames types

The public hostname type options depend on the IP addresses associated with the network interface:

- If the network interface has only a public IPv4 address, the hostname type must be **IPv4 - IP based name**.
- If the network interface has only an IPv6 address, the hostname type must be **IPv6 - IP based name**.
- If the network interface has both a public IPv4 address and an IPv6 address, the hostname type can be **Dual-stack - IP based name**.

Prerequisites

- The network interface must have an associated public IPv4 address or an IPv6 address.
- The VPC that the network interface is in must have EnableDnsHostnames and EnableDnsSupport enabled. See [View and update DNS attributes for your VPC](#) in the *Amazon VPC User Guide*.

Console

To modify the public hostname type

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. In the search field, enter the ID of the instance. Select the checkbox for the network interface.

   Alternatively, from the instance detail page, choose the **Networking** tab and select the ID of the network interface for device index 0.
4. Choose **Actions**, **Modify public hostname type**.
5. Choose an option:
   - **Dual-stack - IP based name**: A dual-stack public hostname for a network interface. Requests from within the VPC resolve to both the private IPv4 address and the IPv6 Global Unicast Address of the network interface. Requests from the internet resolve to both the public IPv4 and the IPv6 GUA address of the network interface.
   - **IPv4 - IP based name**: An IPv4-enabled public hostname for a network interface. Requests from within the VPC resolve to the private primary IPv4 address of the network interface. Requests from the internet resolve to the public IPv4 address of the network interface.
   - **IPv6 - IP based name**: An IPv6-enabled public hostname for a network interface. Requests from within the VPC or from the internet resolve to the IPv6 GUA of the network interface.
6. Choose **Modify**.

AWS CLI

To modify the public hostname type

Use the modify-public-ip-dns-name-options command.

```
aws ec2 modify-public-ip-dns-name-options \
    --network-interface-id eni-1234567890abcdef0 \
    --hostname-type public-dual-stack-dns-name
```

The following is example output.

```
{
    "Successful": true
}
```

PowerShell

To modify the public hostname type

Use the Edit-EC2PublicIpDnsNameOption cmdlet.

```
Edit-EC2PublicIpDnsNameOption `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -HostNameType public-dual-stack-dns-name
```

Private hostnames

This section describes the Amazon EC2 instance private hostnames available when you launch instances into your VPC subnets.

The private hostname distinguishes the EC2 instances on your network. You may use the private hostname of an instance if, for example, you want to run scripts to communicate with some or all of the instances on your network.

Contents

Private hostname types

There are two private hostname types for the guest OS hostname when EC2 instances are launched in a VPC:

- **IP name**: The legacy naming scheme where, when you launch an instance, the *private IPv4 address* of the instance is included in the hostname of the instance. The IP name exists for the life of the EC2 instance. When used as the Private DNS hostname, it will only return the private IPv4 address (A record).
- **Resource name**: When you launch an instance, the *EC2 instance ID* is included in the hostname of the instance. The resource name exists for the life of the EC2 instance. When used as the Private DNS hostname, it can return both the private IPv4 address (A record) and/or the IPv6 Global Unicast Address (AAAA record).

The EC2 instance guest OS hostname type depends on the subnet settings:

- If the instance is launched into an IPv4-only subnet, you can select either IP name or resource name.
- If the instance is launched into a dual-stack (IPv4+IPv6) subnet, you can select either IP name or resource name.
- If the instance is launched into an IPv6-only subnet, resource name is used automatically.

Contents

IP name

When you launch an EC2 instance with the **Hostname type** of **IP name**, the guest OS hostname is configured to use the private IPv4 address.

- Format for an instance in us-east-1: `*private-ipv4-address*.ec2.internal`
- Example: `*ip-10-24-34-0*.ec2.internal`
- Format for an instance in any other AWS Region: `*private-ipv4-address.region*.compute.internal`

- Example: *ip-10-24-34-0.us-west-2*.compute.internal

Resource name

When you launch EC2 instances in IPv6-only subnets, the **Hostname type** of **Resource name** is selected by default. When you launch an instance in IPv4-only or dual-stack (IPv4+IPv6) subnets, **Resource name** is an option that you can select. After you launch an instance, you can manage the hostname configuration. For more information, see Change resource based naming options for Amazon EC2.

When you launch an EC2 instance with a **Hostname type** of **Resource name**, the guest OS hostname is configured to use the EC2 instance ID.

- Format for an instance in us-east-1: *ec2-instance-id*.ec2.internal
- Example: *i-0123456789abcdef*.ec2.internal
- Format for an instance in any other AWS Region: *ec2-instance-id.region*.compute.internal
- Example: *i-0123456789abcdef.us-west-2*.compute.internal

The difference between IP name and Resource name

DNS queries for both IP names and resource names coexist to ensure backward compatibility and to allow you to migrate from IP based-naming for hostnames to resource-based naming. For private DNS hostnames based on IP names, you cannot configure whether a DNS A record query for the instance is responded to or not. DNS A record queries are always responded to irrespective of the guest OS hostname settings. In contrast, for private DNS hostnames based on resource name, you can configure whether DNS A and/or DNS AAAA queries for the instance are responded to or not. You configure the response behavior when you launch an instance or modify a subnet. For more information, see Change resource based naming options for Amazon EC2.

Where to find resource names and IP names

You can see the hostname types, resource name and IP name, in the Amazon EC2 console.

Contents

When creating an EC2 instance

When you create an EC2 instance, depending on which type of subnet you select, **Hostname type** of **Resource name** might be available or it might be selected and not be modifiable. This section explains the scenarios in which you see the hostname types resource name and IP name.

Scenario 1

You create an EC2 instance in the wizard (see Launch an EC2 instance using the launch instance wizard in the console) and, when you configure the details, you choose a subnet that you configured to be IPv6-only.

In this case, the **Hostname type** of **Resource name** is selected automatically and is not modifiable. **DNS Hostname** options of **Enable IP name IPv4 (A record) DNS requests** and **Enable resource-based IPv4 (A record) DNS requests** are deselected automatically and are not modifiable. **Enable resource-based IPv6 (AAAA record) DNS requests** is selected by default but is modifiable. If selected, DNS requests to the resource name will resolve to the IPv6 address (AAAA record) of this EC2 instance.

Scenario 2

You create an EC2 instance in the wizard (see Launch an EC2 instance using the launch instance wizard in the console) and, when you configure the details, you choose a subnet configured with an IPv4 CIDR block or both an IPv4 and IPv6 CIDR block ("dual stack").

In this case, **Enable IP name IPv4 (A record) DNS requests** is selected automatically and can't be changed. This means that requests to the IP name will resolve to the IPv4 address (A record) of this EC2 instance.

The options default to the configurations of the subnet, but you can modify the options for this instance depending on the subnet settings:

- **Hostname type**: Determines whether you want the guest OS hostname of the EC2 instance to be the resource name or IP name. The default value is **IP name**.
- **Enable resource-based IPv4 (A record) DNS requests**: Determines whether requests to your resource name resolve to the private IPv4 address (A record) of this EC2 instance. This option is not selected by default.
- **Enable resource-based IPv6 (AAAA record) DNS requests**: Determines whether requests to your resource name resolve to the IPv6 GUA address (AAAA record) of this EC2 instance. This option is not selected by default.

When viewing the details of an existing EC2 instance

You can see the hostname values for an existing EC2 instance in the **Details** tab for the EC2 instance:

- **Hostname type**: The hostname in IP name or resource name format.
- **Private IP DNS name (IPv4 only)**: The IP name that will always resolve to the private IPv4 address of the instance.
- **Private resource DNS name**: The resource name that resolves to the DNS records selected for this instance.
- **Answer private resource DNS name**: The resource name resolves to IPv4 (A), IPv6 (AAAA) or IPv4 and IPv6 (A and AAAA) DNS records.

In addition, if you connect to your EC2 instance directly over SSH and enter the `hostname` command, you'll see the hostname in either the IP name or resource name format.

Choosing between resource names and IP names

When you launch an EC2 instance (see [Launch an EC2 instance using the launch instance wizard in the console](#)), if you choose a **Hostname type** of **Resource name**, the EC2 instance launches with a hostname in the resource name format. In such cases, the DNS record for this EC2 instance can also point to the resource name. This gives you the flexibility to choose whether that hostname resolves to the IPv4 address, the IPv6 address, or both the IPv4 and IPv6 address of the instance. If you plan to use IPv6 in the future or if you are using dual-stack subnets today, it's best to use a **Hostname type** of **Resource name** so that you change DNS resolution for the hostnames of your instances without making any changes to the DNS records themselves. The resource name allows you to add and remove IPv4 and IPv6 DNS resolution on an EC2 instance.

If instead you choose a **Hostname type** of **IP name**, and use it as the DNS hostname, it can only resolve to the IPv4 address of the instance. It will not resolve to the IPv6 address of the instance even if the instance has both an IPv4 address and an IPv6 address associated with it.

Change resource based naming options for Amazon EC2

You can change the hostname type and DNS hostname configurations for subnets, which affects all subsequent instance launches in that subject, or you can change them for an EC2 instances after you launch it.

Resource based naming options

- **Hostname type**: Determines the default setting for the guest OS hostname of EC2 instances launched in the subnet. This is either the resource name or IP name.
- **Enable DNS hostname IPv4 (A record) requests**: Determines whether DNS requests/queries to the resource name resolve to the private IPv4 address (A record) of the EC2 instance.
- **Enable DNS hostname IPv6 (AAAA record) requests**: Determines whether DNS requests/queries to the resource name resolve to the IPv6 address (AAAA record) of the EC2 instance.

Subnets

Changing the subnet settings doesn't change the configuration of EC2 instances that are already launched in the subnet.

Console

To modify the options for a subnet

Open the Amazon VPC console and select the subnet. Choose **Actions**, **Edit subnet settings**. Modify the settings as needed and then save your changes.

AWS CLI

To modify the options for a subnet

Use the modify-subnet-attribute command.

```
aws ec2 modify-subnet-attribute \
    --subnet-id subnet-0abcdef1234567890 \
    --private-dns-hostname-type-on-launch resource-name \
    --enable-resource-name-dns-a-record-on-launch \
    --enable-resource-name-dns-aaaa-record-on-launch
```

PowerShell

To modify the options for a subnet

Use the Edit-EC2SubnetAttribute cmdlet.

```
Edit-EC2SubnetAttribute `
```

```
    -SubnetId subnet-0abcdef1234567890 `
    -PrivateDnsHostnameTypeOnLaunch ResourceName `
    -EnableResourceNameDnsAAAARecordOnLaunch $true `
    -EnableResourceNameDnsARecordOnLaunch $true
```

EC2 instances

Considerations

- To change the hostname type, you must first stop the instance. There is no need to stop an instance to change the other two options.
- Because you can't stop an instance store-backed instance, you can only configure the hostname type and DNS hostname options at instance launch. Only the following instance types support instance store volumes as root volumes: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

Console

To modify the hostname type and DNS hostname options for an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. If you're going to change the **Use resource based naming as guest OS hostname** setting, first stop the EC2 instance. Otherwise, skip this step.

   To stop the instance, select the instance and choose **Instance state**, **Stop instance**.
3. Select the instance and choose **Actions**, **Instance settings**, **Change resource based naming options**.
   - **Use resource based naming as guest OS hostname**: Determines whether you want the guest OS hostname of the EC2 instance to be the resource name or IP name.
   - **Answer DNS hostname IPv4 (A record) requests**: Determines whether DNS requests/ queries to your resource name resolve to the private IPv4 address of this EC2 instance.
   - **Answer DNS hostname IPv6 (AAAA record) requests**: Determines whether DNS requests/queries to your resource name resolve to the IPv6 address (AAAA record) of this EC2 instance.
4. Choose **Save**.
5. If you stopped the instance, start it again.

AWS CLI

To modify the hostname type and DNS hostname options for an instance

Use the modify-private-dns-name-options command.

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --private-dns-hostname-type resource-name \
    --enable-resource-name-dns-a-record \
    --enable-resource-name-dns-aaaa-record
```

PowerShell

To modify the hostname type and DNS hostname options for an instance

Use the Edit-EC2PrivateDnsNameOption cmdlet.

```
Edit-EC2PrivateDnsNameOption `
    -InstanceId i-1234567890abcdef0 `
    -PrivateDnsHostnameType ResourceName `
    -EnableResourceNameDnsAAAARecord $true`
    -EnableResourceNameDnsARecord $true
```

Bring your own IP addresses (BYOIP) to 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 control the address range and you can advertise the address range on the internet through AWS. After you bring the address range to Amazon EC2, it appears in your AWS account as an address pool.

> **Note**
>
> This documentation describes how to bring your own IP address range for use in Amazon EC2 only. 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*. To bring your own IP address range for use with Amazon VPC IP Address Manager, see Tutorial: Bring your IP addresses to IPAM in the *Amazon VPC IPAM User Guide*.

When you bring an IP address range to AWS, AWS validates that you control the IP address range. There are two methods that you can use to show that you control the range:

- If your IP address range is registered with an Internet Registry that supports RDAP (such as ARIN, RIPE and APNIC), you can verify control of your domain with an X.509 certificate by using the process on this page. The certificate must only be valid for the duration of the provisioning process. You can remove the certificate from your RIR's record after provisioning is complete.
- Regardless of whether your Internet Registry supports RDAP, you can use Amazon VPC IPAM to verify control of your domain with a DNS TXT record. That process is documented in Tutorial: Bring your IP addresses to IPAM in the *Amazon VPC IPAM User guide*.

For more information, see the AWS Online Tech talk Deep Dive on Bring Your Own IP.

Contents

BYOIP definitions

- **X.509 Self-sign certificate** — A certificate standard most commonly used to encrypt and authenticate data within a network. It is a certificate used by AWS to validate control over IP space from an RDAP record. For more information about X.509 certificates, see RFC 3280.
- **Autonomous System Number (ASN)** – A globally unique identifier that defines a group of IP prefixes run by one or more network operators that maintain a single, clearly-defined routing policy.
- **Regional Internet Registry (RIR)** – An organization that manages allocation and registration of IP addresses and ASNs within a region of the world.
- **Registry Data Access Protocol (RDAP)** — A read-only protocol to query current registration data within a RIR. Entries within the queried RIR database are referred to as "RDAP records". Certain

record types need to be updated by customers via a RIR-provided mechanism. These records are queried by AWS to verify control of an address space in the RIR.

- **Route Origin Authorization (ROA)** — An object created by RIRs for customers to authenticate IP advertisement in particular autonomous systems. For an overview, see Route Origin Authorizations (ROAs) on the ARIN website.
- **Local Internet Registry (LIR)** — Organizations such as internet service providers that allocate a block of IP addresses from an RIR for their customers.

Requirements and quotas

- The address range must be registered with your Regional Internet Registry (RIR). See your RIR for any policies regarding geographic regions. BYOIP currently supports registration in 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 advertisable and /60 for CIDRs that are not publicly advertisable.
- ROAs are not required for CIDR ranges that are not publicly advertisable, but the RDAP records still need to be updated.
- You can bring each address range to one AWS Region at a time.
- You can bring a total of five BYOIP IPv4 and IPv6 address ranges per AWS Region to your AWS account. You cannot adjust the quotas for BYOIP CIDRs using the Service Quotas console, but you can request a quota increase by contacting the AWS Support Center as described in AWS service quotas in the *AWS General Reference*.
- You cannot share your IP address range with other accounts using AWS RAM unless you use Amazon VPC IP Address Manager (IPAM) and integrate IPAM with AWS Organizations. For more information, see Integrate IPAM with AWS Organizations in the *Amazon VPC IPAM User Guide*.
- 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.
- Legacy address space, the IPv4 address space that was distributed by the Internet Assigned Numbers Authority's (IANA) central registry prior to the formation of the Regional Internet Registry (RIR) system, still requires a corresponding ROA object.

- For LIRs, it is common that they use a manual process to update their records. This can take days to deploy depending on the LIR.
- A single ROA object and RDAP record are needed for a large CIDR block. You can bring multiple smaller CIDR blocks from that range to AWS, even across multiple AWS Regions, using the single object and record.
- BYOIP is not supported for Wavelength Zones or on AWS Outposts.
- Do not make any manual changes for BYOIP in RADb or any other IRR. BYOIP will automatically update RADb. Any manual changes that include the BYOIP ASN will cause the BYOIP provision operation to fail.
- Once you bring an IPv4 address range to AWS, you can use all of the IP addresses in the range, including the first address (the network address) and the last address (the broadcast address).

Regional availability

The BYOIP feature is currently available in all commercial AWS Regions except for China Regions.

Local Zone availability

A Local Zone is an extension of an AWS Region in geographic proximity to your users. Local Zones are grouped into "network border groups". In AWS, a network border group is a collection of Availability Zones (AZs), Local Zones, or Wavelength Zones from which AWS advertises a public IP address. Local Zones may have different network border groups than the AZs in an AWS Region to ensure minimum latency or physical distance between the AWS network and the customers accessing the resources in these Zones.

You can provision BYOIPv4 address ranges to and advertise them in the following Local Zone network border groups using the `--network-border-group` option:

- af-south-1-los-1
- ap-northeast-1-tpe-1
- ap-south-1-ccu-1
- ap-south-1-del-1
- ap-southeast-1-bkk-1
- ap-southeast-1-mnl-1
- ap-southeast-2-akl-1

- ap-southeast-2-per-1
- eu-central-1-ham-1
- eu-central-1-waw-1
- eu-north-1-cph-1
- eu-north-1-hel-1
- me-south-1-mct-1
- us-east-1-atl-2
- us-east-1-bos-1
- us-east-1-bue-1
- us-east-1-chi-2
- us-east-1-dfw-2
- us-east-1-iah-2
- us-east-1-lim-1
- us-east-1-mci-1
- us-east-1-mia-2
- us-east-1-msp-1
- us-east-1-nyc-1
- us-east-1-nyc-2
- us-east-1-phl-1
- us-east-1-qro-1
- us-east-1-scl-1
- us-west-2-den-1
- us-west-2-hnl-1
- us-west-2-las-1
- us-west-2-lax-1
- us-west-2-pdx-1
- us-west-2-phx-2
- us-west-2-sea-1

If you have Local Zones enabled (see Enable a Local Zone), you can choose a network border group for Local Zones when you provision and advertise a BYOIPv4 CIDR. Choose the network border group carefully as the EIP and the AWS resource it is associated with must reside in the same network border group.

> **Note**
>
> You cannot provision or advertise BYOIPv6 address ranges in Local Zones at this time.

Prerequisites for BYOIP in Amazon EC2

The onboarding process for BYOIP has two phases, for which you must perform three steps. These steps correspond to the steps depicted in the following diagram. We include manual steps in this documentation, but your RIR might offer managed services to help you with these steps.

> **Tip**
>
> The tasks in this section require a Linux terminal and may be performed using Linux, the AWS CloudShell, or the Windows Subsystem for Linux.

Contents

Overview

Preparation phase

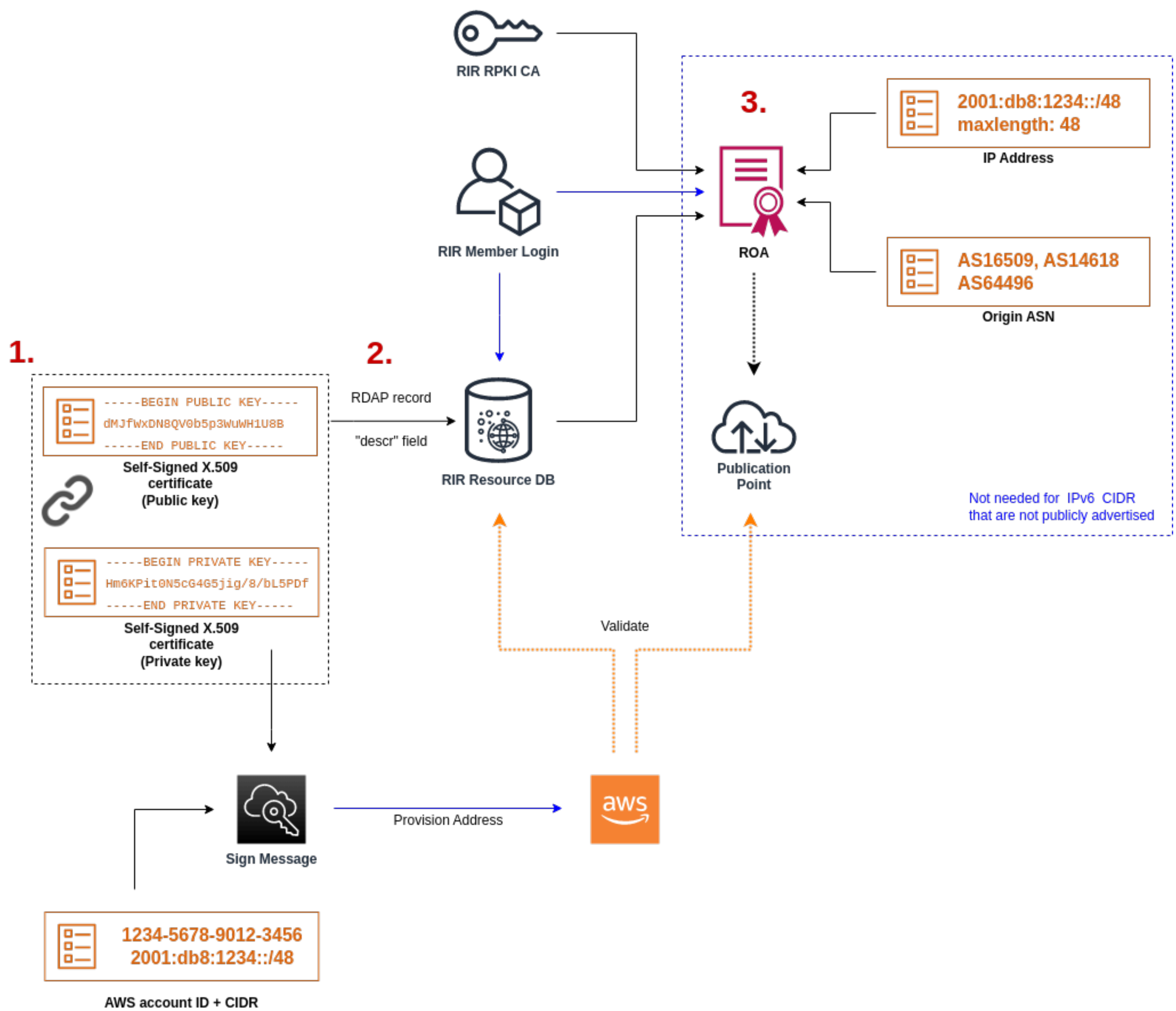
[1] Create a private key and use it to generate a self-signed X.509 certificate for authentication purposes. This certificate is only used during the provisioning phase. You can remove the certificate from your RIR's record after provisioning is complete

RIR configuration phase

[2] Upload the self-signed certificate to your RDAP record comments.

[3] Create a ROA object in your RIR. The ROA defines the desired address range, the Autonomous System Numbers (ASNs) allowed to advertise the address range, and an expiration date to register with the Resource Public Key Infrastructure (RPKI) of your RIR.

> **Note**
>
> A ROA is not required for non-publicly advertisable IPv6 address space.



To bring on multiple non-contiguous address ranges, you must repeat this process with each address range. However, the preparation and RIR configuration steps don't need to be repeated if splitting a contiguous block across several different AWS Regions.

Bringing on an address range has no effect on any address ranges that you brought on previously.

Create a private key and generate an X.509 certificate

Use the following procedure to create a self-signed X.509 certificate and add it to the RDAP record for your RIR. This key pair is used to authenticate the address range with the RIR. The **openssl** commands require OpenSSL version 1.0.2 or later.

Copy the following commands and replace only the placeholder values (in colored italic text).

This procedure follows the best practice of encrypting your private RSA key and requiring a passphrase to access it.

1. Generate an RSA 2048-bit private key 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 passphrase:

   ```
   ......+++
   .+++
   Enter PEM pass phrase: xxxxxxx
   Verifying - Enter PEM pass phrase: xxxxxxx
   ```

   You can inspect the key using the following command:

   ```
   $ openssl pkey -in private-key.pem -text
   ```

   This returns a passphrase prompt and the contents of the key, which should be similar to the following:

   ```
   Enter pass phrase for private-key.pem: xxxxxxx
   -----BEGIN PRIVATE KEY-----
   MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDFBXHRI4HVKAhh
   ```

```
3seiciooizCRTbJe1+YsxNTja4XyKypVGIFWDGhZs44FCHlPOOSVJ+NqP74w96oM
7DPS3xo9kaQyZBFn2YEp2EBq5vf307KHNRmZZUmkn0zHOSEpNmY2fMxISBxewlxR
FAniwmSd/8TDvHJMY9FvAIvWuTsv5l0tJKk+a91K4+tO3UdDR7Sno5WXExfsBrW3
g1ydo3TBsx8i5/YiVOcNApy7ge2/FiwY3aCXJB6r6nuF6H8mRgI4r4vkMRsOlAhJ
DnZPNeweboo+K3Q3lwbgbmOKD/z9svk8N/+hUTBtIX0fRtbG+PLIw3xWRHGrMSn2
BzsPVuDLAgMBAAECggEACiJUj2hfJkKv47Dc3es3Zex67A5uDVjXmxfox2Xhdupn
fAcNqAptV6fXt0SPUNbhUxbBKNbshoJGufFwXPli1SXnpzvkdU4Hyco4zgbhXFsE
RNYjYfOGzTPwdBLpNMB6k3Tp4RHse6dNrlH0jDhpioL8cQEBdBJyVF5X0wymEbmV
mC0jgH/MxsBAPWW6ZKicg9ULMlWiAZ3MRAZPjHHgpYkAAsUWKAbCBwVQcVjGO59W
jfZjzTX5pQtVVH68ruciH88DTZCwjCkjBhxg+OIkJBLE5wkh82jIHSivZ63flwLw
z+E0+HhELSZJrn2MY6Jxmik3qNNUOF/Z+3msdj2luQKBgQDjwlC/3jxp8zJy6P8o
JQKv7TdvMwUj4VSWOHZBHLv4evJaaia0uQjIo1UDa8AYitqhX1NmCCehGH8yuXj/
v6V3CzMKDkmRr1NrONnSz5QsndQ04Z6ihAQlPmJ96g4wKtgoC7AYpyP0g1a+4/sj
b1+o3YQI4pD/F71c+qaztH7PRwKBgQDDc23yNmT3+Jyptf0fKjEvONK+xwUKzi9c
L/OzBq5yOIC1Pz2T85gOe1i8kwZws+xlpG6uBT6lmIJELd0k59FyupNu4dPvX5SD
6GGqdx4jk9KvI74usGeOBohmF0phTHkrWKBxXiyT0oS8zjnJlEn8ysIpGgO28jjr
LpaHNZ/MXQKBgQDfLNcnS0LzpsS2aK0tzyZU8SMyqVHOGMxj7quhneBq2T6FbiLD
T9TVlYaGNZ0j71vQaLI19qOubWymbautH0Op5KV8owdf4+bf1/NJaPIOzhDUSIjD
Qo01WW31Z9XDSRhKFTnWzmCjBdeIcajyzf10YKsycaAW9lItu8aBrMndnQKBgQDb
nNp/JyRwqjOrNljk7DHEs+SD39kHQzzCfqd+dnTPv2sc06+cpym3yulQcbokULpy
fmRo3bin/pvJQ3aZX/Bdh9woTXqhXDdrrSwWInVYMQPyPk8f/D9mIOJp5FUWMwHD
U+whIZSxsEeE+jtixlWtheKRYkQmzQZXbWdIhYyI3QKBgD+F/6wcZ85QW8nAUykA
3WrSIx/3cwDGdm4NRGct8ZOZjTHjiy9ojMOD1L7iMhRQ/3k3hUsin5LDMp/ryWGG
x4uIaLat40kiC7T4I66DM7P59euqdz3w0PD+VU+h7GSivvsFDdySUt7bNK0AUVLh
dMJfWxDN8QV0b5p3WuWH1U8B
-----END PRIVATE KEY-----
Private-Key: (2048 bit)
modulus:
    00:c5:05:71:d1:23:81:d5:28:08:61:de:c7:a2:72:
    2a:28:8b:30:91:4d:b2:5e:d7:e6:2c:c4:d4:e3:6b:
    85:f2:2b:2a:55:18:81:56:0c:68:59:b3:8e:05:08:
    79:4f:38:e4:95:27:e3:6a:3f:be:30:f7:aa:0c:ec:
    33:d2:df:1a:3d:91:a4:32:64:11:67:d9:81:29:d8:
    40:6a:e6:f7:f7:d3:b2:87:35:19:99:65:49:a4:9f:
    4c:c7:39:21:29:36:66:36:7c:cc:48:48:1c:5e:c2:
    5c:51:14:09:e2:c2:64:9d:ff:c4:c3:bc:72:4c:63:
    d1:6f:00:8b:d6:b9:3b:2f:e6:5d:2d:24:a9:3e:6b:
    dd:4a:e3:eb:4e:dd:47:43:47:b4:a7:a3:95:97:13:
    17:ec:06:b5:b7:83:5c:9d:a3:74:c1:b3:1f:22:e7:
    f6:22:54:e7:0d:02:9c:bb:81:ed:bf:16:2c:18:dd:
    a0:97:24:1e:ab:ea:7b:85:e8:7f:26:46:02:38:af:
    8b:e4:31:1b:0e:94:08:49:0e:76:4f:35:ec:1e:6e:
    8a:3e:2b:74:37:97:06:e0:6e:63:8a:0f:fc:fd:b2:
    f9:3c:37:ff:a1:51:30:6d:21:7d:1f:46:d6:c6:f8:
```

```
    f2:c8:c3:7c:56:44:71:ab:31:29:f6:07:3b:0f:56:
    e0:cb
publicExponent: 65537 (0x10001)
privateExponent:
    0a:22:54:8f:68:5f:26:42:af:e3:b0:dc:dd:eb:37:
    65:ec:7a:ec:0e:6e:0d:58:d7:9b:17:e8:c7:65:e1:
    76:ea:67:7c:07:0d:a8:0a:6d:57:a7:d7:b7:44:8f:
    50:d6:e1:53:16:c1:28:d6:ec:86:82:46:b9:f1:70:
    5c:f9:62:d5:25:e7:a7:3b:e4:75:4e:07:c9:ca:38:
    ce:06:e1:5c:5b:04:44:d6:23:61:f3:86:cd:33:f0:
    74:12:e9:34:c0:7a:93:74:e9:e1:11:ec:7b:a7:4d:
    ae:51:f4:8c:38:69:8a:82:fc:71:01:01:74:12:72:
    54:5e:57:d3:0c:a6:11:b9:95:98:2d:23:80:7f:cc:
    c6:c0:40:3d:65:ba:64:a8:9c:83:d5:0b:32:55:a2:
    01:9d:cc:44:06:4f:8c:71:e0:a5:89:00:02:c5:16:
    28:06:c2:07:05:50:71:58:c6:3b:9f:56:8d:f6:63:
    cd:35:f9:a5:0b:55:54:7e:bc:ae:e7:22:1f:cf:03:
    4d:90:b0:8c:29:23:06:1c:60:f8:e2:24:24:12:c4:
    e7:09:21:f3:68:c8:1d:28:af:67:ad:df:97:02:f0:
    cf:e1:34:f8:78:44:2d:26:49:ae:7d:8c:63:a2:71:
    9a:29:37:a8:d3:54:38:5f:d9:fb:79:ac:76:3d:a5:
    b9
prime1:
    00:e3:c2:50:bf:de:3c:69:f3:32:72:e8:ff:28:25:
    02:af:ed:37:6f:33:05:23:e1:54:96:38:76:41:1c:
    bb:f8:7a:f2:5a:6a:26:b4:b9:08:c8:a3:55:03:6b:
    c0:18:8a:da:a1:5f:53:66:08:27:a1:18:7f:32:b9:
    78:ff:bf:a5:77:0b:33:0a:0e:49:91:af:53:6b:38:
    d9:d2:cf:94:2c:9d:d4:34:e1:9e:a2:84:04:25:3e:
    62:7d:ea:0e:30:2a:d8:28:0b:b0:18:a7:23:f4:83:
    56:be:e3:fb:23:6f:5f:a8:dd:84:08:e2:90:ff:17:
    bd:5c:fa:a6:b3:b4:7e:cf:47
prime2:
    00:dd:73:6d:f2:36:64:f7:f8:9c:a9:b5:fd:1f:2a:
    31:2f:38:d2:be:c7:05:0a:ce:2f:5c:2f:f3:b3:06:
    ae:72:38:80:b5:3f:3d:93:f3:98:0e:7b:58:bc:93:
    06:70:b3:ec:65:a4:6e:ae:05:3e:a5:98:82:44:2d:
    dd:24:e7:d1:72:ba:93:6e:e1:d3:ef:5f:94:83:e8:
    61:aa:77:1e:23:93:d2:af:23:be:2e:b0:67:8e:06:
    88:66:17:4a:61:4c:79:2b:58:a0:71:5e:2c:93:d2:
    84:bc:ce:39:c9:94:49:fc:ca:c2:29:1a:03:b6:f2:
    38:eb:2e:96:87:35:9f:cc:5d
exponent1:
    00:df:2c:d7:27:4b:42:f3:a6:c4:b6:68:ad:2d:cf:
```

```
    26:54:f1:23:32:a9:51:ce:18:cc:63:ee:ab:a1:9d:
    e0:6a:d9:3e:85:6e:22:c3:4f:d4:d5:95:86:86:35:
    9d:23:ef:5b:d0:68:b2:35:f6:a3:ae:6d:6c:a6:6d:
    ab:ad:1f:43:a9:e4:a5:7c:a3:07:5f:e3:e6:df:d7:
    f3:49:68:f2:0e:ce:10:d4:48:88:c3:42:8d:35:59:
    6d:f5:67:d5:c3:49:18:4a:15:39:d6:ce:60:a3:05:
    d7:88:71:a8:f2:cd:fd:74:60:ab:32:71:a0:16:f6:
    52:2d:bb:c6:81:ac:c9:dd:9d
exponent2:
    00:db:9c:da:7f:27:24:70:aa:33:ab:36:58:e4:ec:
    31:c4:b3:e4:83:df:d9:07:43:3c:c2:7e:a7:7e:76:
    74:cf:bf:6b:1c:d3:af:9c:a7:29:b7:ca:e9:50:71:
    ba:24:50:ba:72:7e:64:68:dd:b8:a7:fe:9b:c9:43:
    76:99:5f:f0:5d:87:dc:28:4d:7a:a1:5c:37:6b:ad:
    2c:16:22:75:58:31:03:f2:3e:4f:1f:fc:3f:66:20:
    e2:69:e4:55:16:33:01:c3:53:ec:21:21:94:b1:b0:
    47:84:fa:3b:62:c6:55:ad:85:e2:91:62:44:26:cd:
    06:57:6d:67:48:85:8c:88:dd
coefficient:
    3f:85:ff:ac:1c:67:ce:50:5b:c9:c0:53:29:00:dd:
    6a:d2:23:1f:f7:73:00:c6:76:6e:0d:44:67:2d:f1:
    93:99:8d:31:e3:8b:2f:68:8c:c3:83:d4:be:e2:32:
    14:50:ff:79:37:85:4b:22:9f:92:c3:32:9f:eb:c9:
    61:86:c7:8b:88:68:b6:ad:e3:49:22:0b:b4:f8:23:
    ae:83:33:b3:f9:f5:eb:aa:77:3d:f0:d0:f0:fe:55:
    4f:a1:ec:64:a2:be:fb:05:0d:dc:92:52:de:db:34:
    ad:00:51:52:e1:74:c2:5f:5b:10:cd:f1:05:74:6f:
    9a:77:5a:e5:87:d5:4f:01
```

Keep your private key in a secure location when it is not in use.

2. Generate an X.509 certificate using the private key created in the previous step. In this example, the certificate expires in 365 days, after which time it cannot be trusted. Be sure to set the expiration appropriately. The certificate must only be valid for the duration of the provisioning process. You can remove the certificate from your RIR's record after provisioning is complete. 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.

```
$ 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 []:
```

Note

The Common Name is not needed for AWS provisioning. It can be any internal or public domain name.

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-----`.

Upload the X.509 certificate to 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, use the Account Manager portal to add the certificate in the "Public Comments" section for the "Network Information" object representing your address range. Do not add it to the comments section for your organization.
- For RIPE, add the certificate as a new "descr" field to the "inetnum" or "inet6num" object representing your address range. These can usually be found in the "My Resources" section of the RIPE Database portal. Do not add it to the comments section for your organization or the "remarks" field of the above objects.
- For APNIC, add the certificate by editing remarks on the "inetnum" or "inet6num" record.

You can remove the certificate from your RIR's record after the provisioning stage below has been completed.

Create a ROA object in your RIR

Create a ROA object to authorize the Amazon ASNs 16509 and 14618 to advertise your address range, as well as the ASNs that are currently authorized to advertise the address range. For the AWS GovCloud (US) Regions, authorize ASN 8987 instead of 16509 and 14618. You must set the maximum length to the size of the CIDR that you are bringing in. The most specific IPv4 prefix you can bring is /24. The most specific IPv6 address range that you can bring is /48 for CIDRs that are publicly advertisable and /60 for CIDRs that are not publicly advertisable.

Important

If you are creating a ROA object for Amazon VPC IP Address Manager (IPAM), when you create the ROAs, for IPv4 CIDRs you must set the maximum length of an IP address prefix to /24. For IPv6 CIDRs, if you are adding them to an advertisable pool, the maximum length of an IP address prefix must be /48. This ensures that you have full flexibility to divide your public IP address across AWS Regions. IPAM enforces the maximum length you set. For more information about BYOIP addresses to IPAM, see Tutorial: BYOIP address CIDRs to IPAM in the *Amazon VPC IPAM User Guide*.

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

When you migrate advertisements from an on-premises workload to AWS, you must create a ROA for your existing ASN before creating the ROAs for Amazon's ASNs. Otherwise, you might see an impact to your existing routing and advertisements.

> **Important**
>
> For Amazon to advertise and continue advertising your IP address range, your ROAs with Amazon ASNs must be compliant with the above guidelines. If your ROAs are invalid or not compliant with the above guidelines, Amazon reserves the right to stop advertising your IP address range.

> **Note**
>
> This step is not required for non-publicly advertisable IPv6 address space.

Onboard your address range for use in Amazon EC2

The onboarding process for BYOIP includes the following tasks, depending on your needs.

Tasks

- Provision a publicly advertisable address range in AWS
- Provision an IPv6 address range that's not publicly advertisable
- Advertise the address range through AWS
- Deprovision the address range
- Validate your BYOIP

Provision a publicly advertisable address range in AWS

When you provision an address range for use with AWS, you are confirming that you control the address range and are authorizing Amazon to advertise it. We also verify that you control 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:

   ```
   text_message="1|aws|0123456789AB|198.51.100.0/24|20211231|SHA256|RSAPSS"
   ```

   This is not to be confused with a 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.

   Important

   We recommend that you copy and paste this command. Except for the message content, do not modify or replace any of the values.

   ```
   signed_message=$( echo -n $text_message | 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.

> **Important**
>
> You must specify the AWS Region where the BYOIP range should be provisioned if it differs from your [Configure the AWS CLI](#) `Default region name`.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context 
 Message="$text_message",Signature="$signed_message" --region us-east-1
```

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**

   While most provisioning will be completed within two hours, it might take up to one week 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 --region us-east-1
```

   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 advertisable

By default, an address range is provisioned to be publicly advertisable to the internet. You can provision an IPv6 address range that will not be publicly advertisable. 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 hybrid connectivity options that support IPv6, such as AWS Direct Connect, AWS Site-to-Site VPN, or Amazon VPC Transit Gateways.

A ROA is not required to provision a non-public address range.

Important

- You can only specify whether an address range is publicly advertisable during provisioning. You cannot change the advertisable status later on.
- Amazon VPC doesn't support unique local address (ULA) CIDRs. All VPCs must have unique IPv6 CIDRs. Two VPCs can’t have the same IPv6 CIDR range.

To provision an IPv6 address range that will not be publicly advertisable, use the following provision-byoip-cidr command.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
 Message="$text_message",Signature="$signed_message" --no-publicly-advertisable --
region us-east-1
```

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 or any portion of the range from other locations before you advertise it through AWS. If you keep advertising your IP address range or any portion of it from other locations, we can't reliably support it or troubleshoot issues. Specifically, we can't guarantee that traffic to the address range or a portion of the 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.

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.

```
aws ec2 advertise-byoip-cidr --cidr address-range --region us-east-1
```

To stop advertising the address range, use the following withdraw-byoip-cidr command.

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

Deprovision the address range

To stop using your address range with AWS, first release any Elastic IP addresses and disassociate any IPv6 CIDR blocks that are still allocated from the address pool. Then stop advertising the address range, and finally, deprovision the address range.

You cannot deprovision a portion of the address range. If you want to use a more specific address range with AWS, deprovision the entire address range and provision a more specific address range.

(IPv4) To release each Elastic IP address, use the following release-address command.

```
aws ec2 release-address --allocation-id eipalloc-12345678abcabcabc --region us-east-1
```

(IPv6) To disassociate an IPv6 CIDR block, use the following disassociate-vpc-cidr-block command.

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-12345abcd1234abc1
 --region us-east-1
```

To stop advertising the address range, use the following withdraw-byoip-cidr command.

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

To deprovision the address range, use the following deprovision-byoip-cidr command.

```
aws ec2 deprovision-byoip-cidr --cidr address-range --region us-east-1
```

It can take up to a day to deprovision an address range.

Validate your BYOIP

1. Validate the self-signed x.509 key pair

   Validate that the certificate has been uploaded and is valid via the whois command.

   For ARIN, use `whois -h whois.arin.net r + 2001:0DB8:6172::/48` to look up the RDAP record for your address range. Check the `Public Comments` section for the `NetRange` (network range) in the command output. The certificate should be added in the `Public Comments` section for the address range.

   You can inspect the `Public Comments` containing the certificate using the following command:

   ```
   whois -h whois.arin.net r + 2001:0DB8:6172::/48 | grep Comments | grep BEGIN
   ```

   This returns output with the contents of the key, which should be similar to the following:

   ```
   Public Comments:
   -----BEGIN CERTIFICATE-----
   MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
   LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
   hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
   wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
   NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
   rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
   VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
   W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
   R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
   prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
   mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
   Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
   xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
   3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
   AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
   QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
   08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
   Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
   UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
   ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
   MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
   ```

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

For RIPE, use `whois -r -h whois.ripe.net 2001:0DB8:7269::/48` to look up the RDAP record for your address range. Check the `descr` section for the `inetnum` object (network range) in the command output. The certificate should be added as a new `descr` field for the address range.

You can inspect the `descr` containing the certificate using the following command:

```
whois -r -h whois.ripe.net 2001:0DB8:7269::/48 | grep descr | grep BEGIN
```

This returns output with the contents of the key, which should be similar to the following:

```
descr:
-----BEGIN CERTIFICATE-----MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8
RDAHSP+I1TowDQYJKoZIhvcNAQELBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAg
MCEF1Y2tsYW5kMREwDwYDVQQHDAhBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIF
dlYiBTZXJ2aWNlczETMBEGA1UECwwKQ1lPSVAgRGVtbzETMBEGA1UEAwwKQ1lPS
VAgRGVtbzAeFw0yMTEyMDcyMDI0NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNV
BAYTAk5aMREwDwYDVQQIDAhBdWNrbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDA
aBgNVBAoME0FtYXpvbiBXZWIgU2VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW
8xEzARBgNVBAMMCkJZT0lQIERlbW8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQCfmacvDp0wZ0ceiXXcR/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanA
EskgAseyFypwEEQr4CJijI/5hp9prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3
stsI5QesHVRwOaXUdprAnndaTugmDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq
35wU/x+wXlAqBXg4MZK2KoUu27kYt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp
1ZnVIc7NqnhdeIW48QaYjhMlUEfxdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2r
GlHWkJsbhr0VEUyAGu1bwkgcdww3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBS
tFyujN6SYBr2glHpGt0XGF7GbGTAfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0
XGF7GbGTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6Y
Lhz521lfyVfxY0t6o3410bQAeAF08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyL
xngwMYN0XY5tVhDQqk4/gmDNEKSZy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9
wySL507XQz76Uk5cFypBOzbnk35UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8
mBGqVpPpey+dXpzzzv1iBKN/VY4ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGC
vRDl/qdO/GIDJi77dmZWkh/ic90MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoN
PyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

For APNIC, use `whois -h whois.apnic.net 2001:0DB8:6170::/48` to look up the RDAP record for your BYOIP address range. Check the `remarks` section for the `inetnum` object (network range) in the command output. The certificate should be added as a new `remarks` field for the address range.

You can inspect the `remarks` containing the certificate using the following command:

```
whois -h whois.apnic.net 2001:0DB8:6170::/48 | grep remarks | grep BEGIN
```

This returns output with the contents of the key, which should be similar to the following:

```
remarks:
-----BEGIN CERTIFICATE-----
MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

2. Validate the creation of a ROA object

   Validate the successful creation of the ROA objects using the RIPEstat Data API. Be sure to test your address range against the Amazon ASNs 16509 and 14618, plus the ASNs that are currently authorized to advertise the address range.

   You can inspect the ROA objects from different Amazon ASNs with your address range by using the following command:

```
curl --location --request GET "https://stat.ripe.net/data/rpki-validation/data.json?resource=ASN&prefix=CIDR
```

In this example output, the response has a result of `"status": "valid"` for the Amazon ASN 16509. This indicates the ROA object for the address range was created successfully:


```
{
    "messages": [],
    "see_also": [],
    "version": "0.3",
    "data_call_name": "rpki-validation",
    "data_call_status": "supported",
    "cached": false,
    "data": {
        "validating_roas": [
            {
                "origin": "16509",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "valid"
            },
            {
                "origin": "14618",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            },
            {
                "origin": "64496",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            }
        ],
        "status": "valid",
        "validator": "routinator",
        "resource": "16509",
        "prefix": "2001:0DB8::/32"
    },
    "query_id": "20230224152430-81e6384e-21ba-4a86-852a-31850787105f",
    "process_time": 58,
    "server_id": "app116",
```

```
    "build_version": "live.2023.2.1.142",
    "status": "ok",
    "status_code": 200,
    "time": "2023-02-24T15:24:30.773654"
}
```

A status of “unknown” indicates the ROA object for the address range has not been created. A status of “invalid_asn” indicates that the ROA object for the address range was not created successfully.

Use your BYOIP address range in Amazon EC2

You can view and use the IPv4 and IPv6 address ranges that you've provisioned in your account. For more information, see the section called “Onboard your address range”.

IPv4 address ranges

You can create an Elastic IP address from your IPv4 address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers.

To view information about the IPv4 address pools that you've provisioned in your account, use the following describe-public-ipv4-pools command.

```
aws ec2 describe-public-ipv4-pools --region us-east-1
```

To create an Elastic IP address from your IPv4 address pool, use the allocate-address command. You can use the `--public-ipv4-pool` option to specify the ID of the address pool returned by `describe-byoip-cidrs`. Or you can use the `--address` option to specify an address from the address range that you provisioned.

IPv6 address ranges

To view information about the IPv6 address pools that you've provisioned in your account, use the following describe-ipv6-pools command.

```
aws ec2 describe-ipv6-pools --region us-east-1
```

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 --region us-east-1
```

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 --region us-east-1
```

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 --region us-east-1
```

If you disassociate the IPv6 CIDR block from your VPC, it's released back into your IPv6 address pool.

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 you need to connect to an instance that does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet.

Contents

- Transfer an Elastic IP address between AWS accounts
- Release an Elastic IP address
- Create a reverse DNS record for email on Amazon EC2

Elastic IP address pricing

There is a charge for all Elastic IP addresses whether they are in use (allocated to a resource, like an EC2 instance) or idle (created in your account but unallocated).

AWS charges for all public IPv4 addresses, including public IPv4 addresses associated with running instances and Elastic IP addresses. For more information, see the **Public IPv4 Address** tab on the Amazon VPC 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.
- An Elastic IP address is for use in a specific Region only, and cannot be moved to a different Region.
- An Elastic IP address comes from Amazon's pool of IPv4 addresses, or from a custom IPv4 address pool that you have brought to your AWS account. We do not support Elastic IP addresses for IPv6.
- 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, if the instance already has a public IPv4 address associated with it, that public IPv4 address is released back into Amazon's pool of public IPv4 addresses and the Elastic IP address is associated with the instance instead. You cannot reuse the public IPv4 address previously associated with the instance and you cannot convert that public IPv4 address to an Elastic IP address. For more information, see Public IPv4 addresses.
- 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. You are charged for all Elastic IP addresses in your account, regardless of whether they are associated or disassociated with an instance. For more information, see the **Public IPv4 Address** tab on the Amazon VPC pricing page.
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS host name of the instance changes to match the Elastic IP address.
- We resolve a public DNS host name to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.
- 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 quota.
- 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.

Elastic IP address quota

By default, all AWS accounts have a quota of five (5) Elastic IP addresses per Region, because public (IPv4) internet addresses are a scarce public resource. We strongly recommend that you use Elastic IP addresses primarily for their ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

If you think your architecture warrants additional Elastic IP addresses, you can request a quota increase directly from the Service Quotas console. To request a quota increase, choose **Request increase at account-level**. For more information, see Amazon EC2 service quotas.

Associate an Elastic IP address with an instance

After you allocate an Elastic IP address, you can associate it with an AWS resource, such as an EC2 instance, NAT gateway, or Network Load Balancer. To associate an Elastic IP address with a different AWS resource later on, you can disassociate it from its current resource and then associated it with the new resource.

Tasks

- Allocate an Elastic IP address
- Associate an Elastic IP address
- Disassociate an Elastic IP address

Allocate an Elastic IP address

You can allocate an Elastic IP address for use in a Region. There is a charge for all Elastic IP addresses whether they are in use (associated with a resource, like an EC2 instance) or idle (created in your account but unassociated).

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. (Optional) When you allocate an Elastic IP address (EIP), you choose the **Network border group** in which to allocate the EIP. A network border group is a collection of Availability Zones (AZs), Local Zones, or Wavelength Zones from which AWS advertises a public IP address. Local Zones and Wavelength Zones may have different network border groups than the AZs in a Region to ensure minimum latency or physical distance between the AWS network and the customers accessing the resources in these Zones.

   Important

   You must allocate an EIP in the same network border group as the AWS resource that will be associated with the EIP. An EIP in one network border group can only be advertised in zones in that network border group and not in any other zones represented by other network border groups.

If you have Local Zones or Wavelength Zones enabled (for more information, see Enable a Local Zone or Enable Wavelength Zones), you can choose a network border group for AZs, Local Zones, or Wavelength Zones. Choose the network border group carefully as the EIP and the AWS resource it is associated with must reside in the same network border group. You can use the EC2 console to view the network border group that your Availability Zones, Local Zones, or Wavelength Zones are in. Typically, all Availability Zones in a Region belong to the same network border group, whereas Local Zones or Wavelength Zones belong to their own separate network border groups.

If you don't have Local Zones or Wavelength Zones enabled, when you allocate an EIP, the network border group that represents all of the AZs for the Region (such as `us-west-2`) is predefined for you and you cannot change it. This means that the EIP that you allocate to this network border group will be advertised in all AZs in the Region you're in.

5. 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.
   - **Public IPv4 address that you bring to your AWS account**—If you want to allocate a non-contiguous (non-sequential) public 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. For more information about bringing your own IP address range to your AWS account, see Bring your own IP addresses (BYOIP) to Amazon EC2.
   - **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.
   - **Allocate using an IPAM IPv4 pool**: If you want to allocate sequential Elastic IP addresses from a contiguous public IPv4 block in an IPAM pool. Allocating sequential Elastic IP addresses can significantly reduce management overhead for security access control lists and simplify IP address allocation and tracking for enterprises scaling on AWS. For more information, see Allocate sequential Elastic IP addresses from an IPAM pool in the *Amazon VPC IPAM User Guide*.

6. (Optional) To add a tag, choose **Add new tag** and enter a tag key and a tag value.

AWS CLI

To allocate an Elastic IP address

Use the allocate-address AWS CLI command.

```
aws ec2 allocate-address
```

PowerShell

To allocate an Elastic IP address

Use the New-EC2Address cmdlet.

```
New-EC2Address -Domain Vpc
```

Associate an Elastic IP address

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 Enable internet access using an internet gateway in the *Amazon VPC User Guide*.

Console

To associate an Elastic IP address with an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to associate and choose **Actions**, **Associate Elastic IP address**.
4. For **Resource type**, choose **Instance**.
5. For instance, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
6. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
7. Choose **Associate**.

To associate an Elastic IP address with a network interface

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to associate and choose **Actions**, **Associate Elastic IP address**.
4. For **Resource type**, choose **Network interface**.
5. For **Network interface**, choose the network interface with which to associate the Elastic IP address. You can also enter text to search for a specific network interface.
6. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
7. Choose **Associate**.

AWS CLI

To associate an Elastic IP address

Use the associate-address AWS CLI command.

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-
id eipalloc-64d5890a
```

PowerShell

To associate an Elastic IP address

Use the Register-EC2Address cmdlet.

```
Register-EC2Address `
    -InstanceId i-0b263919b6498b123 `
    -AllocationId eipalloc-64d5890a
```

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 associate it with another resource.

Console

To disassociate and reassociate an Elastic IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to disassociate, choose **Actions**, **Disassociate Elastic IP address**.
4. Choose **Disassociate**.

AWS CLI

To disassociate an Elastic IP address

Use the disassociate-address AWS CLI command.

```
aws ec2 disassociate-address --association-id eipassoc-12345678
```

PowerShell

To disassociate an Elastic IP address

Use the Unregister-EC2Address cmdlet.

```
Unregister-EC2Address -AssociationId eipassoc-12345678
```

Transfer an Elastic IP address between AWS accounts

You can transfer an Elastic IP address from one AWS account to another. This can be helpful in the following situations:

- **Disaster recovery** – Quickly remap the IP addresses for public-facing internet workloads during emergency events.
- **Organizational restructuring** – Quickly move a workload from one AWS account to another. An address transfer avoids the need to wait for new Elastic IP addresses to be allowed by your security groups and network ACLs.
- **Centralized security administration** – Use a centralized AWS security account to track and transfer Elastic IP addresses that have been vetted for security compliance.

Pricing

There is no charge for transferring Elastic IP addresses.

Tasks

- Enable Elastic IP address transfer
- Accept a transferred Elastic IP address
- Disable Elastic IP address transfer

Enable Elastic IP address transfer

This section describes how to accept a transferred Elastic IP address. Note the following limitations related to enabling Elastic IP addresses for transfer:

- You can transfer Elastic IP addresses from any AWS account (source account) to any other AWS account in the same AWS Region (transfer account). You cannot transfer Elastic IP addresses to a different Region.
- When you transfer an Elastic IP address, there is a two-step handshake between the AWS accounts. When the source account starts the transfer, the transfer accounts have seven days to accept the Elastic IP address transfer. During those seven days, the source account can view the pending transfer (for example in the AWS console or by using the describe-address-transfers command). After seven days, the transfer expires and ownership of the Elastic IP address returns to the source account.
- Accepted transfers are visible to the source account (for example in the AWS console or by using the describe-address-transfers command) for 14 days after the transfers have been accepted.
- AWS does not notify transfer accounts about pending Elastic IP address transfer requests. The owner of the source account must notify the owner of the transfer account that there is an Elastic IP address transfer request that they must accept.
- Any tags that are associated with an Elastic IP address being transferred are reset when the transfer is complete.
- You cannot transfer Elastic IP addresses allocated from public IPv4 address pools that you bring to your AWS account – commonly referred to as Bring Your Own IP (BYOIP) address pools.
- You cannot transfer Elastic IP addresses allocated from an Amazon-provided contiguous public IPv4 Amazon VPC IP Address Manager (IPAM) pool. Instead, IPAM allows you to share IPAM pools across AWS accounts by integrating IPAM with AWS Organizations and using AWS RAM. For more

information, see Allocate sequential Elastic IP addresses from an IPAM pool in the *Amazon VPC IPAM User Guide*.

- If you attempt to transfer an Elastic IP address that has a reverse DNS record associated with it, you can begin the transfer process, but the transfer account will not be able to accept the transfer until the associated DNS record is removed.
- If you have enabled and configured AWS Outposts, you might have allocated Elastic IP addresses from a customer-owned IP address pool (CoIP). You cannot transfer Elastic IP addresses allocated from a CoIP. However, you can use AWS RAM to share a CoIP with another account. For more information, see Customer-owned IP addresses in the *AWS Outposts User Guide*.
- You can use Amazon VPC IPAM to track the transfer of Elastic IP addresses to accounts in an organization from AWS Organizations. For more information, see View IP address history. If an Elastic IP address is transferred to an AWS account outside of the organization, the IPAM audit history of the Elastic IP address is lost.

These steps must be completed by the source account.

Console

To enable Elastic IP address transfer

1. Ensure that you're using the source AWS account.
2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the navigation pane, choose **Elastic IPs**.
4. Select one or more Elastic IP address to enable for transfer and choose **Actions**, **Enable transfer**.
5. If you are transferring multiple Elastic IP addresses, you’ll see the **Transfer type** option. Choose one of the following options:
   - Choose **Single account** if you are transferring the Elastic IP addresses to a single AWS account.
   - Choose **Multiple accounts** if you are transferring the Elastic IP addresses to multiple AWS accounts.
6. Under **Transfer account ID**, enter the IDs of the AWS accounts that you want to transfer the Elastic IP addresses to.
7. Confirm the transfer by entering `enable` in the text box.

8. Choose **Submit**.
9. To accept the transfer, see Accept a transferred Elastic IP address. To disable the transfer, see Disable Elastic IP address transfer.

AWS CLI

To enable Elastic IP address transfer

Use the enable-address-transfer command.

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

PowerShell

To enable Elastic IP address transfer

Use the Enable-EC2AddressTransfer cmdlet.

```
Enable-EC2AddressTransfer `
    -AllocationId eipalloc-09ad461b0d03f6aaf `
    -TransferAccountId 123456789012
```

Accept a transferred Elastic IP address

This section describes how to accept a transferred Elastic IP address.

When you transfer an Elastic IP address, there is a two-step handshake between the AWS accounts. When the source account starts the transfer, the transfer accounts have seven days to accept the Elastic IP address transfer. During those seven days, the source account can view the pending transfer (for example in the AWS console or by using the describe-address-transfers command). After seven days, the transfer expires and ownership of the Elastic IP address returns to the source account.

When accepting transfers, note the following exceptions that might occur and how to resolve them:

- **AddressLimitExceeded**: If your transfer account has exceeded the Elastic IP address quota, the source account can enable Elastic IP address transfer, but this exception occurs when the transfer account tries to accept the transfer. By default, all AWS accounts are limited to 5 Elastic IP addresses per Region. See Elastic IP address quota for instructions on increasing the limit.

- **InvalidTransfer.AddressCustomPtrSet**: If you or someone in your organization has configured the Elastic IP address that you are attempting to transfer to use reverse DNS lookup, the source account can enable transfer for the Elastic IP address, but this exception occurs when the transfer account tries to accept the transfer. To resolve this issue, the source account must remove the DNS record for the Elastic IP address. For more information, see Create a reverse DNS record for email on Amazon EC2.
- **InvalidTransfer.AddressAssociated**: If an Elastic IP address is associated with an ENI or EC2 instance, the source account can enable transfer for the Elastic IP address, but this exception occurs when the transfer account tries to accept the transfer. To resolve this issue, the source account must disassociate the Elastic IP address. For more information, see Disassociate an Elastic IP address.

For any other exceptions, contact Support.

These steps must be completed by the transfer account.

Console

To accept an Elastic IP address transfer

1. Ensure that you're using the transfer account.
2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the navigation pane, choose **Elastic IPs**.
4. Choose **Actions**, **Accept transfer**.
5. No tags that are associated with the Elastic IP address being transferred are transferred with the Elastic IP address when you accept the transfer. If you want to define a **Name** tag for the Elastic IP address that you are accepting, select **Create a tag with a key of 'Name' and a value that you specify**.
6. Enter the Elastic IP address that you want to transfer.
7. If you are accepting multiple transferred Elastic IP addresses, choose **Add address** to enter an additional Elastic IP address.
8. Choose **Submit**.

AWS CLI

To accept an Elastic IP address transfer

Use the accept-address-transfer command.

```
aws ec2 accept-address-transfer --address 100.21.184.216
```

PowerShell

To accept an Elastic IP address transfer

Use the Approve-EC2AddressTransfer cmdlet.

```
Approve-EC2AddressTransfer -Address 100.21.184.216
```

Disable Elastic IP address transfer

This section describes how to disable an Elastic IP transfer after the transfer has been enabled.

These steps must be completed by the source account that enabled the transfer.

Console

To disable an Elastic IP address transfer

1. Ensure that you're using the source AWS account.
2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the navigation pane, choose **Elastic IPs**.
4. In the resource list of Elastic IPs, ensure that you have the property enabled that shows the column **Transfer status**.
5. Select one or more Elastic IP address that have a **Transfer status** of **Pending**, and choose **Actions**, **Disable transfer**.
6. Confirm by entering **`disable`** in the text box.
7. Choose **Submit**.

AWS CLI

To disable Elastic IP address transfer

Use the disable-address-transfer command.

```
aws ec2 disable-address-transfer --allocation-id eipalloc-09ad461b0d03f6aaf
```

PowerShell

To disable Elastic IP address transfer

Use the Disable-EC2AddressTransfer cmdlet.

```
Disable-EC2AddressTransfer -AllocationId eipalloc-09ad461b0d03f6aaf
```

Release an Elastic IP address

If you no longer need an Elastic IP address, we recommend that you release it. The Elastic IP address to release must not be currently associated with an AWS resource.

Console

To release an Elastic IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to release and choose **Actions**, **Release Elastic IP addresses**.
4. Choose **Release**.

AWS CLI

To release an Elastic IP address

Use the release-address AWS CLI command.

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

PowerShell

To release an Elastic IP address

Use the Remove-EC2Address cmdlet.

```
Remove-EC2Address -AllocationId eipalloc-64d5890a
```

After you release your Elastic IP address, you might be able to recover. The following rules apply:

- You can't 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 can't recover tags associated with an Elastic IP address.

AWS CLI

To recover an Elastic IP address

Use the allocate-address command.

```
aws ec2 allocate-address \
    --domain vpc \
    --address 203.0.113.3
```

PowerShell

To recover an Elastic IP address

Use the New-EC2Address cmdlet.

```
New-EC2Address `
    -Address 203.0.113.3 `
    -Domain vpc `
    -Region us-east-1
```

Create a reverse DNS record for email on Amazon EC2

If you intend to send email to third parties from an EC2 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. 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.

Considerations

- 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.
- 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.
- If you contacted Support to set up reverse DNS for an Elastic IP address, you can remove the reverse DNS, but you can't release the Elastic IP address because it is locked by Support. To unlock the Elastic IP address, contact AWS Support. After the Elastic IP address is unlocked, you can release it.
- [AWS GovCloud (US) Region] You can't create a reverse DNS record. AWS must assign the static reverse DNS records for you. Open a support case to remove reverse DNS and email sending limitations. You must provide your Elastic IP addresses and reverse DNS records.

Create a reverse DNS record

You can create a reverse DNS record for your Elastic IP address as follows.

Console

To create a reverse DNS record

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**, **Update reverse DNS**.
4. For **Reverse DNS domain name**, enter the domain name.
5. Enter **`update`** to confirm.
6. Choose **Update**.

AWS CLI

To create a reverse DNS record

Use the modify-address-attribute command.

```
aws ec2 modify-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
```

```
    --domain-name example.com
```

The following is example output.

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net.",
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

To create a reverse DNS record

Use the Edit-EC2AddressAttribute cmdlet.

```
Edit-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -DomainName 'example.com' |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

The following is example output.

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.net.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.com.
```

Remove a reverse DNS record

You can remove a reverse DNS record from your Elastic IP address as follows.

If you receive the following error, you can submit a Request to remove email sending restrictions to Support for assistance.

```
The address cannot be released because it is locked to your account.
```

Console

To remove a reverse DNS record

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**, **Update reverse DNS**.
4. For **Reverse DNS domain name**, clear the domain name.
5. Enter **`update`** to confirm.
6. Choose **Update**.

AWS CLI

To remove a reverse DNS record

Use the reset-address-attribute command.

```
aws ec2 reset-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --attribute domain-name
```

The following is example output.

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com.",
            "PtrRecordUpdate": {
                "Value": "example.net.",
```

```
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

To remove a reverse DNS record

Use the Reset-EC2AddressAttribute cmdlet.

```
Reset-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -Attribute domain-name |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

The following is example output.

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.com.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.net.
```

Elastic network interfaces

An *elastic network interface* is a logical networking component in a VPC that represents a virtual network card. You can create and configure network interfaces and attach them to instances that you launch in the same Availability Zone. 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 from the original instance to the new instance.

Note that 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".

Network interface attributes

A network interface can include the following attributes:

- A primary private IPv4 address from the IPv4 address range of your subnet
- A primary IPv6 address from the IPv6 address range of your subnet
- Secondary private IPv4 addresses from the IPv4 address range of your subnet
- One Elastic IP address (IPv4) for each private IPv4 address
- One public IPv4 address
- Secondary IPv6 addresses
- Security groups
- A MAC address
- A source/destination check flag
- A description

Monitoring traffic

You can enable a VPC flow log on your network interface to capture information about the 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*.

Contents

- Prefix delegation for Amazon EC2 network interfaces
- Delete a network interface

Network interface concepts

The following are important concepts to understand as you get started using network interfaces.

Primary network interface

Each instance has a default network interface, called the *primary network interface*. You can't detach a primary network interface from an instance.

Secondary network interfaces

You can create and attach secondary network interfaces to your instance. The maximum number of network interfaces varies by instance type. For more information, see Maximum IP addresses per network interface.

IPv4 addresses for network interfaces

When you launch an EC2 instance into an IPv4-only or dual stack subnet, the instance receives a primary private IP address from the IPv4 address range of the subnet. 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.

Public IPv4 addresses for network interfaces

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 Subnet settings in the *Amazon VPC User Guide*. When you launch an instance, the IP address is assigned to the primary network interface. If you specify an existing network interface as the primary network interface when you launch an instance, the public IPv4 address is determined by this network interface.

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.

We release the public IP address when the instance is stopped, hibernated, or terminated. We assign a new public IP address when you start your stopped or hibernated instance, unless it has a secondary network interface or a secondary private IPv4 address that is associated with an Elastic IP address.

IPv6 addresses for network interfaces

If you associate IPv6 CIDR blocks with your VPC and subnet, you can assign 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. When you launch an instance, the IPv6 address is assigned to the primary network interface.

Elastic IP addresses for network interfaces

You can associate an Elastic IP address 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 or associate it with a different instance.

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.

Requester-managed network interfaces

These network interfaces are created and managed by AWS services to enable you to use some resources and services. You can't manage these network interfaces yourself. For more information, see Requester-managed network interfaces.

Prefix delegation

A 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.

Managed network interfaces

A *managed network interface* is managed by a service provider, such as Amazon EKS Auto Mode. You can't directly modify the settings of a managed network interface. Managed network interface are identified by a **true** value in the **Managed** field. For more information, see Amazon EC2 managed instances.

Network cards

Most instance types support one network card. Instance types that support multiple network cards provide higher network performance, including bandwidth capabilities above 100 Gbps and improved packet rate performance. When you attach a network interface to an instance that supports multiple network cards, you can select the network card for the network interface. The primary network interface must be assigned to network card index 0.

EFA and EFA-only network interfaces count as a network interface. You can assign only one EFA or EFA-only network interface per network card. The primary network interface can't be an EFA-only network interface.

The following instance types support multiple network cards. For information about the number of network interfaces that an instance type supports, see Maximum IP addresses per network interface.

Instance type	Number of network cards
c6in.32xlarge	2
c6in.metal	2
c8gn.48xlarge	2
c8gn.metal-48xl	2
dl1.24xlarge	4
g6e.24xlarge	2
g6e.48xlarge	4
hpc6id.32xlarge	2

Instance type	Number of network cards
hpc7a.12xlarge	2
hpc7a.24xlarge	2
hpc7a.48xlarge	2
hpc7a.96xlarge	2
m6idn.32xlarge	2
m6idn.metal	2
m6in.32xlarge	2
m6in.metal	2
p4d.24xlarge	4
p4de.24xlarge	4
p5.48xlarge	32
p5e.48xlarge	32
p5en.48xlarge	16
p6-b200.48xlarge	8
p6e-gb200.36xlarge	17
r6idn.32xlarge	2
r6idn.metal	2
r6in.32xlarge	2
r6in.metal	2
trn1.32xlarge	8

Instance type	Number of network cards
trn1n.32xlarge	16
trn2.48xlarge	16
trn2u.48xlarge	16
u7in-16tb.224xlarge	2
u7in-24tb.224xlarge	2
u7in-32tb.224xlarge	2
u7inh-32tb.480xlarge	2

Maximum IP addresses per network interface

Each instance type supports a maximum number of network interfaces, maximum number of private IPv4 addresses per network interface, and maximum number of IPv6 addresses per network interface. The limit for IPv6 addresses is separate from the limit for private IPv4 addresses per network interface. Note that all instance types support IPv6 addressing except for the following: C1, M1, M2, M3, and T1.

Available network interfaces

The *Amazon EC2 Instance Types Guide* provides the information about the network interfaces available for each instance type. For more information, see the following:

- Network specifications – General purpose
- Network specifications – Compute optimized
- Network specifications – Memory optimized
- Network specifications – Storage optimized
- Network specifications – Accelerated computing
- Network specifications – High-performance computing
- Network specifications – Previous generation

Console

To retrieve the maximum network interfaces

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instance Types**.
3. Add a filter to specify the instance type (**Instance type=c5.12xlarge**) or instance family (**Instance family=c5**).
4. (Optional) Click the **Preferences** icon and then turn on **Maximum number of network interfaces**. This column indicates the maximum number of network interfaces for each instance type.
5. (Optional) Select the instance type. On the **Networking** tab, find **Maximum number of network interfaces**.

AWS CLI

To retrieve the maximum network interfaces

You can use the [describe-instance-types](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-types.html) command to display information about an instance type, such as its supported network interfaces and IP addresses per interface. The following example displays this information for all C5 instances.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c5.*" \
    --query "InstanceTypes[].{ \
        Type: InstanceType, \
        MaxENI: NetworkInfo.MaximumNetworkInterfaces, \
        IPv4addr: NetworkInfo.Ipv4AddressesPerInterface}" \
    --output table
```

The following is example output.

```
---------------------------------------
|        DescribeInstanceTypes        |
+----------+----------+---------------+
| IPv4addr | MaxENI   |     Type      |
+----------+----------+---------------+
|  30      |  8       |  c5.4xlarge   |
|  50      |  15      |  c5.24xlarge  |
```

```
|  15       |  4        |  c5.xlarge    |
|  30       |  8        |  c5.12xlarge  |
|  10       |  3        |  c5.large     |
|  15       |  4        |  c5.2xlarge   |
|  50       |  15       |  c5.metal     |
|  30       |  8        |  c5.9xlarge   |
|  50       |  15       |  c5.18xlarge  |
+----------+----------+---------------+
```

PowerShell

To retrieve the maximum network interfaces

You can use the Get-EC2InstanceType PowerShell command to display information about an instance type, such as its supported network interfaces and IP addresses per interface. The following example displays this information for all C5 instances.

```
Get-EC2InstanceType -Filter @{Name="instance-type"; Values="c5.*"} | `
Select-Object `
    @{Name='Ipv4AddressesPerInterface';
 Expression={($_.Networkinfo.Ipv4AddressesPerInterface)}},
    @{Name='MaximumNetworkInterfaces';
 Expression={($_.Networkinfo.MaximumNetworkInterfaces)}},
    InstanceType | `
Format-Table -AutoSize
```

The following is example output.

```
Ipv4AddressesPerInterface MaximumNetworkInterfaces InstanceType
------------------------- ------------------------ ------------
                       30                        8 c5.4xlarge
                       15                        4 c5.xlarge
                       30                        8 c5.12xlarge
                       50                       15 c5.24xlarge
                       30                        8 c5.9xlarge
                       50                       15 c5.metal
                       15                        4 c5.2xlarge
                       10                        3 c5.large
                       50                       15 c5.18xlarge
```

Create a network interface for your EC2 instance

You can create a network interface for use by your EC2 instances. When you create a network interface, you specify the subnet for which it is created. You can't move a network interface to another subnet after it's created. You must attach a network interface to an instance in the same Availability Zone. You can detach a secondary network interface from an instance and then attach it to a different instance in the same Availability Zone. You can't detach a primary network interface from an instance. For more information, see the section called “Network interface attachments”.

Console

To create 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. Choose **Create network interface**.
4. (Optional) For **Description**, enter a descriptive name.
5. For **Subnet**, select a subnet. The options available in the subsequent steps change depending on the type of subnet you select (IPv4-only, IPv6-only, or dual-stack (IPv4 and IPv6)).
6. For **Interface type**, choose one of the following:
   - **ENA**: A high-performance network interface designed to handle high throughput and packet-per-second rates for TCP/IP protocols while minimizing CPU usage. This is the default value. For more information about ENA, see Elastic Network Adapter.
   - **EFA with ENA**: A network interface that supports both ENA and EFA devices for traditional TCP/IP based transport along with SRD based transport. If you choose EFA with ENA, the instance you are attaching it to must support EFA. For more information about EFA, see Elastic Fabric Adapter.
   - **EFA-only**: A high-performance network interface designed to handle high throughput, low latency inter-node communication for SRD based transport while bypassing the operating system stack. If you choose this option, the instance you are attaching it to must support EFA. EFA-only network interfaces do not support IP addresses. For more information about EFA, see Elastic Fabric Adapter.
7. 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.

8. (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.
9. (Optional) If you're creating a network interface in a dual-stack or IPv6-only subnet, you have the option to **Assign Primary IPv6 IP**. This assigns a primary IPv6 global unicast address (GUA) to the network interface. Assigning a primary IPv6 address enables you to avoid disrupting traffic to instances or ENIs. Choose **Enable** if the instance that this ENI will be attached to relies on its IPv6 address not changing. AWS will automatically assign an IPv6 address associated with the ENI attached to your instance to be the primary IPv6 address. Once you enable an IPv6 GUA address to be a primary IPv6, you can't disable it. When you enable an IPv6 GUA address to be a primary IPv6, the first IPv6 GUA will be made the primary IPv6 address until the instance is terminated or the network interface is detached. If you have multiple IPv6 addresses associated with an ENI attached to your instance and you enable a primary IPv6 address, the first IPv6 GUA address associated with the ENI becomes the primary IPv6 address.
10. (Optional) To create an Elastic Fabric Adapter, choose **Elastic Fabric Adapter**, **Enable**.
11. (Optional) Under **Advanced settings**, you can optionally set IP prefix delegation. For more information, see [Prefix delegation](#).

    - **Auto-assign** — AWS chooses the prefix from the IPv4 or IPv6 CIDR blocks for the subnet, and assigns it to the network interface.
    - **Custom** — You specify the prefix from the IPv4 or IPv6 CIDR blocks for the subnet, and AWS verifies that the prefix is not already assigned to other resources before assigning it to the network interface.
12. (Optional) Under **Advanced settings**, for **Idle connection tracking timeout**, modify the default idle connection timeouts. For more information, see [Idle connection tracking timeout](#).

    - **TCP established timeout**: Timeout (in seconds) for idle TCP connections in an established state.
      - Min: `60` seconds
      - Max: `432000` seconds

    - Default: `350` seconds for [Nitrov6](#) instance types, excluding P6e-GB200. And `432000` seconds for other instance types, including P6e-GB200.
    - Recommended: Less than `432000` seconds
  - **UDP timeout**: Timeout (in seconds) for idle UDP flows that have seen traffic only in a single direction or a single request-response transaction.
    - Min: `30` seconds
    - Max: `60` seconds
    - Default: `30` seconds
  - **UDP stream timeout**: Timeout (in seconds) for idle UDP flows classified as streams which have seen more than one request-response transaction.
    - Min: `60` seconds
    - Max: `180` seconds
    - Default: `180` seconds

13. For **Security groups**, select one or more security groups.
14. (Optional) For each tag, choose **Add new tag** and enter a tag key and an optional tag value.
15. Choose **Create network interface**.

AWS CLI

Example 1: To create a network interface with IP addresses chosen by Amazon EC2

Use the following create-network-interface command. This example creates a network interface with a public IPv4 address and an IPv6 address chosen by Amazon EC2.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --ipv6-address-count 1 \
    --groups sg-1234567890abcdef0
```

Example 2: To create a network interface with specific IP addresses

Use the following create-network-interface command.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
```

```
    --description "my dual-stack network interface" \
    --private-ip-address 10.251.50.12 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 \
    --groups sg-1234567890abcdef0
```

Example 3: To create a network interface with a count of secondary IP addresses

Use the following create-network-interface command. In this example, Amazon EC2 chooses both the primary IP address and the secondary IP addresses.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --secondary-private-ip-address-count 2 \
    --groups sg-1234567890abcdef0
```

Example 4: To create a network interface with a specific secondary IP address

Use the following create-network-interface command. This example specifies a primary IP address and a secondary IP address.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --private-ip-addresses PrivateIpAddress=10.0.1.30,Primary=true \
                           PrivateIpAddress=10.0.1.31,Primary=false
    --groups sg-1234567890abcdef0
```

PowerShell

Example 1: To create a network interface with IP addresses chosen by Amazon EC2

Use the New-EC2NetworkInterface cmdlet. This example creates a network interface with a public IPv4 address and an IPv6 address chosen by Amazon EC2.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -Ipv6AddresCount 1 `
    -Group sg-1234567890abcdef0
```

Example 2: To create a network interface with specific IP addresses

Use the New-EC2NetworkInterface cmdlet.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -PrivateIpAddress 10.251.50.12 `
    -Ipv6Address $ipv6addr `
    -Group sg-1234567890abcdef0
```

Define the IPv6 addresses as follows.

```
$ipv6addr = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
```

Example 3: To create a network interface with a count of secondary IP addresses

Use the New-EC2NetworkInterface cmdlet. In this example, Amazon EC2 chooses both the primary IP address and the secondary IP addresses.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -SecondaryPrivateIpAddressCount 2 `
    -Group sg-1234567890abcdef0
```

Example 4: To create a network interface with a specific secondary IP address

Use the New-EC2NetworkInterface cmdlet. This example specifies a primary IP address and a secondary IP address.

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -PrivateIpAddresses @($primary, $secondary) `
    -Group sg-1234567890abcdef0
```

Define the secondary addresses as follows.

```
$primary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$primary.PrivateIpAddress = "10.0.1.30"
```

```
$primary.Primary = $true
$secondary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$secondary.PrivateIpAddress = "10.0.1.31"
$secondary.Primary = $false
```

Network interface attachments for your EC2 instance

You can create network interfaces to be used by your EC2 instances as primary or secondary network interfaces. You must attach a network interface to an EC2 instance if it is in the same Availability Zone as the network interface. The instance type of an instance determines how many network interfaces you can attach to the instance. For more information, see the section called "IP addresses per network interface".

Considerations

- 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 detach a secondary network interface from one instance and attach it to another instance.
- When launching an instance using the CLI, API, or an SDK, you can specify the primary network interface and additional network interfaces. Note that you can't enable the auto-assignment of public IPv4 addresses if you add a secondary network interface during launch.
- 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 can't 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 multiple network interfaces from the same subnet to an instance, you might encounter networking issues such as asymmetric routing. If possible, add a secondary private IPv4 address on the primary network interface instead.

- For EC2 instances in an IPv6-only subnet, if you attach a secondary network interface, the private DNS hostname of the secondary network interface resolves to the primary IPv6 address for the primary network interface.
- [Windows instances] – If you add multiple network interfaces to an instance, you must configure the network interfaces to use static routing.

Attach a network interface

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.

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.

Console

To attach a network interface 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. Choose a VPC. The network interface can reside in the same VPC as your instance or in a different VPC that you own, as long as the network interface is in the same Availability Zone as the instance. This enables you to create multi-homed instances across VPCs with different networking and security configurations.
6. Select a network interface. If the instance supports multiple network cards, you can choose a network card.
7. Choose **Attach**.

To attach a network interface 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**.

AWS CLI

To attach a network interface

Use the following attach-network-interface command.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

PowerShell

To attach a network interface

Use the Add-EC2NetworkInterface cmdlet.

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1
```

Detach a network interface

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.

Console

To detach a network interface 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.
4. Choose **Actions**, **Networking**, **Detach network interface**.
5. Select the network interface and choose **Detach**.

To detach a network interface 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.

AWS CLI

To detach a network interface

Use the following detach-network-interface command.

```
aws ec2 detach-network-interface --attachment-id eni-attach-016c93267131892c9
```

PowerShell

To detach a network interface

Use the Dismount-EC2NetworkInterface cmdlet.

```
Dismount-EC2NetworkInterface -AttachmentId eni-attach-016c93267131892c9
```

Manage the IP addresses for your network interface

You can manage the following IP addresses for your network interfaces:

- Elastic IP addresses (one per private IPv4 address)
- IPv4 addresses
- IPv6 addresses

Console

To manage the IP addresses of 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 checkbox for the network interface.
4. To manage the IPv4 and IPv6 addresses, do the following:

   a. Choose **Actions**, **Manage IP addresses**.
   b. Expand the network interface.
   c. For **IPv4 addresses**, edit the IP addresses as needed. To assign an additional IPv4 address, choose **Assign new IP address** and then specify an IPv4 address from the subnet range or let AWS choose one for you.
   d. (Dual stack or IPv6 only) For **IPv6 addresses**, edit the IP addresses as needed. To assign an additional IPv6 address, choose **Assign new IP address** and then specify an IPv6 address from the subnet range or let AWS choose one for you.
   e. To assign or unassign a public IPv4 address to a network interface, choose **Auto-assign public IP**. You can enable or disable this for any network interface, but it only affects the primary network interface.

f. (Dual stack or IPv6-only) For **Assign primary IPv6 IP**, choose **Enable** to assign a primary IPv6 address. The first GUA associated with the network interface is chosen as the primary IPv6 address. After you assign a primary IPv6 address, you can't change it. This address is the primary IPv6 address until the instance is terminated or the network interface is detached.

g. Choose **Save**.

5. 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 reassociated** if the network interface is currently associated with another instance or network interface.
   e. Choose **Associate**.

6. 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**.

AWS CLI

To manage the IPv4 addresses

Use the following assign-private-ip-addresses command to assign an IPv4 address.

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

Use the following unassign-private-ip-addresses command to unassign an IPv4 address.

```
aws ec2 unassign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

To manage the IPv6 addresses

Use the following assign-ipv6-addresses command to assign an IPv6 address.

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

Use the following unassign-ipv6-addresses command to unassign an IPv6 address.

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

To manage the Elastic IP address for the primary private IPv4 address

Use the following associate-address command to associate an Elastic IP address with the primary private IPv4 address.

```
aws ec2 associate-address \
    --allocation-id eipalloc-0b263919b6EXAMPLE \
    --network-interface-id eni-1234567890abcdef0
```

Use the following disassociate-address command to disassociate an Elastic IP address from the primary private IPv4 address.

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745a1EXAMPLE
```

PowerShell

To manage the IPv4 addresses

Use the Register-EC2PrivateIpAddress cmdlet to assign an IPv4 address.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

Use the Unregister-EC2PrivateIpAddress cmdlet to unassign an IPv4 address.

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

To manage the IPv6 addresses

Use the Register-EC2Ipv6AddressList cmdlet to assign an IPv6 address.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

Use the Unregister-EC2Ipv6AddressList cmdlet to unassign an IPv6 address.

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

To manage the Elastic IP address for the primary private IPv4 address

Use the Register-EC2Address cmdlet to associate an Elastic IP address with the primary private IPv4 address.

```
Register-EC2Address `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -AllocationId eipalloc-0b263919b6EXAMPLE
```

Use the Unregister-EC2Address cmdlet to disassociate an Elastic IP address from the primary private IPv4 address.

```
Unregister-EC2Address -AssociationId eipassoc-2bebb745a1EXAMPLE
```

Modify network interface attributes

You can change the following network interface attributes:

- Description

- Security groups
- Delete on termination
- Source/destination check
- Idle connection tracking timeout

Considerations

You can't change the attributes of a requester-managed network interface.

Console

To modify network interface attributes

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 change the description, do the following
   a. Choose **Actions**, **Change description**.
   b. For **Description**, enter a description.
   c. Choose **Save**.
5. To change the security groups, do the following:
   a. Choose **Actions**, **Change security groups**.
   b. For **Associated security groups**, add and remove security groups as needed. The security group and network interface must be created for the same VPC.
   c. Choose **Save**.
6. To change the termination behavior, do the following:
   a. Choose **Actions**, **Change termination behavior**.
   b. Select or clear **Delete on termination**, **Enable**.
   c. Choose **Save**.
7. To change source/destination checking, do the following:
   a. Choose **Actions**, **Change source/dest check**.
   b. Select or clear **Source/destination check**, **Enable**.

c. Choose **Save**.

8. To change idle connection tracking timeouts, do the following:

   a. Choose **Actions**, **Modify idle connection tracking timeout**.

   b. Modify timeout values as needed. For more information, see Idle connection tracking timeout.

      - **TCP established timeout**: Timeout (in seconds) for idle TCP connections in an established state.
        - Min: `60` seconds
        - Max: `432000` seconds
        - Default: `350` seconds for Nitrov6 instance types, excluding P6e-GB200. And `432000` seconds for other instance types, including P6e-GB200.
        - Recommended: Less than `432000` seconds
      - **UDP timeout**: Timeout (in seconds) for idle UDP flows that have seen traffic only in a single direction or a single request-response transaction.
        - Min: `30` seconds
        - Max: `60` seconds
        - Default: `30` seconds
      - **UDP stream timeout**: Timeout (in seconds) for idle UDP flows classified as streams which have seen more than one request-response transaction.
        - Min: `60` seconds
        - Max: `180` seconds
        - Default: `180` seconds

   c. Choose **Save**.

AWS CLI

Example Example: To modify the description

Use the following modify-network-interface-attribute command.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --description "my updated description"
```

Example Example: To modify the security groups

Use the following modify-network-interface-attribute command.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

Example Example: To modify the termination behavior

Use the following modify-network-interface-attribute command.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --attachment AttachmentId=eni-attach-43348162abEXAMPLE,DeleteOnTermination=false
```

Example Example: To enable source/destination checking

Use the following modify-network-interface-attribute command.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --source-dest-check
```

Example Example: To modify idle connection tracking timout

Use the following modify-network-interface-attribute command. For more information, see Idle connection tracking timeout.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --connection-tracking-specification
 TcpEstablishedTimeout=172800,UdpStreamTimeout=90,UdpTimeout=60
```

PowerShell

Example Example: To modify the description

Use the Edit-EC2NetworkInterfaceAttribute cmdlet.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
```

```
    -Description "my updated description"
```

Example Example: To modify the security groups

Use the Edit-EC2NetworkInterfaceAttribute cmdlet.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

Example Example: To modify the termination behavior

Use the Edit-EC2NetworkInterfaceAttribute cmdlet.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Attachment_AttachmentId eni-attach-43348162abEXAMPLE `
    -Attachment_DeleteOnTermination $false
```

Example Example: To enable source/destination checking

Use the Edit-EC2NetworkInterfaceAttribute cmdlet.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SourceDestCheck $true
```

Example Example: To modify idle connection tracking timeouts

Use the Edit-EC2NetworkInterfaceAttribute cmdlet. For more information, see Idle connection tracking timeout.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -ConnectionTrackingSpecification_TcpEstablishedTimeout 172800 `
    -ConnectionTrackingSpecification_UdpStreamTimeout 90 `
    -ConnectionTrackingSpecification_UdpTimeout 60
```

Multiple network interfaces for your Amazon EC2 instances

Attaching multiple network interfaces to an instance is useful when you need the following:

- A management network.
- Network and security appliances.
- Dual-homed instances with workloads in different subnets or VPCs.
- A low-budget, high-availability solution.

Management network

The following overview describes a management network created using multiple network interfaces.

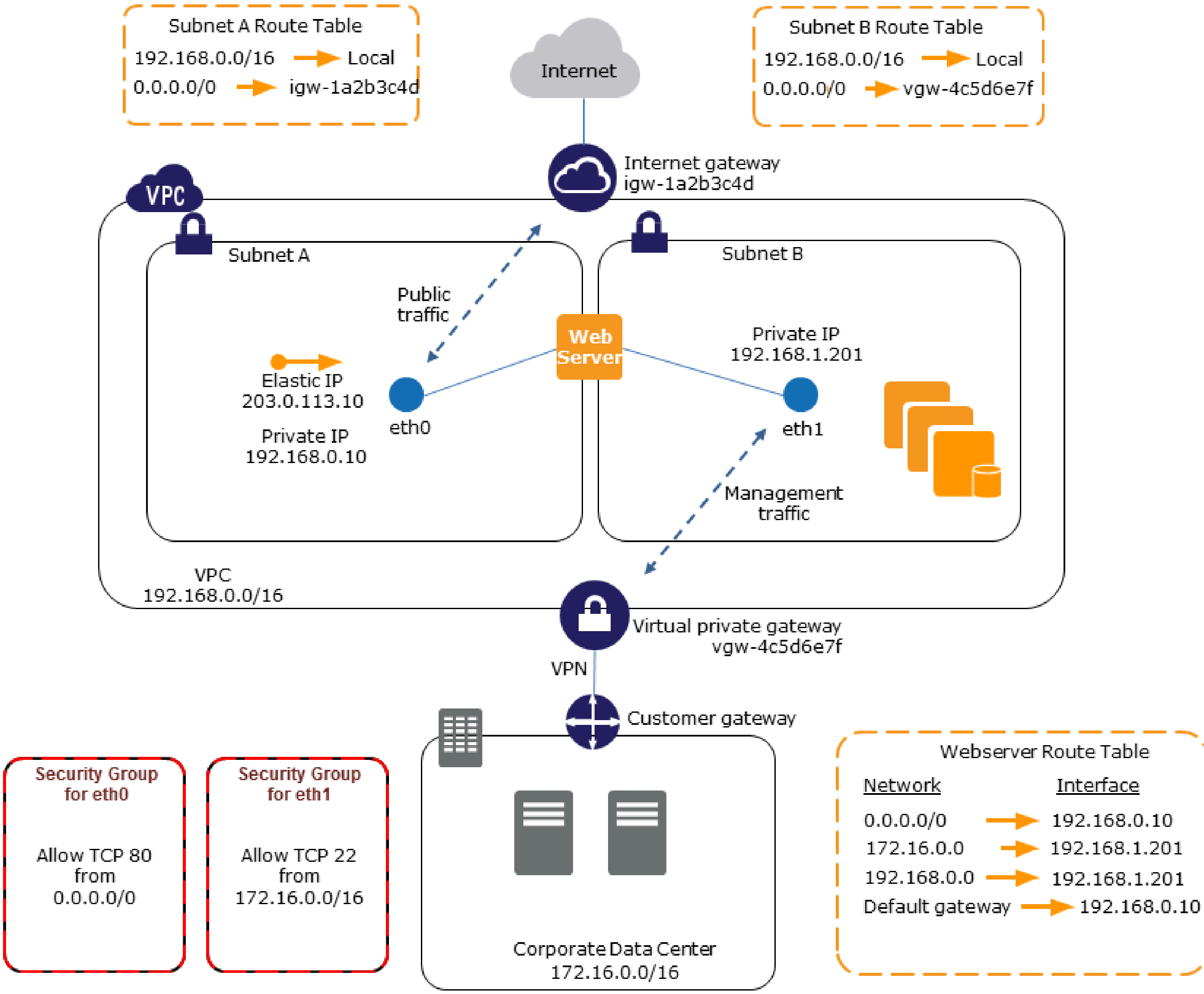
Criteria

- The primary network interface on the instance (for example, eth0) handles public traffic.
- The secondary network interface on the instance (for example, eth1) handles backend management traffic. It's connected to a separate subnet that has more restrictive access controls, and is located within the same Availability Zone as the primary network interface.

Settings

- The primary network 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 secondary network interface has an associated security group that allows SSH access only, initiated from one of the following locations:
  - An allowed range of IP addresses, either within the VPC, or from the internet.
  - A private subnet within the same Availability Zone as the primary network interface.
  - A virtual private gateway.

> **Note**
>
> 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.



Network and security appliances

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.

Dual-homed instances with workloads in different 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.

Dual-homed instances with workloads in different VPCs in the same account

You can launch an EC2 instance in one VPC and attach a secondary ENI from a different VPC, as long as the network interface is in the same Availability Zone as the instance. This enables you to create multi-homed instances across VPCs with different networking and security configurations. You can't create multi-homed instances across VPCs in different AWS accounts.

You can use dual-homed instances across VPCs in the following use cases:

- **Overcome CIDR overlaps between two VPCs that can’t be peered together**: You can leverage a secondary CIDR in a VPC and allow an instance to communicate across two non-overlapping IP ranges.
- **Connect multiple VPCs within a single account**: Enable communication between individual resources that would normally be separated by VPC boundaries.

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.

Requester-managed network interfaces

A requester-managed network interface is a network interface that an AWS service creates in your VPC on your behalf. The network interface is associated with a resource for another service, such as a DB instance from Amazon RDS, a NAT gateway, or an interface VPC endpoint from AWS PrivateLink.

Considerations

- You can view the requester-managed network interfaces in your account. You can add or remove tags, but you can't change other properties of a requester-managed network interface.
- You can't detach a requester-managed network interface.
- When you delete the resource associated with a requester-managed network interface, the AWS service detaches the network interface and deletes it. If the service detached a network interface but didn't delete it, you can delete the detached network interface.

Console

To view requester-managed network interfaces

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network & Security**, **Network Interfaces**.
3. Select the ID of the network interface to open its details page.
4. The following are the key fields that you can use to determine the purpose of the network interface:

   - **Description**: A description provided by the AWS service that created the interface. For example, "VPC Endpoint Interface vpce 089f2123488812123".
   - **Requester-managed**: Indicates whether the network interface is managed by AWS.
   - **Requester ID**: The alias or AWS account ID of the principal or service that created the network interface. If you created the network interface, this is your AWS account ID. Otherwise, another principal or service created it.

AWS CLI

To view requester-managed network interfaces

Use the describe-network-interfaces command as follows.

```
aws ec2 describe-network-interfaces \
    --filters "Name=requester-managed,Values=true" \
    --query "NetworkInterfaces[*].[Description, InterfaceType]" \
    --output table
```

The following is example output that shows the key fields that you can use to determine the purpose of the network interface: `Description` and `InterfaceType`.

```
-------------------------------------------------------------------------------
|                          DescribeNetworkInterfaces                          |
+---------------------------------------------------+-------------------------+
|  VPC Endpoint Interface: vpce-0f00567fa8477a1e6   |  interface              |
|  VPC Endpoint Interface vpce-0d8ddce4be80e4474    |  interface              |
|  VPC Endpoint Interface vpce-078221a1e27d1ea5b    |  vpc_endpoint           |
|  Resource Gateway Interface rgw-0bba03f3d56060135 |  interface              |
|  VPC Endpoint Interface: vpce-0cc199f605eaeace7   |  interface              |
|  VPC Endpoint Interface vpce-019b90d6f16d4f958    |  interface              |
+---------------------------------------------------+-------------------------+
```

PowerShell

To view requester-managed network interfaces

Use the Get-EC2NetworkInterface cmdlet as follows.

```
Get-EC2NetworkInterface -Filter @{Name="requester-managed"; Values="true"} | Select
 Description, InterfaceType
```

The following is example output that shows the key fields that you can use to determine the purpose of a network interface: `Description` and `InterfaceType`.

```
Description                                        InterfaceType
-----------                                        -------------
VPC Endpoint Interface: vpce-0f00567fa8477a1e6   interface
VPC Endpoint Interface vpce-0d8ddce4be80e4474    interface
VPC Endpoint Interface vpce-078221a1e27d1ea5b    vpc_endpoint
Resource Gateway Interface rgw-0bba03f3d56060135 interface
VPC Endpoint Interface: vpce-0cc199f605eaeace7   interface
VPC Endpoint Interface vpce-019b90d6f16d4f958    interface
```

Prefix delegation for 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. For more information about IPv4 and IPv6 addresses, see Amazon EC2 instance IP addressing.

The following assignment options are available:

- **Automatic assignment** — AWS chooses the prefix and assigns it to your network interface. If the subnet for the network interface has a subnet CIDR reservation of type `prefix`, we select the prefixes from the subnet CIDR reservation. Otherwise, we select them from the subnet CIDR range.
- **Manual assignment** — You specify the prefix and AWS verifies that it 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 for 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.

Contents

Basics

- You can assign a prefix to new or existing network interfaces.
- To use prefixes, you assign a prefix to your network interface, 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 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.
- You can also assign an Elastic IP address to the IP address part of the assigned prefix.
- We resolve the private DNS host name of an instance to the primary private IPv4 address.
- We assign each private IPv4 address for a network interface, including those from prefixes, using the following format:
  - `us-east-1` Region

    ```
    ip-private-ipv4-address.ec2.internal
    ```

  - All other Regions

    ```
    ip-private-ipv4-address.region.compute.internal
    ```

Considerations

Take the following into consideration when you use prefixes:

- Network interfaces with prefixes are supported with Nitro-based instances.
- Prefixes for network interfaces must use IPv6 addresses or private IPv4 addresses.
- The maximum number of IP addresses that you can assign to a network interface depends on the instance type. Each prefix that you assign to a network interface counts as one IP address. For example, a `c5.large` instance has a limit of `10` IPv4 addresses per network interface. Each

network interface for this instance has a primary IPv4 address. If a network interface has no secondary IPv4 addresses, you can assign up to 9 prefixes to the network interface. For each additional IPv4 address that you assign to a network interface, you can assign one less prefix to the network interface. For more information, see Maximum IP addresses per network interface.

- Prefixes are included in source/destination checks.
- You must configure your operating system to work with network interfaces with prefixes. Note the following:
  - Some Amazon Linux AMIs contain additional scripts installed by AWS, known as `ec2-net-utils`. These scripts optionally automate the configuration of your network interfaces. They are for use only on Amazon Linux.
  - For containers, you can use a Container Network Interface (CNI) for the Kubernetes plug-in, or `dockerd` if you use Docker to manage your containers.

Manage prefixes for your network interfaces

When you assign prefixes to a network interface, you can choose whether to let us automatically assign the prefixes or you can specify custom prefixes. If you let us automatically assign prefixes and the subnet for the network interface has a subnet CIDR reservation of type `prefix`, we select the prefixes from the subnet CIDR reservation. Otherwise, we select them from the subnet CIDR range.

Tasks

- Assign prefixes during network interface creation
- Assign prefixes to an existing network interface
- Remove prefixes from your network interfaces

Assign prefixes during network interface creation

You can assign automatic or custom prefixes when you create a network interface.

Console

To assign automatic prefixes during network interface creation

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. Enter a description for the network interface, select the subnet in which to create the network interface, and configure the private IPv4 and IPv6 addresses.
5. Expand **Advanced settings**.
6. For **IPv4 prefix delegation** do one of the following:
   - To automatically assign an IPv4 prefix, choose **Auto-assign**. For **Number of IPv4 prefixes**, enter the number of prefixes to assign.
   - To assign a specific IPv4 prefix, choose **Custom**. Choose **Add new prefix** and enter the prefix.
7. For **IPv6 prefix delegation** do one of the following:
   - To automatically assign an IPv6 prefix, choose **Auto-assign**. For **Number of IPv6 prefixes**, enter the number of prefixes to assign.
   - To assign a specific IPv6 prefix, choose **Custom**. Choose **Add new prefix** and enter the prefix.

   > **Note**
   >
   > **IPv6 prefix delegation** appears only if the selected subnet is enabled for IPv6.

8. Select the security groups to associate with the network interface and assign resource tags if needed.
9. Choose **Create network interface**.

AWS CLI

To assign automatic IPv4 prefixes during network interface creation

Use the create-network-interface command and set `--ipv4-prefix-count` to the number of IPv4 prefixes for AWS to assign. In the following example, AWS assigns one IPv4 prefix.

```
aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv4 automatic example" \
--ipv4-prefix-count 1
```

To assign specific IPv4 prefixes during network interface creation

Use the create-network-interface command and set `--ipv4-prefixes` to the prefixes. AWS selects IPv4 addresses from this range. In the following example, the prefix CIDR is 10.0.0.208/28.

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv4 manual example" \
    --ipv4-prefixes Ipv4Prefix=10.0.0.208/28
```

To assign automatic IPv6 prefixes during network interface creation

Use the create-network-interface command and set `--ipv6-prefix-count` to the number of IPv6 prefixes for AWS to assign. In the following example, AWS assigns one IPv6 prefix.

```
aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv6 automatic example" \
--ipv6-prefix-count 1
```

To assign specific IPv6 prefixes during network interface creation

Use the create-network-interface command and set `--ipv6-prefixes` to the prefixes. AWS selects IPv6 addresses from this range. In the following example, the prefix CIDR is 2600:1f13:fc2:a700:1768::/80.

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv6 manual example" \
    --ipv6-prefixes Ipv6Prefix=2600:1f13:fc2:a700:1768::/80
```

PowerShell

To assign automatic IPv4 prefixes during network interface creation

Use the New-EC2NetworkInterface cmdlet and set `Ipv4PrefixCount` to the number of IPv4 prefixes for AWS to assign. In the following example, AWS assigns one IPv4 prefix.

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 automatic example' `
    -Ipv4PrefixCount 1
```

To assign specific IPv4 prefixes during network interface creation

Use the [New-EC2NetworkInterface] cmdlet and set `Ipv4Prefix` to the prefixes. AWS selects IPv4 addresses from this range. In the following example, the prefix CIDR is 10.0.0.208/28.

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 manual example' `
    -Ipv4Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv4PrefixSpecificationRequest `
        -Property @{Ipv4Prefix = '10.0.0.208/28'})
```

To assign automatic IPv6 prefixes during network interface creation

Use the [New-EC2NetworkInterface] cmdlet and set `Ipv6PrefixCount` to the number of IPv6 prefixes for AWS to assign. In the following example, AWS assigns one IPv6 prefix.

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 automatic example' `
    -Ipv6PrefixCount 1
```

To assign specific IPv6 prefixes during network interface creation

Use the [New-EC2NetworkInterface] cmdlet and set `Ipv6Prefixes` to the prefixes. AWS selects IPv6 addresses from this range. In the following example, the prefix CIDR is 2600:1f13:fc2:a700:1768::/80.

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 manual example' `
    -Ipv6Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv6PrefixSpecificationRequest `
        -Property @{Ipv6Prefix = '2600:1f13:fc2:a700:1768::/80'})
```

Assign prefixes to an existing network interface

You can assign automatic or custom prefixes to an existing network interface.

Console

To assign automatic prefixes to an existing 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 to which to assign the prefixes, and choose **Actions**, **Manage prefixes**.
4. For **IPv4 prefix delegation** do one of the following:
   - To automatically assign an IPv4 prefix, choose **Auto-assign**. For **Number of IPv4 prefixes**, enter the number of prefixes to assign.
   - To assign a specific IPv4 prefix, choose **Custom**. Choose **Add new prefix** and enter the prefix.
5. For **IPv6 prefix delegation** do one of the following:
   - To automatically assign an IPv6 prefix, choose **Auto-assign**. For **Number of IPv6 prefixes**, enter the number of prefixes to assign.
   - To assign a specific IPv6 prefix, choose **Custom**. Choose **Add new prefix** and enter the prefix.

   > **Note**
   >
   > **IPv6 prefix delegation** appears only if the selected subnet is enabled for IPv6.

6. Choose **Save**.

AWS CLI

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.

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 IPv4 prefixes for AWS to assign. In the following example, AWS assigns one IPv4 prefix.

```
aws ec2 assign-private-ip-addresses \
--network-interface-id eni-081fbb4095EXAMPLE \
```

```
--ipv4-prefix-count 1
```

To 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.

```
aws ec2 assign-private-ip-addresses \
--network-interface-id eni-081fbb4095EXAMPLE \
--ipv4-prefixes 10.0.0.208/28
```

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 IPv6 prefixes for AWS to assign. In the following example, AWS assigns one IPv6 prefix.

```
aws ec2 assign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefix-count 1
```

To 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.

```
aws ec2 assign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

To assign automatic IPv4 prefixes to an existing network interface

Use the Register-EC2PrivateIpAddress cmdlet and set `Ipv4PrefixCount` to the number of IPv4 prefixes for AWS to assign. In the following example, AWS assigns one IPv4 prefix.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
```

```
    -Ipv4PrefixCount 1
```

To assign specific IPv4 prefixes to an existing network interface

Use the Register-EC2PrivateIpAddress cmdlet and set `Ipv4Prefix` to the prefix. AWS selects IPv4 addresses from this range. In the following example, the prefix CIDR is 10.0.0.208/28.

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

To assign automatic IPv6 prefixes to an existing network interface

Use the Register-EC2Ipv6AddressList cmdlet and set `Ipv6PrefixCount` to the number of IPv4 prefixes for AWS to assign. In the following example, AWS assigns one IPv6 prefix.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6PrefixCount 1
```

To assign specific IPv6 prefixes to an existing network interface

Use the Register-EC2Ipv6AddressList cmdlet and set `Ipv6Prefix` to the prefix. AWS selects IPv6 addresses from this range. In the following example, the prefix CIDR is 2600:1f13:fc2:a700:18bb::/80.

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

Remove prefixes from your network interfaces

You can remove prefixes from an existing network interface.

Console

To remove the prefixes 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.
4. Choose **Actions**, **Manage prefixes**.
5. For **IPv4 prefix delegation**, to remove specific prefixes, choose **Unassign** next to the prefixes to remove. To remove all prefixes, choose **Do not assign**.
6. For **IPv6 prefix delegation**, to remove specific prefixes, choose **Unassign** next to the prefixes to remove. To remove all prefixes, choose **Do not assign**.

> **Note**
>
> **IPv6 prefix delegation** appears only if the selected subnet is enabled for IPv6.

7. Choose **Save**.

AWS CLI

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 prefix CIDR to remove.

```
aws ec2 unassign-private-ip-addresses \
--network-interface-id eni-081fbb4095EXAMPLE \
--ipv4-prefixes 10.0.0.176/28
```

To remove IPv6 prefixes from a network interface

Use the unassign-ipv6-addresses command and set `--ipv6-prefix` to the prefix CIDR to remove.

```
aws ec2 unassign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefix 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

To remove IPv4 prefixes from a network interface

Use the Unregister-EC2PrivateIpAddress cmdlet and set Ipv4Prefix to the prefix CIDR to remove.

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

To remove IPv6 prefixes from a network interface

Use the Unregister-EC2Ipv6AddressList cmdlet and set Ipv6Prefix to the prefix CIDR to remove.

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

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 can't delete a network interface that is in use. First, you must detach the network interface.

Console

To delete 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 checkbox for the network interface, and then choose **Actions**, **Delete**.
4. When prompted for confirmation, choose **Delete**.

AWS CLI

To delete a network interface

Use the following delete-network-interface command.

```
aws ec2 delete-network-interface --network-interface-id eni-1234567890abcdef0
```

PowerShell

To delete a network interface

Use the Remove-EC2NetworkInterface cmdlet.

```
Remove-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0
```

Amazon EC2 instance network bandwidth

Instance bandwidth specifications apply to both inbound and outbound traffic for the instance. For example, if an instance specifies up to 10 Gbps of bandwidth, that means it has up to 10 Gbps of bandwidth for inbound traffic and, simultaneously, up to 10 Gbps for outbound traffic. The network bandwidth that's available to an EC2 instance depends on several factors, as follows.

Multi-flow traffic

Bandwidth for multi-flow traffic is limited to 50% of the available bandwidth for traffic that goes through an internet gateway or a local gateway for instances with 32 or more vCPUs, or 5 Gbps, whichever is larger. For instances with fewer than 32 vCPUs, bandwidth is limited to 5 Gbps.

Single-flow traffic

Bandwidth for single-flow traffic is limited to 5 Gbps when instances are not in the same cluster placement group. To reduce latency and increase single-flow bandwidth, try one of the following:

- Use a cluster placement group to achieve up to 10 Gbps bandwidth for instances within the same placement group.
- Set up multiple paths between two endpoints to achieve higher bandwidth with Multipath TCP (MPTCP).
- Configure ENA Express for eligible instances within the same Availability Zone to achieve up to 25 Gbps between those instances.

> **Note**
>
> A single-flow is considered a unique 5-tuple TCP or UDP flow. For other protocols following the IP header, such as `GRE` or `IPsec`, the 3 tuple of source IP, destination IP, and next protocol is used to define a flow.

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. How much of the available bandwidth the traffic can utilize depends on the number of vCPUs and the destination. For example, an `m5.16xlarge` instance has 64 vCPUs, so traffic to another instance in the Region can utilize the full bandwidth available (20 Gbps). However, traffic that goes through an internet gateway or a local gateway can utilize only 50% of the bandwidth available (10 Gbps).

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.

There are separate network I/O credit buckets for inbound and outbound traffic.

Base and burst network performance

The *Amazon EC2 Instance Types Guide* describes the network performance for each instance type, plus the baseline network bandwidth available for instances that can use burst bandwidth. For more information, see the following:

- Network specifications – General purpose
- Network specifications – Compute optimized
- Network specifications – Memory optimized
- Network specifications – Storage optimized
- Network specifications – Accelerated computing
- Network specifications – High-performance computing

- Network specifications – Previous generation

Alternatively, you can use a command line tool to get this information. The Amazon EC2 console does not display the baseline network bandwidth of an instance type.

AWS CLI

You can use the describe-instance-types command to display information about an instance type. The following example displays network performance information for all C5 instances.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c5.*" \
    --query "InstanceTypes[].[InstanceType, NetworkInfo.NetworkPerformance,
 NetworkInfo.NetworkCards[0].BaselineBandwidthInGbps] | sort_by(@,&[2])" \
    --output table
```

The following is example output. If your output is missing the baseline bandwidth, update to the latest version of the AWS CLI.

```
--------------------------------------------
|           DescribeInstanceTypes          |
+--------------+--------------------+-------+
|  c5.large    |  Up to 10 Gigabit  |  0.75 |
|  c5.xlarge   |  Up to 10 Gigabit  |  1.25 |
|  c5.2xlarge  |  Up to 10 Gigabit  |  2.5  |
|  c5.4xlarge  |  Up to 10 Gigabit  |  5.0  |
|  c5.9xlarge  |  12 Gigabit        |  12.0 |
|  c5.12xlarge |  12 Gigabit        |  12.0 |
|  c5.18xlarge |  25 Gigabit        |  25.0 |
|  c5.24xlarge |  25 Gigabit        |  25.0 |
|  c5.metal    |  25 Gigabit        |  25.0 |
+--------------+--------------------+-------+
```

PowerShell

You can use the Get-EC2InstanceType PowerShell command to display information about an instance type. The following example displays network performance information for all C5 instances.

```
Get-EC2InstanceType -Filter @{Name = "instance-type"; Values = "c5.*" } | `
```

```
    Select-Object `
    InstanceType,
    @{Name = 'NetworkPerformance'; Expression =
 {($_.Networkinfo.NetworkCards.NetworkPerformance)}},
    @{Name = 'BaselineBandwidthInGbps'; Expression =
 {($_.Networkinfo.NetworkCards.BaselineBandwidthInGbps)}} | `
Format-Table -AutoSize
```

The following is example output.

```
InstanceType NetworkPerformance BaselineBandwidthInGbps
------------ ------------------ -----------------------
c5.4xlarge   Up to 10 Gigabit                      5.00
c5.xlarge    Up to 10 Gigabit                      1.25
c5.12xlarge  12 Gigabit                           12.00
c5.9xlarge   12 Gigabit                           12.00
c5.24xlarge  25 Gigabit                           25.00
c5.metal     25 Gigabit                           25.00
c5.2xlarge   Up to 10 Gigabit                      2.50
c5.large     Up to 10 Gigabit                      0.75
c5.18xlarge  25 Gigabit                           25.00
```

EC2 instance bandwidth weighting configuration

Some instance types support configurable bandwidth weighting, where you can select baseline bandwidth weighting that favors either network processing or EBS operations. Default settings for baseline bandwidth are determined by your instance type. You can configure the bandwidth weighting during launch, or modify your instance settings with the following weighting preferences:

- **default** – This option uses the standard bandwidth configuration for your instance type.
- **vpc-1** – This option increases the baseline bandwidth available for networking and decreases the baseline bandwidth for EBS operations.
- **ebs-1** – This option increases the baseline bandwidth available for EBS operations, and decreases the baseline bandwidth for networking.

Bandwidth weighting considerations

The following are some considerations that might affect your bandwidth weighting strategy.

- Setting bandwidth weighting preferences only affects bandwidth specifications. The network packets per second (PPS) and EBS input/output operations per second (IOPS) specifications don't change.
- The combined bandwidth specification between networking and EBS does not change. When you select a bandwidth weighting configuration, the baseline bandwidth available for the selected option increases, and the baseline bandwidth for the remaining option is reduced by the same absolute amount. For all instances except Flex instances, the available burst bandwidth remains the same for your selected option, and is reduced for the remaining option. For Flex instances up to 4xlarge, burst bandwidth remains unchanged. For Flex instances 8xlarge and larger, burst bandwidth increases by the same amount as the baseline bandwidth.
- It's important to understand how changes in bandwidth allocation can affect I/O performance for EBS. For EC2 instances that have `vpc-1` configuration (increased networking bandwidth), you might experience lower IOPS for EBS volumes if you reach the EBS bandwidth limit before you've reached the IOPS limit. This is more noticeable with larger I/O sizes.

  For example, on an instance type that normally supports 240,000 IOPS with 16 KiB I/O size, if you select `vpc-1` weighting, that might reduce the achievable IOPS due to the adjusted EBS baseline bandwidth limit.

  When planning your workload, consider your I/O size and patterns. Smaller I/O sizes are less likely to be affected by bandwidth limitations, while larger I/O sizes or sequential workloads might see more impact from bandwidth changes. Always test your specific workload to ensure optimal performance with your chosen configuration.
- The networking multi-flow bandwidth specification for traffic that goes through an internet gateway or a local gateway is adjusted to 50% of the baseline bandwidth of the configured option or 5 Gbps, where applicable. For more information, see [Amazon EC2 instance network bandwidth](#).

  The following example is based on an instance type that has a default baseline bandwidth of 40 Gbps, and a default border bandwidth of 20 Gbps. If you choose `vpc-1` bandwidth weighting for this instance, the weighted baseline bandwidth changes to 50 Gbps, and the border bandwidth changes to 25 Gbps.
- This feature is available in all commercial regions, aligned with EC2 instance availablilty and support.
- This feature adds no additional cost to your EC2 instance.

Supported instance types for bandwidth weighting

Instance types in the following instance families support configurable bandwidth weighting.

- **General purpose:** M8g, M8gd, M8i, M8i-flex
- **Compute optimized:** C8g, C8gd
- **Memory optimized:** R8g, R8gd, R8i, R8i-flex, X8g

Check current bandwidth settings

To see the current bandwidth settings for your instance, select one of the tabs for instructions.

Console

To get the bandwidth setting 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 that you want to check from the list, and navigate to the **Networking** tab. Your current setting is shown in the **Configured bandwidth** field. Amazon EC2 uses default settings for your instance type if the bandwidth is not set to a specific value.

AWS CLI

To get the bandwidth setting for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].NetworkPerformanceOptions.BandwidthWeighting
 \
    --output text
```

The following is example output.

```
default
```

This example lists all of your instances that have the bandwidth weighting preference set to vpc-1, for higher networking bandwidth.

```
aws ec2 describe-instances \
    --filters "Name=network-performance-options.bandwidth-weighting,Values=vpc-1" \
    --query Reservations[].Instances[].InstanceId \
    --output text
```

PowerShell

To get the bandwidth setting for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.NetworkPerformanceOptions.BandwidthWeighting.Value
```

The following is example output.

```
default
```

This example lists all of your instances that have the bandwidth weighting preference set to vpc-1, for higher networking bandwidth.

```
(Get-EC2Instance `
    -Filter @{Name="network-performance-options.bandwidth-
weighting";Values="vpc-1"}).Instances.InstanceId
```

Configure bandwidth weighting for your instance

You can configure bandwidth weighting either at launch or by modifying existing instances from the EC2 console, API/SDKs or CLI.

Configure bandwidth weighting when you launch an instance

To configure bandwidth settings when you launch an instance, select one of the tabs for instructions.

You can also specify bandwidth weighting in a launch template. To create a launch template, see Create an Amazon EC2 launch template. The parameter to set is in the same location as it is for launching an instance directly from the console. Expand the **Advanced details** section, and set the **Instance bandwidth configuration**.

To launch an instance with your launch template, see [Launch EC2 instances using a launch template](#).

Console

To launch an instance with configurable bandwidth weighting

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose **Launch instances**. This opens the **Launch an instance** dialog. There are several additional ways that you can get to the the launch dialog, depending on your preference. For example, you can launch an instance directly from an AMI or from the Amazon EC2 dashboard itself.
4. The Amazon Machine Image (AMI) that you launch from must be based on `Arm` architecture. Many **Quick Start** images support both `x86` and `Arm` architectures, After you choose the operating system for your instance, select the `Arm` option from the **Architecture** list.
5. The instance type must be one of the [Supported instance types](#) for this feature.
6. When you expand the **Advanced details** section, you can scroll down to find the **Instance bandwidth configuration** settings. Select the bandwidth configuration option for your instance.
7. Configure all of the other settings for your instance as you normally would, and choose **Launch instance**.

AWS CLI

To launch an instance with configurable bandwidth weighting

Use the [run-instances](#) command with the following option to launch instances that are configured for higher network bandwidth weighting.

```
--network-performance-options BandwidthWeighting=vpc-1
```

Use the [run-instances](#) command with the following option to launch instances that are configured for higher EBS bandwidth weighting.

```
--network-performance-options BandwidthWeighting=ebs-1
```

PowerShell

To launch an instance with configurable bandwidth weighting

Use the New-EC2Instance cmdlet with the following parameter to launch instances that are configured for higher network bandwidth weighting.

```
-NetworkPerformanceOptions_BandwidthWeighting vpc-1
```

Use the New-EC2Instance cmdlet with the following parameter to launch instances that are configured for higher EBS bandwidth weighting.

```
-NetworkPerformanceOptions_BandwidthWeighting ebs-1
```

Update bandwidth weighting for an existing instance

To update bandwidth weighting for an existing instance, your instance must be in the `Stopped` state.

Console

To update bandwidth weighting

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 you want to update from the list.
4. Before you change the bandwidth configuration, your instance must be in a `Stopped` state. If your instance is running, select **Stop instance** from the **Instance state** menu.
5. Choose **Manage bandwidth** from the **Actions > Networking** menu. This opens the **Manage bandwidth** dialog.

> **Note**
>
> If your instance type doesn't support configuration for bandwidth weighting, that menu item is disabled.

6. Select the option to update your instance, and choose **Change** to save your settings.

AWS CLI

To update bandwidth weighting

Use the modify-instance-network-performance-options command to configure higher network bandwidth weighting for the specified instance.

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=vpc-1
```

The following example configures higher EBS bandwidth weighting for the specified instance.

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=ebs-1
```

PowerShell

To update bandwidth weighting

Use the Edit-EC2InstanceNetworkPerformanceOption cmdlet to configure higher network bandwidth weighting for the specified instance.

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting vpc-1
```

The following example configures higher EBS bandwidth weighting for the specified instance.

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting ebs-1
```

Bandwidth weighting impact for networking

The following table shows the impact of bandwidth weighting on networking bandwidth, in Gbps, for the supported instance types.

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
c8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
c8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
c8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8g.8xlarge	15.0	18.75	12.5
c8g.12xlarge	22.5	28.125	18.75
c8g.16xlarge	30.0	37.5	25.0
c8g.24xlarge	40.0	50.0	32.5
c8g.48xlarge	50.0	62.5	40.0
c8g.metal-24xl	40.0	50.0	32.5
c8g.metal-48xl	50.0	62.5	40.0
c8gd.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
c8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
c8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8gd.8xlarge	15.0	18.75	12.5
c8gd.12xlarge	22.5	28.125	18.75

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
c8gd.16xlarge	30.0	37.5	25.0
c8gd.24xlarge	40.0	50.0	32.5
c8gd.48xlarge	50.0	62.5	40.0
c8gd.metal-24xl	40.0	50.0	32.5
c8gd.metal-48xl	50.0	62.5	40.0
m8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
m8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
m8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8g.8xlarge	15.0	18.75	12.5
m8g.12xlarge	22.5	28.125	18.75
m8g.16xlarge	30.0	37.5	25.0
m8g.24xlarge	40.0	50.0	32.5
m8g.48xlarge	50.0	62.5	40.0
m8g.metal-24xl	40.0	50.0	32.5
m8g.metal-48xl	50.0	62.5	40.0
m8gd.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
m8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
m8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8gd.8xlarge	15.0	18.75	12.5
m8gd.12xlarge	22.5	28.125	18.75
m8gd.16xlarge	30.0	37.5	25.0
m8gd.24xlarge	40.0	50.0	32.5
m8gd.48xlarge	50.0	62.5	40.0
m8gd.metal-24xl	40.0	50.0	32.5
m8gd.metal-48xl	50.0	62.5	40.0
m8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8i.8xlarge	15.0	18.75	12.5
m8i.12xlarge	22.5	28.125	18.75
m8i.16xlarge	30.0	37.5	25.0
m8i.24xlarge	40.0	50.0	32.5
m8i.32xlarge	50.0	62.5	40.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
m8i.48xlarge	75.0	93.75	60.0
m8i.96xlarge	100.0	125.0	80.0
m8i.metal-48xl	75.0	93.75	60.0
m8i.metal-96xl	100.0	125.0	80.0
m8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
m8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
m8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
m8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
m8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
m8i-flex.16xlarge	15.0 / 30.0	18.75 / 37.5	12.5 / 25.0
r8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
r8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
r8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8g.8xlarge	15.0	18.75	12.5
r8g.12xlarge	22.5	28.125	18.75
r8g.16xlarge	30.0	37.5	25.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
r8g.24xlarge	40.0	50.0	32.5
r8g.48xlarge	50.0	62.5	40.0
r8g.metal-24xl	40.0	50.0	32.5
r8g.metal-48xl	50.0	62.5	40.0
r8gd.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
r8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
r8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8gd.8xlarge	15.0	18.75	12.5
r8gd.12xlarge	22.5	28.125	18.75
r8gd.16xlarge	30.0	37.5	25.0
r8gd.24xlarge	40.0	50.0	32.5
r8gd.48xlarge	50.0	62.5	40.0
r8gd.metal-24xl	40.0	50.0	32.5
r8gd.metal-48xl	50.0	62.5	40.0
r8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
r8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8i.8xlarge	15.0	18.75	12.5
r8i.12xlarge	22.5	28.125	18.75
r8i.16xlarge	30.0	37.5	25.0
r8i.24xlarge	40.0	50.0	32.5
r8i.32xlarge	50.0	62.5	40.0
r8i.48xlarge	75.0	93.75	60.0
r8i.96xlarge	100.0	125.0	80.0
r8i.metal-48xl	75.0	93.75	60.0
r8i.metal-96xl	100.0	125.0	80.0
r8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
r8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
r8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
r8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
r8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
r8i-flex.16xlarge	15.0 / 30.0	18.75 / 37.5	12.5 / 25.0
x8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
x8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
x8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
x8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
x8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
x8g.8xlarge	15.0	18.75	12.5
x8g.12xlarge	22.5	28.125	18.75
x8g.16xlarge	30.0	37.5	25.0
x8g.24xlarge	40.0	50.0	32.5
x8g.48xlarge	50.0	62.5	40.0
x8g.metal-24xl	40.0	50.0	32.5
x8g.metal-48xl	50.0	62.5	40.0

Bandwidth weighting impact for EBS

The following table shows the impact of bandwidth weighting on Amazon EBS operations, in Gbps, for the supported instance types.

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
c8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
c8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
c8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
c8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8g.8xlarge	10.0	6.25	12.5
c8g.12xlarge	15.0	9.375	18.75
c8g.16xlarge	20.0	12.5	25.0
c8g.24xlarge	30.0	20.0	37.5
c8g.48xlarge	40.0	27.5	50.0
c8g.metal-24xl	30.0	20.0	37.5
c8g.metal-48xl	40.0	27.5	50.0
c8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
c8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
c8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8gd.8xlarge	10.0	6.25	12.5
c8gd.12xlarge	15.0	9.375	18.75
c8gd.16xlarge	20.0	12.5	25.0
c8gd.24xlarge	30.0	20.0	37.5
c8gd.48xlarge	40.0	27.5	50.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
c8gd.metal-24xl	30.0	20.0	37.5
c8gd.metal-48xl	40.0	27.5	50.0
m8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
m8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
m8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8g.8xlarge	10.0	6.25	12.5
m8g.12xlarge	15.0	9.375	18.75
m8g.16xlarge	20.0	12.5	25.0
m8g.24xlarge	30.0	20.0	37.5
m8g.48xlarge	40.0	27.5	50.0
m8g.metal-24xl	30.0	20.0	37.5
m8g.metal-48xl	40.0	27.5	50.0
m8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
m8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
m8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
m8gd.8xlarge	10.0	6.25	12.5
m8gd.12xlarge	15.0	9.375	18.75
m8gd.16xlarge	20.0	12.5	25.0
m8gd.24xlarge	30.0	20.0	37.5
m8gd.48xlarge	40.0	27.5	50.0
m8gd.metal-24xl	30.0	20.0	37.5
m8gd.metal-48xl	40.0	27.5	50.0
m8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8i.8xlarge	10.0	6.25	12.5
m8i.12xlarge	15.0	9.375	18.75
m8i.16xlarge	20.0	12.5	25.0
m8i.24xlarge	30.0	20.0	37.5
m8i.32xlarge	40.0	27.5	50.0
m8i.48xlarge	60.0	41.25	75.0
m8i.96xlarge	80.0	55.0	100.0
m8i.metal-48xl	60.0	41.25	75.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
m8i.metal-96xl	80.0	55.0	100.0
m8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
m8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
m8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
m8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75
m8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
r8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
r8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
r8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8g.8xlarge	10.0	6.25	12.5
r8g.12xlarge	15.0	9.375	18.75
r8g.16xlarge	20.0	12.5	25.0
r8g.24xlarge	30.0	20.0	37.5
r8g.48xlarge	40.0	27.5	50.0
r8g.metal-24xl	30.0	20.0	37.5

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
r8g.metal-48xl	40.0	27.5	50.0
r8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
r8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
r8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8gd.8xlarge	10.0	6.25	12.5
r8gd.12xlarge	15.0	9.375	18.75
r8gd.16xlarge	20.0	12.5	25.0
r8gd.24xlarge	30.0	20.0	37.5
r8gd.48xlarge	40.0	27.5	50.0
r8gd.metal-24xl	30.0	20.0	37.5
r8gd.metal-48xl	40.0	27.5	50.0
r8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8i.8xlarge	10.0	6.25	12.5
r8i.12xlarge	15.0	9.375	18.75

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
r8i.16xlarge	20.0	12.5	25.0
r8i.24xlarge	30.0	20.0	37.5
r8i.32xlarge	40.0	27.5	50.0
r8i.48xlarge	60.0	41.25	75.0
r8i.96xlarge	80.0	55.0	100.0
r8i.metal-48xl	60.0	41.25	75.0
r8i.metal-96xl	80.0	55.0	100.0
r8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
r8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
r8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
r8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75
r8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
x8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
x8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
x8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
x8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
x8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0

Instance type	Default Baseline / Burst	vpc-1 Baseline / Burst	ebs-1 Baseline / Burst
x8g.8xlarge	10.0	6.25	12.5
x8g.12xlarge	15.0	9.375	18.75
x8g.16xlarge	20.0	12.5	25.0
x8g.24xlarge	30.0	20.0	37.5
x8g.48xlarge	40.0	27.5	50.0
x8g.metal-24xl	30.0	20.0	37.5
x8g.metal-48xl	40.0	27.5	50.0

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
- Monitor network performance

Enhanced networking on Amazon EC2 instances

Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types. 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 latency between instances. 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.

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.

All Nitro-based instances use ENA for enhanced networking. In addition, the following Xen-based instances use ENA: H1, I3, G3, `m4.16xlarge`, P2, P3, P3dn, and R4.

For more information, see Enable enhanced networking with ENA on your EC2 instances.

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 more information, see Enhanced networking with the Intel 82599 VF interface.

Contents

- Improve network latency for Linux based EC2 instances
- Nitro system considerations for performance tuning
- Optimize network performance on EC2 Windows instances

Enable enhanced networking with ENA on your EC2 instances

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA). To use enhanced networking, you must use an AMI that includes the required ENA driver or manually install it. Then you can enable ENA support on your instance.

To review release notes or install instructions for an ENA driver, see the tab that matches your instance operating system platform.

Linux

You can review the following documentation on GitHub:

- Review ENA Linux kernel driver release notes on GitHub.
- For an overview of the ENA Linux kernel driver that includes install instructions see Linux kernel driver for Elastic Network Adapter (ENA) family on GitHub.

Windows

You can review the following documentation from the **Manage device drivers** section of this guide:

- Track ENA Windows driver version releases.
- Install the ENA driver on EC2 Windows instances.

For Nitro-based instances, enhanced networking capabilities vary by the Nitro version that the instance type implements.

To review network specifications for your instance, choose the instance family link for your instance type. If you're not sure which instance family applies, see Naming conventions in the *Amazon EC2 Instance Types* guide.

- Network specifications for accelerated computing instances

Contents

Prerequisites for enhanced networking with ENA

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Launch a Nitro-based instance.
- Ensure that the instance has internet connectivity.
- 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 the ENA kernel driver and 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.
- **Linux instances** – Launch the instance using a supported version of the Linux kernel and a supported distribution, so that ENA enhanced networking is enabled for your instance automatically. For more information, see ENA Linux Kernel Driver Release Notes.
- **Windows instances** – If the instance is running Windows Server 2008 R2 SP1, ensure that is has the SHA-2 code signing support update.
- 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 or the AWS CloudShell User Guide. Enhanced networking cannot be managed from the Amazon EC2 console.

Test whether enhanced networking is enabled

You can test whether enhanced networking is enabled in your instances or your AMIs.

Instance attribute

Check the value of the `enaSupport` instance attribute.

AWS CLI

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].EnaSupport"
```

If enhanced networking is enabled, the output is as follows.

```
[
    true
]
```

PowerShell

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

If enhanced networking is enabled, the output is as follows.

```
True
```

Image attribute

Check the value of the `enaSupport` image attribute.

AWS CLI

Use the describe-images command.

```
aws ec2 describe-images \
```

```
    --image-id ami-0abcdef1234567890 \
    --query "Images[].EnaSupport"
```

If enhanced networking is enabled, the output is as follows.

```
[
    true
]
```

PowerShell

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).EnaSupport
```

If enhanced networking is enabled, the output is as follows.

```
True
```

Linux network interface driver

Use the following command to verify that the `ena` kernel driver is being used on a particular interface, substituting the interface name that you want to check. If you are using a single interface (default), this is `eth0`. If your Linux distribution supports predictable network names, this could be a name like `ens5`. For more information, expand the section for RHEL, SUSE, and CentOS in Enable enhanced networking on your instance.

In the following example, the `ena` kernel driver is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

In this example, the ena kernel driver is loaded and at the minimum recommended version. This instance has enhanced networking properly configured.

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 1.5.0g
firmware-version:
expansion-rom-version:
bus-info: 0000:00:05.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

Enable enhanced networking on your instance

The procedure that you use depends on the operating system of the instance.

Amazon Linux

The AMIs for Amazon Linux include the kernel driver required for enhanced networking with ENA installed and have ENA support enabled. Therefore, if you launch an instance with an HVM version of Amazon Linux on a supported instance type, enhanced networking is already enabled for your instance. For more information, see Test whether enhanced networking is enabled.

Ubuntu

The latest Ubuntu HVM AMIs include the kernel driver required for enhanced networking with ENA installed and have ENA support enabled. Therefore, if you launch an instance with the latest Ubuntu HVM AMI on a supported instance type, enhanced networking is already enabled for your instance. For more information, see Test whether enhanced networking is enabled.

If you launched your instance using an older AMI and it does not have enhanced networking enabled already, you can install the `linux-aws` kernel package to get the latest enhanced networking drivers and update the required attribute.

To install the `linux-aws` kernel package (Ubuntu 16.04 or later)

Ubuntu 16.04 and 18.04 ship with the Ubuntu custom kernel (`linux-aws` kernel package). To use a different kernel, contact Support.

To install the `linux-aws` kernel package (Ubuntu Trusty 14.04)

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

> **Important**
>
> If during the update process you are prompted to install `grub`, use `/dev/xvda` to install `grub` onto, and then choose to keep the current version of `/boot/grub/menu.lst`.

3. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI) or Stop-EC2Instance (AWS Tools for Windows PowerShell).

   [Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: To enable enhanced networking on Ubuntu (instance store-backed instances).

4. From your local computer, enable the enhanced networking attribute using one of the following commands:

   - modify-instance-attribute (AWS CLI)

   ```
   aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
   ```

   - Edit-EC2InstanceAttribute (Tools for Windows PowerShell)

   ```
   Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
   ```

5. (Optional) Create an AMI from the instance, as described in Create an Amazon EBS-backed AMI. The AMI inherits the enhanced networking `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

6. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or Start-EC2Instance (AWS Tools for Windows PowerShell).

To enable enhanced networking on Ubuntu (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in Create an instance store-backed AMI, making sure to enable the enhanced networking attribute when you register the AMI.

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport $true ...
```

RHEL, SUSE, CentOS

The latest AMIs for Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and CentOS include the kernel driver required for enhanced networking with ENA and have ENA support enabled. Therefore, if you launch an instance with the latest AMI on a supported instance type, enhanced networking is already enabled for your instance. For more information, see Test whether enhanced networking is enabled.

The following procedure provides the general steps for enabling enhanced networking on a Linux distribution other than Amazon Linux AMI or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the documentation for your Linux distribution.

To enable enhanced networking on Linux

1. Connect to your instance.
2. Clone the source code for the `ena` kernel driver on your instance from GitHub at https://github.com/amzn/amzn-drivers. (SUSE Linux Enterprise Server 12 SP2 and later include ENA 2.02 by default, so you are not required to download and compile the ENA driver. For SUSE Linux Enterprise Server 12 SP2 and later, you should file a request to add the driver version you want to the stock kernel).

   ```
   git clone https://github.com/amzn/amzn-drivers
   ```

3. Compile and install the `ena` kernel driver on your instance. These steps depend on the Linux distribution. For more information about compiling the kernel driver on Red Hat Enterprise Linux, see [How do I install the latest ENS driver for enhanced network support on an Amazon EC2 instance that runs RHEL?]

4. Run the **sudo depmod** command to update kernel driver dependencies.

5. Update `initramfs` on your instance to ensure that the new kernel driver loads at boot time. For example, if your distribution supports **dracut**, you can use the following command.

```
dracut -f -v
```

6. Determine if your system uses predictable network interface names by default. Systems that use **systemd** or **udev** versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see [Predictable Network Interface Names] on the freedesktop.org website.

   a. You can check the **systemd** or **udev** versions on RPM-based systems with the following command.

   ```
   rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
   systemd-208-11.el7_0.2.x86_64
   ```

   In the above Red Hat Enterprise Linux 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

   b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

   ```
   sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /etc/default/
   grub
   ```

   c. Rebuild the grub configuration file.

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

7. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances] (AWS CLI), [Stop-EC2Instance] (AWS Tools for Windows PowerShell).

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: To enable enhanced networking on Linux (instance store–backed instances).

8. From your local computer, enable the enhanced networking `enaSupport` attribute using one of the following commands:

   - modify-instance-attribute (AWS CLI)

   ```
   aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
   ```

   - Edit-EC2InstanceAttribute (Tools for Windows PowerShell)

   ```
   Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
   ```

9. (Optional) Create an AMI from the instance, as described in Create an Amazon EBS-backed AMI. The AMI inherits the enhanced networking `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

   If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` might fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or Start-EC2Instance (AWS Tools for Windows PowerShell).

11. (Optional) Connect to your instance and verify that the kernel driver is installed.

    If you are unable to connect to your instance after enabling enhanced networking, see Troubleshoot the ENA kernel driver on Linux.

To enable enhanced networking on Linux (instance store–backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in Create an instance store-backed AMI, making sure to enable the enhanced networking attribute when you register the AMI.

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport ...
```

Ubuntu with DKMS

This method is for testing and feedback purposes only. It is not intended for use with production deployments. For production deployments, see Ubuntu.

> **Important**
>
> Using DKMS voids the support agreement for your subscription. It should not be used for production deployments.

To enable enhanced networking with ENA on Ubuntu (EBS-backed instances)

1. Follow steps 1 and 2 in Ubuntu.
2. Install the `build-essential` packages to compile the kernel driver and the `dkms` package so that your `ena` kernel driver is rebuilt every time your kernel is updated.

   ```
   ubuntu:~$ sudo apt-get install -y build-essential dkms
   ```

3. Clone the source for the `ena` kernel driver on your instance from GitHub at https://github.com/amzn/amzn-drivers.

   ```
   ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
   ```

4. Move the `amzn-drivers` package to the `/usr/src/` directory so DKMS can find it and build it for each kernel update. Append the version number (you can find the current version number in the release notes) of the source code to the directory name. For example, version `1.0.0` is shown in the following example.

   ```
   ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
   ```

5. Create the DKMS configuration file with the following values, substituting your version of `ena`.

   Create the file.

   ```
   ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
   ```

   Edit the file and add the following values.

   ```
   ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
   PACKAGE_NAME="ena"
   PACKAGE_VERSION="1.0.0"
   CLEAN="make -C kernel/linux/ena clean"
   MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
   BUILT_MODULE_NAME[0]="ena"
   BUILT_MODULE_LOCATION="kernel/linux/ena"
   DEST_MODULE_LOCATION[0]="/updates"
   DEST_MODULE_NAME[0]="ena"
   AUTOINSTALL="yes"
   ```

6. Add, build, and install the `ena` kernel driver on your instance using DKMS.

   Add the kernel driver to DKMS.

   ```
   ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
   ```

   Build the kernel driver using the **dkms** command.

   ```
   ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
   ```

   Install the kernel driver using **dkms**.

   ```
   ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
   ```

7. Rebuild `initramfs` so the correct kernel driver is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. Verify that the ena kernel driver is installed using the modinfo ena command from [Test whether enhanced networking is enabled](#).

```
ubuntu:~$ modinfo ena
filename:    /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko
version:  1.0.0
license:  GPL
description: Elastic Network Adapter (ENA)
author:   Amazon.com, Inc. or its affiliates
srcversion:  9693C876C54CA64AE48F0CA
alias:    pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
vermagic:    3.13.0-74-generic SMP mod_unload modversions
parm:     debug:Debug level (0=none,...,16=all) (int)
parm:     push_mode:Descriptor / header push mode (0=automatic,1=disable,3=enable)
    0 - Automatically choose according to device capability (default)
    1 - Don't push anything to device memory
    3 - Push descriptors and header buffer to device memory (int)
parm:     enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1) (int)
parm:     enable_missing_tx_detection:Enable missing Tx completions. (default=1)
 (int)
parm:     numa_node_override_array:Numa node override map
(array of int)
parm:     numa_node_override:Enable/Disable numa node override (0=disable)
(int)
```

9. Continue with Step 3 in [Ubuntu](#).

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.

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.
2. [Windows Server 2016 and 2019 only] Run the following EC2Launch PowerShell script to configure the instance after the driver is installed.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

3. From the instance, install the driver as follows:

    a. Download the latest driver to the instance.

    b. Extract the zip archive.

    c. Install the driver by running the `install.ps1` PowerShell script.

    > **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) or Stop-EC2Instance (AWS Tools for Windows PowerShell).
5. Enable ENA support on your instance as follows:

    a. From your local computer, check the EC2 instance ENA support attribute on your instance by running one of the following commands. If the attribute is not enabled, the output will be "[]" or blank. `EnaSupport` is set to `false` by default.

    - describe-instances (AWS CLI)

    ```
    aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
     "Reservations[].Instances[].EnaSupport"
    ```

    - Get-EC2Instance (Tools for Windows PowerShell)

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

b. To enable ENA support, run one of the following commands:

- modify-instance-attribute (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
```

If you encounter problems when you restart the instance, you can also disable ENA support using one of the following commands:

- modify-instance-attribute (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --no-ena-support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -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:

```
[
 true
]
```

6. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or Start-EC2Instance (AWS Tools for Windows PowerShell).

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.

ENA queues

ENA queues are allocated to network interfaces with default static limits based on the instance type and size. On supported instance types, you can dynamically allocate these queues across Elastic Network Interfaces (ENIs). While the total queue count per instance depends on its type and size, you can configure multiple ENIs with ENA queues until you meet the maximum queue count for the ENI and the instance.

Flexible ENA queue allocation optimizes resource distribution, enabling maximum vCPU utilization. High network performance workloads typically require multiple ENA queues. You can fine-tune network performance and packets per second (PPS) by adjusting queue counts according to your specific workload needs. For example, network-intensive applications may require more queues compared to CPU-intensive applications.

Topics

Supported instances

The following instances support dynamic allocation of multiple ENA queues.

General purpose

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
	M6i		
`m6i.large`	2	2	6

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
m6i.xlarge	4	4	16
m6i.2xlarge	8	8	32
m6i.4xlarge	8	16	64
m6i.8xlarge	8	32	64
m6i.12xlarge	8	32	64
m6i.16xlarge	8	32	120
m6i.24xlarge	8	32	120
m6i.32xlarge	8	32	120
M6id			
m6id.large	2	2	6
m6id.xlarge	4	4	16
m6id.2xlarge	8	8	32
m6id.4xlarge	8	16	64
m6id.8xlarge	8	32	64
m6id.12xlarge	8	32	64
m6id.16xlarge	8	32	120
m6id.24xlarge	8	32	120
m6id.32xlarge	8	32	120
M6idn			

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
m6idn.large	2	2	6
m6idn.xlarge	4	4	16
m6idn.2xlarge	8	8	32
m6idn.4xlarge	8	16	64
m6idn.8xlarge	16	32	128
m6idn.12xlarge	16	32	128
m6idn.16xlarge	16	32	240
m6idn.24xlarge	32	32	480
m6idn.32xlarge	32	32	256
M6in			
m6in.large	2	2	6
m6in.xlarge	4	4	16
m6in.2xlarge	8	8	32
m6in.4xlarge	8	16	64
m6in.8xlarge	16	32	128
m6in.12xlarge	16	32	128
m6in.16xlarge	16	32	240
m6in.24xlarge	32	32	480
m6in.32xlarge	32	32	256

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
M8i			
m8i.large	2	2	6
m8i.xlarge	4	4	16
m8i.2xlarge	8	8	32
m8i.4xlarge	8	16	64
m8i.8xlarge	8	32	128
m8i.12xlarge	16	64	192
m8i.16xlarge	16	64	256
m8i.24xlarge	16	128	384
m8i.32xlarge	16	128	512
m8i.48xlarge	32	128	768
m8i.96xlarge	32	128	1536
M8i-flex			
m8i-flex.large	1	1	3
m8i-flex.xlarge	2	2	8
m8i-flex.2xlarge	4	4	16
m8i-flex.4xlarge	4	8	32
m8i-flex.8xlarge	4	16	64
m8i-flex.12xlarge	8	32	96

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
m8i-flex.16xlarge	8	32	128

Compute optimized

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
C6i			
c6i.large	2	2	6
c6i.xlarge	4	4	16
c6i.2xlarge	8	8	32
c6i.4xlarge	8	16	64
c6i.8xlarge	8	32	64
c6i.12xlarge	8	32	64
c6i.16xlarge	8	32	120
c6i.24xlarge	8	32	120
c6i.32xlarge	8	32	120
C6id			
c6id.large	2	2	6
c6id.xlarge	4	4	16
c6id.2xlarge	8	8	32

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
c6id.4xlarge	8	16	64
c6id.8xlarge	8	32	64
c6id.12xlarge	8	32	64
c6id.16xlarge	8	32	120
c6id.24xlarge	8	32	120
c6id.32xlarge	8	32	120
C6in			
c6in.large	2	2	6
c6in.xlarge	4	4	16
c6in.2xlarge	8	8	32
c6in.4xlarge	8	16	64
c6in.8xlarge	16	32	128
c6in.12xlarge	16	32	128
c6in.16xlarge	16	32	240
c6in.24xlarge	32	32	480
c6in.32xlarge	32	32	256
C8gn			
c8gn.medium	1	1	2
c8gn.large	2	2	6

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
c8gn.xlarge	4	4	16
c8gn.2xlarge	8	8	32
c8gn.4xlarge	8	16	64
c8gn.8xlarge	8	32	128
c8gn.12xlarge	16	64	192
c8gn.16xlarge	16	64	256
c8gn.24xlarge	16	128	384
c8gn.48xlarge	32	128	384

Memory optimized

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
R6i			
r6i.large	2	2	6
r6i.xlarge	4	4	16
r6i.2xlarge	8	8	32
r6i.4xlarge	8	16	64
r6i.8xlarge	8	32	64
r6i.12xlarge	8	32	64

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
r6i.16xlarge	8	32	120
r6i.24xlarge	8	32	120
r6i.32xlarge	8	32	120
R6idn			
r6idn.large	2	2	6
r6idn.xlarge	4	4	16
r6idn.2xlarge	8	8	32
r6idn.4xlarge	8	16	64
r6idn.8xlarge	16	32	128
r6idn.12xlarge	16	32	128
r6idn.16xlarge	16	32	240
r6idn.24xlarge	32	32	480
r6idn.32xlarge	32	32	256
R6in			
r6in.large	2	2	6
r6in.xlarge	4	4	16
r6in.2xlarge	8	8	32
r6in.4xlarge	8	16	64
r6in.8xlarge	16	32	128

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
r6in.12xlarge	16	32	128
r6in.16xlarge	16	32	240
r6in.24xlarge	32	32	480
r6in.32xlarge	32	32	256
R6id			
r6id.large	2	2	6
r6id.xlarge	4	4	16
r6id.2xlarge	8	8	32
r6id.4xlarge	8	16	64
r6id.8xlarge	8	32	64
r6id.12xlarge	8	32	64
r6id.16xlarge	8	32	120
r6id.24xlarge	8	32	120
r6id.32xlarge	8	32	120
R8i			
r8i.large	2	2	6
r8i.xlarge	4	4	16
r8i.2xlarge	8	8	32
r8i.4xlarge	8	16	64

Instance type	Default ENA queues per interface	Maximum ENA queues per interface	Maximum ENA queues per instance
r8i.8xlarge	8	32	128
r8i.12xlarge	16	64	192
r8i.16xlarge	16	64	256
r8i.24xlarge	16	128	384
r8i.32xlarge	16	128	512
r8i.48xlarge	32	128	768
r8i.96xlarge	32	128	1536
R8i-flex			
r8i-flex.large	1	1	3
r8i-flex.xlarge	2	2	8
r8i-flex.2xlarge	4	4	16
r8i-flex.4xlarge	4	8	32
r8i-flex.8xlarge	4	16	64
r8i-flex.12xlarge	8	32	96
r8i-flex.16xlarge	8	32	128

Modify the number of queues

You can modify the number of ENA queues using AWS Management Console or AWS CLI. In the AWS Management Console, the ENA queues configuration is available under each **Network interface** setting.

To modify the number of ENA queues using the AWS CLI, use either one of the following commands. Before modifying the queue count, use the following command to check your current queue count.

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

> **Note**
>
> - Your instance must be stopped before modifying the number of ENA queues.
> - The value for ENA queues must be a power of 2, such as, 1, 2, 4, 8, 16, 32, etc.
> - The number of queues allocated to any single ENI cannot exceed the number of vCPUs available on your instance.

attach-network-interface

In the following example, 32 ENA queues are configured on an ENI.

```
aws ec2 attach-network-interface \
  --network-interface-id eni-001aa1bb223cdd4e4 \
  --instance-id i-1234567890abcdef0 \
  --device-index 1 \
  --ena-queue-count 32
```

run-instances

In the following example, 2 ENA queues each are configured on 3 ENIs.

```
aws ec2 run-instances \
  --image-id ami-12ab3c30 \
  --instance-type c6i.large \
  --min-count 1 \
  --max-count 1 \
  --network-interfaces \
    "[{\"DeviceIndex\":0,\"SubnetId\":\"subnet-123456789012a345a\",\"EnaQueueCount
\":2},
     {\"DeviceIndex\":1,\"SubnetId\":\"subnet-123456789012a345a\",\"EnaQueueCount
\":2},
     {\"DeviceIndex\":2,\"SubnetId\":\"subnet-123456789012a345a\",\"EnaQueueCount
\":2}]"
```

modify-network-interface-attribute

In the following example, 32 ENA queues are configured on an ENI.

```
aws ec2 modify-network-interface-attribute \
--network-interface-id eni-1234567890abcdef0 \
--attachment AttachmentId=eni-attach-12345678,EnaQueueCount=32
```

In the following example, the ENA count is reset to the default value.

```
aws ec2 modify-network-interface-attribute \
--network-interface-id eni-1234567890abcdef0 \
--attachment AttachmentId=eni-attach-12345678,DefaultEnaQueueCount=true
```

Troubleshoot the ENA kernel driver on Linux

The Elastic Network Adapter (ENA) is designed to improve operating system health and reduce the chances of long-term disruption because of unexpected hardware behavior and or failures. The ENA architecture keeps device or driver failures as transparent to the system as possible. This topic provides troubleshooting information for ENA.

If you are unable to connect to your instance, start with the Troubleshoot connectivity issues section.

If you experience performance degradation after migrating to a sixth generation instance type, see the article What do I need to do before I migrate my EC2 instance to a sixth generation instance to make sure that I get maximum network performance?

If you are able to connect to your instance, you can gather diagnostic information by using the failure detection and recovery mechanisms that are covered in the later sections of this topic.

Contents

- Sub-optimal configuration notifications

Troubleshoot connectivity issues

If you lose connectivity while enabling enhanced networking, the `ena` module might be incompatible with your instance's current running kernel. This can happen if you install the module for a specific kernel version (without **dkms**, or with an improperly configured **dkms.conf** file) and then your instance kernel is updated. If the instance kernel that is loaded at boot time does not have the `ena` module properly installed, your instance will not recognize the network adapter and your instance becomes unreachable.

If you enable enhanced networking for a PV instance or AMI, this can also make your instance unreachable.

If your instance becomes unreachable after enabling enhanced networking with ENA, you can disable the `enaSupport` attribute for your instance and it will fall back to the stock network adapter.

To disable enhanced networking with ENA (EBS-backed instances)

1. From your local computer, stop the instance by using the Amazon EC2 console, the stop-instances command (AWS CLI), or the Stop-EC2Instance cmdlet (AWS Tools for PowerShell).
2. From your local computer, disable the enhanced networking attribute by using the modify-instance-attribute command with the `--no-ena-support` option or the Edit-EC2InstanceAttribute cmdlet with the `-EnaSupport $false` parameter.
3. From your local computer, start the instance using the Amazon EC2 console, the start-instances command, or the Start-EC2Instance cmdlet.
4. (Optional) Connect to your instance and try reinstalling the `ena` module with your current kernel version by following the steps in Enable enhanced networking with ENA on your EC2 instances.

To disable enhanced networking with ENA (instance store-backed instances)

1. Create a new AMI as described in Create an instance store-backed AMI.
2. When you register the AMI, be sure to include the `--no-ena-support` option in the stop-instances command (AWS CLI) or the `-EnaSupport $false` parameter in the Register-EC2Image cmdlet.

Keep-alive mechanism

The ENA device posts keep-alive events at a fixed rate (usually once every second). The ENA driver implements a watchdog mechanism, which checks for the presence of these keep-alive messages. If a message or messages are present, the watchdog is rearmed, otherwise the driver concludes that the device experienced a failure and then does the following:

- Dumps its current statistics to syslog
- Resets the ENA device
- Resets the ENA driver state

The above reset procedure may result in some traffic loss for a short period of time (TCP connections should be able to recover), but should not otherwise affect the user.

The ENA device may also indirectly request a device reset procedure, by not sending a keep-alive notification, for example, if the ENA device reaches an unknown state after loading an irrecoverable configuration.

The following is an example of the reset procedure:

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The watchdog process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the current statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.......
........
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx 0x7 // At the end of the down process, the driver discards incomplete packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version: 0.10 //The driver begins its up process
```

```
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version: 0.0.1
 implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8] Build date
 [Wed Apr  6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't supported
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The reset
 process is complete
```

Register read timeout

The ENA architecture suggests a limited usage of memory mapped I/O (MMIO) read operations. MMIO registers are accessed by the ENA device driver only during its initialization procedure.

If the driver logs (available in **dmesg** output) indicate failures of read operations, this may be caused by an incompatible or incorrectly compiled driver, a busy hardware device, or hardware failure.

Intermittent log entries that indicate failures on read operations should not be considered an issue; the driver will retry them in this case. However, a sequence of log entries containing read failures indicate a driver or hardware problem.

Below is an example of driver log entry indicating a read operation failure due to a timeout:

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout.
 expected: req id[1] offset[88] actual: req id[57006] offset[0]
[ 47.333715] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout.
 expected: req id[2] offset[8] actual: req id[57007] offset[0]
[ 47.346221] [ENA_COM: ena_com_dev_reset] Reg read32 timeout occurred
```

Statistics

If you experience insufficient network performance or latency issues, you should retrieve the device statistics and examine them. These statistics can be obtained using **ethtool** as follows.

```
[ec2-user ~]$ ethtool -S ethN
NIC statistics:
 tx_timeout: 0
 suspend: 0
 resume: 0
 wd_expired: 0
 interface_up: 1
 interface_down: 0
 admin_q_pause: 0
 bw_in_allowance_exceeded: 0
 bw_out_allowance_exceeded: 0
 pps_allowance_exceeded: 0
 conntrack_allowance_available: 450878
 conntrack_allowance_exceeded: 0
 linklocal_allowance_exceeded: 0
 queue_0_tx_cnt: 4329
 queue_0_tx_bytes: 1075749
 queue_0_tx_queue_stop: 0
 ...
```

The following command output parameters are described below:

`tx_timeout`: *N*

The number of times that the Netdev watchdog was activated.

`suspend`: *N*

The number of times the driver performed a suspend operation.

`resume`: *N*

The number of times the driver performed a resume operation.

`wd_expired`: *N*

The number of times that the driver did not receive the keep-alive event in the preceding three seconds.

`interface_up`: *N*

The number of times that the ENA interface was brought up.

`interface_down`: *N*

The number of times that the ENA interface was brought down.

`admin_q_pause`: *N*

The number of times the admin queue was not found in a running state.

`bw_in_allowance_exceeded`: *N*

The number of packets queued or dropped because the inbound aggregate bandwidth exceeded the maximum for the instance.

`bw_out_allowance_exceeded`: *N*

The number of packets queued or dropped because the outbound aggregate bandwidth exceeded the maximum for the instance.

`pps_allowance_exceeded`: *N*

The number of packets queued or dropped because the bidirectional PPS exceeded the maximum for the instance. *

`conntrack_allowance_available`: *N*

The number of tracked connections that can be established by the instance before hitting the Connections Tracked allowance of that instance type. Only available for Nitro-based instances. Not supported with FreeBSD instances or DPDK environments.

`conntrack_allowance_exceeded`: *N*

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.

`linklocal_allowance_exceeded`: *N*

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 Amazon DNS service, the Instance Metadata Service, and the Amazon Time Sync Service, but does not impact traffic to custom DNS resolvers.

`queue_N_tx_cnt`: *N*

The number of transmitted packets for this queue.

`queue_N_tx_bytes`: *N*

The number of transmitted bytes for this queue.

`queue_N_tx_queue_stop`: *N*

The number of times that queue *N* was full and stopped.

`queue_N_tx_queue_wakeup`: *N*

The number of times that queue *N* resumed after being stopped.

`queue_N_tx_dma_mapping_err`: *N*

Direct memory access error count. If this value is not 0, it indicates low system resources.

`queue_N_tx_linearize`: *N*

The number of times SKB linearization was attempted for this queue.

`queue_N_tx_linearize_failed`: *N*

The number of times SKB linearization failed for this queue.

`queue_N_tx_napi_comp`: *N*

The number of times the `napi` handler called `napi_complete` for this queue.

`queue_N_tx_tx_poll`: *N*

The number of times the `napi` handler was scheduled for this queue.

`queue_N_tx_doorbells`: *N*

The number of transmission doorbells for this queue.

`queue_N_tx_prepare_ctx_err`: *N*

The number of times `ena_com_prepare_tx` failed for this queue.

`queue_N_tx_bad_req_id`: *N*

Invalid `req_id` for this queue. The valid `req_id` is zero, minus the `queue_size`, minus 1.

`queue_N_tx_llq_buffer_copy`: *N*

The number of packets whose headers size are larger than llq entry for this queue.

`queue_N_tx_missed_tx`: *N*

The number of packets that were left uncompleted for this queue.

`queue_N_tx_unmask_interrupt`: *N*

The number of times the tx interrupt was unmasked for this queue.

`queue_N_rx_cnt`: *N*

The number of received packets for this queue.

`queue_N_rx_bytes`: *N*

The number of received bytes for this queue.

`queue_N_rx_rx_copybreak_pkt`: *N*

The number of times the rx queue received a packet that is less than the rx_copybreak packet size for this queue.

`queue_N_rx_csum_good`: *N*

The number of times the rx queue received a packet where the checksum was checked and was correct for this queue.

`queue_N_rx_refil_partial`: *N*

The number of times the driver did not succeed in refilling the empty portion of the rx queue with the buffers for this queue. If this value is not zero, it indicates low memory resources.

`queue_N_rx_bad_csum`: *N*

The number of times the `rx` queue had a bad checksum for this queue (only if rx checksum offload is supported).

`queue_N_rx_page_alloc_fail`: *N*

The number of time that page allocation failed for this queue. If this value is not zero, it indicates low memory resources.

`queue_N_rx_skb_alloc_fail`: *N*

The number of time that SKB allocation failed for this queue. If this value is not zero, it indicates low system resources.

`queue_N_rx_dma_mapping_err`: *N*

Direct memory access error count. If this value is not 0, it indicates low system resources.

`queue_N_rx_bad_desc_num`: *N*

Too many buffers per packet. If this value is not 0, it indicates the use of very small buffers.

`queue_N_rx_bad_req_id`: *N*

The req_id for this queue is not valid. The valid req_id is from [0, queue_size - 1].

`queue_N_rx_empty_rx_ring`: *N*

The number of times the rx queue was empty for this queue.

`queue_N_rx_csum_unchecked`: *N*

The number of times the rx queue received a packet whose checksum wasn't checked for this queue.

`queue_N_rx_xdp_aborted`: *N*

The number of times that an XDP packet was classified as XDP_ABORT.

`queue_N_rx_xdp_drop`: *N*

The number of times that an XDP packet was classified as XDP_DROP.

`queue_N_rx_xdp_pass`: *N*

The number of times that an XDP packet was classified as XDP_PASS.

`queue_N_rx_xdp_tx`: *N*

The number of times that an XDP packet was classified as XDP_TX.

`queue_N_rx_xdp_invalid`: *N*

The number of times that the XDP return code for the packet was not valid.

`queue_N_rx_xdp_redirect`: *N*

The number of times that an XDP packet was classified as XDP_REDIRECT.

`queue_N_xdp_tx_cnt`: *N*

The number of transmitted packets for this queue.

`queue_N_xdp_tx_bytes`: *N*

The number of transmitted bytes for this queue.

`queue_N_xdp_tx_queue_stop`: *N*

The number of times that this queue was full and stopped.

`queue_N_xdp_tx_queue_wakeup`: *N*

The number of times that this queue resumed after being stopped.

`queue_N_xdp_tx_dma_mapping_err`: *N*

Direct memory access error count. If this value is not 0, it indicates low system resources.

`queue_N_xdp_tx_linearize`: *N*

The number of times XDP buffer linearization was attempted for this queue.

`queue_N_xdp_tx_linearize_failed`: *N*

The number of times XDP buffer linearization failed for this queue.

`queue_N_xdp_tx_napi_comp`: *N*

The number of times the napi handler called napi_complete for this queue.

`queue_N_xdp_tx_tx_poll`: *N*

The number of times the napi handler was scheduled for this queue.

`queue_N_xdp_tx_doorbells`: *N*

The number of transmission doorbells for this queue.

`queue_N_xdp_tx_prepare_ctx_err`: *N*

The number of times ena_com_prepare_tx failed for this queue. This value should always be zero; if not, see the driver logs.

`queue_N_xdp_tx_bad_req_id`: *N*

The req_id for this queue is not valid. The valid req_id is from [0, queue_size - 1].

`queue_N_xdp_tx_llq_buffer_copy`: *N*

The number of packets that had their headers copied using llq buffer copy for this queue.

`queue_N_xdp_tx_missed_tx`: *N*

The number of times a tx queue entry missed a completion timeout for this queue.

`queue_N_xdp_tx_unmask_interrupt`: *N*

The number of times the tx interrupt was unmasked for this queue.

`ena_admin_q_aborted_cmd`: *N*

The number of admin commands that were aborted. This usually happens during the auto-recovery procedure.

`ena_admin_q_submitted_cmd`: *N*

The number of admin queue doorbells.

`ena_admin_q_completed_cmd`: *N*

The number of admin queue completions.

`ena_admin_q_out_of_space`: *N*

The number of times that the driver tried to submit new admin command, but the queue was full.

`ena_admin_q_no_completion`: *N*

The number of times that the driver did not get an admin completion for a command.

Driver error logs in syslog

The ENA driver writes log messages to **syslog** during system boot. You can examine these logs to look for errors if you are experiencing issues. Below is an example of information logged by the ENA driver in **syslog** during system boot, along with some annotations for select messages.

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.416939] [ENA_COM:
 ena_com_validate_version] ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.420915] [ENA_COM:
 ena_com_validate_version] ena controller version: 0.0.1 implementation version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.256831] ena 0000:00:03.0: Device
 watchdog is Enabled
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.672947] ena 0000:00:03.0: creating 8 io
 queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.680885] [ENA_COM:
 ena_com_init_interrupt_moderation] Feature 20 isn't supported  // Interrupt moderation
 is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.691609] [ENA_COM:
 ena_com_get_feature_ex] Feature 10 isn't supported // RSS HASH function configuration
 is not supported by the
 device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.694583] [ENA_COM:
```

```
 ena_com_get_feature_ex] Feature 18 isn't supported //RSS HASH input source
 configuration is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.697433] [ENA_COM:
 ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.701064] ena 0000:00:03.0 (unnamed
 net_device) (uninitialized): Cannot set host attributes
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.704917] ena 0000:00:03.0: Elastic
 Network Adapter (ENA) found at mem f3000000, mac addr 02:8a:3c:1e:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  480.805037] EXT4-fs (xvda1): re-mounted.
 Opts: (null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  481.025842] NET: Registered protocol family
 10
```

Which errors can I ignore?

The following warnings that may appear in your system's error logs can be ignored for the Elastic Network Adapter:

Set host attribute isn't supported

Host attributes are not supported for this device.

failed to alloc buffer for rx queue

This is a recoverable error, and it indicates that there may have been a memory pressure issue when the error was thrown.

Feature *X* isn't supported

The referenced feature is not supported by the Elastic Network Adapter. Possible values for *X* include:

- 10: RSS Hash function configuration is not supported for this device.
- 12: RSS Indirection table configuration is not supported for this device.
- 18: RSS Hash Input configuration is not supported for this device.
- 20: Interrupt moderation is not supported for this device.
- 27: The Elastic Network Adapter driver does not support polling the Ethernet capabilities from snmpd.

Failed to config AENQ

The Elastic Network Adapter does not support AENQ configuration.

Trying to set unsupported AENQ events

This error indicates an attempt to set an AENQ events group that is not supported by the Elastic Network Adapter.

Sub-optimal configuration notifications

The ENA device detects sub-optimal configuration settings in the driver that you can change. The device notifies the ENA driver and logs a warning to the console. The following example shows the format of the warning message.

```
Sub-optimal configuration notification code: 1. Refer to AWS ENA documentation for
 additional details and mitigation options.
```

The following list shows notification code details and recommended actions for sub-optimal configuration findings.

- **Code 1: ENA Express with wide LLQ configuration is not recommended**

  ENA Express ENI is configured with wide LLQ. This configuration is sub-optimal and could impact performance for ENA Express. We recommend that you disable wide LLQ settings when you use ENA Express ENIs as follows.

  ```
  sudo rmmod ena && sudo modprobe ena force_large_llq_header=0
  ```

  For more information about optimal configuration for ENA Express, see Improve network performance between EC2 instances with ENA Express.
- **Code 2: ENA Express ENI with sub-optimal Tx queue depth is not recommended**

  ENA Express ENI is configured with sub-optimal Tx queue depth. This configuration could impact performance for ENA Express. We recommend that you enlarge all Tx queues to the maximum value for the network interface when you use ENA Express ENIs as follows.

  You can run the following **ethtool** commands to adjust LLQ size. To learn more about how to control, query, and enable wide-LLQ, see the Large Low-Latency Queue (Large LLQ) topic of the Linux kernel driver for ENA documentation in the *Amazon Drivers GitHub repository*.

  ```
  ethtool -g interface
  ```

Set your Tx queues to the maximum depth:

```
ethtool -G interface tx depth
```

For more information about optimal configuration for ENA Express, see [Improve network performance between EC2 instances with ENA Express].

- **Code 3: ENA with regular LLQ size and Tx packet traffic exceeds the maximum header supported size**

  By default, ENA LLQ supports Tx packet header size up to 96 bytes. If the packet header size is larger than 96 bytes, the packet is dropped. To mitigate this issue, we recommend that you enable wide-LLQ, which increases the supported Tx packet header size to a maximum of 224 bytes.

  However, when you enable wide-LLQ, the maximum Tx ring size is reduced from 1000 to 512 entries. Wide-LLQ is enabled by default for all Nitro v4 and later instance types.

  - Nitro v4 instance types have a default maximum wide-LLQ Tx ring size of 512 entries, which can't be changed.
  - Nitro v5 instance types have a default wide-LLQ Tx ring size of 512 entries, which you can increase up to 1000 entries.

  You can run the following **ethtool** commands to adjust LLQ size. To learn more about how to control, query, and enable wide-LLQ, see the [Large Low-Latency Queue (Large LLQ)] topic of the Linux kernel driver for ENA documentation in the *Amazon Drivers GitHub repository*.

  Find the maximum depth for your Tx queues:

```
ethtool -g interface
```

  Set your Tx queues to the maximum depth:

```
ethtool -G interface tx depth
```

Troubleshoot the Elastic Network Adapter Windows driver

The Elastic Network Adapter (ENA) is designed to improve operating system health and to reduce unexpected hardware behavior or failures that can disrupt the operation of your Windows instance. The ENA architecture keeps device or driver failures as transparent to the operating system as possible.

Collect diagnostic information on the instance

The steps to open Windows operating system (OS) tools vary, depending on what version of the OS is installed on your instance. In the following sections, we use the **Run** dialog to open the tools, which works the same across all OS versions. However, you can access these tools using any method that you prefer.

Access the Run dialog

- Using the Windows logo key combination: `Windows` + `R`
- Using the search bar:
  - Enter `run` in the search bar.
  - Select the **Run** application from the search results.

Some steps require the context menu to access properties or context-sensitive actions. There are several ways to do this, depending on your OS version and hardware.

Access the context menu

- Using your mouse: right-click an item to bring up its context menu.
- Using your keyboard:
  - Depending on your OS version, use `Shift` + `F10`, or `Ctrl` + `Shift` + `F10`.
  - If you have the context key on your keyboard (three horizontal lines in a box), select the item you want and then press the context key.

If you can connect to your instance, use the following techniques to gather diagnostic information for troubleshooting.

Check ENA device status

To check the status of your ENA Windows driver using the Windows Device Manager, follow these steps:

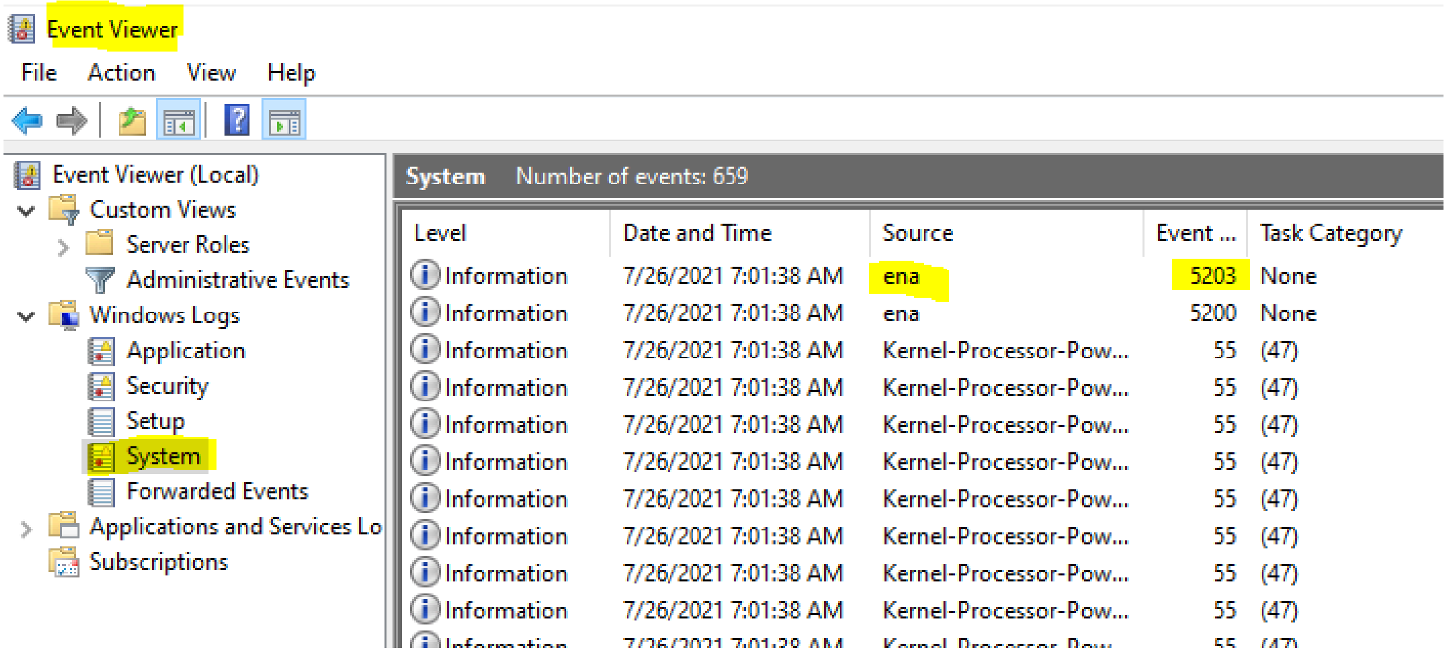
1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.
6. Verify that the message in the **General** tab says "This device is working properly."

Investigate driver event messages

To review ENA Windows driver event logs using the Windows Event Viewer, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Event Viewer, enter `eventvwr.msc` in the **Run** box.
3. Choose **OK**. This opens the Event Viewer window.
4. Expand the **Windows Logs** menu, and then choose **System**.
5. Under **Actions**, in the top-right panel, choose **Filter Current Log**. This displays the filtering dialog.
6. In the **Event sources** box, enter `ena`. This limits results to events that were generated by the ENA Windows driver.
7. Choose **OK**. This shows filtered event log results in the detail sections of the window.
8. To drill down into the details, select an event message from the list.

The following example shows an ENA driver event in the Windows Event Viewer system events list:



Event message summary

The following table shows event messages that the ENA Windows driver generates.

Input

Event ID	ENA driver event description	Type
5001	Hardware is out of resources	Error
5002	Adapter has detected a hardware error	Error
5005	Adapter has timed out on NDIS operation that did not complete in a timely manner	Error
5032	Adapter has failed to reset the device	Error
5200	Adapter has been initialized	Informational
5201	Adapter has been halted	Informational
5202	Adapter has been paused	Informational

Event ID	ENA driver event description	Type
5203	Adapter has been restarted	Informational
5204	Adapter has been shut down	Informational
5205	Adapter has been reset	Error
5206	Adapter has been surprise removed	Error
5208	Adapter initialization routine has failed	Error
5210	Adapter has encountered and successfully recovered an internal issue	Error

Review performance metrics

The ENA Windows driver publishes network performance metrics from the instances where metrics are enabled. You can view and enable metrics on the instance using the native Performance Monitor application. For more information about the metrics that the ENA Windows driver produces, see Monitor network performance for ENA settings on your EC2 instance.

On instances where ENA metrics are enabled, and the Amazon CloudWatch agent is installed, CloudWatch collects the metrics that are associated with the counters in Windows Performance Monitor, as well as some advanced metrics for ENA. These metrics are collected in addition to the metrics enabled by default on EC2 instances. For more information about the metrics, see Metrics collected by the CloudWatch agent in the *Amazon CloudWatch User Guide*.

Note

Performance metrics are available for ENA driver versions 2.4.0 and later (also for version 2.2.3). ENA driver version 2.2.4 was rolled back due to potential performance degradation on the sixth generation EC2 instances. We recommend that you upgrade to the current version of the driver to ensure that you have the latest updates.

Some of the ways that you can use performance metrics include:

- Troubleshoot instance performance issues.
- Choose the right instance size for a workload.
- Proactively plan scaling activities.
- Benchmark applications to determine if they maximize the performance available on an instance.

Refresh rate

By default, the driver refreshes metrics using a 1-second interval. However, the application that retrieves the metrics might use a different interval for polling. You can change the refresh interval in Device Manager, using the advanced properties for the driver.

To change the metrics refresh interval for the ENA Windows driver, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.
6. Open the **Advanced** tab in the pop-up window.
7. From the **Property** list, choose **Metrics Refresh Interval** to change the value.
8. When you are done, choose **OK**.

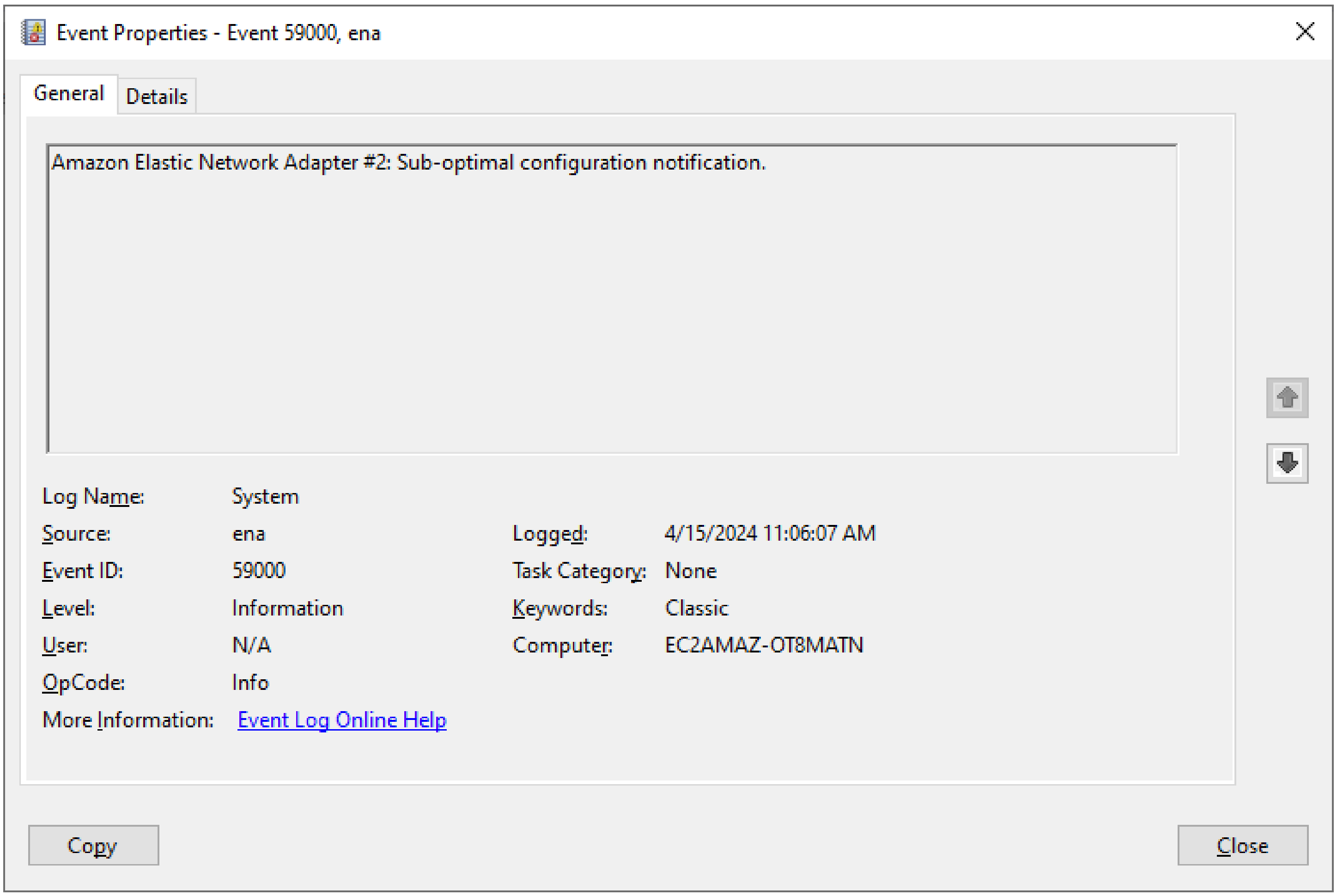
Investigate sub-optimal configuration notifications

The ENA device detects sub-optimal configuration settings in the driver that you can change. The device notifies the ENA driver and logs an event notification. To review sub-optimal events in the Windows Event Viewer

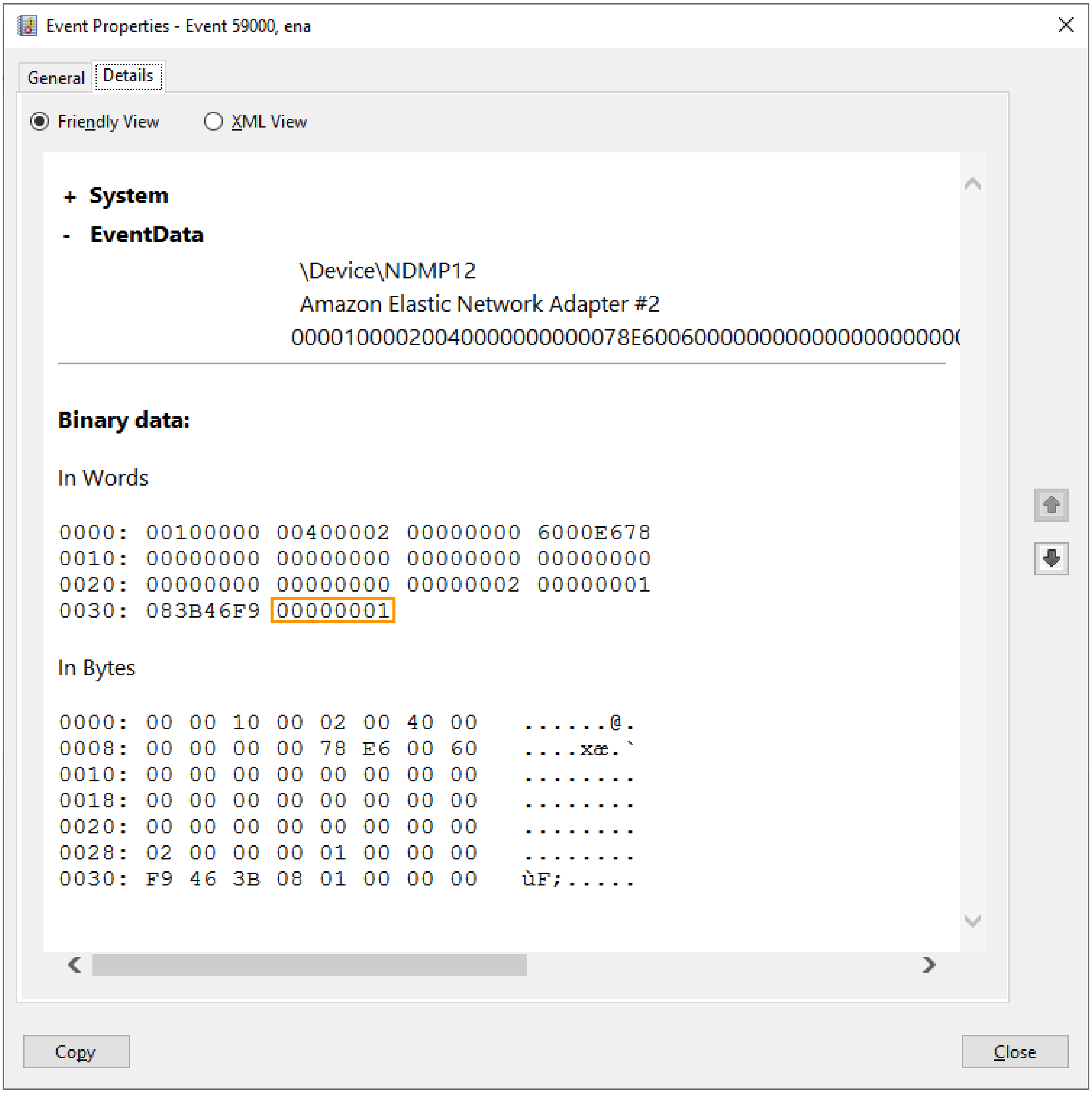
1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Event Viewer, enter `eventvwr.msc` in the **Run** box.
3. Choose **OK**. This opens the Event Viewer window.
4. Expand the **Windows Logs** menu, and then choose **System**.

5. Under **Actions**, in the top-right panel, choose **Filter Current Log**. This displays the filtering dialog.
6. In the **Event sources** box, enter `ena`. This limits results to events that were generated by the ENA Windows driver.
7. Choose **OK**. This shows filtered event log results in the detail sections of the window.

Events with ID `59000` notify you of sub-optimal configuration findings. Right-click an event and choose **Event Properties** to open a detailed view, or select **Preview Pane** from the **View** menu to see the same detail.



Open the **Details** tab to see the event code. In the **Binary Data: In words** section, the last word is the code.



The following list shows notification code details and recommended actions for sub-optimal configuration findings.

- **Code 1: ENA Express with wide LLQ configuration is not recommended**

ENA Express ENI is configured with wide LLQ. This configuration is sub-optimal and could impact performance for ENA Express. We recommend that you disable wide LLQ settings when you use ENA Express ENIs as follows.

1. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
2. Choose **OK**. This opens the Device Manager window.
3. Select the arrow to the left of **Network adapters** to expand the list.
4. Open the device properties for the `Amazon Elastic Network Adapter`.
5. From there, open the **Advanced** tab to make your changes.
6. Select the **LLQ Header Size Policy** property, and set its value to `Normal (128 Bytes)`.
7. Choose **OK** to save your changes.

- **Code 2: ENA Express ENI with sub-optimal Tx queue depth is not recommended**

  ENA Express ENI is configured with sub-optimal Tx queue depth. This configuration could impact performance for ENA Express. We recommend that you enlarge all Tx queues to the maximum value for the network interface when you use ENA Express ENIs as follows.

  Follow these steps to enlarge the Tx queues to the maximum depth:

1. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
2. Choose **OK**. This opens the Device Manager window.
3. Select the arrow to the left of **Network adapters** to expand the list.
4. Open the device properties for the `Amazon Elastic Network Adapter`.
5. From there, open the **Advanced** tab to make your changes.
6. Select the **Transmit Buffers** property, and set its value to the maximum supported.
7. Choose **OK** to save your changes.

ENA adapter reset

The reset process starts when the ENA Windows driver detects an error on an adapter, and marks the adapter as unhealthy. The driver cannot reset itself, so it depends on the operating system to check the adapter health status, and call the reset handle for the ENA Windows driver. The reset process might result in a brief period of time where traffic loss occurs. However, TCP connections should be able to recover.

The ENA adapter might also indirectly request a device reset procedure, by failing to send a keep-alive notification. For example, if the ENA adapter reaches an unknown state after loading an irrecoverable configuration, it might stop sending keep-alive notifications.

Common causes for ENA adapter reset

- Keep-alive messages are missing

  The ENA adapter posts keep-alive events at a fixed rate (usually once every second). The ENA Windows driver implements a watchdog mechanism, which periodically checks for the presence of these keep-alive messages. If it detects one or more new messages since the last time it checked, it records a successful outcome. Otherwise, the driver concludes that the device experienced a failure, and initiates a reset sequence.
- Packets are stuck in transmit queues

  The ENA adapter verifies that packets are flowing through the transmit queues as expected. The ENA Windows driver detects if packets are getting stuck, and initiates a reset sequence if they are.
- Read timeout for Memory Mapped I/O (MMIO) registers

  To limit memory mapped I/O (MMIO) read operations, the ENA Windows driver accesses MMIO registers only during initialization and reset processes. If the driver detects a timeout, it takes one of the following actions, depending on what process was running:
  - If a timeout is detected during initialization, it fails the flow, which results in the driver displaying a yellow exclamation mark by the ENA adapter in Windows Device Manager.
  - If a timeout is detected during reset, it fails the flow. The OS then initiates a surprise removal of the ENA adapter, and recovers it by stopping and starting the adapter that was removed. For more information about surprise removal of a network interface card (NIC), see Handling the Surprise Removal of a NIC in the *Microsoft Windows Hardware Developer* documentation.

Troubleshooting scenarios

The following scenarios can help you troubleshoot issues that you might experience with the ENA Windows driver. We recommend that you start with upgrading your ENA driver, if you don't have the latest version. To find the latest driver for your Windows OS version, see Track ENA Windows driver version releases.

Unexpected ENA driver version installed

Description

After you go through the steps to install a specific version of the ENA driver, the Windows Device Manager shows that Windows installed a different version of the ENA driver.

Cause

When you run the install for a driver package, Windows ranks all of the driver packages that are valid for the given device in the local Driver Store before it begins. Then it selects the package with the lowest rank value as the best match. This can be different from the package that you intended to install. For more information about the device driver package selection process, see How Windows selects a driver package for a device on the *Microsoft documentation website*.

Solution

To ensure that Windows installs your chosen driver package version, you can remove lower ranked driver packages from the Driver Store with the PnPUtil command line tool.

Follow these steps to update the ENA driver:

1. Connect to your instance and log in as the local administrator.
2. Open the Device Manager properties window, as described in the Check ENA device status section. This opens the **General** tab of the **Amazon Elastic Network Adapter Properties** window.
3. Open the **Driver** tab.
4. Choose **Update Driver**. This opens the **Update Driver Software – Amazon Elastic Network Adapter** dialog box.

    a. On the **How do you want to search for driver software?** page, choose **Browse my computer for driver software**.

    b. On the **Browse for driver software on your computer** page, choose **Let me pick from a list of device drivers on my computer**, located below the search bar.

    c. On the **Select the device driver you want to install for this hardware** page, choose **Have Disk...**.

    d. In the **Install from Disk** window, choose **Browse...**, next to the file location from the dropdown list.

   e. Navigate to the location where you downloaded the target ENA driver package. Select the file named `ena.inf` and choose **Open**.

   f. To start the install, choose **OK**, and then choose **Next**.

5. If the installer doesn't automatically reboot your instance, run the **Restart-Computer** PowerShell cmdlet.

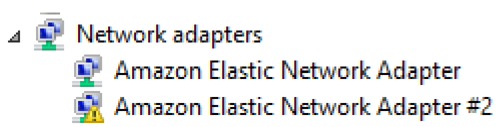
```
PS C:\> Restart-Computer
```

Device warning for ENA driver

Description

The ENA adapter icon in the Device Manager **Network adapters** section displays a warning sign (a yellow triangle with an exclamation mark inside).

The following example shows an ENA adapter with the warning icon in Windows Device Manager:



Cause

This device warning is commonly caused by environment issues, which might require more research, and often require a process of elimination to determine the underlying cause. For a full list of device errors, see Device Manager Error Messages in the Microsoft documentation.

Solution

The solution for this device warning depends on the root cause. The process of elimination described here includes a few basic steps to help identify and resolve the most common issues that might have a simple solution. Additional root cause analysis is required when these steps do not resolve the issue.

Follow these steps to help identify and resolve common issues:

1. **Stop and start the device**

   Open the Device Manager properties window, as described in the Check ENA device status section. This opens the **General** tab of the **Amazon Elastic Network Adapter Properties** window, where the **Device status** displays the error code and a short message.

   a. Open the **Driver** tab.
   b. Choose **Disable Device**, and respond **Yes** to the warning message that displays.
   c. Choose **Enable Device**.

2. **Stop and start the EC2 instance**

   If the adapter still shows the warning icon in Device Manager, the next step is to stop and start the EC2 instance. This relaunches the instance on different hardware in most cases.

3. **Investigate possible instance resource issue**

   If you have stopped and started your EC2 instance, and the problem persists, this might indicate a resource issue on your instance, such as insufficient memory.

Connection timeout with adapter reset (error codes 5007, 5205)

Description

The Windows Event Viewer shows adapter timeout and reset events occurring in combination for ENA adapters. Messages resemble the following examples:

- **Event ID 5007**: Amazon Elastic Network Adapter : Timed out during an operation.
- **Event ID 5205**: Amazon Elastic Network Adapter : Adapter reset has been started.

Adapter resets cause minimal traffic disruption. Even when there are multiple resets, it would be unusual for them to cause any severe network disruption.

Cause

This sequence of events indicates that the ENA Windows driver initiated a reset for an ENA adapter that was unresponsive. However, the mechanism that the device driver uses to detect this issue is subject to false positives resulting from CPU 0 starvation.

Solution

If this combination of errors happens frequently, check your resource allocations to see where adjustments might be helpful.

1. Open the **Run** dialog using one of the methods described in the preceding section.

2. To open the Windows Resource Monitor, enter `resmon` in the **Run** box.
3. Choose **OK**. This opens the Resource Monitor window.
4. Open the **CPU** tab. Per-CPU usage graphs are shown along the right side of the Resource Monitor window.
5. Check the usage levels for CPU 0 to see if they are too high.

We recommend that you configure RSS to exclude CPU 0 for the ENA adapter on larger instance types (more than 16 vCPU). For smaller instance types, configuring RSS might improve the experience, but due to the lower number of available cores, testing is necessary to ensure that constraining CPU cores does not negatively impact performance.

Use the **Set-NetAdapterRss** command to configure RSS for your ENA adapter, as shown in the following example.

```
Set-NetAdapterRss -name (Get-NetAdapter | Where-Object {$_.InterfaceDescription -like
 "*Elastic*"}).Name -Baseprocessorgroup 0 -BaseProcessorNumber 1
```

Migrating to a sixth generation instance infrastructure impacts performance or attachment

Description

If you migrate to a sixth generation EC2 instance, you might experience reduced performance or ENA attachment failures if you haven't updated your ENA Windows driver version.

Cause

The sixth generation EC2 instance types require the following minimum version of the ENA Windows driver, based on the instance operating system (OS).

Minimum version

Windows Server version	ENA driver version
Windows Server 2008 R2	2.2.3 or 2.4.0
Windows Server 2012 and later	2.2.3 and later

Windows Server version	ENA driver version
Windows Workstation	2.2.3 and later

Solution

Before you upgrade to a sixth generation EC2 instance, make sure that the AMI you launch from has compatible drivers based on the instance OS as shown in the previous table. For more information, see [What do I need to do before migrating my EC2 instance to a sixth generation instance to make sure that I get maximum network performance?] in the *AWS re:Post Knowledge Center*.

Suboptimal performance for the elastic network interface

Description

The ENA interface is not performing as expected.

Cause

Root cause analysis for performance issues is a process of elimination. There are too many variables involved to name a common cause.

Solution

The first step in your root cause analysis is to review the diagnostic information for the instance that is not performing as expected, to determine if there are errors that might be causing the issue. For more information, see the [Collect diagnostic information on the instance] section.

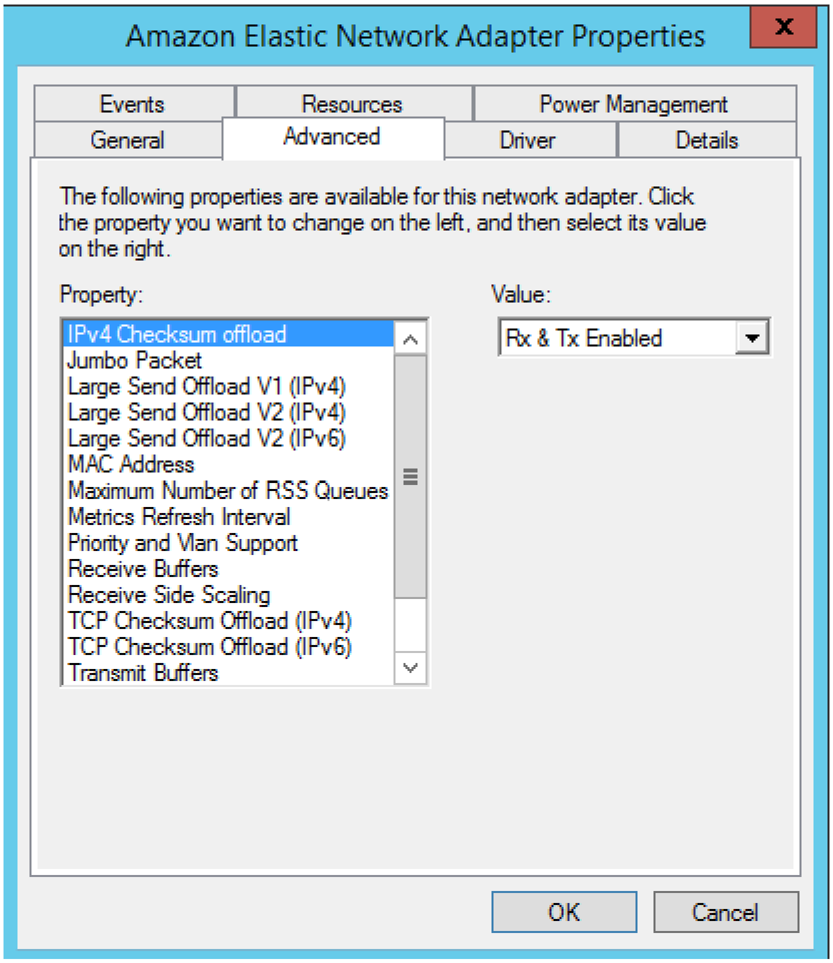
You might need to modify the default operating system configuration to achieve maximum network performance on instances with enhanced networking. Some optimizations, such as turning on checksum offloading and enabling RSS, are configured by default in official Windows AMIs. For other optimizations that you can apply to the ENA adapter, see the performance adjustments shown in [ENA adapter performance adjustments].

We recommend that you proceed with caution, and limit device property adjustments to those that are listed in this section, or to specific changes that are recommended by the AWS support team.

To change ENA adapter properties, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.
6. To make your changes, open the **Advanced** tab.
7. When you're done, choose **OK** to save your changes.

The following example shows an ENA adapter property in the Windows Device Manager:



ENA adapter performance adjustments

The following table includes properties that can be adjusted to improve performance for the ENA interface.

Input

Property	Description	Default value	Adjustment
Receive Buffers	Controls the number of entries in the software receive queues.	1024	Can be increased up to a maximum of 8192.
Receive Side Scaling (RSS)	Enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems.	Enabled	You can spread the load across multiple processors. To learn more, see [Optimize network performance on EC2 Windows instances](#).
Maximum Number of RSS Queues	Sets the maximum number of RSS queues allowed when RSS is enabled.	32	The number of RSS queues is determined during driver initialization, and includes the following limitations (among others): • RSS queue limit set by this property • Instance limits (vCPU count) • Hardware generation limits (up to 8 RSS queues in ENAv1,

Property	Description	Default value	Adjustment
			and up to 32 RSS queues in ENAv2) You can set the value from 1-32, depending on your instance and hardware generation limits. To learn more, see Optimize network performance on EC2 Windows instances.
Jumbo packet	Enables the use of jumbo ethernet frames (more than 1500 bytes of payload).	Disabled (this limits payload to 1500 bytes or less)	Value can be set up to `9015`, which translates to 9001 bytes of payload. This is the maximum payload for jumbo ethernet frames. See Considerations for using jumbo ethernet frames.

Considerations for using jumbo ethernet frames

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, which increases 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.
- Traffic over VPN connections.
- Traffic over an internet gateway.

> **Note**
>
> Packets over 1500 bytes are fragmented. If you have the `Don't Fragment` flag set in the IP header, these packets are dropped.
> Jumbo frames should be used with caution for internet-bound traffic, or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside of a VPC without impacting outbound traffic that's leaving the VPC, try one of the following options:
>
> - Configure the MTU size by route.
> - Use multiple network interfaces with different MTU sizes and different routes.

Recommended use cases for jumbo frames

Jumbo frames can be useful for traffic inside of and between VPCs. We recommend using jumbo frames for the following use cases:

- For instances that are collocated inside of a cluster placement group, jumbo frames help to achieve the maximum network throughput possible. For more information, see Placement groups for your Amazon EC2 instances.
- You can use jumbo frames for traffic between your VPCs and your on-premises networks over AWS Direct Connect. For more information about using AWS Direct Connect, and verifying jumbo frame capability, see MTU for private virtual interfaces or transit virtual interfaces in the *AWS Direct Connect User Guide*.
- For more information about supported MTU sizes for transit gateways, see Quotas for your transit gateways in the *Amazon VPC Transit Gateways*.

Improve network performance between EC2 instances with ENA Express

ENA Express is powered by AWS Scalable Reliable Datagram (SRD) technology. SRD is a high performance network transport protocol that uses dynamic routing to increase throughput and minimize tail latency. With ENA Express, you can communicate between two EC2 instances in the same Availability Zone.

Benefits of ENA Express

- Increases the maximum bandwidth a single flow can use from 5 Gbps up to 25 Gbps within the Availability Zone, up to the aggregate instance limit.
- Reduces tail latency of network traffic between EC2 instances, especially during periods of high network load.
- Detects and avoids congested network paths.
- Handles some tasks directly in the network layer, such as packet reordering on the receiving end, and most retransmits that are needed. This frees up the application layer for other work.

> **Note**
>
> - If your application sends or receives a high volume of packets per second, and needs to optimize for latency most of the time, especially during periods when there is no congestion on the network, Enhanced networking might be a better fit for your network.
> - ENA Express traffic can't be sent in a Local Zone.

After you've enabled ENA Express for the network interface attachment on an instance, the sending instance initiates communication with the receiving instance, and SRD detects if ENA Express is operating on both the sending instance and the receiving instance. If ENA Express is operating, the communication can use SRD transmission. If ENA Express is not operating, the communication falls back to standard ENA transmission.

During periods of time when network traffic is light, you might notice a slight increase in packet latency (tens of microseconds) when the packet uses ENA Express. During those times, applications that prioritize specific network performance characteristics can benefit from ENA Express as follows:

- Processes can benefit from increased maximum single flow bandwidth from 5 Gbps up to 25 Gbps within the same Availability Zone, up to the aggregate instance limit. For example, if a specific instance type supports up to 12.5 Gbps, the single flow bandwidth is also limited to 12.5 Gbps.
- Longer running processes should experience reduced tail latency during periods of network congestion.
- Processes can benefit from a smoother and more standard distribution for network response times.

Topics

How ENA Express works

ENA Express is powered by AWS Scalable Reliable Datagram (SRD) technology. It distributes packets for each network flow across different AWS network paths, and dynamically adjusts distribution when it detects signs of congestion. It also manages packet reordering on the receiving end.

To ensure that ENA Express can manage network traffic as intended, sending and receiving instances and the communication between them must meet all of the following requirements:

- Both sending and receiving instance types are supported. See the Supported instance types for ENA Express table for more information.
- Both sending and receiving instances must have ENA Express configured. If there are differences in the configuration, you can run into situations where traffic defaults to standard ENA transmission. The following scenario shows what can happen.

  Scenario: Differences in configuration

Instance	ENA Express Enabled	UDP uses ENA Express
Instance 1	Yes	Yes
Instance 2	Yes	No

In this case, TCP traffic between the two instances can use ENA Express, as both instances have enabled it. However, since one of the instances does not use ENA Express for UDP traffic, communication between these two instances over UDP uses standard ENA transmission.

- The sending and receiving instances must run in the same Availability Zone.
- The network path between the instances must not include middleware boxes. ENA Express doesn't currently support middleware boxes.
- (Linux instances only) To utilize full bandwidth potential, use driver version 2.2.9 or higher.
- (Linux instances only) To produce metrics, use driver version 2.8 or higher.

If any requirement is unmet, the instances use the standard TCP/UDP protocol but without SRD to communicate.

To ensure that your instance network driver is configured for optimum performance, review the recommended best practices for ENA drivers. These best practices apply to ENA Express, as well. For more information, see the ENA Linux Driver Best Practices and Performance Optimization Guide on the GitHub website.

Note

Amazon EC2 refers to the relationship between an instance and a network interface that's attached to it as an *attachment*. ENA Express settings apply to the attachment. If the network interface is detached from the instance, the attachment no longer exists, and the ENA Express settings that applied to it are no longer in force. The same is true when an instance is terminated, even if the network interface remains.

After you've enabled ENA Express for the network interface attachments on both the sending instance and the receiving instance, you can use ENA Express metrics to help ensure that your instances take full advantage of the performance improvements that SRD technology provides. For more information about ENA Express metrics, see Metrics for ENA Express.

Supported instance types for ENA Express

The following instance types support ENA Express.

General purpose

Instance type	Architecture
m6a.12xlarge	x86_64
m6a.16xlarge	x86_64
m6a.24xlarge	x86_64
m6a.32xlarge	x86_64
m6a.48xlarge	x86_64
m6a.metal	x86_64
m6i.8xlarge	x86_64
m6i.12xlarge	x86_64
m6i.16xlarge	x86_64
m6i.24xlarge	x86_64
m6i.32xlarge	x86_64
m6i.metal	x86_64
m6id.8xlarge	x86_64
m6id.12xlarge	x86_64
m6id.16xlarge	x86_64
m6id.24xlarge	x86_64

Instance type	Architecture
m6id.32xlarge	x86_64
m6id.metal	x86_64
m6idn.8xlarge	x86_64
m6idn.12xlarge	x86_64
m6idn.16xlarge	x86_64
m6idn.24xlarge	x86_64
m6idn.32xlarge	x86_64
m6idn.metal	x86_64
m6in.8xlarge	x86_64
m6in.12xlarge	x86_64
m6in.16xlarge	x86_64
m6in.24xlarge	x86_64
m6in.32xlarge	x86_64
m6in.metal	x86_64
m7a.12xlarge	x86_64
m7a.16xlarge	x86_64
m7a.24xlarge	x86_64
m7a.32xlarge	x86_64
m7a.48xlarge	x86_64
m7a.metal-48xl	x86_64

Instance type	Architecture
m7g.12xlarge	arm64
m7g.16xlarge	arm64
m7g.metal	arm64
m7gd.12xlarge	arm64
m7gd.16xlarge	arm64
m7gd.metal	arm64
m7i.12xlarge	x86_64
m7i.16xlarge	x86_64
m7i.24xlarge	x86_64
m7i.48xlarge	x86_64
m7i.metal-24xl	x86_64
m7i.metal-48xl	x86_64
m8g.12xlarge	arm64
m8g.16xlarge	arm64
m8g.24xlarge	arm64
m8g.48xlarge	arm64
m8g.metal-24xl	arm64
m8g.metal-48xl	arm64
m8gd.12xlarge	arm64
m8gd.16xlarge	arm64

Instance type	Architecture
m8gd.24xlarge	arm64
m8gd.48xlarge	arm64
m8gd.metal-24xl	arm64
m8gd.metal-48xl	arm64
m8i.24xlarge	x86_64
m8i.32xlarge	x86_64
m8i.48xlarge	x86_64
m8i.96xlarge	x86_64
m8i.metal-48xl	x86_64
m8i.metal-96xl	x86_64

Compute optimized

Instance type	Architecture
c6a.12xlarge	x86_64
c6a.16xlarge	x86_64
c6a.24xlarge	x86_64
c6a.32xlarge	x86_64
c6a.48xlarge	x86_64
c6a.metal	x86_64
c6gn.4xlarge	arm64

Instance type	Architecture
c6gn.8xlarge	arm64
c6gn.12xlarge	arm64
c6gn.16xlarge	arm64
c6i.8xlarge	x86_64
c6i.12xlarge	x86_64
c6i.16xlarge	x86_64
c6i.24xlarge	x86_64
c6i.32xlarge	x86_64
c6i.metal	x86_64
c6id.8xlarge	x86_64
c6id.12xlarge	x86_64
c6id.16xlarge	x86_64
c6id.24xlarge	x86_64
c6id.32xlarge	x86_64
c6id.metal	x86_64
c6in.8xlarge	x86_64
c6in.12xlarge	x86_64
c6in.16xlarge	x86_64
c6in.24xlarge	x86_64
c6in.32xlarge	x86_64

Instance type	Architecture
c6in.metal	x86_64
c7a.12xlarge	x86_64
c7a.16xlarge	x86_64
c7a.24xlarge	x86_64
c7a.32xlarge	x86_64
c7a.48xlarge	x86_64
c7a.metal-48xl	x86_64
c7g.12xlarge	arm64
c7g.16xlarge	arm64
c7g.metal	arm64
c7gd.12xlarge	arm64
c7gd.16xlarge	arm64
c7gd.metal	arm64
c7gn.4xlarge	arm64
c7gn.8xlarge	arm64
c7gn.12xlarge	arm64
c7gn.16xlarge	arm64
c7gn.metal	arm64
c7i.12xlarge	x86_64
c7i.16xlarge	x86_64

Instance type	Architecture
c7i.24xlarge	x86_64
c7i.48xlarge	x86_64
c7i.metal-24xl	x86_64
c7i.metal-48xl	x86_64
c8g.12xlarge	arm64
c8g.16xlarge	arm64
c8g.24xlarge	arm64
c8g.48xlarge	arm64
c8g.metal-24xl	arm64
c8g.metal-48xl	arm64
c8gd.12xlarge	arm64
c8gd.16xlarge	arm64
c8gd.24xlarge	arm64
c8gd.48xlarge	arm64
c8gd.metal-24xl	arm64
c8gd.metal-48xl	arm64
c8gn.8xlarge	arm64
c8gn.12xlarge	arm64
c8gn.16xlarge	arm64
c8gn.24xlarge	arm64

Instance type	Architecture
c8gn.48xlarge	arm64
c8gn.metal-24xl	arm64
c8gn.metal-48xl	arm64

Memory optimized

Instance type	Architecture
r6a.12xlarge	x86_64
r6a.16xlarge	x86_64
r6a.24xlarge	x86_64
r6a.32xlarge	x86_64
r6a.48xlarge	x86_64
r6a.metal	x86_64
r6i.8xlarge	x86_64
r6i.12xlarge	x86_64
r6i.16xlarge	x86_64
r6i.24xlarge	x86_64
r6i.32xlarge	x86_64
r6i.metal	x86_64
r6idn.8xlarge	x86_64
r6idn.12xlarge	x86_64

Instance type	Architecture
r6idn.16xlarge	x86_64
r6idn.24xlarge	x86_64
r6idn.32xlarge	x86_64
r6idn.metal	x86_64
r6in.8xlarge	x86_64
r6in.12xlarge	x86_64
r6in.16xlarge	x86_64
r6in.24xlarge	x86_64
r6in.32xlarge	x86_64
r6in.metal	x86_64
r6id.8xlarge	x86_64
r6id.12xlarge	x86_64
r6id.16xlarge	x86_64
r6id.24xlarge	x86_64
r6id.32xlarge	x86_64
r6id.metal	x86_64
r7a.12xlarge	x86_64
r7a.16xlarge	x86_64
r7a.24xlarge	x86_64
r7a.32xlarge	x86_64

Instance type	Architecture
r7a.48xlarge	x86_64
r7a.metal-48xl	x86_64
r7g.12xlarge	arm64
r7g.16xlarge	arm64
r7g.metal	arm64
r7gd.12xlarge	arm64
r7gd.16xlarge	arm64
r7gd.metal	arm64
r7i.12xlarge	x86_64
r7i.16xlarge	x86_64
r7i.24xlarge	x86_64
r7i.48xlarge	x86_64
r7i.metal-24xl	x86_64
r7i.metal-48xl	x86_64
r8g.12xlarge	arm64
r8g.16xlarge	arm64
r8g.24xlarge	arm64
r8g.48xlarge	arm64
r8g.metal-24xl	arm64
r8g.metal-48xl	arm64

Instance type	Architecture
r8gd.12xlarge	arm64
r8gd.16xlarge	arm64
r8gd.24xlarge	arm64
r8gd.48xlarge	arm64
r8gd.metal-24xl	arm64
r8gd.metal-48xl	arm64
r8i.24xlarge	x86_64
r8i.32xlarge	x86_64
r8i.48xlarge	x86_64
r8i.96xlarge	x86_64
r8i.metal-48xl	x86_64
r8i.metal-96xl	x86_64
u7i-6tb.112xlarge	x86_64
u7i-8tb.112xlarge	x86_64
u7i-12tb.224xlarge	x86_64
u7in-16tb.224xlarge	x86_64
u7in-24tb.224xlarge	x86_64
u7in-32tb.224xlarge	x86_64
u7inh-32tb.480xlarge	x86_64
x2idn.16xlarge	x86_64

Instance type	Architecture
x2idn.24xlarge	x86_64
x2idn.32xlarge	x86_64
x2idn.metal	x86_64
x2iedn.8xlarge	x86_64
x2iedn.16xlarge	x86_64
x2iedn.24xlarge	x86_64
x2iedn.32xlarge	x86_64
x2iedn.metal	x86_64
x8g.12xlarge	arm64
x8g.16xlarge	arm64
x8g.24xlarge	arm64
x8g.48xlarge	arm64
x8g.metal-24xl	arm64
x8g.metal-48xl	arm64

Accelerated computing

Instance type	Architecture
g6.48xlarge	x86_64
g6e.12xlarge	x86_64
g6e.24xlarge	x86_64

Instance type	Architecture
g6e.48xlarge	x86_64
p5.48xlarge	x86_64
p5e.48xlarge	x86_64
p5en.48xlarge	x86_64
p6-b200.48xlarge	x86_64

Storage optimized

Instance type	Architecture
i4g.4xlarge	arm64
i4g.8xlarge	arm64
i4g.16xlarge	arm64
i4i.8xlarge	x86_64
i4i.12xlarge	x86_64
i4i.16xlarge	x86_64
i4i.24xlarge	x86_64
i4i.32xlarge	x86_64
i4i.metal	x86_64
i7i.12xlarge	x86_64
i7i.16xlarge	x86_64
i7i.24xlarge	x86_64

Instance type	Architecture
i7i.48xlarge	x86_64
i7i.metal-24xl	x86_64
i7i.metal-48xl	x86_64
i7ie.12xlarge	x86_64
i7ie.18xlarge	x86_64
i7ie.24xlarge	x86_64
i7ie.48xlarge	x86_64
i7ie.metal-24xl	x86_64
i7ie.metal-48xl	x86_64
i8g.12xlarge	arm64
i8g.16xlarge	arm64
i8g.24xlarge	arm64
i8g.48xlarge	arm64
i8g.metal-24xl	arm64
i8ge.12xlarge	arm64
i8ge.18xlarge	arm64
i8ge.24xlarge	arm64
i8ge.48xlarge	arm64
i8ge.metal-24xl	arm64
i8ge.metal-48xl	arm64

Instance type	Architecture
im4gn.4xlarge	arm64
im4gn.8xlarge	arm64
im4gn.16xlarge	arm64

Prerequisites for Linux instances

To ensure that ENA Express can operate effectively, update the settings for your Linux instance as follows.

- If your instance uses jumbo frames, run the following command to set your maximum transmission unit (MTU) to `8900`.

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 8900
```

- Increase the receiver (Rx) ring size, as follows:

```
[ec2-user ~]$ ethtool -G device rx 8192
```

- To maximize ENA Express bandwidth, configure your TCP queue limits as follows:

  1. Set the TCP small queue limit to 1MB or higher. This increases the amount of data that's queued for transmission on a socket.

  ```
  sudo sh -c 'echo 1048576 > /proc/sys/net/ipv4/tcp_limit_output_bytes'
  ```

  2. Disable byte queue limits on the eth device if they're enabled for your Linux distribution. This increases data queued for transmission for the device queue.

  ```
  sudo sh -c 'for txq in /sys/class/net/eth0/queues/tx-*; do echo max > ${txq}/
  byte_queue_limits/limit_min; done'
  ```

> **Note**
>
> The ENA driver for the Amazon Linux distribution disables byte queue limits by default.

Tune performance for ENA Express settings on Linux instances

To check your Linux instance configuration for optimal ENA Express performance, you can run the following script that's available on the Amazon GitHub repository:

https://github.com/amzn/amzn-ec2-ena-utilities/blob/main/ena-express/check-ena-express-settings.sh

The script runs a series of tests and suggests both recommended and required configuration changes.

Review ENA Express settings for your EC2 instance

You can verify the ENA Express settings by instance or by network interface. To update the ENA Express settings, see the section called “Configure instance settings”.

Console

To view ENA Express settings for a network interface

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Network interfaces**.
3. Select a network interface to see the details for that instance. You can choose the **Network interface ID** link to open the detail page, or you can select the checkbox on the left side of the list to view details in the detail pane at the bottom of the page.
4. In the **Network interface attachment** section on the the **Details** tab or detail page, review settings for **ENA Express** and **ENA Express UDP**.

To view ENA Express settings for an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**.

3. Select an instance to see the details for that instance. You can choose the **Instance ID** link to open the detail page, or you can select the checkbox on the left side of the list to view details in the detail pane at the bottom of the page.
4. In the **Network interfaces** section on the **Networking** tab, scroll right to review settings for **ENA Express** and **ENA Express UDP**.

AWS CLI

To get the ENA Express settings for an instance

Use the describe-instances command. This command example returns a list of ENA Express configurations for the network interfaces attached to each of the running instances that are specified by the `--instance-ids` parameter.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7 \
    --query 'Reservations[*].Instances[*].[InstanceId,
 NetworkInterfaces[*].Attachment.EnaSrdSpecification]'
```

The following is example output.

```
[
    [
        [
            "i-1234567890abcdef0",
            [
                {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ],
    [
        [
            "i-0598c7d356eba48d7",
            [
            {
                    "EnaSrdEnabled": true,
```

```
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ]
]
```

To get the ENA Express settings for a network interface

Use the describe-network-interfaces command.

```
aws ec2 describe-network-interfaces \
    --network-interface-ids eni-1234567890abcdef0 \
    --query NetworkInterfaces[].[NetworkInterfaceId,Attachment.EnaSrdSpecification]
```

The following is example output.

```
[
    [
        "eni-1234567890abcdef0",
        {
            "EnaSrdEnabled": true,
            "EnaSrdUdpSpecification": {
                "EnaSrdUdpEnabled": false
            }
        }
    ]
]
```

PowerShell

To get the ENA Express settings for a network interface

Use the Get-EC2NetworkInterface cmdlet.

```
Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    Association,
    NetworkInterfaceId,
```

```
    OwnerId,
    @{Name = 'AttachTime'; Expression = { $_.Attachment.AttachTime } },
    @{Name = 'AttachmentId'; Expression = { $_.Attachment.AttachmentId } },
    @{Name = 'DeleteOnTermination'; Expression = 
{ $_.Attachment.DeleteOnTermination } },
    @{Name = 'NetworkCardIndex'; Expression = { $_.Attachment.NetworkCardIndex } },
    @{Name = 'InstanceId'; Expression = { $_.Attachment.InstanceId } },
    @{Name = 'InstanceOwnerId'; Expression = { $_.Attachment.InstanceOwnerId } },
    @{Name = 'Status'; Expression = { $_.Attachment.Status } },
    @{Name = 'EnaSrdEnabled'; Expression = 
{ $_.Attachment.EnaSrdSpecification.EnaSrdEnabled } },
    @{Name = 'EnaSrdUdpEnabled'; Expression = 
{ $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled } }
```

The following is example output.

```
Association         : 
NetworkInterfaceId  : eni-0d1234e5f6a78901b
OwnerId             : 111122223333
AttachTime          : 6/11/2022 1:13:11 AM
AttachmentId        : eni-attach-0d1234e5f6a78901b
DeleteOnTermination : True
NetworkCardIndex    : 0
InstanceId          : i-1234567890abcdef0
InstanceOwnerId     : 111122223333
Status              : attached
EnaSrdEnabled       : True
EnaSrdUdpEnabled    : False
```

Configure ENA Express settings for your EC2 instance

You can configure ENA Express for supported EC2 instance types without needing to install any additional software. For more information, see the section called “Supported instance types for ENA Express”.

Console

To manage ENA Express for a network interface

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Network interfaces**.

3. Select a network interface that is attached to an instance. You can choose the **Network interface ID** link to open the detail page, or you can select the checkbox on the left side of the list.
4. Choose **Manage ENA Express** from the **Action** menu at the top right side of the page. This opens the **Manage ENA Express** dialog, with the selected network interface ID and current settings displayed.

   If the network interface you selected is not attached to an instance, this action does not appear in the menu.
5. To use **ENA Express**, select the **Enable** checkbox.
6. When ENA Express is enabled, you can configure UDP settings. To use **ENA Express UDP**, select the **Enable** checkbox.
7. To save your settings, choose **Save**.

To manage ENA Express for an 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 that you want to manage. You can choose the **Instance ID** to open the detail page, or you can select the checkbox on the left side of the list.
4. Select the **Network interface** to configure for your instance.
5. Choose **Manage ENA Express** from the **Action** menu at the top right side of the page.
6. To configure ENA Express for a network interface that's attached to your instance, select it from the **Network interface** list.
7. To use **ENA Express** for the selected network interface attachment, select the **Enable** checkbox.
8. When ENA Express is enabled, you can configure UDP settings. To use **ENA Express UDP**, select the **Enable** checkbox.
9. To save your settings, choose **Save**.

To configure ENA Express when you attach a network interface

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Network interfaces**.

3. Select a network interface that is not attached to an instance (**Status** is **Available**). You can choose the **Network interface ID** link to open the detail page, or you can select the checkbox on the left side of the list.
4. Select the **Instance** that you'll attach to.
5. To use **ENA Express** after you attach the network interface to the instance, select the **Enable** checkbox.
6. When ENA Express is enabled, you can configure UDP settings. To use **ENA Express UDP**, select the **Enable** checkbox.
7. To attach the network interface to the instance and save your ENA Express settings, choose **Attach**.

AWS CLI

To configure ENA Express when you attach a network interface

Use the attach-network-interface command, as shown in the following examples.

Example 1: Use ENA Express for TCP traffic, but not UDP traffic

This example configures `EnaSrdEnabled` as `true`, and allows `EnaSrdUdpEnabled` to default to `false`.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

This example configures both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

To update ENA Express settings for a network interface attachment

Use the modify-network-interface-attribute command as shown in the following examples.

Example 1: Use ENA Express for TCP traffic, but not UDP traffic

This example configures `EnaSrdEnabled` as `true`, and allows `EnaSrdUdpEnabled` to default to `false` if it has never been set previously.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

This example configures both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

Example 3: Stop using ENA Express for UDP traffic

This example configures `EnaSrdUdpEnabled` as `false`.

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdUdpSpecification={EnaSrdUdpEnabled=false}'
```

PowerShell

To configure ENA Express when you attach a network interface

Use the Add-EC2NetworkInterface cmdlet as shown in the following examples.

Example 1: Use ENA Express for TCP traffic, but not UDP traffic

This example configures `EnaSrdEnabled` as `true`, and allows `EnaSrdUdpEnabled` to default to `false`.

```
Add-EC2NetworkInterface `
```

```
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

This example configures both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdUdpSpecification_EnaSrdUdpEnabled $true
```

To configure ENA Express settings for your network interface attachment

Use the Edit-EC2NetworkInterfaceAttribute cmdlet as shown in the following examples.

Example 1: Use ENA Express for TCP traffic, but not UDP traffic

This example configures `EnaSrdEnabled` as `true`, and allows `EnaSrdUdpEnabled` to default to `false` if it has never been set previously.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

This example configures both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
Edit-EC2NetworkInterfaceAttribute `
```

```
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

Example 3: Stop using ENA Express for UDP traffic

This example configures `EnaSrdUdpEnabled` as `false`.

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-0123f4567890a1b23 `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $false ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

Configure ENA Express at launch

You can use one of the following methods to configure ENA Express directly when you launch an instance. The specified links refer you to the AWS Management Console instructions for these methods.

- **Launch instance wizard** – You can configure ENA Express at launch with the launch instance wizard. For more information, see **Advanced network configuration** in the Network settings for the launch instance wizard.
- **Launch template** – You can configure ENA Express at launch when you use a launch template. For more information, see the Create an Amazon EC2 launch template page, then expand the **Network settings** section and review the **Advanced network configuration**.

Enhanced networking with the Intel 82599 VF interface

For Xen-based instances, the Intel 82599 Virtual Function (VF) interface provides enhanced networking capabilities. The interface uses the Intel `ixgbevf` driver.

The following tabs show how to verify the network adapter driver that's installed for your instance operating system.

Linux

Linux network interface driver

Use the following command to verify that the module is being used on a particular interface, substituting the interface name that you want to check. If you are using a single interface (default), this is `eth0`. If the operating system supports predictable network names, this could be a name like `ens5`.

In the following example, the `ixgbevf` module is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

In this example, the `ixgbevf` module is loaded. This instance has enhanced networking properly configured.

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 4.0.3
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
```

```
supports-register-dump: yes
supports-priv-flags: no
```

Windows

Windows network adapter

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**.

Contents

Prepare your instance for enhanced networking

To prepare for enhanced networking using the Intel 82599 VF interface, set up your instance as follows:

- Verify that the instance type is one of the following: C3, C4, D2, I2, M4 (excluding `m4.16xlarge`), and R3.
- Ensure that the instance has internet connectivity.
- 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.
- **Linux instances** – Launch the instance from an HVM AMI using Linux kernel version of 2.6.32 or later. The latest Amazon Linux HVM AMIs have the modules required for enhanced networking installed and have the required attributes set. Therefore, if you launch an Amazon EBS–backed, enhanced networking–supported instance using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance.

> **Warning**
>
> Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- **Windows instances –** Launch the instance from a 64-bit HVM AMI. You can't enable enhanced networking on Windows Server 2008. 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.
- 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 or the AWS CloudShell User Guide. Enhanced networking cannot be managed from the Amazon EC2 console.

Test whether enhanced networking is enabled

Verify that the `sriovNetSupport` attribute is set on the instance or the image.

AWS CLI

To check the instance attribute (sriovNetSupport)

Use the following describe-instance-attribute command. If the attribute is set, the value is `simple`.

```
aws ec2 describe-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --attribute sriovNetSupport
```

To check the image attribute (sriovNetSupport)

Use the following describe-images command. If the attribute is set, the value is `simple`.

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
```

```
    --query "Images[].SriovNetSupport"
```

PowerShell

To check the instance attribute (sriovNetSupport)

Use the Get-EC2InstanceAttribute cmdlet. If the attribute is set, the value is `simple`.

```
Get-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Attribute sriovNetSupport
```

To check the image attribute (sriovNetSupport)

Use the following describe-images command. If the attribute is set, the value is `simple`.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).SriovNetSupport
```

Enable enhanced networking on your instance

The procedure that you use depends on the operating system of the instance.

> **Warning**
>
> There is no way to disable the enhanced networking attribute after you've enabled it.

Amazon Linux

The latest Amazon Linux HVM AMIs have the `ixgbevf` module required for enhanced networking installed and have the required `sriovNetSupport` attribute set. Therefore, if you launch an instance type using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance. For more information, see Test whether enhanced networking is enabled.

If you launched your instance using an older Amazon Linux AMI and it does not have enhanced networking enabled already, use the following procedure to enable enhanced networking.

To enable enhanced networking

1. Connect to your instance.

2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including `ixgbevf`:

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

3. From your local computer, reboot your instance using the Amazon EC2 console or one of the following commands: reboot-instances (AWS CLI) or Restart-EC2Instance (AWS Tools for Windows PowerShell).

4. Connect to your instance again and verify that the `ixgbevf` module is installed and at the minimum recommended version using the **modinfo ixgbevf** command from Test whether enhanced networking is enabled.

5. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI) or Stop-EC2Instance (AWS Tools for Windows PowerShell).

   [Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, skip to the next procedure.

6. From your local computer, enable the enhanced networking attribute using one of the following commands:

   AWS CLI

   Use the modify-instance-attribute command as follows.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

   PowerShell

   Use the Edit-EC2InstanceAttribute cmdlet as follows.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

7. (Optional) Create an AMI from the instance, as described in Create an Amazon EBS-backed AMI. The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

8. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or Start-EC2Instance (AWS Tools for Windows PowerShell).
9. Connect to your instance and verify that the `ixgbevf` module is installed and loaded on your network interface using the **ethtool -i eth*n*** command from Test whether enhanced networking is enabled.

To enable enhanced networking (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in Create an instance store-backed AMI, making sure to enable the enhanced networking attribute when you register the AMI.

AWS CLI

Use the register-image command as follows.

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

Use Register-EC2Image as follows.

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Ubuntu

Before you begin, check if enhanced networking is already enabled on your instance.

The Quick Start Ubuntu HVM AMIs include the necessary drivers for enhanced networking. If you have a version of `ixgbevf` earlier than 2.16.4, you can install the `linux-aws` kernel package to get the latest enhanced networking drivers.

The following procedure provides the general steps for compiling the `ixgbevf` module on an Ubuntu instance.

To install the `linux-aws` kernel package

1. Connect to your instance.

2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

> **Important**
>
> If during the update process, you are prompted to install `grub`, use `/dev/xvda` to install `grub`, and then choose to keep the current version of `/boot/grub/menu.lst`.

Other Linux distributions

Before you begin, check if enhanced networking is already enabled on your instance. The latest Quick Start HVM AMIs include the necessary drivers for enhanced networking, therefore you do not need to perform additional steps.

The following procedure provides the general steps if you need to enable enhanced networking with the Intel 82599 VF interface on a Linux distribution other than Amazon Linux or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the specific documentation for your Linux distribution.

To enable enhanced networking on Linux

1. Connect to your instance.
2. Download the source for the `ixgbevf` module on your instance from Sourceforge at https://sourceforge.net/projects/e1000/files/ixgbevf%20stable/.

   Versions of `ixgbevf` earlier than 2.16.4, including version 2.14.2, do not build properly on some Linux distributions, including certain versions of Ubuntu.
3. Compile and install the `ixgbevf` module on your instance.

> **Warning**
>
> If you compile the `ixgbevf` module for your current kernel and then upgrade your kernel without rebuilding the driver for the new kernel, your system might revert to the distribution-specific `ixgbevf` module at the next reboot. This could make your system unreachable if the distribution-specific version is incompatible with enhanced networking.

4. Run the **sudo depmod** command to update module dependencies.
5. Update `initramfs` on your instance to ensure that the new module loads at boot time.
6. Determine if your system uses predictable network interface names by default. Systems that use **systemd** or **udev** versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see Predictable Network Interface Names on the freedesktop.org website.

   a. You can check the **systemd** or **udev** versions on RPM-based systems with the following command:

   ```
   [ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
       systemd-208-11.el7_0.2.x86_64
   ```

   In the above Red Hat Enterprise Linux 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

   b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

   ```
   [ec2-user ~]$ sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /
   etc/default/grub
   ```

   c. Rebuild the grub configuration file.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

7. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI) or Stop-EC2Instance (AWS Tools for Windows PowerShell).

   [Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, skip to the next procedure.

8. From your local computer, enable the enhanced networking attribute using one of the following commands:

   AWS CLI

   Use the modify-instance-attribute command as follows.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 -\
    -sriov-net-support simple
```

PowerShell

Use the Edit-EC2InstanceAttribute cmdlet as follows.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

9. (Optional) Create an AMI from the instance, as described in Create an Amazon EBS-backed AMI. The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

   If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` might fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.
10. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or Start-EC2Instance (AWS Tools for Windows PowerShell).
11. (Optional) Connect to your instance and verify that the module is installed.

To enable enhanced networking (instance store–backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in Create an instance store-backed AMI, making sure to enable the enhanced networking attribute when you register the AMI.

AWS CLI

Use the register-image command as follows.

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

Use Register-EC2Image as follows.

```
Register-EC2Image -SriovNetSupport "simple" ...
```

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 on Amazon EC2 instances.

> **Important**
>
> To view the latest driver updates in the Windows AMIs, see Windows AMI version history in the *AWS Windows AMI Reference*.

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 EC2 Launch PowerShell script to configure the instance after the driver is installed.

   ```
   PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   Schedule
   ```

   > **Important**
   >
   > The administrator password will reset when you enable the initialize instance EC2 Launch script. You can modify the configuration file to disable the administrator password reset by specifying it in the settings for the initialization tasks.

3. From the instance, download the Intel network adapter driver for your operating system:

   - **Windows Server 2022**

Visit the [download page] and download `Wired_driver_*version*_x64.zip`.

- **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.

- **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.

        ```
        C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
        ```

   - **Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012**

     1. In the **Downloads** folder, extract the contents of the Wired_driver_*version*_x64.zip file.
     2. 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 2022

```
pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
```

- Windows Server 2019

```
pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
```

- Windows Server 2016

```
pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
```

- Windows Server 2012 R2

```
pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
```

- Windows Server 2012

```
pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
```

5. From your local computer, enable the enhanced networking attribute using one of the following commands:

   AWS CLI

   Use the modify-instance-attribute command as follows.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

   PowerShell

   Use the Edit-EC2InstanceAttribute cmdlet as follows.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

6. (Optional) Create an AMI from the instance, as described in Create an Amazon EBS-backed AMI. The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
7. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI) or Start-EC2Instance (AWS Tools for Windows PowerShell).

Troubleshoot connectivity issues

If you lose connectivity while enabling enhanced networking, the `ixgbevf` module might be incompatible with the kernel. Try installing the version of the `ixgbevf` module included with the distribution of Linux for your instance.

If you enable enhanced networking for a PV instance or AMI, this can make your instance unreachable.

For more information, see How do I turn on and configure enhanced networking on my EC2 instances?

Monitor network performance for ENA settings on 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. For more information, see Amazon EC2 instance network bandwidth.
- **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. For more information, see [Amazon EC2 security group connection tracking](#)
- **Link-local service access** – Amazon EC2 provides a maximum PPS per network interface for traffic to local proxy services such as the Amazon 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

Linux instances

- Install ENA driver version 2.2.10 or later. To verify the installed version, use the **ethtool** command. In the following example, the version meets the minimum requirement.

```
[ec2-user ~]$ ethtool -i eth0 | grep version
version: 2.2.10
```

  To upgrade your ENA driver, see Enhanced networking.

- To import these metrics to Amazon CloudWatch, install the CloudWatch agent. For more information, see Collect network performance metrics in the *Amazon CloudWatch User Guide*.
- To support the `conntrack_allowance_available` metric, install ENA driver version 2.8.1 or later.

- To override the egress fragment PPS limit of 1024, install ENA driver version 2.13.3 or later.

Windows instances

- 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**.
  3. Choose **Amazon Elastic Network Adapter**, **Properties**.
  4. On the **Driver** tab, locate **Driver Version**.

  To upgrade your ENA driver, see Enhanced networking.
- 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.

Metric	Description	Supported on
`bw_in_allowance_exceeded`	The number of packets queued or dropped because the inbound aggregate bandwidth exceeded the maximum for the instance.	All instance types
`bw_out_allowance_exceeded`	The number of packets queued or dropped because the outbound aggregate bandwidth exceeded the maximum for the instance.	All instance types
`conntrack_allowance_exceeded`	The number of packets dropped because connection tracking exceeded the maximum for the	All instance types

Metric	Description	Supported on
	instance and new connections could not be established. This can result in packet loss for traffic to or from the instance.	
`conntrack_allowance_available`	The number of tracked connections that can be established by the instance before hitting the Connections Tracked allowance of that instance type.	Nitro-based instances only
`linklocal_allowance_exceeded`	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 Amazon DNS service, the Instance Metadata Service, and the Amazon Time Sync Service, but does not impact traffic to custom DNS resolvers.	All instance types
`pps_allowance_exceeded`	The number of packets queued or dropped because the bidirectional PPS exceeded the maximum for the instance. *	All instance types

* Depending on the fragment proxy mode setting for ENA Linux driver v2.13.3 or later, this limit can also include egress fragment drops that exceed 1024 PPS for the network interface. If fragment proxy mode is enabled for the Linux driver, egress fragment drops bypass the 1024 PPS limit that usually applies and are counted within standard PPS allowances. Fragment proxy mode is disabled by default.

View the network performance metrics for your instance

The procedure that you use depends on the operating system of the instance.

Linux instances

You can publish metrics to your favorite tools to visualize the metric data. For example, you can publish the metrics to Amazon CloudWatch using the CloudWatch agent. The agent enables you to select individual metrics and control publication.

You can also use the **ethtool** to retrieve the metrics for each network interface, such as eth0, as follows.

```
[ec2-user ~]$ ethtool -S eth0
     bw_in_allowance_exceeded: 0
     bw_out_allowance_exceeded: 0
     pps_allowance_exceeded: 0
     conntrack_allowance_exceeded: 0
     linklocal_allowance_exceeded: 0
     conntrack_allowance_available: 136812
```

Windows instances

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.

To install the manifest

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:

1. Remove the existing manifest using the following command:

   ```
   unlodctr /m:EnaPerfCounters.man
   ```

2. Copy the manifest file `EnaPerfCounters.man` from the driver installation package to `%SystemRoot%\System32\drivers`.
3. Install the new manifest using the following command:

```
lodctr /m:EnaPerfCounters.man
```

To view metrics using Performance Monitor

1. Open Performance Monitor.
2. Press Ctrl+N to add new counters.
3. Choose **ENA Packets Shaping** from the list.
4. Select the instances to monitor and choose **Add**.
5. Choose **OK**.

Metrics for ENA Express

ENA Express is powered by AWS Scalable Reliable Datagram (SRD) technology. SRD is a high performance network transport protocol that uses dynamic routing to increase throughput and minimize tail latency. If you've enabled ENA Express for the network interface attachments on both the sending instance and receiving instance, you can use ENA Express metrics to help ensure that your instances take full advantage of the performance improvements that SRD technology provides. For example:

- Evaluate your resources to ensure that they have sufficient capacity to establish more SRD connections.
- Identify where there are potential issues that prevent eligible outgoing packets from using SRD.
- Calculate the percentage of outgoing traffic that uses SRD for the instance.
- Calculate the percentage of incoming traffic that uses SRD for the instance.

> **Note**
>
> To produce metrics, use driver version 2.8 or higher.

To see a list of metrics for your Linux instance that's filtered for ENA Express, run the following **ethtool** command for your network interface (shown here as `eth0`). Take note of the value of the `ena_srd_mode` metric.

```
[ec2-user ~]$ ethtool -S eth0 | grep ena_srd
NIC statistics:
 ena_srd_mode: 1
 ena_srd_tx_pkts: 0
 ena_srd_eligible_tx_pkts: 0
 ena_srd_rx_pkts: 0
 ena_srd_resource_utilization: 0
```

The following metrics are available for all instances that have ENA Express enabled.

ena_srd_mode

Describes which ENA Express features are enabled. Values are as follows:

- `0` = ENA Express off, UDP off
- `1` = ENA Express on, UDP off
- `2` = ENA Express off, UDP on

> **Note**
>
> This only happens when ENA Express was originally enabled, and UDP was configured to use it. The prior value is retained for UDP traffic.

- `3` = ENA Express on, UDP on

ena_srd_eligible_tx_pkts

The number of network as follows:

- Both sending and receiving instance types are supported. See the Supported instance types for ENA Express table for more information.
- Both sending and receiving instances must have ENA Express configured.
- The sending and receiving instances must run in the same Availability Zone.
- The network path between the instances must not include middleware boxes. ENA Express doesn't currently support middleware boxes.

> **Note**
>
> The ENA Express eligibility metric covers source and destination requirements, and the network between the two endpoints. Eligible packets can still be disqualified after

they've already been counted. For example, if an eligible packet is over the maximum transmission unit (MTU) limit, it falls back to standard ENA transmission, though the packet is still reflected as eligible in the counter.

ena_srd_tx_pkts

The number of SRD packets transmitted within a given time period.

ena_srd_rx_pkts

The number of SRD packets received within a given time period.

ena_srd_resource_utilization

The percentage of the maximum allowed memory utilization for concurrent SRD connections that the instance has consumed.

To confirm if packet transmission is using SRD, you can compare the number of eligible packets (`ena_srd_eligible_tx_pkts` metric) to the number of SRD packets transmitted (`ena_srd_tx_pkts` metric) during a given time period.

Egress traffic (outgoing packets)

To ensure that your egress traffic uses SRD as expected, compare the number of SRD eligible packets (`ena_srd_eligible_tx_pkts`) with the number of SRD packets sent (`ena_srd_tx_pkts`) over a given time period.

Significant differences between the number of eligible packets and the number of SRD packets sent are often caused by resource utilization issues. When the network card attached to the instance has used up its maximum resources, or if packets are over the MTU limit, eligible packets are not able to transmit via SRD, and must fall back to standard ENA transmission. Packets can also fall into this gap during live migrations or live server updates. Additional troubleshooting is required to determine the root cause.

> **Note**
>
> You can ignore occasional minor differences between the number of eligible packets and the number of SRD packets. This can happen when your instance establishes a connection to another instance for SRD traffic, for example.

To find out what percentage of your total egress traffic over a given time period uses SRD, compare the number of SRD packets sent (`ena_srd_tx_pkts`) to the total number of packets sent for the instance (`NetworkPacketOut`) during that time.

Ingress traffic (incoming packets)

To find out what percentage of your ingress traffic uses SRD, compare the number of SRD packets received (`ena_srd_rx_pkts`) over a given time period to the total number of packets received for the instance (`NetworkPacketIn`) during that time.

Resource utilization

Resource utilization is based on the number of concurrent SRD connections a single instance can hold at a given time. The resource utilization metric (`ena_srd_resource_utilization`) keeps track of your current utilization for the instance. As utilization approaches 100%, you can expect to see performance issues. ENA Express falls back from SRD to standard ENA transmission, and the possibility of dropped packets increases. High resource utilization is a sign that it's time to scale the instance out to improve network performance.

> **Note**
>
> 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.

Persistence

Egress and ingress metrics accrue while ENA Express is enabled for the instance. Metrics stop accruing if ENA Express is deactivated, but persist as long as the instance is still running. Metrics reset if the instance reboots or is terminated, or if the network interface is detached from the instance.

Network performance metrics with the DPDK driver for ENA

The ENA driver version 2.2.0 and later supports network metrics reporting. DPDK 20.11 includes the ENA driver 2.2.0 and is the first DPDK version to support this feature.

DPDK driver v25.03 or later supports fragment proxy mode. If fragment proxy mode is enabled for the DPDK driver, egress fragment drops bypass the 1024 PPS limit that usually applies and are counted within standard PPS allowances. Fragment proxy mode is disabled by default.

You can use an example application to view DPDK statistics. To start an interactive version of the example application, run the following command.

```
./app/dpdk-testpmd -- -i
```

Within this interactive session, you can enter a command to retrieve extended statistics for a port. The following example command retrieves the statistics for port 0.

```
show port xstats 0
```

The following is an example of an interactive session with the DPDK example application.

```
[root@ip-192.0.2.0 build]# ./app/dpdk-testpmd -- -i
        EAL: Detected 4 lcore(s)
        EAL: Detected 1 NUMA nodes
        EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
        EAL: Selected IOVA mode 'PA'
        EAL: Probing VFIO support...
        EAL:   Invalid NUMA socket, default to 0
        EAL:   Invalid NUMA socket, default to 0
        EAL: Probe PCI driver: net_ena (1d0f:ec20) device: 0000:00:06.0
(socket 0)
        EAL: No legacy callbacks, legacy socket not created
        Interactive-mode selected

        Port 0: link state change event
        testpmd: create a new mbuf pool <mb_pool_0>: n=171456,
size=2176, socket=0
        testpmd: preferred mempool ops selected: ring_mp_mc

        Warning! port-topology=paired and odd forward ports number, the
last port will pair with itself.

        Configuring Port 0 (socket 0)
        Port 0: 02:C7:17:A2:60:B1
        Checking link statuses...
        Done
        Error during enabling promiscuous mode for port 0: Operation
not supported - ignore
        testpmd> show port xstats 0
        ###### NIC extended statistics for port 0
        rx_good_packets: 0
```

```
        tx_good_packets: 0
        rx_good_bytes: 0
        tx_good_bytes: 0
        rx_missed_errors: 0
        rx_errors: 0
        tx_errors: 0
        rx_mbuf_allocation_errors: 0
        rx_q0_packets: 0
        rx_q0_bytes: 0
        rx_q0_errors: 0
        tx_q0_packets: 0
        tx_q0_bytes: 0
        wd_expired: 0
        dev_start: 1
        dev_stop: 0
        tx_drops: 0
        bw_in_allowance_exceeded: 0
        bw_out_allowance_exceeded: 0
        pps_allowance_exceeded: 0
        conntrack_allowance_exceeded: 0
        linklocal_allowance_exceeded: 0
        rx_q0_cnt: 0
        rx_q0_bytes: 0
        rx_q0_refill_partial: 0
        rx_q0_bad_csum: 0
        rx_q0_mbuf_alloc_fail: 0
        rx_q0_bad_desc_num: 0
        rx_q0_bad_req_id: 0
        tx_q0_cnt: 0
        tx_q0_bytes: 0
        tx_q0_prepare_ctx_err: 0
        tx_q0_linearize: 0
        tx_q0_linearize_failed: 0
        tx_q0_tx_poll: 0
        tx_q0_doorbells: 0
        tx_q0_bad_req_id: 0
        tx_q0_available_desc: 1023
        testpmd>
```

For more information about the example application and using it to retrieve extended statistics. see Testpmd Application User Guide in the DPDK documentation.

Metrics on instances running FreeBSD

Starting with version 2.3.0, the ENA FreeBSD driver supports collecting network performance metrics on instances running FreeBSD. To enable the collection of FreeBSD metrics, enter the following command and set *interval* to a value between 1 and 3600. This specifies how often, in seconds, to collect FreeBSD metrics.

```
sysctl dev.ena.network_interface.eni_metrics.sample_interval=interval
```

For example, the following command sets the driver to collect FreeBSD metrics on network interface 1 every 10 seconds:

```
sysctl dev.ena.1.eni_metrics.sample_interval=10
```

To turn off the collection of FreeBSD metrics, you can run the preceding command and specify `0` as the *interval*.

After you enable collecting FreeBSD metrics, you can retrieve the latest set of collected metrics by running the following command.

```
sysctl dev.ena.network_interface.eni_metrics
```

Improve network latency for Linux based EC2 instances

Network latency is the amount of time it takes for a packet of data to travel from its source to its destination. Applications that send data across the network depend on timely responses to provide a positive user experience. High network latency can lead to various issues, such as the following:

- Slow load times for web pages
- Video stream lag
- Difficulty accessing online resources

This section outlines steps that you can take to improve the network latency on Amazon EC2 instances that run on Linux. To achieve optimal latency, follow these steps to configure your instance, kernel, and ENA driver settings. For additional configuration guidance, see the [ENA Linux Driver Best Practices and Performance Optimization Guide] on GitHub.

> **Note**
>
> Steps and settings may vary slightly, depending on your specific network hardware, the AMI that you launched your instance from, and your application use case. Before you make any changes, thoroughly test and monitor your network performance to ensure that you're getting the desired results.

Reduce the number of network hops for data packets

Each hop that a data packet takes as it moves from router to router increases network latency. Typically, traffic must take multiple hops to reach your destination. There are two ways to reduce network hops for your Amazon EC2 instances, as follows:

- **Cluster placement group** – When you specify a cluster placement group, Amazon EC2 launches instances that are in close proximity to each other, physically within the same Availability Zone (AZ) with tighter packing. The physical proximity of the instances in the group allows them to take advantage of high-speed connectivity, resulting in low latency and high single flow throughput.
- **Dedicated Host** – A Dedicated Host is a physical server that's dedicated for your use. With a Dedicated Host, you can launch your instances to run on the same physical server. Communication between instances that run on the same Dedicated Host can happen without extra network hops.

How Linux kernel configuration affects latency

Linux kernel configuration can increase or decrease network latency. To achieve your latency optimization goals, it's important to fine-tune the Linux kernel configuration according to the specific requirements of your workload.

There are many configuration options for the Linux kernel that might help decrease network latency. The most impactful options are as follows.

- **Enable busy poll mode** – Busy poll mode reduces latency on the network receive path. When you enable busy poll mode, the socket layer code can directly poll the receive queue of a network device. The downside of busy polling is higher CPU usage in the host that comes from polling for new data in a tight loop. There are two global settings that control the number of microseconds to wait for packets for all interfaces.

`busy_read`

A low latency busy poll timeout for socket reads. This controls the number of microseconds to wait for the socket layer to read packets on the device queue. To enable the feature globally with the **sysctl** command, the Linux Kernel organization recommends a value of 50 microseconds. For more information, see [busy_read](https://) in the *Linux kernel user's and administrator's guide*.

```
[ec2-user ~]$ sudo sysctl -w net.core.busy_read=50
```

`busy_poll`

A low latency busy poll timeout for poll and select. This controls the number of microseconds to wait for events. The recommended value is between 50-100 microseconds, depending on the number of sockets you're polling. The more sockets you add, the higher the number should be.

```
[ec2-user ~]$ sudo sysctl -w net.core.busy_poll=50
```

- **Configure CPU power states (C-states) –** C-states control the sleep levels that a core may enter when it's inactive. You might want to control C-states to tune your system for latency versus performance. In deeper C-states, the CPU is essentially "asleep" and can't respond to requests until it wakes up and transitions back to an active state. Putting cores to sleep takes time, and although a sleeping core allows more headroom for another core to boost to a higher frequency, it takes time for that sleeping core to wake back up and perform work.

  For example, if a core that's assigned to handle network packet interrupts is asleep, there might be a delay in servicing that interrupt. You can configure the system so that it doesn't use deeper C-states. However, while this configuration reduces the processor reaction latency, it also reduces the headroom available to other cores for Turbo Boost.

  To reduce the processor reaction latency, you can limit deeper C-states. For more information, see High performance and low latency by limiting deeper C-states in the *Amazon Linux 2 User Guide*.

Interrupt moderation

The ENA network driver enables communication between an instance and a network. The driver processes network packets and passes them on to the network stack or to the Nitro card. When a network packet comes in, the Nitro card generates an interrupt for the CPU to notify the software of an event.

Interrupt

An interrupt is a signal that a device or application sends to the processor. The interrupt tells the processor that an event has occurred or a condition has been met that requires immediate attention. Interrupts can handle time-sensitive tasks such as receiving data from a network interface, handling hardware events, or servicing requests from other devices.

Interrupt moderation

Interrupt moderation is a technique that reduces the number of interrupts a device generates by aggregating or delaying them. The purpose of interrupt moderation is to improve system performance by reducing the overhead associated with handling a large number of interrupts. Too many interrupts increase CPU usage, impacting the throughput adversely, while too few interrupts increase the latency.

Dynamic interrupt moderation

Dynamic interrupt moderation is an enhanced form of interrupt moderation that dynamically adjusts the interrupt rate based on the current system load and traffic patterns. It aims to strike a balance between reducing interrupt overhead and packets per second, or bandwidth.

> **Note**
>
> Dynamic interrupt moderation is enabled by default in some AMIs (but can be enabled or disabled in all AMIs).

To minimize network latency, it might be necessary to disable interrupt moderation. However, this can also increase the overhead of interrupt processing. It's important to find the right balance between reducing latency and minimizing overhead. `ethtool` commands can help you configure interrupt moderation. By default, `rx-usecs` is set to `20`, and `tx-usecs` is set to `64`.

To get the current interrupt modification configuration, use the following command.

```
[ec2-user ~]$ ethtool -c interface | egrep "rx-usecs:|tx-usecs:|Adaptive RX"
Adaptive RX: on  TX: off
rx-usecs: 20
tx-usecs: 64
```

To disable interrupt modification and dynamic interrupt moderation, use the following command.

```
[ec2-user ~]$ sudo ethtool -C interface adaptive-rx off rx-usecs 0 tx-usecs 0
```

Nitro system considerations for performance tuning

The Nitro System is a collection of hardware and software components built by AWS that enable high performance, high availability, and high security. The Nitro System provides bare metal-like capabilities that eliminate virtualization overhead and support workloads that require full access to host hardware. For more detailed information, see AWS Nitro System.

All current generation EC2 instance types perform network packet processing on EC2 Nitro Cards. This topic covers high level packet handling on the Nitro card, common aspects of network architecture and configuration that impact packet handling performance, and what actions you can take to achieve peak performance for your Nitro based instances.

Nitro Cards handle all input and output (I/O) interfaces, such as those needed for Virtual Private Clouds (VPCs). For all of the components that send or receive information over the network, the Nitro cards act as a self-contained computing device for I/O traffic that's physically separate from the system main board on which customer workloads run.

Network packet flow on Nitro cards

EC2 instances built on the Nitro system have hardware acceleration capabilities that enable faster packet processing, as measured by packets per second (PPS) throughput rates. When a Nitro card performs the initial evaluation for a new flow, it saves information that's the same for all packets in the flow, such as security groups, access control lists, and route table entries. When it processes additional packets for the same flow, it can use the saved information to reduce overhead for those packets.

Your connection rate is measured by the connections per second (CPS) metric. Each new connection requires additional processing overhead that must be factored into workload capability estimates. It's important to consider both the CPS and PPS metrics when you design your workloads.

How a connection is established

When a connection is established between a Nitro based instance and another endpoint, the Nitro card evaluates the full flow for the first packet that's sent or received between the two endpoints. For subsequent packets of the same flow, full reevaluation is usually not necessary. However, there are exceptions. For more information about the exceptions, see Packets that don't use hardware acceleration.

The following properties define the two endpoints and the packet flow between them. These five properties together are known as a 5-tuple flow.

- Source IP
- Source port
- Destination IP
- Destination port
- Communication protocol

The direction of the packet flow is known as *ingress* (inbound) and *egress* (outbound). The following high level descriptions summarize end to end network packet flow.

- **Ingress –** When a Nitro card handles an inbound network packet, it evaluates the packet against stateful firewall rules and access control lists. It tracks the connection, meters it, and performs other actions as applicable. Then it forwards the packet to its destination on the host CPU.
- **Egress –** When a Nitro card handles an outbound network packet, it looks up the remote interface destination, evaluates various VPC functions, applies rate limits, and performs other actions that apply. Then it forwards the packet to its next hop destination on the network.

Design your network for optimal performance

To take advantage of your Nitro system's performance capabilities, you must understand what your network processing needs are and how those needs affect the workload for your Nitro resources. Then you can design for optimal performance for your network landscape. Your infrastructure settings and application workload design and configuration can impact both the packet processing and connection rates. For example, if your application has a high rate of connection establishment, such as a DNS service, firewall, or virtual router, it will have less opportunity to take advantage of the hardware acceleration that only occurs after the connection is established.

You can configure application and infrastructure settings to streamline workloads and improve network performance. However, not all packets are eligible for acceleration. The Nitro system uses the full network flow for new connections and for packets that aren't eligible for acceleration.

The remainder of this section will focus on application and infrastructure design considerations to help ensure that packets flow within the accelerated path as much as possible.

Network design considerations for the Nitro system

When you configure network traffic for your instance, there are many aspects to consider that can affect PPS performance. After a flow is established, the majority of packets that regularly come in or go out are eligible for acceleration. However, exceptions exist to ensure that infrastructure designs and packet flows continue to meet protocol standards.

To get the best performance from your Nitro card, you should carefully consider the pros and cons of the following configuration details for your infrastructure and applications.

Infrastructure considerations

Your infrastructure configuration can affect your packet flow and processing efficiency. The following list includes some important considerations.

Network interface configuration with asymmetry

Security groups use connection tracking to track information about traffic that flows to and from the instance. Asymmetric routing, where traffic comes into an instance through one network interface and leaves through a different network interface, can reduce the peak performance that an instance can achieve if flows are tracked. For more information about security group connection tracking, untracked connections, and automatically tracked connections, see [Amazon EC2 security group connection tracking](#).

Network drivers

Network drivers are updated and released on a regular basis. If your drivers are out of date, that can significantly impair performance. Keep your drivers up to date to ensure that you have the latest patches and can take advantage of performance improvements, such as the accelerated path feature that's only available for the latest generation of drivers. Earlier drivers don't support the accelerated path feature.

To take advantage of the accelerated path feature, we recommend that you install the latest ENA driver on your instances.

Linux instances – ENA Linux driver 2.2.9 or later. To install or update the ENA Linux driver from the Amazon Drivers GitHub repository, see the Driver compilation section of the readme file.

Windows instances – ENA Windows driver 2.0.0 or later. To install or update the ENA Windows driver, see Install the ENA driver on EC2 Windows instances.

Distance between endpoints

A connection between two instances in the same Availability Zone can process more packets per second than a connection across Regions as a result of TCP windowing at the application layer, which determines how much data can be in flight at any given time. Long distances between instances increase latency and decrease the number of packets that the endpoints can process.

Byte queue limit (BQL)

BQL is a feature that limits the number of bytes passed to the Nitro card to reduce queuing. BQL is disabled by default in ENA drivers, in Amazon Linux operating systems, and in most Linux distributions. If BQL and the fragment proxy override are both enabled, it can result in performance limitations by restricting the number of bytes passed to Nitro before all fragments are processed.

Application design considerations

There are aspects of application design and configuration that can affect your processing efficiency. The following list includes some important considerations.

Packet size

Larger packet sizes can increase throughput for the data that an instance can send and receive on the network. Amazon EC2 supports jumbo frames of 9001 bytes, however other services may enforce different limits. Smaller packet sizes can increase the packet process rate, but this can reduce the maximum achieved bandwidth when the number of packets exceed PPS allowances.

If the size of a packet exceeds the Maximum Transmission Unit (MTU) of a network hop, a router along the path might fragment it. The resulting packet fragments are considered exceptions, and are normally processed at the standard rate (not accelerated). This can cause variations in your performance. However, you can override the standard behavior for outbound fragmented packets with the fragment proxy mode setting. For more information, see Maximize network performance on your Nitro system. We recommended that you evaluate your topology when you configure MTU.

Protocol trade-offs

Reliable protocols like TCP have more overhead than unreliable protocols like UDP. The lower overhead and simplified network processing for the UDP transport protocol can result in a higher PPS rate, but at the expense of reliable packet delivery. If reliable packet delivery isn't critical for your application, UDP might be a good option.

Micro-bursting

Micro-bursting occurs when traffic exceeds allowances during brief periods of time rather than being evenly distributed. This typically happens on a microsecond scale.

For example, say that you have an instance that can send up to 10 Gbps, and your application sends the full 10 Gb in half a second. This micro-burst exceeds the allowance during the first half second and leaves nothing for the remainder of the second. Even though you sent 10Gb in the 1 second timeframe, allowances in the first half second can result in packets being queued or dropped.

You can use a network scheduler such as Linux Traffic Control to help pace your throughput and avoid causing queued or dropped packets as a result of micro-bursting.

Number of flows

A single flow is limited to 5 Gbps unless it's inside of a cluster placement group that supports up to 10 Gbps, or if it uses ENA Express, which supports up to 25 Gbps.

Similarly, a Nitro card can process more packets across multiple flows as opposed to using a single flow. To achieve the peak packet processing rate per instance, we recommend at least 100 flows on instances with 100 Gbps or higher aggregate bandwidth. As aggregate bandwidth capabilities increase, the number of flows needed to achieve peak processing rates also increases. Benchmarking will help you determine what configuration you need to achieve peak rates on your network.

Elastic Network Adapter (ENA) queues

ENA (Elastic Network Adapter) uses multiple receive (Rx) and transmit (Tx) queues (ENA queues) to improve network performance and scalability on EC2 instances. These queues efficiently manage network traffic by load-balancing sent and received data across available queues.

For more information, see the section called "ENA queues".

Feature process overhead

Features like Traffic Mirroring and ENA Express can add more processing overhead, which can reduce absolute packet processing performance. You can limit feature use or disable features to increase packet processing rates.

Connection tracking to maintain state

Your security groups use connection tracking to store information about traffic to and from the instance. Connection tracking applies rules against each individual flow of network traffic to determine if the traffic is allowed or denied. The Nitro card uses flow tracking to maintain state for the flow. As more security group rules are applied, more work is required to evaluate the flow.

> **Note**
>
> Not all network traffic flows are tracked. If a security group rule is configured with [Untracked connections](), no additional work is required except for connections that are automatically tracked to ensure symmetric routing when there are multiple valid reply paths.

Packets that don't use hardware acceleration

Not all packets can take advantage of hardware acceleration. Handling these exceptions involves some processing overhead which is necessary to ensure the health of your network flows. Network flows must reliably meet protocol standards, conform to changes in the VPC design, and route packets only to allowed destinations. However, the overhead reduces your performance.

Packet fragments

As mentioned under **Application considerations**, packet fragments that result from packets that exceed network MTU are normally handled as exceptions, and can't take advantage of hardware acceleration. However, you can bypass egress fragment limitations with the fragment proxy mode, depending on your driver version. For more information, see actions you can take in the Maximize network performance on your Nitro system section.

Idle connections

When a connection has no activity for a while, even if the connection hasn't reached its timeout limit, the system can de-prioritize it. Then, if data comes in after the connection is de-prioritized, the system needs to handle it as an exception in order to reconnect.

To manage your connections, you can use connection tracking timeouts to close idle connections. You can also use TCP keepalives to keep idle connections open. For more information, see Idle connection tracking timeout.

VPC mutation

Updates to security groups, route tables, and access control lists all need to be reevaluated in the processing path to ensure that route entries and security group rules still apply as expected.

ICMP flows

Internet Control Message Protocol (ICMP) is a network layer protocol that network devices use to diagnose network communication issues. These packets always use the full flow.

Asymmetric L2 flows

NitroV3 and earlier platforms do not use hardware acceleration for traffic between two ENIs in the same subnet where one ENI is using the default gateway router and the other is not. NitroV4 and later platforms utilize hardware acceleration in this scenario. For better performance on NitroV3 or earlier platforms, ensure that either the default gateway router used matches between both ENIs, or those ENIs are in different subnets.

Maximize network performance on your Nitro system

You can maximize your network performance on Nitro system by adjusting network settings.

Topics

Considerations

Before you make any design decisions or adjust any network settings on your instance, we recommend that you take the following steps to help ensure that you have the best outcome:

1. Understand the pros and cons of the actions that you can take to improve performance by reviewing Network design considerations for the Nitro system.

   For more considerations and best practices for your instance configuration on Linux, see ENA Linux Driver Best Practices and Performance Optimization Guide on GitHub.
2. Benchmark your workloads with peak active flow count to determine a baseline for your application performance. With a performance baseline, you can test variations in your settings or application design to understand which considerations will have the most impact, especially if you plan to scale up or scale out.

Tune PPS performance

The following list contains actions that you can take to tune your PPS performance, depending on your system needs.

- Reduce the physical distance between two instances. When sending and receiving instances are located in same Availability Zone or use cluster placement groups, you can reduce the number of hops a packet needs to take to travel from one endpoint to another.
- Use Untracked connections.
- Use the UDP protocol for network traffic.
- For EC2 instances with aggregate bandwidth of 100 Gbps or more, distribute the workload over 100 or more individual flows to spread the work evenly across the Nitro card.
- To overcome the egress fragment PPS limit on EC2 instances, you can enable fragment proxy mode (depending on your driver version). This setting allows fragmented packets to be evaluated in the processing path, thereby overcoming the egress PPS limit of 1024. When loading the driver, run one of the following commands to enable or disable fragment proxy mode:

  Enable fragment proxy mode

  ```
  sudo insmod ena.ko enable_frag_bypass=1
  ```

  Disable fragment proxy mode

```
sudo insmod ena.ko enable_frag_bypass=0
```

Configure ENA queue allocation

On supported instance types, you can dynamically allocate these queues across Elastic Network Interfaces (ENIs). Flexible ENA queue allocation optimizes resource distribution, enabling maximum vCPU utilization. High network performance workloads typically require multiple ENA queues. For more information, see [ENA queues](#).

Monitor performance on Linux instances

You can use Ethtool metrics on Linux instances to monitor instance networking performance indicators such as bandwidth, packet rate, and connection tracking. For more information, see [Monitor network performance for ENA settings on your EC2 instance](#).

Optimize network performance on EC2 Windows instances

To achieve the maximum network performance on your Windows 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 configured 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](#).

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 Receive side scaling 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 elastic network interfaces 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 that you use the `Set-NetAdapterRSS` PowerShell cmdlet, which manually excludes the boot processor (logical processor 0 and 1 when hyper-threading is enabled) from the RSS configuration for all elastic network interfaces, in order to prevent contention with various system components.

Windows is hyper-thread aware and ensures that the RSS queues of a single network interface card (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 among a range of 16 logical processors. The `Set-NetAdapterRss` cmdlet 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 elastic network interface ranges depending on the expected workload of the interface (in other words, a low volume administrative network interface may not need as many RSS queues assigned). Also, as previously noted, 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 elastic network interfaces 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.

```
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 elastic network interface. The complete Microsoft documentation for the cmdlet can be found here: Set-NetAdapterRss.

Special note for SQL workloads: We also recommend that you review your I/O thread affinity settings along with your elastic network interface RSS configuration to minimize I/O and network contention for the same CPUs. See Server configuration: affinity mask.

Elastic Fabric Adapter for AI/ML and HPC workloads on Amazon EC2

An Elastic Fabric Adapter (EFA) is a network device that you can attach to your Amazon EC2 instance to accelerate Artificial Intelligence (AI), Machine Learning (ML), and High Performance Computing (HPC) applications. EFA enables you to achieve the application performance of an on-premises AI/ML or HPC cluster, with the scalability, flexibility, and elasticity provided by the AWS Cloud.

EFA provides lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems. It enhances the performance of inter-instance communication that is critical for scaling AI/ML and HPC applications. It is optimized to work on the existing AWS network infrastructure and it can scale depending on application requirements.

EFA integrates with Libfabric 1.7.0 and later, and it supports Nvidia Collective Communications Library (NCCL) for AI and ML applications, and Open MPI 4.1 and later and Intel MPI 2019 Update 5 and later for HPC applications.

EFA supports RDMA (Remote Direct Memory Access) write on most supported instance types that have Nitro version 4 and later. RDMA read is supported on all instances with Nitro version 4 and later. For more information, see Supported instance types.

Contents

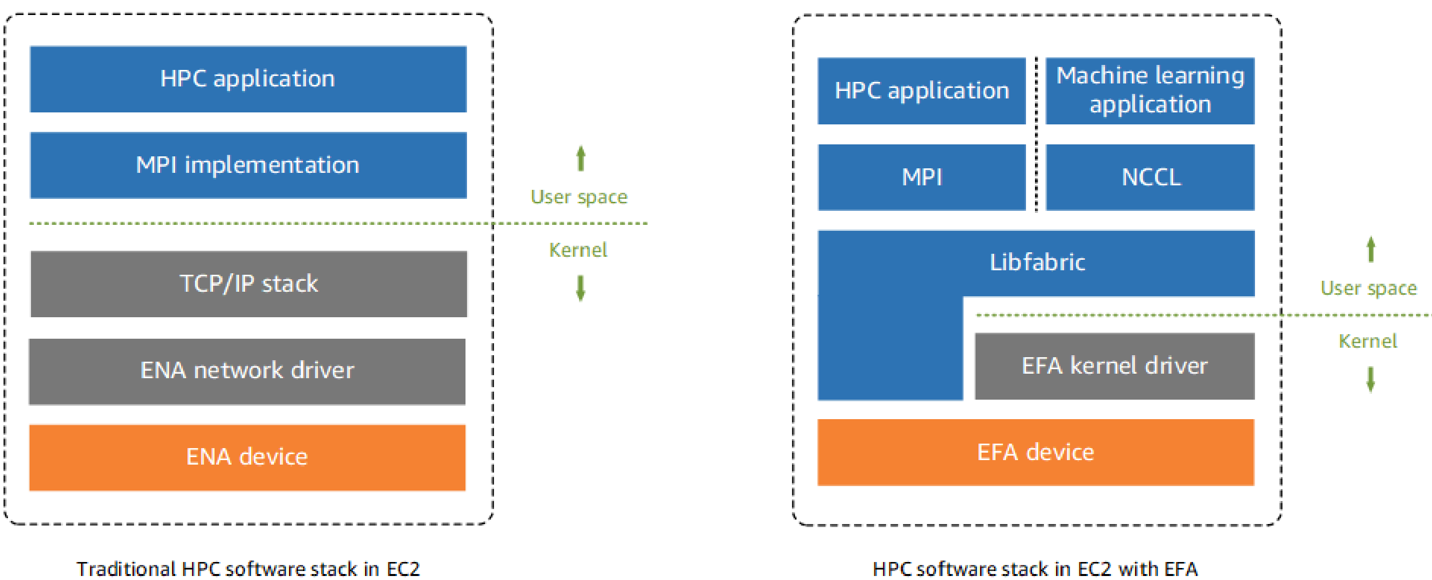
- Get started with EFA and NCCL for ML workloads on Amazon EC2
- Maximize network bandwidth on Amazon EC2 instances with multiple network cards
- Create and attach an Elastic Fabric Adapter to an Amazon EC2 instance
- Detach and delete an EFA from an Amazon EC2 instance
- Monitor an Elastic Fabric Adapter on Amazon EC2
- Verify the EFA installer using a checksum
- Elastic Fabric Adapter release notes

EFA basics

An EFA device can be attached to an EC2 instance in two ways:

1. Using a traditional EFA interface, also called EFA with ENA, which creates both an EFA device and an ENA device.
2. Using an EFA-only interface, which creates just the EFA device.

The EFA device provides capabilities like built-in OS-bypass and congestion control through the Scalable Reliable Datagram (SRD) protocol. The EFA device features enable low-latency, reliable transport functionality that allows EFA interface to provide better application performance for HPC and ML applications on Amazon EC2. While the ENA device offers traditional IP networking.



Traditionally, AI/ML applications use NCCL and HPC applications use the Message Passing Interface (MPI) to interface with the system's network transport. In the AWS cloud, this has meant that applications interface with NCCL or MPI, which then uses the operating system's TCP/IP stack and the ENA device driver to enable network communication between instances.

With a traditional EFA (EFA with ENA) or EFA-only interface, AI/ML applications use NCCL and HPC applications use MPI, to interface directly with the Libfabric API. The Libfabric API bypasses the operating system kernel and communicates directly with the EFA device to put packets on the network. This reduces overhead and enables AI/ML and HPC applications to run more efficiently.

Note

Libfabric is a core component of the OpenFabrics Interfaces (OFI) framework, which defines and exports the user-space API of OFI. For more information, see the Libfabric OpenFabrics website.

Differences between ENA, EFA, and EFA-only network interfaces

Amazon EC2 provides two types of network interfaces:

- **ENA** interfaces provide all of the traditional IP networking and routing features that are required to support IP networking for a VPC. For more information, see Enable enhanced networking with ENA on your EC2 instances.
- **EFA** (EFA with ENA) interfaces provide both the ENA device for IP networking and the EFA device for low-latency, high-throughput communication.
- **EFA-only** interfaces support only the EFA device capabilities, without the ENA device for traditional IP networking.

The following table provides a comparison of ENA, EFA, and EFA-only network interfaces.

	ENA	EFA (EFA with ENA)	EFA-only
Supports IP networking functionality	Yes	Yes	No
Can be assigned	Yes	Yes	No

	ENA	EFA (EFA with ENA)	EFA-only
IPv4 or IPv6 addresses			
Can be used as primary network interface for instance	Yes	Yes	No
Counts towards ENI attacheme nt limit for instance	Yes	Yes	Yes
Instance type support	Supported on all Nitro-based instances types	Supported instance types	Supported instance types
Parameter naming in EC2 APIs	`interface`	`efa`	`efa-only`
Field naming in EC2 console	No selection	EFA with ENA	EFA-only

Supported interfaces and libraries

EFAs support the following interfaces and libraries:

- Open MPI 4.1 and later
- Intel MPI 2019 Update 5 and later
- NVIDIA Collective Communications Library (NCCL) 2.4.2 and later
- AWS Neuron SDK version 2.3 and later

Supported instance types

All of the following instance types support EFA. Additionally, the tables indicate RDMA read and RDMA write support for the instance types.

Nitro v6

Instance type	RDMA read support	RDMA write support
General Purpose		
m8i.48xlarge	No	No
m8i.96xlarge	No	No
m8i.metal-48xl	No	No
m8i.metal-96xl	No	No
Compute Optimized		
c8gn.16xlarge	Yes	Yes
c8gn.24xlarge	Yes	Yes
c8gn.48xlarge	Yes	Yes
c8gn.metal-24xl	Yes	Yes
c8gn.metal-48xl	Yes	Yes

Instance type	RDMA read support	RDMA write support
Memory Optimized		
r8i.48xlarge	No	No
r8i.96xlarge	No	No
r8i.metal-48xl	No	No
r8i.metal-96xl	No	No
Storage Optimized		
i8ge.48xlarge	Yes	No
i8ge.metal-48xl	Yes	No
Accelerated Computing		
p6-b200.48xlarge	Yes	Yes

Nitro v5

Instance type	RDMA read support	RDMA write support
General Purpose		
m8g.24xlarge	Yes	No

Instance type	RDMA read support	RDMA write support
m8g.48xlarge	Yes	No
m8g.metal-24xl	Yes	No
m8g.metal-48xl	Yes	No
m8gd.24xlarge	No	No
m8gd.48xlarge	No	No
m8gd.metal-24xl	No	No
m8gd.metal-48xl	No	No
Compute Optimized		
c7gn.16xlarge	Yes	No
c7gn.metal	Yes	No
c8g.24xlarge	Yes	No
c8g.48xlarge	Yes	No
c8g.metal-24xl	Yes	No

Instance type	RDMA read support	RDMA write support
c8g.metal-48xl	Yes	No
c8gd.24xlarge	No	No
c8gd.48xlarge	No	No
c8gd.metal-24xl	No	No
c8gd.metal-48xl	No	No
Memory Optimized		
r8g.24xlarge	No	No
r8g.48xlarge	No	No
r8g.metal-24xl	No	No
r8g.metal-48xl	No	No
r8gd.24xlarge	No	No
r8gd.48xlarge	No	No
r8gd.metal-24xl	No	No

Instance type	RDMA read support	RDMA write support
r8gd.metal-48xl	No	No
x8g.24xlarge	No	No
x8g.48xlarge	No	No
x8g.metal-24xl	No	No
x8g.metal-48xl	No	No
Storage Optimized		
i7ie.48xlarge	Yes	No
i7ie.metal-48xl	Yes	No
i8g.48xlarge	No	No
Accelerated Computing		
p5en.48xlarge	Yes	Yes
p6e-gb200.36xlarge	Yes	Yes
trn2.48xlarge	Yes	Yes

Instance type	RDMA read support	RDMA write support
trn2u.48xlarge	Yes	Yes
High Performance Computing		
hpc7g.4xlarge	Yes	No
hpc7g.8xlarge	Yes	No
hpc7g.16xlarge	Yes	No

Nitro v4

Instance type	RDMA read support	RDMA write support
General Purpose		
m6a.48xlarge	Yes	Yes
m6a.metal	Yes	Yes
m6i.32xlarge	Yes	Yes
m6i.metal	Yes	Yes
m6id.32xlarge	Yes	Yes
m6id.metal	Yes	Yes

Instance type	RDMA read support	RDMA write support
m6idn.32xlarge	Yes	Yes
m6idn.metal	Yes	Yes
m6in.32xlarge	Yes	Yes
m6in.metal	Yes	Yes
m7a.48xlarge	Yes	No
m7a.metal-48xl	Yes	No
m7g.16xlarge	Yes	No
m7g.metal	Yes	No
m7gd.16xlarge	Yes	No
m7gd.metal	Yes	No
m7i.48xlarge	Yes	No
m7i.metal-48xl	Yes	No
Compute Optimized		

Instance type	RDMA read support	RDMA write support
c6a.48xlarge	Yes	Yes
c6a.metal	Yes	Yes
c6gn.16xlarge	Yes	Yes
c6i.32xlarge	Yes	Yes
c6i.metal	Yes	Yes
c6id.32xlarge	Yes	Yes
c6id.metal	Yes	Yes
c6in.32xlarge	Yes	Yes
c6in.metal	Yes	Yes
c7a.48xlarge	Yes	No
c7a.metal-48xl	Yes	No
c7g.16xlarge	Yes	Yes
c7g.metal	Yes	Yes

Instance type	RDMA read support	RDMA write support
c7gd.16xlarge	Yes	No
c7gd.metal	Yes	No
c7i.48xlarge	Yes	No
c7i.metal-48xl	Yes	No
Memory Optimized		
r6a.48xlarge	Yes	Yes
r6a.metal	Yes	Yes
r6i.32xlarge	Yes	Yes
r6i.metal	Yes	Yes
r6idn.32xlarge	Yes	Yes
r6idn.metal	Yes	Yes
r6in.32xlarge	Yes	Yes
r6in.metal	Yes	Yes

Instance type	RDMA read support	RDMA write support
r6id.32xlarge	Yes	Yes
r6id.metal	Yes	Yes
r7a.48xlarge	No	No
r7a.metal-48xl	No	No
r7g.16xlarge	No	No
r7g.metal	No	No
r7gd.16xlarge	No	No
r7gd.metal	No	No
r7i.48xlarge	No	No
r7i.metal-48xl	No	No
r7iz.32xlarge	No	No
r7iz.metal-32xl	No	No
u7i-6tb.112xlarge	Yes	Yes

Instance type	RDMA read support	RDMA write support
u7i-8tb.112xlarge	Yes	Yes
u7i-12tb.224xlarge	Yes	Yes
u7in-16tb.224xlarge	Yes	Yes
u7in-24tb.224xlarge	Yes	Yes
u7in-32tb.224xlarge	Yes	Yes
u7inh-32tb.480xlarge	Yes	Yes
x2idn.32xlarge	Yes	Yes
x2idn.metal	Yes	Yes
x2iedn.32xlarge	Yes	Yes
x2iedn.metal	Yes	Yes
Storage Optimized		
i4g.16xlarge	Yes	Yes
i4i.32xlarge	Yes	Yes

Instance type	RDMA read support	RDMA write support
i4i.metal	Yes	Yes
i7i.24xlarge	Yes	No
i7i.48xlarge	Yes	No
i7i.metal-48xl	Yes	No
im4gn.16xlarge	Yes	Yes
Accelerated Computing		
f2.48xlarge	Yes	Yes
g6.8xlarge	Yes	Yes
g6.12xlarge	Yes	Yes
g6.16xlarge	Yes	Yes
g6.24xlarge	Yes	Yes
g6.48xlarge	Yes	Yes
g6e.8xlarge	Yes	Yes

Instance type	RDMA read support	RDMA write support
g6e.12xlarge	Yes	Yes
g6e.16xlarge	Yes	Yes
g6e.24xlarge	Yes	Yes
g6e.48xlarge	Yes	Yes
gr6.8xlarge	Yes	Yes
p5.4xlarge	Yes	Yes
p5.48xlarge	Yes	Yes
p5e.48xlarge	Yes	Yes
trn1.32xlarge	Yes	Yes
trn1n.32xlarge	Yes	Yes
High Performance Computing		
hpc6a.48xlarge	Yes	Yes
hpc6id.32xlarge	Yes	Yes

Instance type	RDMA read support	RDMA write support
hpc7a.12xlarge	Yes	No
hpc7a.24xlarge	Yes	No
hpc7a.48xlarge	Yes	No
hpc7a.96xlarge	Yes	No

Nitro v3

Instance type	RDMA read support	RDMA write support
General Purpose		
m5dn.24xlarge	No	No
m5dn.metal	No	No
m5n.24xlarge	No	No
m5n.metal	No	No
m5zn.12xlarge	No	No
m5zn.metal	No	No
Compute Optimized		

Instance type	RDMA read support	RDMA write support
c5n.9xlarge	No	No
c5n.18xlarge	No	No
c5n.metal	No	No
Memory Optimized		
r5dn.24xlarge	No	No
r5dn.metal	No	No
r5n.24xlarge	No	No
r5n.metal	No	No
x2iezn.12xlarge	No	No
x2iezn.metal	No	No
Storage Optimized		
i3en.12xlarge	No	No
i3en.24xlarge	No	No

Instance type	RDMA read support	RDMA write support
i3en.metal	No	No
Accelerated Computing		
dl1.24xlarge	Yes	No
dl2q.24xlarge	No	No
g4dn.8xlarge	No	No
g4dn.12xlarge	No	No
g4dn.16xlarge	No	No
g4dn.metal	No	No
g5.8xlarge	No	No
g5.12xlarge	No	No
g5.16xlarge	No	No
g5.24xlarge	No	No
g5.48xlarge	No	No

Instance type	RDMA read support	RDMA write support
inf1.24xlarge	No	No
p3dn.24xlarge	No	No
p4d.24xlarge	Yes	No
p4de.24xlarge	Yes	No
vt1.24xlarge	No	No

To see the available instance types that support EFAs in a specific Region

The available instance types vary by Region. To see the available instance types that support EFAs in a Region, use the describe-instance-types command with the `--region` parameter. Include the `--filters` parameter to scope the results to the instance types that support EFA and the `--query` parameter to scope the output to the value of `InstanceType`.

```
aws ec2 describe-instance-types \
    --region us-east-1  \
    --filters Name=network-info.efa-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]"  \
    --output text | sort
```

Supported operating systems

Operating system support differs depending on the processor type. The following table shows the supported operating systems.

Operating system	Intel/AMD (x86_64) instance types	AWS Graviton (arm64) instance types
Amazon Linux 2023	✓	✓
Amazon Linux 2	✓	✓
RHEL 8 and 9	✓	✓
Debian 11 and 12	✓	✓
Rocky Linux 8 and 9	✓	✓
Ubuntu 22.04 and 24.04	✓	✓
SUSE Linux Enterprise 15 SP2 and later	✓	✓
OpenSUSE Leap 15.5 and later	✓	

Note

Some of the listed operating systems might not be supported with Intel MPI. If you are using Intel MPI, refer to the Intel MPI documentation to verify support for your operating system.

EFA limitations

EFAs have the following limitations:

- RDMA write is not supported with all instance types. For more information, see Supported instance types.
- EFA traffic[1] between P4d/P4de/DL1 instances and other instance types is currently not supported.
- Instance types that support multiple network cards can be configured with one EFA per network card. All other supported instance types support only one EFA per instance.
- `c7g.16xlarge`, `m7g.16xlarge`, and `r7g.16xlarge` Dedicated Instances and Dedicated Hosts are not supported when an EFA is attached.
- EFA traffic[1] can't cross Availability Zones or VPCs. This does not apply to normal IP traffic from the ENA device of an EFA interface.
- EFA traffic[1] is not routable. Normal IP traffic from the ENA device of an EFA interface remains routable.
- EFA is not supported on AWS Outposts.
- The EFA device of an EFA (EFA with ENA) interface is supported on Windows instances only for AWS Cloud Digital Interface Software Development Kit (AWS CDI SDK) based applications. If you attach an EFA (EFA with ENA) interface to a Windows instance for non-CDI SDK based applications, it functions as an ENA interface, without the added EFA device capabilities. The EFA-only interface is not supported by AWS CDI based applications on Windows or Linux. For more information, see the AWS Cloud Digital Interface Software Development Kit (AWS CDI SDK) User Guide.

[1]*EFA traffic* refers to the traffic transmitted through the EFA device of either an EFA (EFA with ENA) or EFA-only interface.

EFA pricing

EFA is available as an optional Amazon EC2 networking feature that you can enable on any supported instance at no additional cost.

Get started with EFA and MPI for HPC workloads on Amazon EC2

This tutorial helps you to launch an EFA and MPI-enabled instance cluster for HPC workloads.

Note

The `u7i-12tb.224xlarge`, `u7in-16tb.224xlarge`, `u7in-24tb.224xlarge`, and `u7in-32tb.224xlarge` instances can run up to 128 parallel MPI processes with Open MPI or up to 256 parallel MPI processes with Intel MPI.

Tasks

Step 1: Prepare an EFA-enabled security group

An EFA requires a security group that allows all inbound and outbound traffic to and from the security group itself. The following procedure creates a security group that allows all inbound and outbound traffic to and from itself, and that allows inbound SSH traffic from any IPv4 address for SSH connectivity.

Important

This security group is intended for testing purposes only. For your production environments, we recommend that you create an inbound SSH rule that allows traffic only from the IP address from which you are connecting, such as the IP address of your computer, or a range of IP addresses in your local network.

For other scenarios, see [Security group rules for different use cases](#).

To create an EFA-enabled security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups** and then choose **Create security group**.
3. In the **Create security group** window, do the following:
   a. For **Security group name**, enter a descriptive name for the security group, such as `EFA-enabled security group`.
   b. (Optional) For **Description**, enter a brief description of the security group.
   c. For **VPC**, select the VPC into which you intend to launch your EFA-enabled instances.
   d. Choose **Create security group**.
4. Select the security group that you created, and on the **Details** tab, copy the **Security group ID**.
5. With the security group still selected, choose **Actions**, **Edit inbound rules**, and then do the following:
   a. Choose **Add rule**.
   b. For **Type**, choose **All traffic**.
   c. For **Source type**, choose **Custom** and paste the security group ID that you copied into the field.
   d. Choose **Add rule**.
   e. For **Type**, choose **SSH**.
   f. For **Source type**, choose **Anywhere-IPv4**.
   g. Choose **Save rules**.
6. With the security group still selected, choose **Actions**, **Edit outbound rules**, and then do the following:
   a. Choose **Add rule**.
   b. For **Type**, choose **All traffic**.
   c. For **Destination type**, choose **Custom** and paste the security group ID that you copied into the field.
   d. Choose **Save rules**.

Step 2: Launch a temporary instance

Launch a temporary instance that you can use to install and configure the EFA software components. You use this instance to create an EFA-enabled AMI from which you can launch your EFA-enabled instances.

To launch a temporary instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then choose **Launch Instances** to open the new launch instance wizard.
3. (*Optional*) In the **Name and tags** section, provide a name for the instance, such as `EFA-instance`. The name is assigned to the instance as a resource tag (`Name=`*`EFA-instance`*).
4. In the **Application and OS Images** section, select an AMI for one of the supported operating systems.
5. In the **Instance type** section, select a supported instance type.
6. In the **Key pair** section, select the key pair to use for the instance.
7. In the **Network settings** section, choose **Edit**, and then do the following:

   a. For **Subnet**, choose the subnet in which to launch the instance. If you do not select a subnet, you can't enable the instance for EFA.

   b. For **Firewall (security groups)**, choose **Select existing security group**, and then select the security group that you created in the previous step.

   c. Expand the **Advanced network configuration** section.

      For **Network interface 1**, select **Network card index = 0**, **Device index = 0**, and **Interface type = EFA with ENA**.

      (Optional) If you are using a multi-card instance type, such as `p4d.24xlarge` or `p5.48xlarge`, for each additional network interface required, choose **Add network interface**, for **Network card index** select the next unused index, and then select **Device index = 1** and **Interface type = EFA with ENA** or **EFA-only**.
8. In the **Storage** section, configure the volumes as needed.
9. In the **Summary** panel on the right, choose **Launch instance**.

Note

Consider requiring the use of IMDSv2 for the temporary instance as well as the AMI that you will create in Step 9 unless you have already set IMDSv2 as the default for the account. For more information about IMDSv2 configuration steps, see Configure instance metadata options for new instances.

Step 3: Install the EFA software

Install the EFA-enabled kernel, EFA drivers, Libfabric, and Open MPI stack that is required to support EFA on your temporary instance.

The steps differ depending on whether you intend to use EFA with Open MPI, with Intel MPI, or with Open MPI and Intel MPI.

Note

Some operating systems might not be supported with Intel MPI. If you are using Intel MPI, refer to the Intel MPI documentation to verify support for your operating system.

To install the EFA software

1. Connect to the instance you launched. For more information, see Connect to your Linux instance using SSH.
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes.
   - Amazon Linux 2023, Amazon Linux 2, RHEL 8/9, Rocky Linux 8/9

     ```
     $ sudo yum update -y
     ```
   - Ubuntu and Debian

     ```
     $ sudo apt-get update && sudo apt-get upgrade -y
     ```
   - SUSE Linux Enterprise

```
$ sudo zypper update -y
```

3. Reboot the instance and reconnect to it.

4. Download the EFA software installation files. The software installation files are packaged into a compressed tarball (`.tar.gz`) file. To download the latest *stable* version, use the following command.

   You can also get the latest version by replacing the version number with `latest` in the preceding command.

   ```
   $ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.43.2.tar.gz
   ```

5. (*Optional*) Verify the authenticity and integrity of the EFA tarball (`.tar.gz`) file.

   We recommend that you do this to verify the identity of the software publisher and to check that the file has not been altered or corrupted since it was published. If you do not want to verify the tarball file, skip this step.

   > **Note**
   >
   > Alternatively, if you prefer to verify the tarball file by using an MD5 or SHA256 checksum instead, see Verify the EFA installer using a checksum.

   a. Download the public GPG key and import it into your keyring.

      ```
      $ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
      import aws-efa-installer.key
      ```

      The command should return a key value. Make a note of the key value, because you need it in the next step.

   b. Verify the GPG key's fingerprint. Run the following command and specify the key value from the previous step.

      ```
      $ gpg --fingerprint key_value
      ```

The command should return a fingerprint that is identical to `4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC`. If the fingerprint does not match, don't run the EFA installation script, and contact Support.

c. Download the signature file and verify the signature of the EFA tarball file.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.43.2.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.43.2.tar.gz.sig
```

The following shows example output.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

If the result includes `Good signature`, and the fingerprint matches the fingerprint returned in the previous step, proceed to the next step. If not, don't run the EFA installation script, and contact Support.

6. Extract the files from the compressed `.tar.gz` file and navigate into the extracted directory.

```
$ tar -xf aws-efa-installer-1.43.2.tar.gz && cd aws-efa-installer
```

7. Install the EFA software. Do one of the following depending on your use case.

> **Note**
>
> **EFA does not support NVIDIA GPUDirect with SUSE Linux**. If you are using SUSE Linux, you must additionally specify the `--skip-kmod` option to prevent kmod installation. By default, SUSE Linux does not allow out-of-tree kernel modules.

Open MPI and Intel MPI

If you intend to use EFA with Open MPI and Intel MPI, you must install the EFA software with Libfabric and Open MPI, and you **must complete Step 5: Install Intel MPI**.

To install the EFA software with Libfabric and Open MPI, run the following command.

Note

From EFA 1.30.0, both Open MPI 4.1 and Open MPI 5 are installed by default. You can optionally specify the version of Open MPI that you want to install. To install only Open MPI 4.1, include `--mpi=openmpi4`. To install only Open MPI 5, include `--mpi=openmpi5`. To install both, omit the `--mpi` option.

```
$ sudo ./efa_installer.sh -y
```

Libfabric is installed to `/opt/amazon/efa`. Open MPI 4.1 is installed to `/opt/amazon/openmpi`. Open MPI 5 is installed to `/opt/amazon/openmpi5`.

Open MPI only

If you intend to use EFA with Open MPI only, you must install the EFA software with Libfabric and Open MPI, and you can **skip Step 5: Install Intel MPI**. To install the EFA software with Libfabric and Open MPI, run the following command.

Note

From EFA 1.30.0, both Open MPI 4.1 and Open MPI 5 are installed by default. You can optionally specify the version of Open MPI that you want to install. To install only Open MPI 4.1, include `--mpi=openmpi4`. To install only Open MPI 5, include `--mpi=openmpi5`. To install both, omit the `--mpi` option.

```
$ sudo ./efa_installer.sh -y
```

Libfabric is installed to `/opt/amazon/efa`. Open MPI 4.1 is installed to `/opt/amazon/openmpi`. Open MPI 5 is installed to `/opt/amazon/openmpi5`.

Intel MPI only

If you intend to use EFA with Intel MPI only, you can install the EFA software without Libfabric and Open MPI. In this case, Intel MPI uses its embedded Libfabric. If you choose to do this, you **must complete Step 5: Install Intel MPI**.

To install the EFA software without Libfabric and Open MPI, run the following command.

```
$ sudo ./efa_installer.sh -y --minimal
```

8. If the EFA installer prompts you to reboot the instance, do so and then reconnect to the instance. Otherwise, log out of the instance and then log back in to complete the installation.
9. Delete the uncompressed tarball and the tarball itself. Otherwise, these will be included in the EFA-enabled AMI that you create, increasing its size.

Step 4: (*Optional*) Enable Open MPI 5

> **Note**
>
> Perform this step only if you intend to use Open MPI 5.

From EFA 1.30.0, both Open MPI 4.1 and Open MPI 5 are installed by default. Alternatively, you can choose to install only Open MPI 4.1 or Open MPI 5.

If you chose to install Open MPI 5 in **Step 3: Install the EFA software**, and you intend to use it, you must perform the following steps to enable it.

To enable Open MPI 5

1. Add Open MPI 5 to the PATH environment variable.

```
$ module load openmpi5
```

2. Verify that Open MPI 5 is enabled for use.

```
$ which mpicc
```

The command should return the Open MPI 5 installation directory - `/opt/amazon/openmpi5`.

3. (*Optional*) To ensure that Open MPI 5 is added to PATH environment variable each time the instance starts, do the following:

bash shell

Add `module load openmpi5` to `/home/username/.bashrc` and `/home/username/.bash_profile`.

csh and tcsh shells

Add `module load openmpi5` to `/home/username/.cshrc`.

If you need to remove Open MPI 5 from the PATH environment variable, run the following command and remove the command from the shell startup scripts.

```
$ module unload openmpi5
```

Step 5: (*Optional*) Install Intel MPI

Important

Perform this step only if you intend to use Intel MPI. If you intend to only use Open MPI, skip this step.

Intel MPI requires an additional installation and environment variable configuration.

Prerequisite

Ensure that the user performing the following steps has sudo permissions.

To install Intel MPI

1. To download the Intel MPI installation script, do the following

   a. Visit the Intel website.

   b. In the **Intel MPI Library** section of the webpage, choose the link for the **Intel MPI Library for Linux Offline** installer.

2. Run the installation script that you downloaded in the previous step.

   ```
   $ sudo bash installation_script_name.sh
   ```

3. In the installer, choose **Accept & install**.

4. Read the Intel Improvement Program, choose the appropriate option, and then choose **Begin Installation**.

5. When the installation completes, choose **Close**.

6. By default, Intel MPI uses its embedded (internal) Libfabric. You can configure Intel MPI to use the Libfabric that ships with the EFA installer instead. Typically, the EFA installer ships with a later version of Libfabric than Intel MPI. In some cases, the Libfabric that ships with the EFA installer is more performant than that of Intel MPI. To configure Intel MPI to use the Libfabric that ships with the EFA installer, do one of the following depending on your shell.

   bash shells

   Add the following statement to `/home/username/.bashrc` and `/home/username/.bash_profile`.

   ```
   export I_MPI_OFI_LIBRARY_INTERNAL=0
   ```

   csh and tcsh shells

   Add the following statement to `/home/username/.cshrc`.

   ```
   setenv I_MPI_OFI_LIBRARY_INTERNAL 0
   ```

7. Add the following **source** command to your shell script to source the `vars.sh` script from the installation directory to set up the compiler environment each time the instance starts. Do one of the following depending on your shell.

   bash shells

   Add the following statement to `/home/username/.bashrc` and `/home/username/.bash_profile`.

   ```
   source /opt/intel/oneapi/mpi/latest/env/vars.sh
   ```

   csh and tcsh shells

   Add the following statement to `/home/username/.cshrc`.

   ```
   source /opt/intel/oneapi/mpi/latest/env/vars.csh
   ```

8. By default, if EFA is not available due to a misconfiguration, Intel MPI defaults to the TCP/IP network stack, which might result in slower application performance. You can prevent this by setting `I_MPI_OFI_PROVIDER` to `efa`. This causes Intel MPI to fail with the following error if EFA is not available:

    ```
    Abort (XXXXXX) on node 0 (rank 0 in comm 0): Fatal error in PMPI_Init: OtherMPI
     error,
    MPIR_Init_thread (XXX)........:
    MPID_Init (XXXX)..............:
    MPIDI_OFI_mpi_init_hook (XXXX):
    open_fabric (XXXX)............:
    find_provider (XXXX)..........:
    OFI fi_getinfo() failed (ofi_init.c:2684:find_provider:
    ```

    Do one of the following depending on your shell.

    bash shells

    Add the following statement to `/home/username/.bashrc` and `/home/username/.bash_profile`.

    ```
    export I_MPI_OFI_PROVIDER=efa
    ```

    csh and tcsh shells

    Add the following statement to `/home/username/.cshrc`.

    ```
    setenv I_MPI_OFI_PROVIDER efa
    ```

9. By default, Intel MPI doesn't print debugging information. You can specify different verbosity levels to control the debugging information. Possible values (in order of the amount of detail they provide) are: `0` (default), 1, 2, 3, 4, 5. Level 1 and higher prints the `libfabric version` and `libfabric provider`. Use `libfabric version` to check whether Intel MPI is using the internal Libfabric or the Libfabric that ships with the EFA installer. If it's using the internal Libfabric, the version is suffixed with `impi`. Use `libfabric provider` to check with Intel MPI is using EFA or the TCP/IP network. If it's using EFA, the value is `efa`. If it's using TCP/IP, the value is `tcp;ofi_rxm`.

    To enable debugging information, do one of the following depending on your shell.

bash shells

Add the following statement to /home/*username*/.bashrc and /home/*username*/.bash_profile.

```
export I_MPI_DEBUG=value
```

csh and tcsh shells

Add the following statement to /home/*username*/.cshrc.

```
setenv I_MPI_DEBUG value
```

10. By default, Intel MPI uses the operating system's shared memory (shm) for intra-node communication, and it uses Libfabric (ofi) only for inter-node communication. Generally, this configuration provides the best performance. However, in some cases the Intel MPI shm fabric can cause certain applications to hang indefinitely.

    To resolve this issue, you can force Intel MPI to use Libfabric for both intra-node and inter-node communication. To do this, do one of the following depending on your shell.

    bash shells

    Add the following statement to /home/*username*/.bashrc and /home/*username*/.bash_profile.

    ```
    export I_MPI_FABRICS=ofi
    ```

    csh and tcsh shells

    Add the following statement to /home/*username*/.cshrc.

    ```
    setenv I_MPI_FABRICS ofi
    ```

Note

The EFA Libfabric provider uses the operating system's shared memory for intra-node communication. This means that setting `I_MPI_FABRICS` to `ofi` yields similar performance to the default `shm:ofi` configuration.

11. Log out of the instance and then log back in.

If you no longer want to use Intel MPI, remove the environment variables from the shell startup scripts.

Step 6: Disable ptrace protection

To improve your HPC application's performance, Libfabric uses the instance's local memory for interprocess communications when the processes are running on the same instance.

The shared memory feature uses Cross Memory Attach (CMA), which is not supported with *ptrace protection*. If you are using a Linux distribution that has ptrace protection enabled by default, such as Ubuntu, you must disable it. If your Linux distribution does not have ptrace protection enabled by default, skip this step.

To disable ptrace protection

Do one of the following:

- To temporarily disable ptrace protection for testing purposes, run the following command.

```
$ sudo sysctl -w kernel.yama.ptrace_scope=0
```

- To permanently disable ptrace protection, add `kernel.yama.ptrace_scope = 0` to `/etc/sysctl.d/10-ptrace.conf` and reboot the instance.

Step 7. Confirm installation

To confirm successful installation

1. To confirm that MPI was successfully installed, run the following command:

```
$ which mpicc
```

- For Open MPI, the returned path should include `/opt/amazon/`
- For Intel MPI, the returned path should include `/opt/intel/`. If you do not get the expected output, ensure that you have sourced the Intel MPI `vars.sh` script.

2. To confirm that the EFA software components and Libfabric were successfully installed, run the following command.

```
$ fi_info -p efa -t FI_EP_RDM
```

The command should return information about the Libfabric EFA interfaces. The following example shows the command output.

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
```

Step 8: Install your HPC application

Install the HPC application on the temporary instance. The installation procedure varies depending on the specific HPC application. For more information, see [Manage software on your AL2 instance] in the *Amazon Linux 2 User Guide*.

> **Note**
>
> Refer to your HPC application's documentation for installation instructions.

Step 9: Create an EFA-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions**, **Image**, **Create image**.
4. For **Create image**, do the following:

   a. For **Image name**, enter a descriptive name for the AMI.

   b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.

   c. Choose **Create image**.
5. In the navigation pane, choose **AMIs**.
6. Locate the AMI tht you created in the list. Wait for the status to change from `pending` to `available` before continuing to the next step.

Step 10: Launch EFA-enabled instances into a cluster placement group

Launch your EFA-enabled instances into a cluster placement group using the EFA-enabled AMI that you created in **Step 7**, and the EFA-enabled security group that you created in **Step 1**.

> **Note**
>
> - It is not an absolute requirement to launch your EFA-enabled instances into a cluster placementgroup. However, we do recommend running your EFA-enabled instances in a cluster placement group as it launches the instances into a low-latency group in a single Availability Zone.
> - To ensure that capacity is available as you scale your cluster's instances, you can create a Capacity Reservation for your cluster placement group. For more information, see Use Capacity Reservations with cluster placement groups.

To launch an instance

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** to open the new launch instance wizard.

3. (*Optional*) In the **Name and tags** section, provide a name for the instance, such as `EFA-instance`. The name is assigned to the instance as a resource tag (`Name=EFA-instance`).
4. In the **Application and OS Images** section, choose **My AMIs**, and then select the AMI that you created in the previous step.
5. In the **Instance type** section, select a supported instance type.
6. In the **Key pair** section, select the key pair to use for the instance.
7. In the **Network settings** section, choose **Edit**, and then do the following:

    a. For **Subnet**, choose the subnet in which to launch the instance. If you do not select a subnet, you can't enable the instance for EFA.

    b. For **Firewall (security groups)**, choose **Select existing security group**, and then select the security group that you created in the previous step.

    c. Expand the **Advanced network configuration** section.

    For **Network interface 1**, select **Network card index = 0**, **Device index = 0**, and **Interface type = EFA with ENA**.

    (*Optional*) If you are using a multi-card instance type, such as `p4d.24xlarge` or `p5.48xlarge`, for each additional network interface required, choose **Add network interface**, for **Network card index** select the next unused index, and then select **Device index = 1** and **Interface type = EFA with ENA** or **EFA-only**.
8. (*Optional*) In the **Storage** section, configure the volumes as needed.
9. In the **Advanced details** section, for **Placement group name**, select the cluster placement group into which to launch the instances. If you need to create a new cluster placement group, choose **Create new placement group**.
10. In the **Summary** panel on the right, for **Number of instances**, enter the number of EFA-enabled instances that you want to launch, and then choose **Launch instance**.

Step 11: Terminate the temporary instance

At this point, you no longer need the instance that you launched in Step 2. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

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

2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions**, **Instance state**, **Terminate (delete) instance**.
4. When prompted for confirmation, choose **Terminate (delete)**.

Step 12: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

   The key pair is created in the `$HOME/.ssh/` directory.
4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.
6. For each member node in the cluster, do the following:

   a. Connect to the instance.

b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.

7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Get started with EFA and NCCL for ML workloads on Amazon EC2

The NVIDIA Collective Communications Library (NCCL) is a library of standard collective communication routines for multiple GPUs across a single node or multiple nodes. NCCL can be used together with EFA, Libfabric, and MPI to support various machine learning workloads. For more information, see the NCCL website.

Requirements

- Only accelerated computing P series instance types are supported. For more information, see Amazon EC2 accelerated computing instances.
- Only Amazon Linux 2023, Amazon Linux 2, Ubuntu 24.04, and Ubuntu 22.04 base AMIs are supported.
- Only NCCL 2.4.2 and later is supported with EFA.

For more information about running machine learning workloads with EFA and NCCL using an AWS Deep Learning AMIs, see Using EFA on the DLAMI in the *AWS Deep Learning AMIs Developer Guide*.

Steps

- Step 1: Prepare an EFA-enabled security group
- Step 2: Launch a temporary instance
- Step 3: Install Nvidia GPU drivers, Nvidia CUDA toolkit, and cuDNN
- Step 4: Install GDRCopy
- Step 5: Install the EFA software
- Step 6: Install NCCL
- Step 7: Install the NCCL tests

Step 1: Prepare an EFA-enabled security group

An EFA requires a security group that allows all inbound and outbound traffic to and from the security group itself. The following procedure creates a security group that allows all inbound and outbound traffic to and from itself, and that allows inbound SSH traffic from any IPv4 address for SSH connectivity.

> **Important**
>
> This security group is intended for testing purposes only. For your production environments, we recommend that you create an inbound SSH rule that allows traffic only from the IP address from which you are connecting, such as the IP address of your computer, or a range of IP addresses in your local network.

For other scenarios, see Security group rules for different use cases.

To create an EFA-enabled security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups** and then choose **Create security group**.
3. In the **Create security group** window, do the following:

   a. For **Security group name**, enter a descriptive name for the security group, such as `EFA-enabled security group`.
   b. (Optional) For **Description**, enter a brief description of the security group.
   c. For **VPC**, select the VPC into which you intend to launch your EFA-enabled instances.
   d. Choose **Create security group**.
4. Select the security group that you created, and on the **Details** tab, copy the **Security group ID**.

5. With the security group still selected, choose **Actions**, **Edit inbound rules**, and then do the following:

   a. Choose **Add rule**.
   b. For **Type**, choose **All traffic**.
   c. For **Source type**, choose **Custom** and paste the security group ID that you copied into the field.
   d. Choose **Add rule**.
   e. For **Type**, choose **SSH**.
   f. For **Source type**, choose **Anywhere-IPv4**.
   g. Choose **Save rules**.

6. With the security group still selected, choose **Actions**, **Edit outbound rules**, and then do the following:

   a. Choose **Add rule**.
   b. For **Type**, choose **All traffic**.
   c. For **Destination type**, choose **Custom** and paste the security group ID that you copied into the field.
   d. Choose **Save rules**.

Step 2: Launch a temporary instance

Launch a temporary instance that you can use to install and configure the EFA software components. You use this instance to create an EFA-enabled AMI from which you can launch your EFA-enabled instances.

To launch a temporary instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then choose **Launch Instances** to open the new launch instance wizard.
3. (*Optional*) In the **Name and tags** section, provide a name for the instance, such as `EFA-instance`. The name is assigned to the instance as a resource tag (`Name=`*`EFA-instance`*).
4. In the **Application and OS Images** section, select an AMI for one of the supported operating systems.

5. In the **Instance type** section, select a supported instance type.
6. In the **Key pair** section, select the key pair to use for the instance.
7. In the **Network settings** section, choose **Edit**, and then do the following:

    a. For **Subnet**, choose the subnet in which to launch the instance. If you do not select a subnet, you can't enable the instance for EFA.

    b. For **Firewall (security groups)**, choose **Select existing security group**, and then select the security group that you created in the previous step.

    c. Expand the **Advanced network configuration** section.

    For **Network interface 1**, select **Network card index = 0**, **Device index = 0**, and **Interface type = EFA with ENA**.

    (*Optional*) If you are using a multi-card instance type, such as `p4d.24xlarge` or `p5.48xlarge`, for each additional network interface required, choose **Add network interface**, for **Network card index** select the next unused index, and then select **Device index = 1** and **Interface type = EFA with ENA** or **EFA-only**.

8. In the **Storage** section, configure the volumes as needed.

> **Note**
>
> You must provision an additional 10 to 20 GiB of storage for the Nvidia CUDA Toolkit. If you do not provision enough storage, you will receive an `insufficient disk space` error when attempting to install the Nvidia drivers and CUDA toolkit.

9. In the **Summary** panel on the right, choose **Launch instance**.

Step 3: Install Nvidia GPU drivers, Nvidia CUDA toolkit, and cuDNN

Amazon Linux 2023 and Amazon Linux 2

To install the Nvidia GPU drivers, Nvidia CUDA toolkit, and cuDNN

1. To ensure that all of your software packages are up to date, perform a quick software update on your instance.

```
$ sudo yum upgrade -y && sudo reboot
```

After the instance has rebooted, reconnect to it.

2. Install the utilities that are needed to install the Nvidia GPU drivers and the Nvidia CUDA toolkit.

```
$ sudo yum groupinstall 'Development Tools' -y
```

3. Disable the nouveau open source drivers.

   a. Install the required utilities and the kernel headers package for the version of the kernel that you are currently running.

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

   b. Add nouveau to the /etc/modprobe.d/blacklist.conf deny list file.

   ```
   $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   c. Append GRUB_CMDLINE_LINUX="rdblacklist=nouveau" to the grub file and rebuild the Grub configuration.

   ```
   $ echo 'GRUB_CMDLINE_LINUX="rdblacklist=nouveau"' | sudo tee -a /etc/
   default/grub \
   && sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

4. Reboot the instance and reconnect to it.

5. Prepare the required repositories

   a. Enable the EPEL repository and set the distribution to rhel7.

   ```
   $ sudo amazon-linux-extras install epel \
   && distribution='rhel7'
   ```

   b. Set up the CUDA network repository and update the repository cache.

```
$ ARCH=$( /bin/arch ) \
&& sudo yum-config-manager --add-repo http://developer.download.nvidia.com/
compute/cuda/repos/$distribution/${ARCH}/cuda-$distribution.repo \
&& sudo yum clean expire-cache
```

c. (*Kernel version 5.10 only*) Perform these steps only if you are using Amazon Linux 2 with kernel version 5.10. If you are using Amazon Linux 2 with kernel version 4.12, skip these steps. To check your kernel version, run **uname -r**.

   i. Create the Nvidia driver configuration file named `/etc/dkms/nvidia.conf`.

```
$ sudo mkdir -p /etc/dkms \
&& echo "MAKE[0]=\"'make' -j2 module SYSSRC=\${kernel_source_dir}
 IGNORE_XEN_PRESENCE=1 IGNORE_PREEMPT_RT_PRESENCE=1 IGNORE_CC_MISMATCH=1
 CC=/usr/bin/gcc10-gcc\"" | sudo tee /etc/dkms/nvidia.conf
```

   ii. (`p4d.24xlarge` and `p5.48xlarge` only) Copy the Nvidia driver configuration file.

```
$ sudo cp /etc/dkms/nvidia.conf /etc/dkms/nvidia-open.conf
```

6. Install the Nvidia GPU drivers, NVIDIA CUDA toolkit, and cuDNN.

```
$ sudo yum clean all \
&& sudo yum -y install nvidia-driver-latest-dkms \
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcudnn8-devel
```

7. Reboot the instance and reconnect to it.

8. (`p4d.24xlarge` and `p5.48xlarge` only) Start the Nvidia Fabric Manager service, and ensure that it starts automatically when the instance starts. Nvidia Fabric Manager is required for NV Switch Management.

```
$ sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
fabricmanager
```

9. Ensure that the CUDA paths are set each time that the instance starts.

   - For *bash* shells, add the following statements to `/home/username/.bashrc` and `/home/username/.bash_profile`.

```
export PATH=/usr/local/cuda/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

- For *tcsh* shells, add the following statements to `/home/username/.cshrc`.

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

10. To confirm that the Nvidia GPU drivers are functional, run the following command.

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

The command should return information about the Nvidia GPUs, Nvidia GPU drivers, and Nvidia CUDA toolkit.

Ubuntu 24.04 and Ubuntu 22.04

To install the Nvidia GPU drivers, Nvidia CUDA toolkit, and cuDNN

1. To ensure that all of your software packages are up to date, perform a quick software update on your instance.

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

2. Install the utilities that are needed to install the Nvidia GPU drivers and the Nvidia CUDA toolkit.

```
$ sudo apt-get update && sudo apt-get install build-essential -y
```

3. To use the Nvidia GPU driver, you must first disable the `nouveau` open source drivers.

   a. Install the required utilities and the kernel headers package for the version of the kernel that you are currently running.

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

   b. Add nouveau to the `/etc/modprobe.d/blacklist.conf` deny list file.

   ```
   $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   ```

```
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

c. Open `/etc/default/grub` using your preferred text editor and add the following.

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

d. Rebuild the Grub configuration.

```
$ sudo update-grub
```

4. Reboot the instance and reconnect to it.

5. Add the CUDA repository and install the Nvidia GPU drivers, NVIDIA CUDA toolkit, and cuDNN.

- `p3dn.24xlarge`

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-dkms-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535
 cuda-toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

- `p4d.24xlarge` and `p5.48xlarge`

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-kernel-open-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535
 cuda-toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

6. Reboot the instance and reconnect to it.

7. (`p4d.24xlarge` and `p5.48xlarge` only) Install the Nvidia Fabric Manager.

   a. You must install the version of the Nvidia Fabric Manager that matches the version of the Nvidia kernel module that you installed in the previous step.

      Run the following command to determine the version of the Nvidia kernel module.

      ```
      $ cat /proc/driver/nvidia/version | grep "Kernel Module"
      ```

      The following is example output.

      ```
      NVRM version: NVIDIA UNIX x86_64 Kernel Module  450.42.01  Tue Jun 15
       21:26:37 UTC 2021
      ```

      In the example above, major version `450` of the kernel module was installed. This means that you need to install Nvidia Fabric Manager version `450`.

   b. Install the Nvidia Fabric Manager. Run the following command and specify the major version identified in the previous step.

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-major_version_number
```

For example, if major version 450 of the kernel module was installed, use the following command to install the matching version of Nvidia Fabric Manager.

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-450
```

c. Start the service, and ensure that it starts automatically when the instance starts. Nvidia Fabric Manager is required for NV Switch Management.

```
$ sudo systemctl start nvidia-fabricmanager && sudo systemctl enable nvidia-
fabricmanager
```

8. Ensure that the CUDA paths are set each time that the instance starts.

   - For *bash* shells, add the following statements to `/home/username/.bashrc` and `/home/username/.bash_profile`.

   ```
   export PATH=/usr/local/cuda/bin:$PATH
   export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
   lib64:$LD_LIBRARY_PATH
   ```

   - For *tcsh* shells, add the following statements to `/home/username/.cshrc`.

   ```
   setenv PATH=/usr/local/cuda/bin:$PATH
   setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
   lib64:$LD_LIBRARY_PATH
   ```

9. To confirm that the Nvidia GPU drivers are functional, run the following command.

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

   The command should return information about the Nvidia GPUs, Nvidia GPU drivers, and Nvidia CUDA toolkit.

Step 4: Install GDRCopy

Install GDRCopy to improve the performance of Libfabric. For more information about GDRCopy, see the GDRCopy repository.

Amazon Linux 2023 and Amazon Linux 2

To install GDRCopy

1. Install the required dependencies.

```
$ sudo yum -y install dkms rpm-build make check check-devel subunit subunit-devel
```

2. Download and extract the GDRCopy package.

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz ; cd gdrcopy-2.4/packages
```

3. Build the GDRCopy RPM package.

```
$ CUDA=/usr/local/cuda ./build-rpm-packages.sh
```

4. Install the GDRCopy RPM package.

```
$ sudo rpm -Uvh gdrcopy-kmod-2.4-1dkms.noarch*.rpm \
&& sudo rpm -Uvh gdrcopy-2.4-1.x86_64*.rpm \
&& sudo rpm -Uvh gdrcopy-devel-2.4-1.noarch*.rpm
```

Ubuntu 24.04 and Ubuntu 22.04

To install GDRCopy

1. Install the required dependencies.

```
$ sudo apt -y install build-essential devscripts debhelper check libsubunit-dev
 fakeroot pkg-config dkms
```

2. Download and extract the GDRCopy package.

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
```

```
&& tar xf v2.4.tar.gz \
&& cd gdrcopy-2.4/packages
```

3. Build the GDRCopy RPM package.

```
$ CUDA=/usr/local/cuda ./build-deb-packages.sh
```

4. Install the GDRCopy RPM package.

```
$ sudo dpkg -i gdrdrv-dkms_2.4-1_amd64.*.deb \
&& sudo dpkg -i libgdrapi_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy-tests_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy_2.4-1_amd64.*.deb
```

Step 5: Install the EFA software

Install the EFA-enabled kernel, EFA drivers, Libfabric, aws-ofi-nccl plugin, and Open MPI stack that is required to support EFA on your instance.

To install the EFA software

1. Connect to the instance you launched. For more information, see Connect to your Linux instance using SSH.
2. Download the EFA software installation files. The software installation files are packaged into a compressed tarball (`.tar.gz`) file. To download the latest *stable* version, use the following command.

   You can also get the latest version by replacing the version number with `latest` in the preceding command.

   ```
   $ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.43.2.tar.gz
   ```
3. (*Optional*) Verify the authenticity and integrity of the EFA tarball (`.tar.gz`) file.

   We recommend that you do this to verify the identity of the software publisher and to check that the file has not been altered or corrupted since it was published. If you do not want to verify the tarball file, skip this step.

> **Note**
>
> Alternatively, if you prefer to verify the tarball file by using an MD5 or SHA256 checksum instead, see Verify the EFA installer using a checksum.

a. Download the public GPG key and import it into your keyring.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
import aws-efa-installer.key
```

The command should return a key value. Make a note of the key value, because you need it in the next step.

b. Verify the GPG key's fingerprint. Run the following command and specify the key value from the previous step.

```
$ gpg --fingerprint key_value
```

The command should return a fingerprint that is identical to `4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC`. If the fingerprint does not match, don't run the EFA installation script, and contact Support.

c. Download the signature file and verify the signature of the EFA tarball file.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.43.2.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.43.2.tar.gz.sig
```

The following shows example output.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

If the result includes `Good signature`, and the fingerprint matches the fingerprint returned in the previous step, proceed to the next step. If not, don't run the EFA installation script, and contact Support.

4. Extract the files from the compressed `.tar.gz` file and navigate into the extracted directory.

```
$ tar -xf aws-efa-installer-1.43.2.tar.gz && cd aws-efa-installer
```

5. Run the EFA software installation script.

> **Note**
>
> From EFA 1.30.0, both Open MPI 4.1 and Open MPI 5 are installed by default. Unless you need Open MPI 5, we recommend that you install only Open MPI 4.1. The following command installs Open MPI 4.1 only. If you want to install Open MPI 4.1 and Open MPI 5, remove `--mpi=openmpi4`.

```
$ sudo ./efa_installer.sh -y --mpi=openmpi4
```

Libfabric is installed in the `/opt/amazon/efa` directory. The **aws-ofi-nccl plugin** is installed in the `/opt/amazon/ofi-nccl` directory. **Open MPI** is installed in the `/opt/amazon/openmpi` directory.

6. If the EFA installer prompts you to reboot the instance, do so and then reconnect to the instance. Otherwise, log out of the instance and then log back in to complete the installation.

7. Confirm that the EFA software components were successfully installed.

```
$ fi_info -p efa -t FI_EP_RDM
```

The command should return information about the Libfabric EFA interfaces. The following example shows the command output.

- `p3dn.24xlarge` with single network interface

```
provider: efa
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-rdm
version: 2.0
```

```
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

- `p4d.24xlarge` and `p5.48xlarge` with multiple network interfaces

```
provider: efa
fabric: EFA-fe80::c6e:8fff:fef6:e7ff
domain: efa_0-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c34:3eff:feb2:3c35
domain: efa_1-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c0f:7bff:fe68:a775
domain: efa_2-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::ca7:b0ff:fea6:5e99
domain: efa_3-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

Step 6: Install NCCL

Install NCCL. For more information about NCCL, see the NCCL repository.

To install NCCL

1. Navigate to the `/opt` directory.

```
$ cd /opt
```

2. Clone the official NCCL repository to the instance and navigate into the local cloned repository.

```
$ sudo git clone https://github.com/NVIDIA/nccl.git -b v2.23.4-1 && cd nccl
```

3. Build and install NCCL and specify the CUDA installation directory.

```
$ sudo make -j src.build CUDA_HOME=/usr/local/cuda
```

Step 7: Install the NCCL tests

Install the NCCL tests. The NCCL tests enable you to confirm that NCCL is properly installed and that it is operating as expected. For more information about the NCCL tests, see the nccl-tests repository.

To install the NCCL tests

1. Navigate to your home directory.

```
$ cd $HOME
```

2. Clone the official nccl-tests repository to the instance and navigate into the local cloned repository.

```
$ git clone https://github.com/NVIDIA/nccl-tests.git && cd nccl-tests
```

3. Add the Libfabric directory to the LD_LIBRARY_PATH variable.

   - Amazon Linux 2023 and Amazon Linux 2

   ```
   $ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
   ```

   - Ubuntu 24.04 and Ubuntu 22.04

   ```
   $ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
   ```

4. Install the NCCL tests and specify the MPI, NCCL, and CUDA installation directories.

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=/opt/nccl/build CUDA_HOME=/usr/
local/cuda
```

Step 8: Test your EFA and NCCL configuration

Run a test to ensure that your temporary instance is properly configured for EFA and NCCL.

To test your EFA and NCCL configuration

1. Create a host file that specifies the hosts on which to run the tests. The following command creates a host file named `my-hosts` that includes a reference to the instance itself.

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
    "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
   meta-data/local-ipv4 >> my-hosts
   ```

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-
   hosts
   ```

2. Run the test and specify the host file (`--hostfile`) and the number of GPUs to use (`-n`). The following command runs the `all_reduce_perf` test on 8 GPUs on the instance itself, and specifies the following environment variables.

   - `FI_EFA_USE_DEVICE_RDMA=1`—(`p4d.24xlarge` only) uses the device's RDMA functionality for one-sided and two-sided transfer.
   - `NCCL_DEBUG=INFO`—enables detailed debugging output. You can also specify `VERSION` to print only the NCCL version at the start of the test, or `WARN` to receive only error messages.

   For more information about the NCCL test arguments, see the NCCL Tests README in the official nccl-tests repository.

   - `p3dn.24xlarge`

   ```
   $ /opt/amazon/openmpi/bin/mpirun \
   -x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/
   lib:/opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
   -x NCCL_DEBUG=INFO \
   --hostfile my-hosts -n 8 -N 8 \
   ```

```
--mca pml ^cm --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to
 none \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

- `p4d.24xlarge` and `p5.48xlarge`

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_EFA_USE_DEVICE_RDMA=1 \
-x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/
lib:/opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
--hostfile my-hosts -n 8 -N 8 \
--mca pml ^cm --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to
 none \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

3. You can confirm that EFA is active as the underlying provider for NCCL when the `NCCL_DEBUG` log is printed.

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Selected Provider is efa*
```

The following additional information is displayed when using a `p4d.24xlarge` instance.

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Running on P4d platform, Setting
 NCCL_TOPO_FILE environment variable to /home/ec2-user/install/plugin/share/aws-
ofi-nccl/xml/p4d-24xl-topo.xml
```

Step 9: Install your machine learning applications

Install the machine learning applications on the temporary instance. The installation procedure varies depending on the specific machine learning application. For more information about installing software on your Linux instance, see Manage software on your Amazon Linux 2 instance.

Note

Refer to your machine learning application's documentation for installation instructions.

Step 10: Create an EFA and NCCL-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions**, **Image**, **Create image**.
4. For **Create image**, do the following:
   a. For **Image name**, enter a descriptive name for the AMI.
   b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.
   c. Choose **Create image**.
5. In the navigation pane, choose **AMIs**.
6. Locate the AMI tht you created in the list. Wait for the status to change from `pending` to `available` before continuing to the next step.

Step 11: Terminate the temporary instance

At this point, you no longer need the temporary instance that you launched. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions**, **Instance state**, **Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

Step 12: Launch EFA and NCCL-enabled instances into a cluster placement group

Launch your EFA and NCCL-enabled instances into a cluster placement group using the EFA-enabled AMI and the EFA-enabled security group that you created earlier.

Note

- It is not an absolute requirement to launch your EFA-enabled instances into a cluster placement group. However, we do recommend running your EFA-enabled instances in a cluster placement group as it launches the instances into a low-latency group in a single Availability Zone.
- To ensure that capacity is available as you scale your cluster's instances, you can create a Capacity Reservation for your cluster placement group. For more information, see Use Capacity Reservations with cluster placement groups.

New console

To launch a temporary instance

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** to open the new launch instance wizard.
3. (*Optional*) In the **Name and tags** section, provide a name for the instance, such as `EFA-instance`. The name is assigned to the instance as a resource tag (`Name=`*`EFA-instance`*).
4. In the **Application and OS Images** section, choose **My AMIs**, and then select the AMI that you created in the previous step.
5. In the **Instance type** section, select either `p3dn.24xlarge` or `p4d.24xlarge`.
6. In the **Key pair** section, select the key pair to use for the instance.
7. In the **Network settings** section, choose **Edit**, and then do the following:

    a. For **Subnet**, choose the subnet in which to launch the instance. If you do not select a subnet, you can't enable the instance for EFA.

    b. For **Firewall (security groups)**, choose **Select existing security group**, and then select the security group that you created in the previous step.

    c. Expand the **Advanced network configuration** section.

       For **Network interface 1**, select **Network card index = 0**, **Device index = 0**, and **Interface type = EFA with ENA**.

(Optional) If you are using a multi-card instance type, such as `p4d.24xlarge` or `p5.48xlarge`, for each additional network interface required, choose **Add network interface**, for **Network card index** select the next unused index, and then select **Device index = 1** and **Interface type = EFA eith ENA** or **EFA-only**.

8. (*Optional*) In the **Storage** section, configure the volumes as needed.
9. In the **Advanced details** section, for **Placement group name**, select the cluster placement group into which to launch the instance. If you need to create a new cluster placement group, choose **Create new placement group**.
10. In the **Summary** panel on the right, for **Number of instances**, enter the number of EFA-enabled instances that you want to launch, and then choose **Launch instance**.

Old console

To launch your EFA and NCCL-enabled instances into a cluster placement group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose **My AMIs**, find the AMI that you created earlier, and then choose **Select**.
4. On the **Choose an Instance Type** page, select **p3dn.24xlarge** and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:

   a. For **Number of instances**, enter the number of EFA and NCCL-enabled instances that you want to launch.

   b. For **Network** and **Subnet**, select the VPC and subnet into which to launch the instances.

   c. For **Placement group**, select **Add instance to placement group**.

   d. For **Placement group name**, select **Add to a new placement group**, and then enter a descriptive name for the placement group. Then for **Placement group strategy**, select **cluster**.

   e. For **EFA**, choose **Enable**.

   f. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**. You can optionally specify a primary IPv4 address and one or more secondary IPv4 addresses. If you are launching the instance into a subnet that has an associated IPv6 CIDR block, you can optionally specify a primary IPv6 address and one or more secondary IPv6 addresses.

g. Choose **Next: Add Storage**.

6. On the **Add Storage** page, specify the volumes to attach to the instances in addition to the volumes specified by the AMI (such as the root device volume). Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify tags for the instances, such as a user-friendly name, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**, and then select the security group that you created earlier.
9. Choose **Review and Launch**.
10. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instances.

Step 13: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

The key pair is created in the `$HOME/.ssh/` directory.

4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.
6. For each member node in the cluster, do the following:

   a. Connect to the instance.

   b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.

7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Maximize network bandwidth on Amazon EC2 instances with multiple network cards

Many instances types that support EFA also have multiple network cards. For more information, see Network cards. If you plan to use EFA with one of these instance types, we recommend the following basic configuration:

- For the primary network interface (network card index `0`, device index `0`), create an EFA (EFA with ENA) interface. You can't use an EFA-only network interface as the primary network interface.
- For each additional network interface, use the next unused network card index, device index 1, and either an EFA (EFA with ENA) or EFA-only network interface, depending on your usecase, such as ENA bandwidth requirements or IP address space. For example use cases, see EFA configuration for a P5 instances.

Note

P5 instances require network interfaces to be configured in a specific manner to enable maximum network bandwidth. For more information, see [EFA configuration for a P5 instances](#).

The following examples show how to launch an instance based on these recommendations.

Instance launch

To specify EFAs during instance launch using the launch instance wizard

1. In the **Network settings** section, choose **Edit**.
2. Expand **Advanced network configuration**.
3. For the primary network interface (**Network interface 1**), select **Network card index = 0**, **Device index = 0**, and **Interface type = EFA with ENA**.
4. For each additional network interface required, choose **Add network interface**. For **Network card index** select the next unused index, and then select **Device index = 1**, and **Interface type = EFA with ENA** or **EFA-only**.

To specify EFAs during instance launch using the run-instances command

For `--network-interfaces`, specify the required number of network interfaces. For the primary network interface, specify `NetworkCardIndex=0`, `DeviceIndex=0`, and `InterfaceType=efa`. For any additional network interfaces, for `NetworkCardIndex` specify the next unused index, `DeviceIndex=1`, and `InterfaceType=efa` or `efa-only`.

The following example command snippet shows a request with 32 EFA devices and one ENA device.

```
$ aws --region $REGION ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami-0abcdef1234567890 \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
 \
```

```
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
efa-only" \
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=13,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=17,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=18,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=19,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=21,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=22,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
```

```
"NetworkCardIndex=23,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=25,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=26,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=27,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=29,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=30,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only" \
"NetworkCardIndex=31,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
efa-only"
...
```

Launch templates

To add EFAs to a launch template using the Amazon EC2 console

1. In the **Network settings** section, expand **Advanced network configuration**.
2. To add the primary network interface (Network interface 1), choose **Add network interface** and then select **Network card index = 0**, **Device index = 0**, and **Interface type = EFA with ENA**.
3. To add additional network interfaces, choose **Add network interface**. For **Network card index**, select the next unused index, and then select **Device index = 1**, and **Interface type = EFA with ENA** or **EFA-only**.

To add EFAs to a launch template using the create-launch-template command

For `NetworkInterfaces`, specify the required number of network interfaces. For the primary network interface, specify `NetworkCardIndex=0`, `DeviceIndex=0`, and `InterfaceType=efa`. For any additional network interfaces, for `NetworkCardIndex` specify the next unused index, `DeviceIndex=1`, and `InterfaceType=efa` or `efa-only`.

The following snippet shows an example with 3 network interfaces out of the possible 32 network interfaces.

```
"NetworkInterfaces":[
{
  "NetworkCardIndex":0,
  "DeviceIndex":0,
  "InterfaceType": "efa",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 1,
  "DeviceIndex": 1,
  "InterfaceType": "efa|efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 2,
  "DeviceIndex": 1,
  "InterfaceType": "efa|efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 3,
  "DeviceIndex": 1,
  "InterfaceType": "efa|efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
}
...
```

EFA configuration for a P5 instances

P5 instances have a total network bandwidth capacity of 3,200 Gbps, of which up to 800 Gbps can be utilized for IP network traffic. Because EFA and IP network traffic share the same underlying resources, bandwidth used by one will reduce the bandwidth that is available to the other. This means that you can distribute the network bandwidth between EFA traffic and IP traffic in any combination, as long as the total bandwidth does not exceed 3,200 Gbps and IP bandwidth does not exceed 800 Gbps. For example, if you use 400 Gbps for IP bandwidth, you can achieve up to 2,800 Gbps of EFA bandwidth at the same time.

Use case 1: Save IP addresses and avoid potential Linux IP issues

This configuration provides up to 3200 Gbps of EFA networking bandwidth and up to 100 Gbps of IP networking bandwidth with one private IP address. This configuration also helps to avoid potential Linux IP issues, such as disallowed auto-assignment of public IP addresses and IP routing challenges (hostname to IP address mapping issues and source IP address mismatches), that can arise if an instance has multiple network interfaces.

- For the primary network interface (network card index 0, device index 0), use an EFA (EFA with ENA) network interface.
- For the remaining network interfaces (network card indexes 1-31, device index 1), use EFA-only network interfaces.

Use case 2: Maximum EFA and IP network bandwidth

This configuration provides up to 3200 Gbps of EFA networking bandwidth and up to 800 Gbps of IP networking bandwidth with 8 private IP address. You can't auto-assign public IP addresses with this configuration. However, you can attach an Elastic IP address to the primary network interface (network card index 0, device index 0) after launch for internet connectivity.

- For the primary network interface (network card index 0, device index 0), use an EFA (EFA with ENA) network interface.
- For the remaining interfaces, do the following:
  - Specify EFA-only network interfaces on network card indexes 1, 2, and 3, and use device index 1.
  - Specify one EFA (EFA with ENA) network interface and three EFA-only network interfaces **in each** of the following network card index subsets, and use device index 1 for all of them:

- [4,5,6,7]
- [8,9,10,11]
- [12,13,14,15]
- [16,17,18,19]
- [20,21,22,23]
- [24,25,26,27]
- [28,29,30,31]

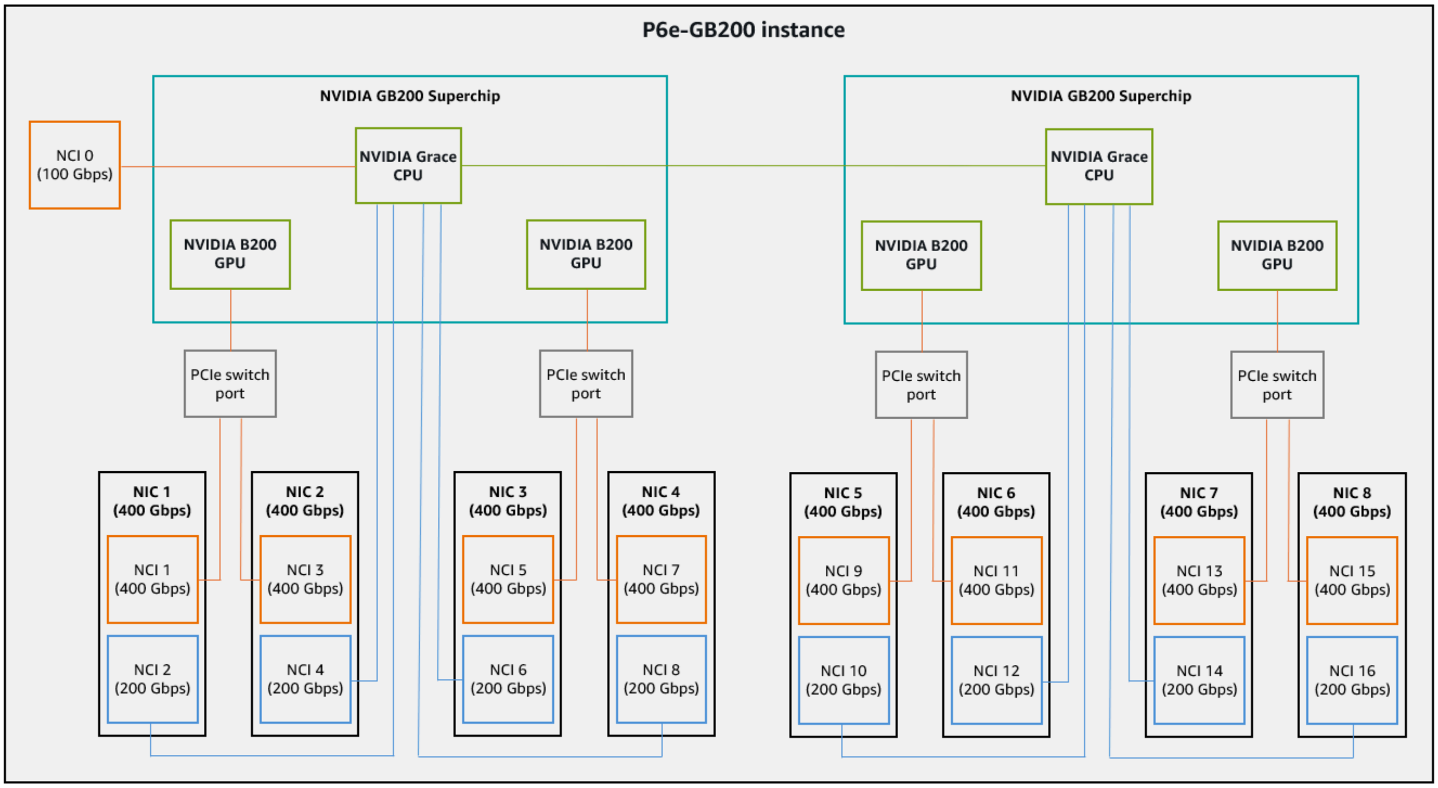
The following example illustrates this configuration:

```
$ aws --region $REGION ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami_id \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
 \
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
 \
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
```

```
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=13,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=17,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=18,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=19,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=21,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=22,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=23,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=25,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=26,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=27,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
 \
"NetworkCardIndex=29,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=30,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=31,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only"
...
```

EFA configuration for a P6e-GB200 instances

P6e-GB200 instances can be configured with up to 17 network cards. The following image shows the physical network interface card (NIC) layout for P6e-GB200 instances, along with the mapping of network card indexes (NCIs).



The primary NCI (index 0) supports up to 100 Gbps of ENA bandwidth. NCIs with the following indexes support EFA-only network interfaces and 400 Gbps EFA bandwidth: [1, 3, 5, 7, 9, 11, 13, 15]. NCIs with the following indexes support up to 200 Gbps ENA or EFA bandwidth: [2, 4, 6, 8, 10, 12, 14, 16].

The NCIs in the following groups share an underlying physical NIC on the host:

- [1 and 2]
- [3 and 4]
- [5 and 6]
- [7 and 8]
- [9 and 10]
- [11 and 12]
- [13 and 14]

- [15 and 16]

Each physical NIC supports up 400 Gbps of bandwidth. Because the NCIs in these groups share the same underlying physical NIC, bandwidth used by one will reduce the bandwidth that is available to the other. For example, if NCI 2 uses 200 Gbps of ENA bandwidth, NCI 1 can use up to 200 Gbps of EFA bandwidth at the same time.

Each underlying GPU on the host can send traffic directly over the following pairs of NCIs:

- [1 and 3]
- [5 and 7]
- [9 and 11]
- [13 and 15]

Each GPU supports up to 400 Gbps of EFA bandwidth. Because the network cards in these groups share the same GPU, bandwidth used by one will reduce the bandwidth that is available to the other. For example, if NCI 1 uses 200 Gbps of EFA bandwidth, NCI 3 can use up to 200 Gbps of EFA bandwidth at the same time. Therefore, to achieve maximum EFA performance, we recommend that you do **one of the following** to achieve a total of 1,600 Gbps EFA bandwidth:

- Add an EFA-only network interface to only one NCI in each group to achieve 400 Gbps per network interface (*4 EFA network interfaces x 400 Gbps*).
- Add an EFA-only network interface to each NCI in each group to achieve 200 Gbps per network interface (*8 EFA network interfaces x 200 Gbps*).

For example, the following configuration provides up to 1,600 Gbps of EFA bandwidth using a single EFA-only network interface in each NCI group, and up to 100 Gbps of ENA networking bandwidth using only the primary NCI (index 0).

- For the primary NCI (network card index 0, device index 0), use an ENA network interface.
- Add EFA-only network interfaces to the following:
  - NCI 1, device index 0
  - NCI 5, device index 0
  - NCI 9, device index 0
  - NCI 13, device index 0

Create and attach an Elastic Fabric Adapter to an Amazon EC2 instance

You can create an EFA and attach it to an Amazon EC2 instance much like any other elastic network interface in Amazon EC2. However, unlike elastic network interfaces, EFAs can't be attached to or detached from an instance in a `running` state.

Considerations

- You can change the security group that is associated with an EFA. To enable OS-bypass functionality, the EFA must be a member of a security group that allows all inbound and outbound traffic to and from the security group itself. For more information, see Step 1: Prepare an EFA-enabled security group.

  You change the security group that is associated with an EFA in the same way that you change the security group that is associated with an elastic network interface. For more information, see the section called "Modify network interface attributes".
- You assign an Elastic IP (IPv4) and IPv6 address to an EFA (EFA with ENA) network interface in the same way that you assign an IP address to an elastic network interface. For more information, see Managing IP addresses.

  You can't assign an IP address to an EFA-only network interface.

Tasks

- Create an EFA
- Attach an EFA to a stopped instance
- Attach an EFA when launching an instance
- Add an EFA to a launch template

Create an EFA

You can create an EFA in a subnet in a VPC. You can't move the EFA to another subnet after it's created, and you can only attach it to stopped instances in the same Availability Zone.

Console

To create an EFA (EFA with ENA or ENA-only) network interface

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

2. In the navigation pane, choose **Network Interfaces** and then choose **Create network interface**.
3. For **Description**, enter a descriptive name for the EFA.
4. For **Subnet**, select the subnet in which to create the EFA.
5. **Interface type**, choose one of the following options:
   - **EFA with ENA** — To create a network interface that supports both ENA and EFA devices.
   - **EFA-only** — To create a network interface with an EFA device only.
6. (For EFA with ENA only) Configure the IP address and prefix assignment for the network interface. The type of IP addresses and prefixes you can assign depend on the selected subnet. For IPv4-only subnets, you can assign IPv4 IP addresses and prefixes only. For IPv6-only subnets, you can assign IPv6 IP addresses and prefixes only. For dual-stack subnets, you can assign both IPv4 and IPv6 IP addresses and prefixes.

   > **Note**
   >
   > You can't assign IP addresses to an EFA-only network interface.

   a. For **Private IPv4 address** and/or **IPv6 address**, choose **Auto-assign** to have Amazon EC2 automatically assign an IP address from the selected subnet, or choose **Custom** to manually specify the IP address to assign.
   b. If you assign an IPv6 address, you can optionally enable **Assign primary IPv6 IP**. Doing this assigns a primary IPv6 global unicast address (GUA) to the network interface. Assigning a primary IPv6 address enables you to avoid disrupting traffic to instances or ENIs. For more information, see IPv6 addresses.
   c. For **IPv4 prefix delegation** and/or **IPv6 prefix delegation**, choose **Auto-assign** to have Amazon EC2 automatically assign a prefix from the subnet's CIDR block, or choose **Custom** to manually specify a prefix from the subnet's CIDR block. If you specify a prefix, AWS verifies that it is not already assigned to another resource. For more information, see Prefix delegation for Amazon EC2 network interfaces
   d. (Optional) Configure the **Idle connection tracking timeout** settings. For more information, see Idle connection tracking timeout

- **TCP established timeout** — The timeout period, in seconds, for idle TCP connections in an established state. Min: 60 seconds. Max: 432000 seconds (5 days). Default: 432000 seconds. Recommended: Less than 432000 seconds.
- **UDP timeout** — The timeout period, in seconds, for idle UDP flows that have seen traffic only in a single direction or a single request-response transaction. Min: 30 seconds. Max: 60 seconds. Default: 30 seconds.
- **UDP stream timeout** — The timeout period, in seconds, for idle UDP flows classified as streams that have seen more than one request-response transaction. Min: 60 seconds. Max: 180 seconds (3 minutes). Default: 180 seconds.

7. For **Security groups**, select one or more security groups.
8. Choose **Create network interface**.

AWS CLI

To create an EFA

Use the create-network-interface command. For `--interface-type`, specify `efa` for an EFA network interface or `efa-only` for an EFA-only network interface.

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --interface-type efa \
    --description "my efa"
```

PowerShell

To create an EFA

Use the New-EC2NetworkInterface cmdlet. For `-InterfaceType`, specify `efa` for an EFA network interface or `efa-only` for an EFA-only network interface

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -InterfaceType efa `
    -Description "my efa"
```

Attach an EFA to a stopped instance

You can attach an EFA to any supported instance that is in the `stopped` state. You cannot attach an EFA to an instance that is in the `running` state. For more information about the supported instance types, see Supported instance types.

You attach an EFA to an instance in the same way that you attach a network interface to an instance. For more information, see Attach a network interface.

Attach an EFA when launching an instance

AWS CLI

To attach an existing EFA when launching an instance

Use the run-instances command with the `--network-interfaces` option. For the primary network interface, specify an EFA network interface and `NetworkCardIndex=0`, `DeviceIndex=0`. To attach multiple EFA network interfaces, see Maximize network bandwidth.

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    NetworkInterfaceId=eni-1234567890abcdef0, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

To attach a new EFA when launching an instance

Use the run-instances command with the `--network-interfaces` option. For the primary network interface, use `NetworkCardIndex=0`, `DeviceIndex=0`, and `InterfaceType=efa`. If you are attaching multiple EFA network interfaces, see Maximize network bandwidth.

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    InterfaceType=efa, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

PowerShell

To attach an existing EFA when launching an instance

Use the New-EC2Instance cmdlet with the `-NetworkInterfaces` parameter.

```
-NetworkInterface $networkInterface
```

Define the network interface as follows.

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.NetworkInterfaceId = "eni-1234567890abcdef0"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

To attach a new EFA when launching an instance

Use the New-EC2Instance cmdlet with the `-NetworkInterfaces` parameter.

```
-NetworkInterface $networkInterface
```

Define the network interface as follows.

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.InterfaceType = "efa"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

Add an EFA to a launch template

You can create a launch template that contains the configuration information needed to launch EFA-enabled instances. You can specify both EFA and EFA-only network interfaces in the launch template. To create an EFA-enabled launch template, create a new launch template and specify a supported instance type, your EFA-enabled AMI, and an EFA-enabled security group. For `NetworkInterfaces`, specify the EFA network interfaces to attach. For the primary network interface, use `NetworkCardIndex=0`, `DeviceIndex=0`, and `InterfaceType=efa`. If you are attaching multiple EFA network interfaces, see Maximize network bandwidth on Amazon EC2 instances with multiple network cards.

You can leverage launch templates to launch EFA-enabled instances with other AWS services, such as AWS Batch or AWS ParallelCluster.

For more information about creating launch templates, see Create an Amazon EC2 launch template.

Detach and delete an EFA from an Amazon EC2 instance

You can detach an EFA from an Amazon EC2 instance and delete it in the same way as any other elastic network interface in Amazon EC2.

Detach an EFA

To detach an EFA from an instance, you must first stop the instance. You cannot detach an EFA from an instance that is in the running state.

You detach an EFA from an instance in the same way that you detach an elastic network interface from an instance. For more information, see Detach a network interface.

Delete an EFA

To delete an EFA, you must first detach it from the instance. You cannot delete an EFA while it is attached to an instance.

You delete EFAs in the same way that you delete elastic network interfaces. For more information, see Delete a network interface.

Monitor an Elastic Fabric Adapter on Amazon EC2

You can use the following features to monitor the performance of your Elastic Fabric Adapters.

Topics

EFA driver metrics for an Amazon EC2 instance

The Elastic Fabric Adapter (EFA) driver publishes multiple metrics from the instances that have EFA interfaces attached. You can use these metrics to troubleshoot application performance issues, choose the right cluster size for a workload, plan scaling activities proactively, and benchmark applications to determine whether they maximize the EFA performance available on an instance.

Topics

Available EFA driver metrics

The EFA driver publishes the following metrics to the instance in real time. They provide the cumulative number of errors and packets or bytes sent, received, or dropped by the attached EFA devices since instance launch or the last driver reset.

Metric	Description
`tx_bytes`	The number of bytes transmitted. Unit: bytes
`rx_bytes`	The number of bytes received. Unit: bytes
`tx_pkts`	The number of packets transmitted. Unit: count
`rx_pkts`	The number of packets received. Unit: count
`rx_drops`	The number of packets that were received and then dropped. Unit: count
`send_bytes`	The number of bytes sent using send operations. Unit: bytes

Metric	Description
recv_bytes	The number of bytes received by send operations. Unit: bytes
send_wrs	The number of packets sent using send operations. Unit: count
recv_wrs	The number of packets received by send operations. Unit: count
rdma_write_wrs	The number of completed rdma write operations. Unit: count
rdma_read_wrs	The number of completed rdma read operations. Unit: count
rdma_write_bytes	The number of bytes written to it by other instances using rdma write operations. Unit: bytes
rdma_read_bytes	The number of bytes received using rdma read operations. Unit: bytes
rdma_write_wr_err	The number of rdma write operations that had local or remote errors. Unit: count

Metric	Description
rdma_read_wr_err	The number of rdma read operations that had local or remote errors. Unit: count
rdma_read_resp_bytes	The number of bytes sent in response to rdma read operations. Unit: bytes
rdma_write_recv_bytes	The number of bytes received by rdma write operations. Unit: bytes

Retrieve EFA driver metrics for your instance

You can use the rdma-tool command line tool to retrieve the metrics for all EFA interfaces attached to an instance as follows:

```
$ rdma -p statistic show
link rdmap0s31/1
    tx_bytes 0
    tx_pkts 0
    rx_bytes 0
    rx_pkts 0
    rx_drops 0
    send_bytes 0
    send_wrs 0
    recv_bytes 0
    recv_wrs 0
    rdma_read_wrs 0
    rdma_read_bytes 0
    rdma_read_wr_err 0
    rdma_read_resp_bytes 0
    rdma_write_wrs 0
    rdma_write_bytes 0
    rdma_write_wr_err 0
```

Alternatively, you can retrieve the metrics for each EFA interface attached to an instance from the sys files using the following command.

```
$ more /sys/class/infiniband/device_number/ports/port_number/hw_counters/* | cat
```

For example

```
$ more /sys/class/infiniband/rdmap0s31/ports/1/hw_counters/* | cat
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/lifespan
::::::::::::::
12
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_resp_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wrs
::::::::::::::
```

```
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_drops
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_pkts
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_pkts
::::::::::::::
0
```

Amazon VPC flow logs

You can create an Amazon VPC Flow Log to capture information about the traffic going to and from an EFA. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you create a flow log, you can retrieve and view its data in the chosen destination. For more information, see VPC Flow Logs in the *Amazon VPC User Guide*.

You create a flow log for an EFA in the same way that you create a flow log for an elastic network interface. For more information, see Create a flow log in the *Amazon VPC User Guide*.

In the flow log entries, EFA traffic is identified by the `srcAddress` and `destAddress`, which are both formatted as MAC addresses, as shown in the following example.

```
version accountId  eniId        srcAddress        destAddress       sourcePort destPort
 protocol packets bytes start      end        action log-status
2       3794735123 eni-10000001 01:23:45:67:89:ab 05:23:45:67:89:ab -          -
 -        9       5689  1521232534 1524512343 ACCEPT OK
```

Amazon CloudWatch

If you are using EFA in an Amazon EKS cluster, you can monitor your EFAs using CloudWatch Container Insights. For more information, see Amazon EKS and Kubernetes Container Insights metrics in the *Amazon CloudWatch User Guide*.

Verify the EFA installer using a checksum

You can optionally verify the EFA tarball (`.tar.gz` file) using an MD5 or SHA256 checksum. We recommend that you do this to verify the identity of the software publisher and to check that the application has not been altered or corrupted since it was published.

To verify the tarball

Use the **md5sum** utility for the MD5 checksum, or the **sha256sum** utility for the SHA256 checksum, and specify the tarball filename. You must run the command from the directory in which you saved the tarball file.

- MD5

  ```
  $  md5sum tarball_filename.tar.gz
  ```

- SHA256

  ```
  $  sha256sum tarball_filename.tar.gz
  ```

The commands should return a checksum value in the following format.

```
checksum_value tarball_filename.tar.gz
```

Compare the checksum value returned by the command with the checksum value provided in the table below. If the checksums match, then it is safe to run the installation script. If the checksums do not match, do not run the installation script, and contact Support.

For example, the following command verifies the EFA 1.9.4 tarball using the SHA256 checksum.

```
$  sha256sum aws-efa-installer-1.9.4.tar.gz
```

Example output:

```
1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14 aws-efa-
installer-1.9.4.tar.gz
```

The following table lists the checksums for recent versions of EFA.

Version	Checksums
EFA 1.43.2	**MD5:** `7287b25a07c9747c0d4001e8fc5f59b2` **SHA256:** `de15c5bdbc83b952afbde876110830c604ad0796680e5157c05f7c1979a41069`
EFA 1.43.1	**MD5:** `7cfafc8debaea51dd4966fa0b2bba673` **SHA256:**54211eda0c193138ee8ed09b5fb41c41fc76fe0a77935fa4ec8d989466342740
EFA 1.43.0	**MD5:** `f2b3dd7dc8670b541f7c23fd58e5e503`

Version	Checksums
	SHA256: `786df3458c499237be 33bb8e50ffd4da7c18c20e25438 0ffc80fb90833d8cc73`
EFA 1.42.0	**MD5:** `94b2b1db09da1dde08ec049db1f 24370` **SHA256:** `4114fe612905ee0508 3ae5cb391a00a012510f3abfecc 642d86c9a5ae4be9008`
EFA 1.41.0	**MD5:** `086181c3ee3f8da512fc6e1c795 e8936` **SHA256:** `3506354cdfbe31ff55 2fe75f5d0d9bb7efd29cf79bd99 457347d29c751c38f9f`
EFA 1.40.0	**MD5:** `f3ec6f73fbeaccba08232750758 1157c` **SHA256:** `30491b0fe7c3470d44 39594538855c981b05fa69862d7 4f8c05eb9b97912368a`
EFA 1.39.0	**MD5:** `c223d5954a85a7fbcd248c942b8 66e43` **SHA256:** `2cbc028c03064633bb 990782b47c36156637769e2f487 04417a9c700a7a32101`

Version	Checksums
EFA 1.38.1	**MD5:** f112569e828ab65187777f794bab542c **SHA256:** 83923374afd388b1cfcf4b3a21a2b1ba7cf46a01a587f7b519b8386cb95e4f81
EFA 1.38.0	**MD5:** 43a2a446b33a2506f40853d55059f1ea **SHA256:** 4f436954f35ad53754b4d005fd8d0be63de3b4184de41a695b504bdce0fecb22
EFA 1.37.0	**MD5:** 6328070192bae920eca45797ad4c1db1 **SHA256:** 2584fc3c8bb99f29b3285e275747ff09d67c18e162c2a652e36c976b72154bfb
EFA 1.36.0	**MD5:** 1bec83180fbffb23452ab6469ca21dfa **SHA256:** de183f333cfb58aeb7908a67bf9106985ba3ccb7f8638b851d2a0d8dbfacaec4
EFA 1.35.0	**MD5:** 252f03c978dca5f8e8d9f34e488b256e **SHA256:** 432b6ad4368ba0cd8b902729d14a908a97be7a3dcc5239422ea994a47f35a5e1

Version	Checksums
EFA 1.34.0	**MD5:** `5cd4b28d27a31677c16139b54c9acb45` **SHA256:** `bd68839e741b0afd3ec2e37d50603803cfa7a279c120f0a736cc57c2ff2d7fdc`
EFA 1.33.0	**MD5:** `e2f61fccbcaa11e2ccfddd3660522276` **SHA256:** `0372877b87c6a7337bb7791d255e1053b907d030489fb2c3732ba70069185fce`
EFA 1.32.0	**MD5:** `db8d65cc028d8d08b5a9f2d88881c1b1` **SHA256:** `5f7233760be57f6fee6de8c09acbfbf59238de848e06048dc54d156ef578fc66`
EFA 1.31.0	**MD5:** `856352f12bef2ccbadcd75e35aa52aaf` **SHA256:** `943325bd37902a4300ac9e5715163537d56ecb4e7b87b37827c3e547aa1897bf`
EFA 1.30.0	**MD5:** `31f48e1a47fe93ede8ebd273fb747358` **SHA256:** `876ab9403e07a0c3c91a1a34685a52eced890ae052df94857f6081c5f6c78a0a`

Version	Checksums
EFA 1.29.1	**MD5:** e1872ca815d752c1d7c2b5c175e52a16 **SHA256:** 178b263b8c25845b63dc93b25bcdff5870df5204ec509af26f43e8d283488744
EFA 1.29.0	**MD5:** 39d06a002154d94cd982ed348133f385 **SHA256:** 836655f87015547e733e7d9f7c760e4e24697f8bbc261bb5f3560abd4206bc36
EFA 1.28.0	**MD5:** 9dc13b7446665822605e66febe074035 **SHA256:** 2e625d2d6d3e073b5178e8e861891273d896b66d03cb1a32244fd56789f1c435
EFA 1.27.0	**MD5:** 98bfb515ea3e8d93f554020f3837fa15 **SHA256:** 1d49a97b0bf8d964d91652a79ac851f2550e33a5bf9d0cf86ec9357ff6579aa3
EFA 1.26.1	**MD5:** 884e74671fdef4725501f7cd2d451d0c **SHA256:** c616994c924f54ebfabfab32b7fe8ac56947fae00a0ff453d975e298d174fc96

Version	Checksums
EFA 1.26.0	**MD5:** `f8839f12ff2e3b9ba09ae8a82b30e663` **SHA256:** `bc1abc1f76e97d204d3755d2a9ca307fc423e51c63141f798c2f15be3715aa11`
EFA 1.25.1	**MD5:** `6d876b894547847a45bb8854d4431f18` **SHA256:** `d2abc553d22b89a4ce92882052c1fa6de450d3a801fe005da718b7d4b9602b06`
EFA 1.25.0	**MD5:** `1993836ca749596051da04694ea0d00c` **SHA256:** `98b7b26ce031a2d6a93de2297cc71b03af647194866369ca53b60d82d45ad342`
EFA 1.24.1	**MD5:** `211b249f39d53086f3cb0c07665f4e6f` **SHA256:** `120cfeec233af0955623ac7133b674143329f9561a9a8193e473060f596aec62`
EFA 1.24.0	**MD5:** `7afe0187951e2dd2c9cc4b572e62f924` **SHA256:** `878623f819a0d9099d76ecd41cf4f569d4c3aac0c9bb7ba9536347c50b6bf88e`

Version	Checksums
EFA 1.23.1	**MD5:** 22491e114b6ee7160a8290145dca0c28 **SHA256:** 5ca848d8e0ff4d1571cd443c36f8d27c8cdf2a0c97e9068ebf000c303fc40797
EFA 1.23.0	**MD5:** 38a6d7c1861f5038dba4e441ca7683ca **SHA256:** 555d497a60f22e3857fdeb3dfc53aa86d05926023c68c916d15d2dc3df6525bd
EFA 1.22.1	**MD5:** 600c0ad7cdbc06e8e846cb763f92901b **SHA256:** f90f3d5f59c031b9a964466b5401e86fd0429272408f6c207c3f9048254e9665
EFA 1.22.0	**MD5:** 8f100c93dc8ab519c2aeb5dab89e98f8 **SHA256:** f329e7d54a86a03ea51da6ea9a5b68fb354fbae4a57a02f9592e21fce431dc3a
EFA 1.21.0	**MD5:** 959ccc3a4347461909ec02ed3ba7c372 **SHA256:** c64e6ca34ccfc3ebe8e82d08899ae8442b3ef552541cf5429c43d11a04333050

Version	Checksums
EFA 1.20.0	**MD5:** 7ebfbb8e85f1b94709df4ab3db47913b **SHA256:** aeefd2681ffd5c4c631d1502867db5b831621d6eb85b61fe3ec80df983d1dcf0
EFA 1.19.0	**MD5:** 2fd45324953347ec5518da7e3fefa0ec **SHA256:** 99b77821b9e72c8dea015cc92c96193e8db307deee05b91a58094cc331f16709
EFA 1.18.0	**MD5:** fc2571a72f5d3c7b7b576ce2de38d91e **SHA256:** acb18a0808aedb9a5e485f1469225b9ac97f21db9af78e4cd6939700debe1cb6
EFA 1.17.3	**MD5:** 0517df4a190356ab559235147174cafd **SHA256:** 5130998b0d2883bbae189b21ab215ecbc1b01ae0231659a9b4a17b0a33ebc6ca
EFA 1.17.2	**MD5:** a329dedab53c4832df218a24449f4c9a **SHA256:** bca1fdde8b32b00346e175e597ffab32a09a08ee9ab136875fb38283cc4cd099

Version	Checksums
EFA 1.17.1	**MD5:** `733ae2cfc9d14b52017eaf0a2ab6b0ff` **SHA256:** `f29322640a88ae9279805993cb836276ea240623820848463ca686c8ce02136f`
EFA 1.17.0	**MD5:** `d430fc841563c11c3805c5f82a4746b1` **SHA256:** `75ab0cee4fb6bd38889dce313183f5d3a83bd233e0a6ef6205d8352821ea901d`
EFA 1.16.0	**MD5:** `399548d3b0d2e812d74dd67937b696b4` **SHA256:** `cecec36495a1bc6fdc82f97761a541e4fb6c9a3cbf3cfcb145acf25ea5dbd45b`
EFA 1.15.2	**MD5:** `955fea580d5170b05823d51acde7ca21` **SHA256:** `84df4fbc1b3741b6c073176287789a601a589313accc8e6653434e8d4c20bd49`
EFA 1.15.1	**MD5:** `c4610267039f72bbe4e35d7bf53519bc` **SHA256:** `be871781a1b9a15fca342a9d169219260069942a8bda7a8ad06d4baeb5e2efd7`

Version	Checksums
EFA 1.15.0	**MD5:** 9861694e1cc00d884fadac07d22898be **SHA256:** b329862dd5729d2d098d0507fb486bf859d7c70ce18b61c302982234a3a5c88f
EFA 1.14.1	**MD5:** 50ba56397d359e57872fde1f74d4168a **SHA256:** c7b1b48e86fe4b3eaa4299d3600930919c4fe6d88cc6e2c7e4a408a3f16452c7
EFA 1.14.0	**MD5:** 40805e7fd842c36ececb9fd7f921b1ae **SHA256:** 662d62c12de85116df33780d40e0533ef7dad92709f4f613907475a7a1b60a97
EFA 1.13.0	**MD5:** c91d16556f4fd53becadbb345828221e **SHA256:** ad6705eb23a3fce44af3afc0f7643091595653a723ad0374084f4f2b715192e1
EFA 1.12.3	**MD5:** 818aee81f097918cfaebd724eddea678 **SHA256:** 2c225321824788b8ca3fbc118207b944cdb096b847e1e0d1d853ef2f0d727172

Version	Checksums
EFA 1.12.2	**MD5:** 956bb1fc5ae0d6f0f87d2e481d49fccf **SHA256:** 083a868a2c212a5a4fcf3e4d732b685ce39cceb3ca7e5d50d0b74e7788d06259
EFA 1.12.1	**MD5:** f5bfe52779df435188b0a2874d0633ea **SHA256:** 5665795c2b4f09d5f3f767506d4d4c429695b36d4a17e5758b27f033aee58900
EFA 1.12.0	**MD5:** d6c6b49fafb39b770297e1cc44fe68a6 **SHA256:** 28256c57e9ecc0b0778b41c1f777a9982b4e8eae782343dfe1246079933dca59
EFA 1.11.2	**MD5:** 2376cf18d1353a4551e35c33d269c404 **SHA256:** a25786f98a3628f7f54f7f74ee2b39bc6734ea9374720507d37d3e8bf8ee1371
EFA 1.11.1	**MD5:** 026b0d9a0a48780cc7406bd51997b1c0 **SHA256:** 6cb04baf5ffc58ddf319e956b5461289199c8dd805fe216f8f9ab8d102f6d02a

Version	Checksums
EFA 1.11.0	**MD5:** 7d9058e010ad65bf2e14259214a36949 **SHA256:** 7891f6d45ae33e822189511c4ea1d14c9d54d000f6696f97be54e915ce2c9dfa
EFA 1.10.1	**MD5:** 78521d3d668be22976f46c6fecc7b730 **SHA256:** 61564582de7320b21de319f532c3a677d26cc46785378eb3b95c636506b9bcb4
EFA 1.10.0	**MD5:** 46f73f5a7afe41b4bb918c81888fefa9 **SHA256:** 136612f96f2a085a7d98296da0afb6fa807b38142e2fc0c548fa986c41186282
EFA 1.9.5	**MD5:** 95edb8a209c18ba8d250409846eb6ef4 **SHA256:** a4343308d7ea4dc943ccc21bcebed913e8868e59bfb2ac93599c61a7c87d7d25
EFA 1.9.4	**MD5:** f26dd5c350422c1a985e35947fa5aa28 **SHA256:** 1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14

Version	Checksums
EFA 1.9.3	**MD5:** 95755765a097802d3e6d5018d1a 5d3d6 **SHA256:** 46ce732d6f3fcc9edf 6a6e9f9df0ad136054328e24675 567f7029edab90c68f1
EFA 1.8.4	**MD5:** 85d594c41e831afc6c930526314 0457e **SHA256:** 0d974655a09b213d78 59e658965e56dc4f23a0eee2dc4 4bb41b6d039cc5bab45

Elastic Fabric Adapter release notes

The following table describes the version history and changelog for the Elastic Fabric Adapter software.

Version	Changes	Release date
1.43.2	• Upgrade to libnccl-ofi 1.16.3	Aug 15 2025
1.43.1	• Upgrade to libfabric 2.1.0amzn5.0 • Upgrade to libnccl-ofi 1.16.2	July 31, 2025
1.43.0	• Deprecate support for Ubuntu 20.04 • Upgrade to libnccl-ofi 1.16.1 • Upgrade to rdma-core 58.amzn0 • Upgrade to libfabric 2.1.0amzn4.0 • Enable optimizations for Graviton platforms • Upgrade to efa driver 2.17.2 • Upgrade to efa-nv-peermem 1.2.2	July 25, 2025

Version	Changes	Release date
1.42.0	• Upgrade to efa driver 2.15.3 • Upgrade to efa-nv-peermem 1.2.1 • Upgrade to rdma-core 57.amzn1 • Upgrade to libfabric 2.1.0amzn3.0 • Upgrade to libnccl-ofi 1.15.0	June 6, 2025
1.41.0	• Upgrade to rdma-core 57.amzn0	May 16, 2025
1.40.0	• Upgrade to libfabric 2.1.0amzn2.0 • Upgrade to rdma-core 56.0 • Upgrade to efa-config 1.18. • Deprecate support for Debian 10 • Upgrade to libnccl-ofi 1.14.2 • Enable CUDA support in Libfabric on ARM for Ubuntu and Amazon Linux 2023 • Add libnccl-ofi on ARM for Ubuntu and Amazon Linux 2023	May 2, 2025
1.39.0	• Upgrade to libfabric 2.1.0 • Upgrade to Open MPI 5.0.6 • Upgrade to libnccl-ofi 1.14.1 • Upgrade to efa driver 2.15.0	April 16, 2025
1.38.1	• Update to `Libfabric 1.22.0amzn5.0`	March 3, 2025
1.38.0	• Add `libnccl-ofi 1.13.2-1`	January 8, 2025
1.37.0	• Upgrade to `libfabric 1.22.0amzn4.0` • Upgrade to `rdma-core 54.amzn0`	November 18, 2024

Version	Changes	Release date
1.36.0	• Add support for Debian 12 • Distribute `efa_test.sh` as a utility script at `/opt/amazon/efa/bin` • Upgrade to efa driver 2.13.0 • Upgrade to `libfabric 1.22.0amzn3.0` • Upgrade to `rdma-core 54.0` • Upgrade to open mpi 4.1.7	November 7, 2024
1.35.0	• Upgrade to Open MPI 5.0.5 • Upgrade to PRRTE 3.0.6 and rename the RPM package to prrte-aws • Rename the OpenPMIx RPM package to pmix-aws • Configure build options for dpkg and RPM builds • Upgrade to `Libfabric 1.22.0amzn2.0` • Upgrade to efa driver 2.12.1	October 14, 2024
1.34.0	• Drop support for CentOS 7 and RHEL 7 • Ingest `Libfabric 1.22.0amzn1.0` • Update efa-config package to 1.17	August 6, 2024
1.33.0	• Upgrade to efa driver 2.10.0 • Upgrade to `rdma-core 52.0` • Upgrade to PRRTE 3.0.5 • Upgrade to Open MPI 5.0.3 • Add support for Ubuntu 24.04 LTS • Remove OpenPMIx, PRRTE and Open MPI HTML documentation. Users should refer to the official website. • Drop support for openSUSE 15.4 or older	June 20, 2024

Version	Changes	Release date
1.32.0	• Upgrade to efa driver 2.8.0 • Upgrade to efa-nv-peermem driver 1.1.1 • Upgrade efa-config package to 1.16 • Upgrade efa-profile package to 1.7 • Upgrade to `rdma-core 50.0` • Upgrade to `libfabric 1.21.0amzn1.0` • Enhance `efa_test.sh` with `fi_pingpong` port selection • Install newer rdma-core from system when required	April 18, 2024
1.31.0	• Upgrade to OpenPMIx 4.2.8 • Upgrade to PRRTE 3.0.3 • Upgrade to Open MPI 5.0.2, and convert MCA components to DSO. • Upgrade to `Libfabric 1.20.1amzn1.0`	March 7, 2024
1.30.0	• Include OpenPMIx 4.2.7, installed at `/opt/amazon/pmix` by default • Include PRRTE 3.0.2, installed at `/opt/amazon/prrte` by default • Include Open MPI 5.0.0, installed at `/opt/amazon/openmpi5` by default • Update efa-profile to 1.6	December 2023
1.29.1	• Ingest `libfabric 1.19.0amzn4.0`	December 2023
1.29.0	• Ingest efa kernel driver 2.6.0 • Ingest `libfabric 1.19.0amzn3.0`	November 2023
1.28.0	• Upgrade Open MPI to 4.1.6 • Ingest `libfabric 1.19.0amzn2.0`	October 2023

Version	Changes	Release date
1.27.0	• Add `Libfabric 1.19.0amzn1.0` to installer • Enable GCC builtin atomics for Open MPI 4	September 2023
1.26.1	• `efa_test.sh` : fix infinite retry bug	September 2023
1.26.0	• Add support for RHEL 9 • Fix debug build on multiple distributions • Ingest `libfabric 1.18.2amzn1.0` • `efa_test.sh` : Add retry for `fi_pingpong`	September 2023
1.25.1	• Ingest `libfabric 1.18.1amzn1.0` • Update timeouts in EFA installer scripts	September 2023
1.25.0	• Add support for Amazon Linux 2023 • Bugfix in post-installation test	July 2023
1.24.1	• Upgrade libfabric to 1.18.1 • Upgrade efa driver to 2.5.0	July 2023
1.24.0	• Ingest `rdma-core 46.0` • Ingest efa driver 2.4.1 • Support Debian 11	June 2023
1.23.1	• Ingest `libfabric 1.18.0amzn2.0`	June 2023
1.23.0	• Add support for Debian 10 • Drop support for Ubuntu 18.04 LTS • Upgrade efa-config package to 1.14 • Ingest `libfabric 1.18.0amzn1.0`	May 2023
1.22.1	• Upgrade libfabric to 1.17.1	March 2023

Version	Changes	Release date
1.22.0	• Upgrade Open MPI to 4.1.5 • Upgrade libfabric to 1.17.0 • Upgrade efa-config package to 1.13	February 2023
1.21.0	• Add support for Rocky Linux 9 OS • Ingest efa driver 2.1.1 • Ingest `libfabric 1.16.1amzn3.0` • Upgrade efa-config package to 1.12	December 2022
1.20.0	• Add support for Rocky Linux 8 OS. • Ingest efa driver 2.1.0. • Ingest `rdma-core 43.0`. • Ingest `libfabric 1.16.1amzn1.0`	November 2022
1.19.0	• Ingest `libfabric 1.16.0` • Build Open MPI with `--enable-orterun-prefix-by-default`	October 2022
1.18.0	• Add support for Ubuntu22.04	August 2022
1.17.3	• Update libfabric to `1.16.0~amzn4.0` . The ~ indicates it is a pre-release version of libfabric 1.16.0. • Extend post-installation pingpong test timeout to 20 seconds	August 2022
1.17.2	• Update libfabric to `1.16.0~amzn3.0` . The ~ indicates it is a pre-release version of libfabric 1.16.0.	July 2022
1.17.1	• Update libfabric to `1.16.0~amzn2.0` . The ~ indicates it is a pre-release version of libfabric 1.16.0. • Disable the experimental net provider when building libfabric	July 2022

Version	Changes	Release date
1.17.0	• Update `rdma-core` to v41.0 • Update Open MPI to 4.1.4 • Update libfabric to `1.16.0~amzn1.0` . The ~ indicates it is a pre-release version of libfabric 1.16.0.	July 2022
1.16.0	• Update libfabric to `1.15.1amzn1.0` , contains neuron library name change • Upgrade efa-config to 1.10 • Exclude opx and rxd providers in the libfabric build	June 2022
1.15.2	• Update libfabric to 1.14.1	May 2022
1.15.1	• Update libfabric to `1.14.0amzn1.0`	March 2022
1.15.0	• Fix a bug that cause installation fail on Open SuSE 15.3 • Drop support of Open SuSE 15.2 (as Open SuSE 15.2 reached end of life) • Drop Support of CentOS 8 (as CentOS 8 reached end of life) • Update libfabric to 1.14.0 • Update efa kernel driver to 1.16.0 • Update `rdma-core` to v39.0 • Update Open MPI to version 4.1.2.	Feburary 2022
1.14.1	• Update libfabric to `1.13.2amzn1.0` .	October 2021

Version	Changes	Release date
1.14.0	• Ingest efa kernel driver 1.14.2. • Make `-g, --enable-gdr` in `efa_installer.sh` as a no-op option as the latest efa kernel driver enables GDR support by default. • Ingest `rdma-core v37.0`. • Ingest `libfabric 1.13.2`. • Add packages list and compare RPM/DEB to list during installation to prevent unknown package installations. • Add sleep in installer script to wait for udev rule to apply after EFA driver reload.	October 2021
1.13.0	• Update `rdma-core` to v35.0. • Update `libfabric` to v1.13.0amzn1.0. • Add EFA support for CentOS/RHEL 8 on Gravition2 platform. • Add version comparison logic in installer script to skip the local package installation when there is one installed in system with higher version.	August 2021
1.12.3	• Update EFA kernel module to 1.13.0. • Update efa-config package to version 1.9. Improve the calculation of huge page reservation to handle large defaulted huge page size.	July 2021
1.12.2	• Update EFA kernel module to 1.12.3. • Build Open MPI debian packages with `--with-libevent=external` and `--with-hwloc=external` . • Bump Open MPI rpm build ID to 2 to fix backward compatibility issue of HWLOC on CentOS 8. • Remove the installation of kernel-devel and kernel-source packages on SLES15SP2 and openSUSE 15.2.	June 2021
1.12.1	• Update Libfabric to version 1.11.2amzon1.1. • Update EFA kernel module to version 1.12.1.	May 2021

Version	Changes	Release date
1.12.0	• Update Open MPI to version 4.1.1. • Update Libfabric to version 1.11.2amzn1.0. • Build rdma-core for Amazon Linux 2 using the same packaging configuration as the AL2 rdma-core. • Do not force `-Wl,--enable-new-dtags` when building Open MPI RPMs. • Build Open MPI with system libraries for hwloc and libevent. • Update EFA kernel module to version 1.12.0 • Update efa-config package to version 1.8. Improve the calculation of huge page reservation for long-lived instances. • Update efa-profile package to version 1.5. Remove the open mpi collective tuning file that worked as a workaround to fix Open MPI 4.1.0 hang on P4d. • Update `rdma-core` to v32.1. • Drop support for Amazon Linux 1 and Ubuntu 16.04.	May 2021
1.11.2	• Fix Open MPI hang when using Open MPI on P4d by changing the default algorithm used to implement MPI_BARRIER via a configuration file. • Disable use of builtin atomics in Open MPI on ARM via `--disable-builtin-atomics` to work around compiler issue.	February 2021
1.11.1	• Update Open MPI to version 4.1.0. • Update efa-config package to version 1.7. Improve calculati on of huge page reservation count. • Update efa-profile package to version 1.3. Removes unneeded collectives decision file now that Open MPI 4.1.0 is used.	December 2020

Version	Changes	Release date
1.11.0	• Add support for Gravition2 platform. • Update `rdma-core` to version 31.2amzn. • Update Libfabric to version 1.11.1amzn1.0. • Update efa-config to version 1.6. • Update efa-profile to version 1.2.	December 2020
1.10.1	• Add support for CentOS / RHEL 8. • Add support for Ubuntu 20.04. • Add support for SUSE Linux Enterprise 15.	November 2020
1.10.0	• Add GPUDirect RDMA support for P4d platform. Use `--enable-gdr` installer option to instal GDR-aware kernel module and userspace. • Update EFA kernel module to version 1.10.2. • Update `rdma-core` to version 31.amzn0. • Update Libfabric to version 1.11.1. • Update Open MPI to version 4.0.5. • Update efa-config to version 1.5. • Update efa-profile to version 1.1. Includes improved Open MPI collectives decision file.	October 2020
1.9.5	• Update efa-config to version 1.4. Fixes bug in Open MPI collective decision file.	September 2020
1.9.4	• Update Open MPI to version 4.0.3. • Update Libfabric to version 1.10.1amazon1.1. • Update `rdma-core` to version 28.amzn0.	July 2020

Version	Changes	Release date
1.9.3	• Update EFA kernel module to version 1.6.0. • Update `rdma-core` to version 28.amzn0. • Update Libfabric to version 1.10.1amzn1.1. • Update efa-config to version 1.3. Adds collectives tuning file for Open MPI. • Skip dkms installation if it is already installed. • Fix `--skip-kmod` installation mode to actually work.	June 2020
1.8.4	• Move configuration files into efa-config and efa-profile packages so that they are tracked by the operating system package manager. • Update Open MPI to version 4.0.3.	April 2020
1.8.3	• Update EFA kernel module to version 1.5.1. • Distributed DKMS on some platforms rather than relying on EPEL repositories for added installation reliability. • On RHEL 7, install RPMs built on CentOS 7 instead of RPMs built on Amazon Linux 2.	February 2020
1.8.2	• Revert `rdma-core` to version 25 due to a mismatch in device naming between kernel module and `rdma-core` .	January 2020
1.8.1	• Update Libfabric to version 1.9.0amzn1.1.	January 2020
1.8.0	• Update `rdma-core` to version 27.0. • Update EFA kernel module to version 1.5.0. • Update Libfabric to version 1.9.0amzn1.0. • Add option `--minimal` to install just the EFA kernel module and `rdma-core` .	December 2019
1.7.1	• Update Libfabric to version 1.8.1amzn1.3.	December 2019

Version	Changes	Release date
1.7.0	• Add Libfabric module file. • Update Libfabric to version 1.8.1amzn1.1.	November 2019
1.6.2	• Update Open MPI to version 1.6.2.	October 2019
1.6.1	• Update Libfabric to version 1.8.1amzn1.0. • Update Open MPI to version 4.0.1. • Update `rdma-core` to version 26.0.	October 2019
1.5.4	• Update EFA kernel module to version 1.4.1.	September 2019
1.5.3	• Update EFA kernel module to version 1.3.1. • Avoid installing `kernel-devel` or `linux-headers` packages unless installing the kernel driver.	September 2019
1.5.1	• Only configure huge pages when EFA device is present.	August 2019
1.5.0	• Update Libfabric to version 1.8.0amzn1.1. • Update `rmda-core` to version 25.0.	August 2019
1.4.1	• Add Libfabric and Open MPI library paths (`/opt/amazon/efa/lib64` and `/opt/amazon/efa/openmpi/lib`) to `/etc/ld.so.conf.d/efa.conf` to ensure the Open MPI and Libfabric shared libraries are properly located.	July 2019
1.4.0	• Update EFA kernel module to version 1.3.0. • Update Libfabric to version 1.8.0amzn1.0 • First release to support Intel MPI 2019 Update 4.	July 2019

Amazon EC2 instance topology

Describing your instance topology provides a hierarchical view of the relative proximity between your Amazon EC2 instances. You can use this information to manage high performance computing

(HPC) and machine learning (ML) compute infrastructure at scale, while optimizing job placement. HPC and ML jobs are sensitive to latency and throughput. You can use the instance topology to detect the location of your instances, and then use this information to optimize HPC and ML jobs by running them on instances that are physically closer to each other.

You can use instance topology to detect the location of your existing instances, but you can't use it to choose to launch a new instance physically close to an existing instance. To influence instance placement, you can create Capacity Reservations in cluster placement groups.

Considerations

- Instance topology views are only available for instances in the `running` state.
- Each instance topology view is unique per account.
- The AWS Management Console does not support viewing the instance topology.

Pricing

There is no additional cost to describe your instance topology.

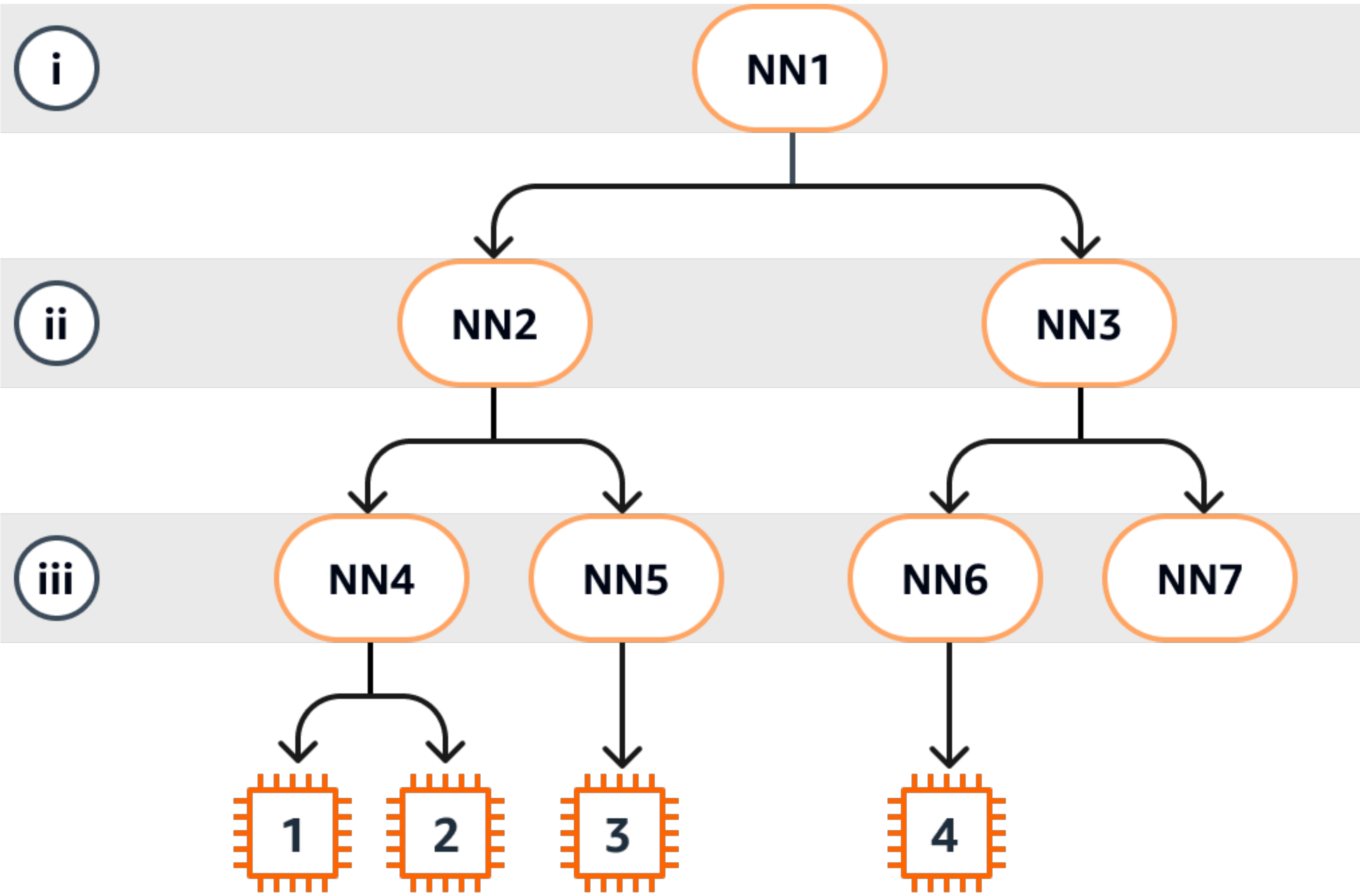
Contents

How Amazon EC2 instance topology works

The AWS network is arranged in a hierarchy of layers. EC2 instances connect into the network at or below the third layer, depending on the instance type. An instance's topology is described by a set of nodes, with one node in each layer of the network. The node set in the DescribeInstanceTopology API response provides a top-down view of the network hierarchy, with the bottom node connected to the instance.

> **Note**
>
> Some instance types comprise 4 network nodes in a node set representing 4 layers in the network, while others comprise 3 network nodes representing 3 layers in the network. For the supported instance types, see Instance types.

The following diagram provides a visual representation that you can use to understand the instance topology. The network nodes are identified as **NN1 – NN7**. The numerals **i**, **ii**, and **iii** identify the network layers. The numbers **1**, **2**, **3**, and **4** identify the EC2 instances. Instances connect to a node in the bottom layer, identified by **iii** in the following diagram. More than one instance can connect to the same node.



In this example:

- Instance 1 connects to network node 4 (NN4) in layer iii. NN4 connects to network node 2 (NN2) in layer ii, and NN2 connects to network node 1 (NN1) in layer i, which is the top of the network hierarchy in this example. The network node set comprises NN1, NN2, and NN4, expressed hierarchically from the upper layers to the bottom layer.
- Instance 2 also connects to network node 4 (NN4). Instance 1 and instance 2 share the same network node set: NN1, NN2, and NN4.
- Instance 3 connects to network node 5 (NN5). NN5 connects to NN2, and NN2 connects to NN1. The network node set for instance 3 is NN1, NN2, and NN5.
- Instance 4 connects to network node 6 (NN6). Its network node set is NN1, NN3, and NN6.

When considering the proximity of instances 1, 2, and 3, instances 1 and 2 are closer to each other because they connect to the same network node (NN4), while instance 3 is further away because it connects to a different network node (NN5).

When considering the proximity of all the instances in this diagram, instances 1, 2, and 3 are closer to each other than they are to instance 4 because they share NN2 in their network node set.

As a general rule, if the network node connected to any two instances is the same, these instances are physically close to each other, as is the case with instances 1 and 2. Furthermore, the fewer the number of hops between network nodes, the closer the instances are to each other. For example, instances 1 and 3 have fewer hops to a common network node (NN2) than they have to the network node (NN1) they have in common with instance 4, and are therefore closer to each other than they are to instance 4.

There are no instances running under network node 7 (NN7) in this example, and therefore the API output won't include NN7.

How to interpret the output

You get the instance topology information using the DescribeInstanceTopology API. The output provides a hierarchical view of the underlying network topology for an instance.

The following example output corresponds to the network topology information of the four instances in the preceding diagram. Comments are included in the example output for the purposes of this example.

The following information in the output is important to note:

- `NetworkNodes` describes the network node set of an instance.
- In each network node set, the network nodes are listed in hierarchical order from top to bottom.
- The network node that is connected to the instance is the last network node in the list (the bottom layer).
- To work out which instances are close to each other, first find common network nodes in the bottom layer. If there are no common network nodes in the bottom layer, then find common network nodes in the upper layers.

In the following example output, `i-1111111111example` and `i-2222222222example` are located closest to each other compared to the other instances in this example because they have the network node `nn-4444444444example` in common in the bottom layer.

> **Note**
>
> The response contains 3 or more network nodes. For information about the number of network nodes in the response for each supported instance type, see [Instance types](#).

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",  //Corresponds to instance 1
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 in layer i
                "nn-2222222222example",           //Corresponds to NN2 in layer ii
                "nn-4444444444example"            //Corresponds to NN4 in layer iii -
 bottom layer, connected to the instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",  //Corresponds to instance 2
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-4444444444example"            //Corresponds to NN4 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",  //Corresponds to instance 3
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
```

```
                "nn-5555555555example"            //Corresponds to NN5 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-444444444example",  //Corresponds to instance 4
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",          //Corresponds to NN1 - layer i
                "nn-3333333333example",          //Corresponds to NN3 - layer ii
                "nn-6666666666example"           //Corresponds to NN6 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

Prerequisites for Amazon EC2 instance topology

Before you describe the instance topology for your instances, ensure that your instances meet the following requirements.

Requirements to describe the topology of your instances

- AWS Regions
- Instance types
- Instance state
- IAM permission

AWS Regions

Supported AWS Regions:

- US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon)

- Asia Pacific (Melbourne), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt), Europe (Ireland), Europe (London), Europe (Paris), Europe (Spain), Europe (Stockholm)
- Israel (Tel Aviv)
- South America (São Paulo)
- AWS GovCloud (US-West)

Instance types

Supported instance types:

- Returns 3 network nodes in the response
  - `hpc6a.48xlarge` | `hpc6id.32xlarge` | `hpc7a.12xlarge` | `hpc7a.24xlarge` | `hpc7a.48xlarge` | `hpc7a.96xlarge` | `hpc7g.4xlarge` | `hpc7g.8xlarge` | `hpc7g.16xlarge`
  - `p3dn.24xlarge` | `p4d.24xlarge` | `p4de.24xlarge` | `p5.48xlarge` | `p5e.48xlarge` | `p5en.48xlarge` | `p6e-gb200.36xlarge`
  - `trn1.2xlarge` | `trn1.32xlarge` | `trn1n.32xlarge` | `trn2.48xlarge` | `trn2u.48xlarge`
- Returns 4 network nodes in the response
  - `p6-b200.48xlarge`

The available instance types vary by Region. For more information, see [Amazon EC2 instance types by Region](#).

Instance state

Instances must be in the `running` state. You can't get instance topology information for instances that are in another state.

IAM permission

Your IAM identity (user, user group, or role) requires the following IAM permission:

- `ec2:DescribeInstanceTopology`

Examples for Amazon EC2 instance topology

You can use the describe-instance-topology command to describe the instance topology for your EC2 instances.

When you use the `describe-instance-topology` command without parameters or filters, the response includes all your instances that match the supported instance types for this command in the specified Region. You can specify the Region by including the `--region` parameter, or by setting a default Region. For more information about setting a default Region, see Select a Region for your Amazon EC2 resources.

You can include parameters to return instances that match specified instance IDs or placement group names. You can also include filters to return instances that match a specified instance type or instance family, or instances in a specified Availability Zone or Local Zone. You can include a single parameter or filter, or a combination of parameters and filters.

The output is paginated, with up to 20 instances per page by default. You can specify up to 100 instances per page using the `--max-results` parameter.

For more information, see describe-instance-topology.

Required permissions

The following permission is required to describe instance topology:

- `ec2:DescribeInstanceTopology`

Examples

- Example 1 – Instance IDs
- Example 2 – Placement group name parameter
- Example 3 – Instance type filter
  - Example 3a – Exact match filter for a specified instance type
  - Example 3b – Wild card filter for an instance family
  - Example 3c – Combined instance family and exact match filters
- Example 4 – Zone ID filter
  - Example 4a – Availability Zone filter
  - Example 4b – Local Zone filter

Example 1 – Instance IDs

AWS CLI

To describe the topology of specific instances

Use the describe-instance-topology command with the `--instance-ids` parameter. The output includes only the instances that match the specified instance IDs.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --instance-ids i-1111111111example i-2222222222example
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
```

```
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of specific instances

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -InstanceId i-1111111111example, i-2222222222example
```

Example 2 – Placement group name parameter

AWS CLI

To describe the topology of instances in a specific placement group

Use the describe-instance-topology command with the `group-names` parameter. The output includes only the instances that are in either of the specified placement groups.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --group-names ML-group HPC-group
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
```

```
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances in a specific placement group

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -GroupName ML-group, HPC-group
```

Example 3 – Instance type filter

You can filter by a specified instance type (exact match) or filter by an instance family (using a wildcard). You can also combine a specified instance type filter and instance family filter.

Example 3a – Exact match filter for a specified instance type

AWS CLI

To describe the topology of instances with a specific instance type

Use the describe-instance-topology command with the `instance-type` filter. The output includes only the instances with the specified instance type.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1n.32xlarge
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances with a specific instance type

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1n.32xlarge"}
```

Example 3b – Wild card filter for an instance family

AWS CLI

To describe the topology of instances with a specific instance family

Use the describe-instance-topology command with the `instance-type` filter. The output includes only the instances with the specified instance family.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1*
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
                "nn-1235301234example"
            ],
```

```
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances with a specific instance family

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1*"}
```

Example 3c – Combined instance family and exact match filters

AWS CLI

To describe the topology of instances with an instance family or instance type

Use the describe-instance-topology command with the `instance-type` filter. The output includes only the instances that meet the specified criteria.

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge"
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
```

```
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-4343434343example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances with an instance family or instance type

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"}
```

Example 4 – Zone ID filter

You can use the `zone-id` filter to filter by an Availability Zone or Local Zone. You can also combine an Availability Zone filter and Local Zone filter.

Example 4a – Availability Zone filter

AWS CLI

To describe the topology of instances in a specific Availability Zone

Use the describe-instance-topology command with the zone-id filter. The output includes only the instances in the specified Availability Zone.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances in a specific Availability Zone

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1"}
```

Example 4b – Local Zone filter

AWS CLI

To describe the topology of instances in a specific Local Zone

Use the describe-instance-topology command with the zone-id filter. The output includes only the instances in the specified Local Zone.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-atl2-az1
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances in a specific Local Zone

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-atl2-az1"}
```

Example 4c – Combined Availability Zone and Local Zone filters

AWS CLI

To describe the topology of instances in a specific zone

Use the describe-instance-topology command with the `zone-id` filter. The output includes only the instances that are in either of the specified zones.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1,use1-atl2-az1
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances in a specific zone

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

Example 5 – Instance type and zone ID filters

You can combine filters in a single command.

AWS CLI

To describe the topology of instances with specific instance types, instance families, and zones

Use the describe-instance-topology command with the `instance-type` and `zone-id` filters. The response contains any instances with either of the specified instance types and are in either of the specified zones.

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge" \
              "Name=zone-id,Values=use1-az1,use1-atl2-az1"
```

The following is example output.

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
```

```
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

To describe the topology of instances with specific instance types, instance families, and zones

Use the Get-EC2InstanceTopology cmdlet.

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"} `
            @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

Placement groups for your Amazon EC2 instances

To meet the needs of your workload, you can launch a group of *interdependent* EC2 instances into a *placement group* to influence their placement.

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 high-performance computing (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.

Placement groups are optional. If you don't launch your instances into a placement group, EC2 tries to place the instances in such a way that all of your instances are spread out across the underlying hardware to minimize correlated failures.

Pricing

There is no charge for creating a placement group.

Rules and limitations

Before you use placement groups, be aware of the following rules:

- An instance can be placed in one placement group at a time; you can't place an instance in multiple placement groups.
- You can't merge placement groups.
- On-Demand Capacity Reservations and zonal Reserved Instances allow you to reserve capacity for EC2 instances in Availability Zones. When you launch an instance, if the instance attributes match those specified by an On-Demand Capacity Reservation or a zonal Reserved Instance, then the reserved capacity is automatically used by the instance. This is also true if you launch the instance into a placement group.
- You can't launch Dedicated Hosts in placement groups.
- You can't launch a Spot Instance that is configured to stop or hibernate on interruption in a placement group.

Contents

Placement strategies for your placement groups

You can create a placement group for your EC2 instances using one of the following placement strategies.

Placement strategies

- Cluster placement groups
- Partition placement groups
- Spread placement groups

Cluster placement groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. Instances are not isolated to a single rack. A cluster placement group can span peered virtual private networks (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.

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.

Rules and limitations

The following rules apply to cluster placement groups:

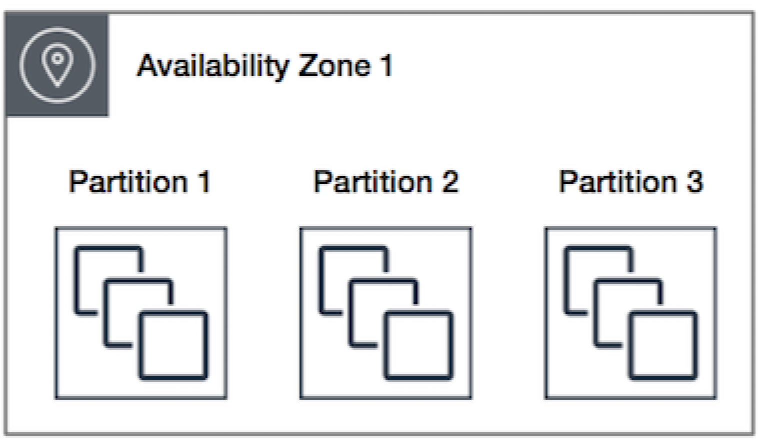
- The following instance types are supported:
  - Current generation instances, except for burstable performance instances (for example, T2), Mac1 instances, and M7i-flex instances.
  - The following previous generation instances: A1, C3, C4, I2, M4, R3, and R4.
- A cluster placement group can't span 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.

- We recommend that you reserve capacity explicitly in the cluster placement group by creating an On-Demand Capacity Reservation in the cluster placement group. Note that you can't reserve capacity using zonal Reserved Instances, as they can't reserve capacity explicitly in a placement group.
- Network traffic to the internet and over an AWS Direct Connect connection to on-premises resources is limited to 5 Gbps for cluster placement groups.

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.

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.
- Capacity Reservations do not reserve capacity in a partition placement group.

Spread placement groups

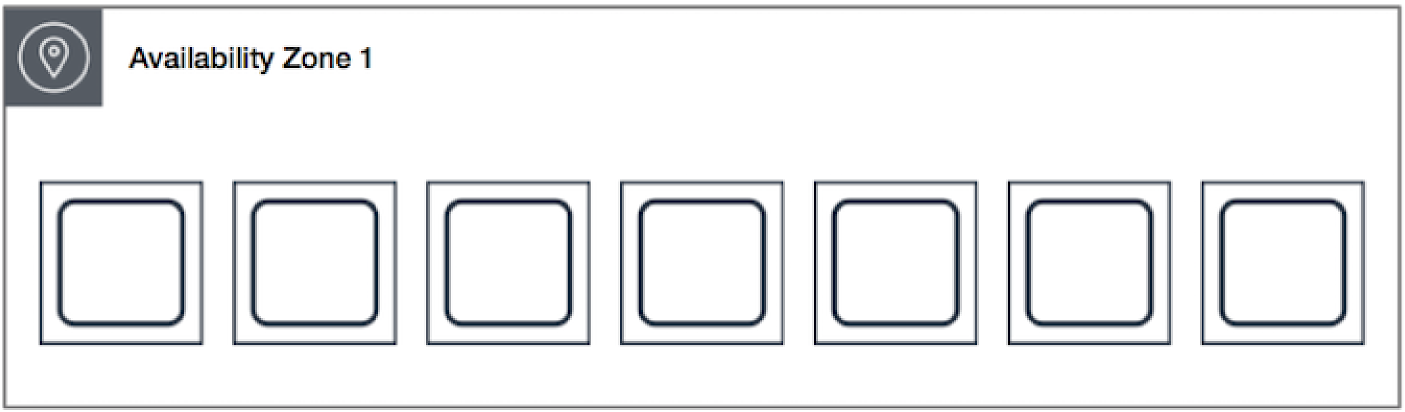
A spread placement group is a group of instances that are each placed on distinct hardware.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread level placement group reduces the risk of simultaneous failures that might occur when instances share the same equipment. Spread level placement groups provide access to distinct hardware, and are therefore suitable for mixing instance types or launching instances over time.

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 groups can spread instances across racks or hosts. Rack level spread placement groups can be used in AWS Regions and on AWS Outposts. Host level spread placement groups can be used with AWS Outposts only.

Rack level spread placement groups

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, each rack has its own network and power source.



A rack level spread placement group can span multiple Availability Zones in the same Region. In a Region, a rack level spread placement group can have a maximum of seven running instances per Availability Zone per group. With Outposts, a rack level spread placement group can hold as many instances as you have racks in your Outpost deployment.

Host level spread placement groups

Host level spread placement groups are only available with AWS Outposts. A host spread level placement group can hold as many instances as you have hosts in your Outpost deployment. For more information, see the section called “Placement groups on AWS Outposts”.

Rules and limitations

The following rules apply to spread placement groups:

- A rack 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, with seven instances in each Availability 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 more than seven instances in an Availability Zone, we recommend that you use multiple spread placement groups. Using multiple spread placement groups does not provide guarantees about the spread of instances between groups, but it does help ensure the spread for each group, thus limiting the impact from certain classes of failures.

- Spread placement groups are not supported for Dedicated Instances.
- Host level spread placement groups are only supported for placement groups on AWS Outposts. A host level spread placement group can hold as many instances as you have hosts in your Outpost deployment.
- In a Region, a rack level spread placement group can have a maximum of seven running instances per Availability Zone per group. With AWS Outposts, a rack level spread placement group can hold as many instances as you have racks in your Outpost deployment.
- Capacity Reservations do not reserve capacity in a spread placement group.

Create a placement group for your EC2 instances

You can use a placement group to control the placement of instances relative to each other. After you create a placement group, you can launch instances in the placement group.

Limitation

You can create a maximum of 500 placement groups per Region.

Console

To create a placement group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Placement Groups**.
3. Choose **Create placement group**.
4. Specify a name for the group.
5. Choose the placement strategy for the group: **Cluster**, **Spread**, or **Partition**.

   If you chose **Spread**, you must choose the spread level: **Rack** or **Host**.

   If you chose **Partition**, you must enter the number of partitions for the group.
6. (Optional) To add a tag, choose **Add new tag**, and then enter a key and value.
7. Choose **Create group**.

AWS CLI

Use the [create-placement-group](#) command.

To create a cluster placement group

The following example creates a placement group 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

The following example creates a placement group that uses the `partition` placement strategy, and specifies the five partitions using the `--partition-count` parameter.

```
aws ec2 create-placement-group \
    --group-name HDFS-Group-A \
    --strategy partition \
    --partition-count 5
```

PowerShell

Use the [New-EC2PlacementGroup](#) cmdlet.

To create a cluster placement group

The following example creates a cluster placement group.

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy cluster
```

To create a partition placement group

The following example creates a partition placement group.

```
New-EC2PlacementGroup `
```

```
    -GroupName my-placement-group `
    -Strategy partition `
    -PartitionCount 5
```

Change the placement for an EC2 instance

You can change the placement group for an instance as follows:

- Add an instance to a placement group
- Move an instance from one placement group to another
- Remove an instance from a placement group

Requirement

Before you can change the placement group for an instance, the instance must be in the `stopped` state.

Console

To change the instance placement

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**, **Modify instance placement**.
5. For **Placement group**, do one of the following:
   - To add the instance to a placement group, choose the placement group.
   - To move the instance from one placement group to another, choose the placement group.
   - To remove the instance from the placement group, choose **None**.
6. Choose **Save**.

AWS CLI

To move an instance to a placement group

Use the following modify-instance-placement command.

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name MySpreadGroup
```

To remove an instance from a placement group

Use the following modify-instance-placement command. When you specify an empty string for the placement group name, this removes the instance from its current placement group.

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

PowerShell

To move an instance to a placement group

Use the Edit-EC2InstancePlacement cmdlet with the name of the placement group.

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName MySpreadGroup
```

To remove an instance from a placement group

Use the Edit-EC2InstancePlacement cmdlet with an empty string for the name of the placement group.

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName ""
```

Delete a placement group

If you need to replace a placement group or no longer need one, you can delete it. Before you can delete a placement group, it must contain no instances. You can terminate the instances, move them to another placement group, or remove them from the placement group.

Console

To delete a 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 the placement group and choose **Actions**, **Delete**.
4. When prompted for confirmation, enter **`Delete`** and then choose **Delete**.

AWS CLI

To delete a placement group

Use the delete-placement-group command.

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

To delete a placement group

Use the Remove-EC2PlacementGroup cmdlet.

```
Remove-EC2PlacementGroup -GroupName my-cluster
```

Shared placement groups

Placement group sharing allows you to influence the placement of interdependent instances that are owned by separate AWS accounts. An owner can share a placement group across multiple AWS accounts or within their organization. A participant can launch instances in a placement group that is shared with their account.

A placement group owner can share a placement group with:

- Specific AWS accounts inside or outside of its organization
- An organizational unit inside its organization
- Its entire organization

You can use VPC peering to connect instances owned by separate AWS accounts and get the full latency benefits offered by shared cluster placement groups.

Contents

Rules and limitations

The following rules and limitations apply when you share a placement group or when a placement group is shared with you.

- To share a placement group, you must own it in your AWS account. You can't share a placement group that has been shared with you.
- When you share a partition or spread placement group, the placement group limits do not change. A shared partition placement group supports a maximum of seven partitions per Availability Zone, and a shared spread placement group supports a maximum of seven running instances per Availability Zone.
- To share a placement group with your organization or an organizational unit in your organization, you must enable sharing with AWS Organizations. For more information, see Sharing your AWS resources.
- When using the AWS Management Console to launch an instance, you can select any placement groups that were shared with you. When using the AWS CLI to launch an instance, you must specify a shared placement group by ID, not by name. You can use the name of a placement group only if you're the owner of the shared placement group.
- You are responsible for managing the instances owned by you in a shared placement group.
- You can't view or modify instances and capacity reservations that are associated with a shared placement group but not owned by you.
- The Amazon Resource Name (ARN) of a placement group contains the ID of the account that owns the placement group. You can use the account ID portion of a placement group ARN to identify the owner of a placement group that is shared with you.

Required permissions

To share a placement group, users must have permissions for following actions:

- `ec2:PutResourcePolicy`
- `ec2:DeleteResourcePolicy`

Sharing across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To specify the location of your Dedicated Hosts 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 Availability Zone ID for the `us-east-1` Region and it is the same location in every AWS account. For more information, see AZ IDs.

Placement group sharing

To share a placement group, 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.

If you are part of an organization in AWS Organizations sharing within your organization is enabled, consumers in your organization are granted access to the shared placement group.

If the placement group is shared with an AWS account outside of your organization, the AWS account owner will receive an invitation to join the resource share. They can access the shared placement group after accepting the invitation.

You can share a placement group across AWS accounts using AWS Resource Access Manager. For more information, see Creating a resource share in the *AWS RAM User Guide*.

Placement group unsharing

The placement group owner can unshare a shared placement group at any time. When you unshare a shared placement group, the following changes occur:

- The AWS accounts with which a placement group was shared are no longer able to launch instances or reserve capacity.
- Any instances running in a shared placement group are disassociated from the placement group, but they continue to run in your AWS account.
- Any capacity reservations in a shared placement group are disassociated from the placement group, but remain available to you in your AWS account.

For more information, see Deleting a resource share in the *AWS RAM User Guide*.

Placement groups 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 placement groups on Outposts that you have created in your account. This allows you to spread out instances across underlying hardware on an Outpost at your site. You create and use placement groups on Outposts in the same way that you create and use placement groups in regular Availability Zones. When you create a placement group with a spread strategy on an Outpost, you can choose to have the placement group spread instances across hosts or racks. Spreading instances across hosts allows you to use a spread strategy with a single rack Outpost.

Considerations

- A rack level spread placement group can hold as many instances as you have racks in your Outpost deployment.
- A host level spread placement group can hold as many instances as you have hosts in your Outpost deployment.

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*.

To use a placement group 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 placement group in the associated Region of the Outpost. If you create a placement group with a spread strategy, you can choose host or rack level spread to determine how the group will spread instances across the underlying hardware on your Outpost. For more information, see the section called “Create a placement group”.
3. Launch an instance into the placement group. For **Subnet** choose the subnet that you created in Step 1, and for **Placement group name**, select the placement group that you created in Step 2. For more information, see Launch an instance on the Outpost in the *AWS Outposts User Guide*.

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 frames consist of the packet, 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 EC2 instance types support 1500 MTU.

Contents

Jumbo frames (9001 MTU)

With jumbo frames, you can increase the payload size per packet, thereby increasing the percentage of the packet that is not packet overhead. With jumbo frames, fewer packets are needed to send the same amount of usable data. However, certain types of traffic are subject to the following maximum payloads:

MTU limit 1500 bytes

- Traffic over an internet gateway
- Traffic over VPN connections
- Traffic between AWS Regions, unless a transit gateway is used

MTU limit 8500 bytes

- Traffic over an inter-region VPC peering connection

If packets are over their MTU limit, they are fragmented, or they are dropped if the `Don't Fragment` flag is set in the IP header.

Jumbo frames should be used with caution for internet-bound traffic or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside a VPC and not slow traffic that's bound for outside the VPC, you can configure the MTU size by route, or use multiple elastic network interfaces with different MTU sizes and different routes.

For instances that are collocated inside a cluster placement group, jumbo frames help to achieve the maximum network throughput possible, and they are recommended in this case. For more information, see Placement groups for your Amazon EC2 instances.

You can use jumbo frames for traffic between your VPCs and your on-premises networks over AWS Direct Connect. For more information, and for how to verify Jumbo Frame capability, see MTU for private virtual interfaces or transit virtual interfaces in the *AWS Direct Connect User Guide*.

All current generation instance types support jumbo frames. The following previous generation instance types support jumbo frames: A1, C3, I2, M3, and R3.

Related resources

- For NAT gateways, see NAT gateway basics in the *Amazon VPC User Guide*.
- For transit gateways, see Maximum transmission unit in the *Amazon VPC Transit Gateways User Guide*.
- For Local Zones, see Considerations in the *AWS Local Zones User Guide*.
- For AWS Wavelength, see Maximum transmission unit in the *AWS Wavelength User Guide*.
- For Outposts see Service link maximum transmission unit requirements in the *AWS Outposts User Guide*.

Path MTU Discovery

Path MTU Discovery (PMTUD) 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. When there is a difference in the MTU size in the network between two hosts, PMTUD enables the receiving host to respond to the originating host with an ICMP message. This ICMP message instructs the originating host to use the lowest MTU size along the network path and to resend the request. Without this negotiation, packet drop can occur because the request is too large for the receiving host to accept.

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.

Connections made through some components, like NAT gateways and load balancers, are automatically tracked. This means that security group tracking is automatically enabled for your outbound connection attempts. If connections are automatically tracked or if your security group rules allow inbound ICMP traffic, you can receive PMTUD responses.

Note that ICMP traffic can be blocked even if the traffic is allowed at the security group level, such as if you have a network access control list entry that denies ICMP traffic to the subnet.

> **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.

For MTU rules over NAT gateways, see Maximum transmission unit (MTU) in the *Amazon VPC User Guide*. For MTU rules over Transit gateways, see Maximum transmission unit (MTU) in the *AWS Transit Gateway User Guide*.

Set the MTU for your Amazon EC2 instances

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. All Amazon EC2 instances support standard frames (1500 MTU) and all current generation instance types support jumbo frames (9001 MTU).

You can view the MTU for your Amazon EC2 instances, view the path MTU between your instance and another host, and configure your instances to use either standard or jumbo frames.

Tasks

- Check the path MTU between two hosts
- Check the MTU for your instance
- Set the MTU for your instance

Check the path MTU between two hosts

You can check the path MTU between your EC2 instance and another host. You can specify a DNS name or an IP address as the destination. If the destination is another EC2 instance, verify that its security group allows inbound UDP traffic.

The procedure that you use depends on the operating system of the instance.

Linux instances

Run the **tracepath** command on your instance to check the path MTU between your EC2 instance and the specified destination. This command is part of the `iputils` package, which is available by default in many Linux distributions.

This example checks the path MTU between the EC2 instance and `amazon.com`.

```
[ec2-user ~]$ tracepath amazon.com
```

In this example output, the path MTU is 1500.

```
 1?: [LOCALHOST]     pmtu 9001
 1:  ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)   0.187ms pmtu 1500
 1:  no reply
 2:  no reply
 3:  no reply
 4:  100.64.16.241 (100.64.16.241)                              0.574ms
 5:  72.21.222.221 (72.21.222.221)                             84.447ms asymm 21
 6:  205.251.229.97 (205.251.229.97)                           79.970ms asymm 19
 7:  72.21.222.194 (72.21.222.194)                             96.546ms asymm 16
 8:  72.21.222.239 (72.21.222.239)                             79.244ms asymm 15
 9:  205.251.225.73 (205.251.225.73)                           91.867ms asymm 16
...
31:  no reply
     Too many hops: pmtu 1500
     Resume: pmtu 1500
```

Windows instances

To check path MTU using mturoute

1. Download **mturoute.exe** to your EC2 instance from https://elifulkerson.com/projects/mturoute.php.
2. Open a Command Prompt window and change to the directory where you downloaded **mturoute.exe**.
3. Use the following command to check the path MTU between your EC2 instance and the specified destination. This example checks the path MTU between the EC2 instance and `www.elifulkerson.com`.

```
.\mturoute.exe www.elifulkerson.com
```

In this example output, the path MTU is 1500.

```
* 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.
```

Check the MTU for your instance

You can check the MTU value for your instance. Some instances are configured to use jumbo frames, and others are configured to use standard frame sizes.

The procedure that you use depends on the operating system of the instance.

Linux instances

To check the MTU setting on a Linux instance

Run the following **ip** command on your EC2 instance. If the primary network interface is not `eth0`, replace `eth0` with your network interface.

```
[ec2-user ~]$ ip link show eth0
```

In this example output, *`mtu 9001`* indicates that the instance uses jumbo frames.

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode
 DEFAULT group default qlen 1000
    link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

Windows instances

The procedure that you use depends on the driver on your instance.

ENA driver

Version 2.1.0 and later

To get the MTU value, use the following **Get-NetAdapterAdvancedProperty** command on your EC2 instance. Use the wildcard (asterisk) to get all Ethernet names. Check the output for the interface name `*JumboPacket`. A value of 9015 indicates that Jumbo frames are enabled. Jumbo frames are disabled by default.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet*"
```

Version 1.5 and earlier

To get the MTU value, use the following **Get-NetAdapterAdvancedProperty** command on your EC2 instance. Check the output 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"
```

Intel SRIOV 82599 driver

To get the MTU value, use the following **Get-NetAdapterAdvancedProperty** command on your EC2 instance. 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"
```

AWS PV driver

To get the MTU value, use the following command on your EC2 instance. 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
```

Set the MTU for your instance

You might want to use jumbo frames for network traffic within your VPC and standard frames for internet traffic. Whatever your use case, we recommend that you verify that your instance behaves as expected.

The procedure that you use depends on the operating system of the instance.

Linux instances

To set the MTU value on a Linux instance

1. Run the following **ip** command on your instance. It sets the desired MTU value to 1500, but you could use 9001 instead. If the primary network interface is not `eth0`, replace `eth0` with the actual network interface.

   ```
   [ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
   ```

2. (Optional) To persist your network MTU setting after a reboot, modify the following configuration files, based on your operating system type.

   - **Amazon Linux 2023** – Modify the `[Link]` section of the config file. The default config file is `/usr/lib/systemd/network/80-ec2.network`, or you can update any custom config file created in /run/systemd/network/, where the file name is *`priority`*-*`interface`*.network. For more information, see [Networking service](#) in the Amazon Linux documentation.

     ```
     MTUBytes=1500
     ```

   - **Amazon Linux 2** – Add the following line to the `/etc/sysconfig/network-scripts/ifcfg-eth0` file:

     ```
     MTU=1500
     ```

     Add the following line to the `/etc/dhcp/dhclient.conf` file:

     ```
     request subnet-mask, broadcast-address, time-offset, routers, domain-name,
      domain-search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-
     servers;
     ```

   - **Other Linux distributions** – Consult their specific documentation.

3. (Optional) Reboot your instance and verify that the MTU setting is correct.

Windows instances

The procedure that you use depends on the driver on your instance.

ENA driver

You can change the MTU using Device Manager or the **Set-NetAdapterAdvancedProperty** command on your instance.

Version 2.1.0 and later

Use the following command to enable jumbo frames.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9015
```

Use the following command to disable jumbo frames.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

Version 1.5 and earlier

Use the following command to enable jumbo frames.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 9001
```

Use the following command to disable jumbo frames.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 1500
```

Intel SRIOV 82599 driver

You can change the MTU using Device Manager or the **Set-NetAdapterAdvancedProperty** command on your instance.

Use the following command to enable jumbo frames.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9014
```

Use the following command to disable jumbo frames.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

AWS PV driver

You can change the MTU using the **netsh** command on your instance. You can't change the MTU using Device Manager.

Use the following command to enable jumbo frames.

```
netsh interface ipv4 set subinterface "Ethernet" mtu=9001
```

Use the following command to disable jumbo frames.

```
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 and sometimes fail to reach the cluster in the *Amazon Redshift Management Guide*.

Virtual private clouds for your EC2 instances

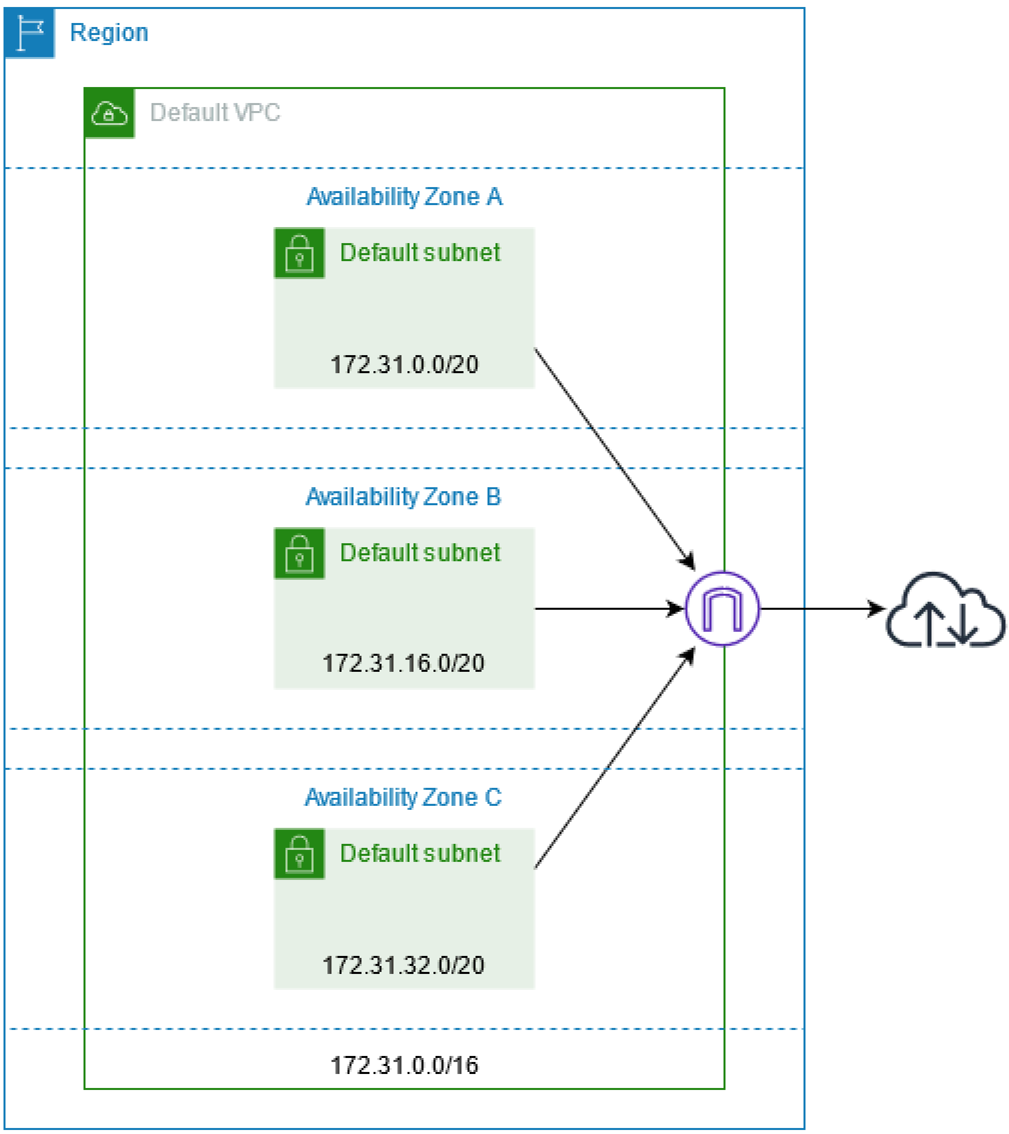
Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud* or *VPC*. You can create AWS resources, such as Amazon EC2 instances, into the subnets of your VPC. Your VPC closely resembles 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.

Contents

- Nondefault VPCs
- Internet access
- Shared subnets
- IPv6-only subnets

Your default VPCs

When you create your AWS account, we create a *default VPC* in each Region. A default VPC is a VPC that is already configured and ready for you to use. For example, there is a default subnet for each Availability Zone in each default VPC, an internet gateway attached to the VPC, and there's a route in the main route table that sends all traffic (0.0.0.0/0) to the internet gateway. You can modify the configuration of your default VPCs as needed. For example, you can add subnets and route tables.



Nondefault VPCs

Instead of using a default VPC for your resources, you can create your own VPC, as described in Create a VPC in the *Amazon VPC User Guide*.

Here are some things to consider when creating a VPC for your EC2 instances.

- You can use the default suggestion for the IPv4 CIDR block or enter the CIDR block required by your application or network.
- To ensure high availability, create subnets in multiple Availability Zones.
- If your instances must be accessible from the internet, do one of the following:
  - If your instances can be in a public subnet, add public subnets. Keep both DNS options enabled. You can optionally add private subnets now or later on.
  - If your instances must be in a private subnet, add only private subnets. You can add a NAT gateway to provide internet access to instances in the private subnets. If your instances send or receive a significant volume of traffic across Availability Zones, create a NAT gateway in each Availability Zone. Otherwise, you can create a NAT gateway in just one of the Availability Zones and launch instances that send or receive cross-zone traffic in the same Availability Zone as the NAT gateway.

Internet access

Instances launched into a default subnet in a default VPC have access to the internet, as default VPCs are configured to assign public IP addresses and DNS hostnames, and the main route table is configured with a route to an internet gateway attached to the VPC.

For instances that you launch in nondefault subnets and VPCs, you can use one of the following options to ensure that the instances that you launch in these subnets have access to the internet:

- Configure an internet gateway. For more information, see Connect to the internet using an internet gateway in the *Amazon VPC User Guide*.
- Configure a public NAT gateway. For more information, see Access the internet from a private subnet in the *Amazon VPC User Guide*.

Shared subnets

When launching EC2 instances into shared VPC subnets, note the following:

- Participants can run instances in a shared subnet by specifying the ID of the shared subnet. Participants must own any network interfaces that they specify.
- Participants can start, stop, terminate, and describe instances that they've created in a shared subnet. Participants can't start, stop, terminate, or describe instances that the VPC owner created in the shared subnet.
- VPC owners can't start, stop, terminate, or describe instances created by participants in a shared subnet.
- Participants can connect to an instance in a shared subnet using EC2 Instance Connect Endpoint. The participant must create the EC2 Instance Connect Endpoint in the shared subnet. Participants can't use an EC2 Instance Connect Endpoint that the VPC owner created in the shared subnet.

For information about shared Amazon EC2 resources, see the following:

- the section called "Manage AMI sharing with an organization or OU"
- the section called "Shared Capacity Reservations"
- the section called "Shared placement groups"
- Cross-account Amazon EC2 Dedicated Host sharing

For more information about shared subnets, see Share your VPC with other accounts in the *Amazon VPC User Guide*.

IPv6-only subnets

An EC2 instance launched in an IPv6-only subnet receives an IPv6 address but not an IPv4 address. Any instances that you launch into an IPv6-only subnet must be Nitro-based instances.

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.
  - 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 for Amazon EC2 instances.
  - 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.

This documentation helps you understand how to apply the shared responsibility model when using Amazon EC2. It shows you how to configure Amazon EC2 to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon EC2 resources.

Contents

- Update management for Amazon EC2 instances
- Security best practices for Windows instances
- Amazon EC2 key pairs and Amazon EC2 instances
- Amazon EC2 security groups for your EC2 instances
- NitroTPM for Amazon EC2 instances
- Credential Guard for Windows instances
- Access Amazon EC2 using an interface VPC endpoint

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. You are also responsible for 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 users with AWS IAM Identity Center or 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 require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- 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 sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 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) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text 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 text 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.

Contents

Amazon EBS data security

Amazon EBS volumes are presented to you as raw, unformatted block devices. These devices are logical devices that are created on the EBS infrastructure and the Amazon EBS service ensures that the devices are logically empty (that is, the raw blocks are zeroed or they contain cryptographically pseudorandom data) prior to any use or re-use by a customer.

If you have procedures that require that all data be erased using a specific method, either after or before use (or both), such as those detailed in **DoD 5220.22-M** (National Industrial Security Program Operating Manual) or **NIST 800-88** (Guidelines for Media Sanitization), you have the ability to do so on Amazon EBS. That block-level activity will be reflected down to the underlying storage media within the Amazon EBS service.

Encryption at rest

EBS volumes

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 in the *Amazon EBS User Guide*.

[Windows instances] You can also use Microsoft EFS and NTFS permissions for folder- and file-level encryption.

Instance store volumes

The data on NVMe instance store volumes is encrypted using an XTS-AES-256 cipher, implemented on a hardware module on the instance. The keys used to encrypt data that's written to locally-attached NVMe storage devices are per-customer, and per volume. The keys are generated by, and only reside within, the hardware module, which is inaccessible to AWS personnel. The 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.

When you stop, hibernate, or terminate an instance, every block of storage in the instance store volume is reset. Therefore, your data cannot be accessed through the instance store of another instance.

Memory

Memory encryption is enabled on the following instances:

- Instances with AWS Graviton2 or later AWS Graviton processors support always-on memory encryption. The encryption keys are securely generated within the host system, do not leave the host system, and are destroyed when the host is rebooted or powered down. For more information, see AWS Graviton Processors.
- Instances with 3rd generation Intel Xeon Scalable processors (Ice Lake), such as M6i instances, and 4th generation Intel Xeon Scalable processors (Sapphire Rapids), such as M7i instances. These processors support always-on memory encryption using Intel Total Memory Encryption (TME).
- Instances with 3rd generation AMD EPYC processors (Milan), such as M6a instances, and 4th generation AMD EPYC processors (Genoa), such as M7a instances. These processors support always-on memory encryption using AMD Secure Memory Encryption (SME).
- AMD Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP) is supported for some AMD-based instance types. For more information, see Find EC2 instance types that support AMD SEV-SNP.

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 peering and Transit Gateway cross-Region peering

All cross-Region traffic that uses Amazon VPC peering 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 traffic before it leaves AWS secured facilities, 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. This encryption uses Authenticated Encryption with Associated Data (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, M6a, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd, M8i, M8i-flex
  - **Compute optimized:** C5n, C6a, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd, C8gn
  - **Memory optimized:** R5dn, R5n, R6a, R6i, R6idn, R6in, R6id, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd, R8i, R8i-flex, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, U7inh-32tb, X2idn, X2iedn, X2iezn, X8g
  - **Storage optimized:** D3, D3en, I3en, I4g, I4i, I7i, I7ie, I8g, I8ge, Im4gn, Is4gen
  - **Accelerated computing:** DL1, DL2q, F2, G4ad, G4dn, G5, G6, G6e, G6f, Gr6, Gr6f, Inf1, Inf2, P3dn, P4d, P4de, P5, P5e, P5en, P6-B200, P6e-GB200, Trn1, Trn1n, Trn2, Trn2u, VT1
  - **High-performance computing:** Hpc6a, Hpc6id, Hpc7a, Hpc7g
- 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 | sort
```

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

The SSH and RDP protocols provides secure communications channels for remote access to your 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.

(Windows 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.

Infrastructure security in Amazon EC2

As a managed service, Amazon Elastic Compute Cloud is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access Amazon EC2 through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by 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.

For more information, see Infrastructure Protection in the *Security Pillar – AWS Well-Architected Framework*.

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 using private IP addresses, use AWS PrivateLink. For more information, see Access Amazon EC2 using an interface VPC endpoint.

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. Configure rules that allow the minimum required network traffic. For example, you can allow traffic only from the address ranges for your corporate network or only for specific protocols, such as HTTPS. For Windows instances, allow 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.
- [Windows instances] 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.
- Use private subnets for your instances if they should not be accessed directly from the internet. Use a bastion host or NAT gateway for internet access from an instance in a private subnet.
- [Windows instances] 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.

- [Windows instances] 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.
- [Windows instances] System 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 network interfaces 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. Using multiple network interfaces also provides additional options for controlling network traffic, including the ability to create host-based routing policies or leverage different VPC subnet routing rules based on the assigned subnet of the network interface.
- 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 GuardDuty Malware Protection to identify suspicious behavior indicative of malicious software on your instances that might compromise your workload, repurpose resources for malicious use, and gain unauthorized access to your data.
- Use GuardDuty Runtime Monitoring to identify and respond to potential threats to your instances. For more information, see How Runtime Monitoring works with Amazon EC2 instances.

- Use AWS Security Hub, Reachability Analyzer, or Network Access Analyzer to check for unintended network accessibility from your instances.
- Use EC2 Instance Connect to connect to your instances using Secure Shell (SSH) without the need to share and manage SSH keys.
- Use AWS Systems Manager Session Manager to access your instances remotely instead of opening inbound SSH or RDP ports and managing key pairs.
- Use AWS Systems Manager Run Command to automate common administrative tasks instead of connecting to your instances.
- [Windows instances] 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.

Amazon VPC supports additional network security controls, such as gateways, proxy servers, and network monitoring options. For more information, see Control network traffic in the *Amazon VPC User Guide*.

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

Compliance validation for Amazon EC2

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. 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 Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.

- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- 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.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Identity and access management for Amazon EC2

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon EC2 resources. IAM is an AWS service that you can use with no additional charge.

Your security credentials identify you to services in AWS and grant you access to AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and 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 or limited use of your Amazon EC2 resources.

If you are a developer, you can use IAM roles to manage the security credentials needed by the applications that you run on your EC2 instances. After you attach an IAM role to your instance, your applications running on the instance can retrieve the credentials from the Instance Metadata Service (IMDS).

For best practices for securing your AWS resources using IAM, see Security best practices in IAM in the *IAM User Guide*.

Contents

Identity-based policies for Amazon EC2

By default, users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API, Amazon EC2 console, or CLI. To allow users to create or modify resources and perform tasks, you must create IAM policies that grant users permission to use the specific resources and API actions they'll need, and then attach those policies to the users, groups, or IAM roles that require those permissions.

When you attach a policy to a user, group of users, or role it allows or denies the users permission to perform the specified tasks on the specified resources. For more general information about IAM policies, see Policies and permissions in IAM in the *IAM User Guide*. For more information about managing and creating IAM policies, see Manage IAM policies.

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.

To get started, you can check whether the AWS managed policies for Amazon EC2 meet your needs. Otherwise, you can create your own custom policies. For more information, see the section called “AWS managed policies”.

Contents

- Actions for Amazon EC2
- Supported resource-level permissions for Amazon EC2 API actions
- Amazon Resource Names (ARNs) for Amazon EC2
- Condition keys for Amazon EC2
- Control access using attribute-based access
- Grant permissions to users, groups, and roles
- Check that users have the required permissions

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, 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.
- **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.

- **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.

For more information about policy requirements, see the IAM JSON policy reference in the *IAM User Guide*. For example IAM policy statements for Amazon EC2, see Example policies to control access the Amazon EC2 API.

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*"
```

Note

Currently, the Amazon EC2 Describe* API actions do not support resource-level permissions. For more information about resource-level permissions for Amazon EC2, see Identity-based policies for Amazon EC2.

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](). 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 Amazon EC2 resources during creation]().

Amazon Resource Names (ARNs) for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

An ARN has the following general syntax:

```
arn:aws:[service]:[region]:[account-id]:resourceType/resourcePath
```

service

The service (for example, `ec2`).

region

The Region for the resource (for example, `us-east-1`).

account-id

The AWS account ID, with no hyphens (for example, `123456789012`).

resourceType

The type of resource (for example, `instance`).

resourcePath

A path that identifies the resource. You can use the * wildcard in your paths.

For example, you can indicate a specific instance (`i-1234567890abcdef0`) in your statement using its ARN as follows.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

You can specify all instances that belong to a specific account by using the * wildcard as follows.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

You can also specify all Amazon EC2 resources that belong to a specific account by using the * wildcard as follows.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:*"
```

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the `Resource` element as follows.

```
"Resource": "*"
```

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so a 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 global 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 global condition keys. For more information, see Information available in all requests in the *IAM User Guide*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see Example: Restrict access to a specific Region.

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"],
     "Resource": "arn:aws:ec2:us-east-1:111122223333:security-group/*",
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": "arn:aws:ec2:us-east-1:111122223333:vpc/
vpc-11223344556677889"
        }
      }
    }
  ]
}
```

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permissions to be granted, all conditions must be met.

You can also use placeholders when you specify conditions. For 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*.

Condition keys

- ec2:Attribute condition key
- ec2:ResourceID condition keys
- ec2:SourceInstanceARN condition key

ec2:Attribute condition key

The `ec2:Attribute` condition key can be used for conditions that filter access by an attribute of a resource.

This condition key supports only properties that are of a primitive data type (such as strings or integers), or complex **AttributeValue** objects that contain only a **Value** property (such as the **Description** or **ImdsSupport** objects of the ModifyImageAttribute API action). The condition key can't be used with complex objects that contain multiple properties, such as the **LaunchPermission** object of ModifyImageAttribute.

For example, the following policy uses the `ec2:Attribute/Description` condition key to filter access by the complex **Description** object of the **ModifyImageAttribute** API action. The condition key allows only requests that modify an image's description to either `Production` or `Development`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute/Description": [
            "Production",
            "Development"
          ]
        }
      }
    }
  ]
}
```

The following example policy uses the `ec2:Attribute` condition key to filter access by the primitive **Attribute** property of the **ModifyImageAttribute** API action. The condition key denies all requests that attempt to modify an image's description.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
```

```
        "Condition": {
          "StringEquals": {
            "ec2:Attribute": "Description"
          }
        }
      }
    ]
}
```

ec2:ResourceID condition keys

When using the following `ec2:`*`Resource`*`ID` condition keys with the specified API actions, the condition key value is used to specify the resulting resource that is created by the API action. `ec2:`*`Resource`*`ID` condition keys can't be used to specify a source resource that is specified in the API request. If you use one of the following `ec2:`*`Resource`*`ID` condition keys with a specified API, then you must always specify the wildcard (`*`). If you specify a different value, the condition always resolves to `*` during runtime. For example, to use the `ec2:ImageId` condition key with the **CopyImage** API, then you must specify the condition key as follows:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:ImageID": "*"
        }
      }
    }
  ]
}
```

We recommend that you avoid using these condition keys with these API actions:

- `ec2:DhcpOptionsID` – `CreateDhcpOptions`
- `ec2:ImageID` – `CopyImage`, `CreateImage`, `ImportImage`, and `RegisterImage`
- `ec2:InstanceID` – `RunInstances` and `ImportInstance`
- `ec2:InternetGatewayID` – `CreateInternetGateway`
- `ec2:NetworkAclID` – `CreateNetworkAcl`
- `ec2:NetworkInterfaceID` – `CreateNetworkInterface`
- `ec2:PlacementGroupName` – `CreatePlacementGroup`
- `ec2:RouteTableID` – `CreateRouteTable`
- `ec2:SecurityGroupID` – `CreateSecurityGroup`
- `ec2:SnapshotID` – `CopySnapshot`, `CreateSnapshot`, `CreateSnapshots`, and `ImportSnapshots`
- `ec2:SubnetID` – `CreateSubnet`
- `ec2:VolumeID` – `CreateVolume` and `ImportVolume`
- `ec2:VpcID` – `CreateVpc`
- `ec2:VpcPeeringConnectionID` – `CreateVpcPeeringConnection`

To filter access based on specific resource IDs, we recommend that you use the `Resource` policy element as follows.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-01234567890abcdef"
    }
  ]
}
```

ec2:SourceInstanceARN condition key

Use `ec2:SourceInstanceARN` to specify the ARN of the instance from which a request is made. This is an AWS global condition key, which means that you can use it with services other than Amazon EC2. For a policy example, see Example: Allow a specific instance to view resources in other AWS services.

Control access using attribute-based access

When you create an IAM policy that grants 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 `aws:ResourceTag` condition key to allow or deny access to the resource based on the tags that are attached to the resource.

```
"StringEquals": { "aws:ResourceTag/environment": "production" }
```

To learn whether an Amazon EC2 API action supports controlling access using the `aws: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 to control access the Amazon EC2 API.

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.

Grant permissions to users, groups, and roles

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.

- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

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 a 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.

Example policies to control access the Amazon EC2 API

You can use IAM policies to grant users the permissions required to work with Amazon EC2. For step-by-step directions, see Creating IAM policies in the *IAM User Guide*.

The following examples show policy statements that you could use to grant users permissions to use Amazon EC2. These policies are designed for requests that are made using the AWS CLI or an AWS SDK. In the following examples, replace each *`user input placeholder`* with your own information.

Examples

- Example: Read-only access
- Example: Restrict access to a specific Region
- Work with instances
- Launch instances (RunInstances)
- Work with Spot Instances
- Example: Work with Reserved Instances
- Example: Tag resources
- Example: Work with IAM roles
- Example: Work with route tables
- Example: Allow a specific instance to view resources in other AWS services
- Example: Work with launch templates
- Work with instance metadata
- Work with Amazon EBS volumes and snapshots

For example policies for working in the Amazon EC2 console, see Example policies to control access to the Amazon EC2 console.

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.

JSON

```
{
   "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.

JSON

```
{
  "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
- Example: Describe all instances, and stop, start, and terminate only particular instances

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",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeImages",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": "*"
    }
   ]
}
```

Example: Describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances i-1234567890abcdef0 and i-0598c7d356eba48d7, and to terminate only instances in the `us-east-1` Region, 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-1` Region 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"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:111122223333:instance/i-0598c7d356eba48d7"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/purpose": "test"
         }
      }
   }

   ]
}
```

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.

Resources

- AMIs
- Instance types
- Subnets
- EBS volumes
- Tags
- Tags in a launch template
- Elastic GPUs
- Launch templates

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).

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:us-east-1::image/ami-9e1670f7",
        "arn:aws:ec2:us-east-1::image/ami-45cf5c3c",
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
        "arn:aws:ec2:us-east-1:111122223333:security-group/*",
        "arn:aws:ec2:us-east-1:111122223333:subnet/*",
        "arn:aws:ec2:us-east-1:111122223333:network-interface/*"
      ]
    }
   ]
}
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon, or certain trusted and verified partners. 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).

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
```

```
                "ec2:Owner": "amazon"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "arn:aws:ec2:us-east-1:111122223333:subnet/*",
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
            "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
            "arn:aws:ec2:us-east-1:111122223333: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:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
         }
```

```
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:us-east-1::image/ami-*",
            "arn:aws:ec2:us-east-1:111122223333:subnet/*",
            "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
            "arn:aws:ec2:us-east-1:111122223333: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:us-east-1:111122223333:instance/*"
        ],
        "Condition": {
            "StringNotEquals": {
                "ec2:InstanceType": ["t2.micro", "t2.small"]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
```

```
            "arn:aws:ec2:us-east-1::image/ami-*",
            "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
            "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "arn:aws:ec2:us-east-1:111122223333:subnet/*",
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
            "arn:aws:ec2:us-east-1:111122223333:security-group/*"
            ]
        }
    ]
}
```

Subnets

The following policy allows users to launch instances using only the specified subnet, `subnet-12345678`. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so).

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:subnet/subnet-12345678",
        "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "arn:aws:ec2:us-east-1::image/ami-*",
        "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
        "arn:aws:ec2:us-east-1:111122223333: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.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*"
      ],
      "Condition": {
         "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-12345678"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:instance/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333: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.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
                {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

Tags

Tag instances on creation

The following policy allows users to launch instances and tag the instances during creation. For resource-creating actions that apply tags, users must have permissions to use the `CreateTags` action. The second statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `RunInstances`, and only for instances. Users cannot tag existing resources, and users cannot tag volumes using the `RunInstances` request.

For more information, see Grant permission to tag Amazon EC2 resources during creation.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333: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`. If users don't pass these specific tags, or if they don't specify tags at all, the request fails.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1::image/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
          "arn:aws:ec2:us-east-1:111122223333:volume/*",
          "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/environment": "production" ,
             "aws:RequestTag/purpose": "webserver"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
            "ec2:CreateTags"
         ],
         "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
         "Condition": {
            "StringEquals": {
                "ec2:CreateAction" : "RunInstances"
             }
          }
      }
    ]
}
```

Tag instances and volumes on creation with at least one specific tag

The following policy uses the `ForAnyValue` modifier on the `aws:TagKeys` condition to indicate that at least one tag must be specified in the request, and it must contain the key `environment` or `webserver`. The tag must be applied to both instances and volumes. Any tag values can be specified in the request.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1::image/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
          "ec2:RunInstances"
```

```
        ],
        "Resource": [
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:instance/*"
        ],
        "Condition": {
            "ForAnyValue:StringEquals": {
                "aws:TagKeys": ["environment","webserver"]
            }
         }
      },
      {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction" : "RunInstances"
            }
         }
      }
    ]
}
```

If instances are tagged on creation, they must be tagged with a specific tag

In the following policy, users do not have to specify tags in the request, but if they do, the tag must be `purpose=test`. No other tags are allowed. Users can apply the tags to any taggable resource in the `RunInstances` request.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "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/*",
```

```
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Only allow specific tags for spot-instances-request. Surprise inconsistency number 2 comes into play here. Under normal circumstances, specifying no tags will result in Unauthenticated. In the case of spot-instances-request, this policy will not be evaluated if there are no spot-instances-request tags, so a non-tag Spot on Run request will succeed.

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": [
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/lt-09477bcd97b0d310e"
          },
          "Bool": {
```

```
                "ec2:IsLaunchTemplateResource": "true"
             }
          }
       },
       {
         "Effect": "Allow",
         "Action": [
            "ec2:CreateTags"
         ],
         "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
         "Condition": {
            "StringEquals": {
                "ec2:CreateAction" : "RunInstances"
             }
          }
       }
    ]
}
```

Elastic GPUs

In the following policy, users can launch an instance and specify an elastic GPU to attach to the instance. Users can launch instances in any Region, but they can only attach an elastic GPU during a launch in the `us-east-2` Region.

The `ec2:ElasticGpuType` condition key ensures that instances use either the `eg1.medium` or `eg1.large` elastic GPU type.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
             {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:elastic-gpu/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2",
                    "ec2:ElasticGpuType": [
                        "eg1.medium",
                        "eg1.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:111122223333:network-interface/*",
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ec2:*:111122223333:subnet/*",
                "arn:aws:ec2:*:111122223333:volume/*",
                "arn:aws:ec2:*:111122223333:key-pair/*",
                "arn:aws:ec2:*:111122223333: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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
         {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
```

```
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333: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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
         {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    }
  ]
}
```

The following example policy allows user to launch instances, but only if they use a launch template. Users cannot override the subnet and network interface parameters in the request; these parameters can only be specified in the launch template. The first part of the statement uses the

NotResource element to allow all other resources except subnets and network interfaces. The second part of the statement allows the subnet and network interface resources, but only if they are sourced from the launch template.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": ["arn:aws:ec2:us-east-1:111122223333:subnet/*",
                      "arn:aws:ec2:us-east-1:111122223333:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          }
       }
    },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": ["arn:aws:ec2:us-east-1:111122223333:subnet/*",
                   "arn:aws:ec2:us-east-1:111122223333:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333: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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
         "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
       "StringEquals": {
           "aws: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"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        }
    ]
}
```

Warning

NOT SUPPORTED – Example: Deny users permission to request Spot Instances using RunInstances

The following policy is not supported for the `spot-instances-request` resource.

The following policy is meant to give users the permission to launch On-Demand Instances, but deny users the permission to request Spot Instances. The `spot-instances-request` resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement is meant to deny the RunInstances action for the `spot-instances-request` resource. However, this condition is not supported because Amazon EC2 does not evaluate the `spot-instances-request` resource in the RunInstances statement if the Spot Instance request is not tagged on create.

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/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequestsNOTSUPPORTEDDONOTUSE",
            "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.

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": "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.

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": "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.

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/*"
            ]
```

```
        },
        {
            "Sid":
 "RequestSpotInstancesOnlyIfTagIsEnvironmentProductionNOTSUPPORTEDDONOTUSE",
            "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"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequests",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Example: 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.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
        "ec2:ModifyReservedInstances",
        "ec2:PurchaseReservedInstancesOffering",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
    }
   ]
}
```

To allow users to view and modify the Reserved Instances in your account, but not purchase new Reserved Instances.

JSON

```
{
  "Version": "2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
        "ec2:ModifyReservedInstances",
```

```
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        }
    ]
}
```

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`. No other tags are allowed and the user cannot tag any other resource types.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
             {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        }
    ]
}
```

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 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:us-east-1:111122223333:*/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/anycompany:environment-type":
 ["test","prod"],
                    "aws: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",
        "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "Condition": {
          "ForAllValues:StringEquals": {
            "aws:TagKeys": ["environment","cost-center"]
          }
        }
      }
    ]
}
```

This policy allows users to delete only the `environment=prod` tag on any resource, and only if the resource is already tagged with a key of `owner` and a value of the username. Users can't delete any other tags for a resource.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "prod",
          "aws:ResourceTag/owner": "${aws:username}"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment"]
        }
      }
    }
  ]
}
```

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.

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"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department":"test"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeIamInstanceProfileAssociations",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/DevTeam*"
    }
  ]
}
```

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.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/TestRole-*"
        }
    ]
}
```

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.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-east-1:111122223333: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` global condition key to specify that the instance from which the request is made must be instance `i-093452212644b0dd6`. If the same IAM role is associated with another instance, the other instance cannot perform any of these actions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:DescribeVolumes",
                "s3:ListAllMyBuckets",
                "dynamodb:ListTables",
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ArnEquals": {
                    "ec2:SourceInstanceARN": "arn:aws:ec2:us-
east-1:111122223333:instance/i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

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:us-east-1:111122223333: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",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Testing"
        }
      }
    }
  ]
}
```

Work with instance metadata

The following policies ensure that users can only retrieve instance metadata 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]() and [Launch instances (RunInstances)]().

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 Migrate your Auto Scaling groups to launch templates 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.

Examples

- Require the use of IMDSv2
- Deny opt-out of IMDSv2
- Specify maximum hop limit
- Limit who can modify the instance metadata options
- Require role credentials to be retrieved from IMDSv2

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.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
               {
            "Sid": "RequireImdsV2",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
```

```
                "StringNotEquals": {
                    "ec2:MetadataHttpTokens": "required"
                }
            }
        }
    ]
}
```

Deny opt-out of IMDSv2

The following policy specifies that you cannot call the `ModifyInstanceMetadataOptions` API and allow the option of IMDSv1 or IMDSv2. If you call the `ModifyInstanceMetadataOptions` API, the `HttpTokens` attribute must be set to `required`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "DenyIMDSv1HttpTokensModification",
        "Effect": "Deny",
        "Action": "ec2:ModifyInstanceMetadataOptions",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringNotEquals": {
                "ec2:Attribute/HttpTokens": "required"
            },
            "Null": {
                "ec2:Attribute/HttpTokens": false
            }
        }
    }]
}
```

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.

JSON

```
{
    "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 permits only users with the role `ec2-imds-admins` to make changes to the instance metadata options. If any principal other than the `ec2-imds-admins` role tries to call the ModifyInstanceMetadataOptions API, it will get an `UnauthorizedOperation` error. This statement could be used to control the use of the ModifyInstanceMetadataOptions API; there are currently no fine-grained access controls (conditions) for the ModifyInstanceMetadataOptions API.

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.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
            {
        "Sid": "RequireAllEc2RolesToUseV2",
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NumericLessThan": {
                "ec2:RoleDelivery": "2.0"
            }
        }
      }
    ]
}
```

Work with Amazon EBS volumes and snapshots

For example policies for working with Amazon EBS volumes and snapshots, see Identity-based policy examples for Amazon EBS.

Example policies to control access to the Amazon EC2 console

You can use IAM policies to grant users the permissions required to work with Amazon EC2. For step-by-step directions, see Creating IAM policies in the *IAM User Guide*.

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. 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.

The following examples show policy statements that you could use to grant users permissions to use Amazon EC2. Replace each *`user input placeholder`* with your own information. These policies are designed for requests that are made using the AWS Management Console. The Amazon EC2 console might call multiple API actions to display a single resource, and it might not be obvious until the user attempts a task and the console displays an error. For more information, see the following AWS Security Blog post: Granting Users Permission to Work in the Amazon EC2 Console.

Examples

- Example: Read-only access
- Example: Use the EC2 launch instance wizard
- Example: Work with security groups
- Example: Work with Elastic IP addresses
- Example: Work with Reserved Instances

To help you work out which API actions are required to perform tasks in the console, you can use a service that logs calls, such as AWS CloudTrail. 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.

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. 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.

JSON

```
{
   "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`, `cloudwatch:DescribeAlarmsForMetric`, `cloudwatch:ListMetrics`, `cloudwatch:GetMetricStatistics`, and `cloudwatch:GetMetricData` actions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeInstances",
            "ec2:DescribeInstanceTypes",
            "cloudwatch:DescribeAlarms",
            "cloudwatch:DescribeAlarmsForMetric",
            "cloudwatch:ListMetrics",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:GetMetricData"
```

```
        ],
        "Resource": "*"
    }
    ]
}
```

Example: Use the EC2 launch instance wizard

The Amazon EC2 launch instance wizard is a screen 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 instance 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"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*"
        }
    ]
}
```

You can add API actions to your policy to provide more options for users, for example:

- `ec2:DescribeAvailabilityZones`: To view and select a specific Availability Zone.
- `ec2:DescribeNetworkInterfaces`: To view and select existing network interfaces for the selected subnet.
- To add outbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupEgress` API action. To modify or delete existing rules, users must be granted permission to use the relevant `ec2:RevokeSecurityGroup*` API action.
- `ec2:CreateTags`: To tag the resources that are created by `RunInstances`. For more information, see Grant permission to tag Amazon EC2 resources during creation. If users do not have permission to use this action and they attempt to apply tags on the tagging page of the launch instance wizard, the launch fails.

Important

Specifying a **Name** while launching an instance creates a tag and requires the `ec2:CreateTags` action. Be careful about granting users permission to use the `ec2:CreateTags` action, because doing so limits your ability to use the `aws: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 using attribute-based access.

- To use Systems Manager parameters when selecting an AMI, you must add `ssm:DescribeParameters` and `ssm:GetParameters` to your policy. `ssm:DescribeParameters` grants your users the permission to view and select Systems Manager parameters. `ssm:GetParameters` grants your 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 instance 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 specified Region. If users select a different Region, or select a different instance type, AMI, or subnet in the launch instance wizard, the launch fails.

The first statement grants users permission to view the options in the launch instance 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). 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, or certain trusted and verified partners.

JSON

```
{
   "Version": "2012-10-17",
```

```
    "Statement": [{
        "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:us-east-2:111122223333:network-interface/*",
            "arn:aws:ec2:us-east-2:111122223333:volume/*",
            "arn:aws:ec2:us-east-2:111122223333:key-pair/*",
            "arn:aws:ec2:us-east-2:111122223333:security-group/*",
            "arn:aws:ec2:us-east-2:111122223333:subnet/subnet-1a2b3c4d"
        ]
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:us-east-2:111122223333:instance/*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:InstanceType": "t2.micro"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
                "arn:aws:ec2:us-east-2::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 instance wizard. The second statement grants users the permission to only use parameters that are named `prod-*`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ssm:DescribeParameters"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ssm:GetParameters"
        ],
       "Resource": "arn:aws:ssm:us-east-2:123456123456:parameter/prod-*"
    }
    ]
}
```

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": [
         "arn:aws:ec2:us-east-2:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/Department": "Test"
         }
```

```
        }
      },
      {
        "Effect": "Allow",
        "Action": [
          "ec2:ModifySecurityGroupRules"
        ],
        "Resource": [
          "arn:aws:ec2:us-east-2:111122223333: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 the least the following API actions:

- `ec2:CreateSecurityGroup`: To create a new security group.
- `ec2:DescribeVpcs`: To view a list of existing VPCs in the **VPC** list.

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.
- `ec2:AuthorizeSecurityGroupEgress`: To add outbound rules to VPC security groups.
- `ec2:RevokeSecurityGroupIngress`: To modify or delete existing inbound rules. This is useful to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.
- `ec2:RevokeSecurityGroupEgress`: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
- `ec2:DeleteSecurityGroup`: To cater for when invalid rules cannot be saved. The console first creates the security group, and then adds the specified rules. If the rules are invalid, the action

fails, and the console attempts to delete the security group. The user remains in the **Create Security Group** dialog box so that they can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user must add them afterward.

- `ec2:UpdateSecurityGroupRuleDescriptionsIngress`: To add or update descriptions of ingress (inbound) security group rules.
- `ec2:UpdateSecurityGroupRuleDescriptionsEgress`: To add or update descriptions of egress (outbound) security group rules.
- `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 a 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"
          ],
          "Resource": "arn:aws:ec2:us-east-2:111122223333:security-group/*",
          "Condition":{
             "ArnEquals": {
                "ec2:Vpc": "arn:aws:ec2:us-east-2:111122223333:vpc/vpc-1a2b3c4d"
             }
          }
       }
    ]
}
```

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 allows users 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": [
         "ec2:DescribeReservedInstances",
         "ec2:ModifyReservedInstances",
         "ec2:PurchaseReservedInstancesOffering",
         "ec2:DescribeInstances",
         "ec2:DescribeInstanceTypes",
         "ec2:DescribeAvailabilityZones",
         "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
   }
   ]
}
```

The `ec2:DescribeAvailabilityZones` action is necessary to ensure that the Amazon EC2 console can display information about the Availability Zones in which you can purchase Reserved Instances. The `ec2:DescribeInstances` action is not required, but ensures that the user can view the instances in the account and purchase reservations to match the correct specifications.

You can adjust the API actions to limit user access, for example removing `ec2:DescribeInstances` and `ec2:DescribeAvailabilityZones` means the user has read-only access.

AWS managed policies for Amazon EC2

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*.

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*.

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 Managed Policy Reference*.

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 Managed Policy Reference*.

AWS managed policy: AmazonEC2ImageReferencesAccessPolicy

You can attach the `AmazonEC2ImageReferencesAccessPolicy` policy to your IAM identities. This policy grants the permissions needed to use the EC2 DescribeImageReferences API, including permission to view EC2 instances, launch templates, Systems Manager parameters, and Image Builder recipes. The policy supports the `IncludeAllResourceTypes` flag and will continue to work when AWS adds support for new resource types, removing the need for future policy updates.

To view the permissions for this policy, see AmazonEC2ImageReferencesAccessPolicy in the *AWS Managed Policy Reference*.

AWS managed policy: AWSEC2CapacityReservationFleetRolePolicy

This policy is attached to the service-linked role named **AWSServiceRoleForEC2CapacityReservationFleet** to allow the service to create, modify, and cancel Capacity Reservations in a Capacity Reservation Fleet on your behalf. For more information, see Using service-linked roles for Capacity Reservation Fleet.

To view the permissions for this policy, see AWSEC2CapacityReservationFleetRolePolicy in the *AWS Managed Policy Reference*.

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.

To view the permissions for this policy, see AWSEC2FleetServiceRolePolicy in the *AWS Managed Policy Reference*.

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.

To view the permissions for this policy, see AWSEC2SpotFleetServiceRolePolicy in the *AWS Managed Policy Reference*.

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.

To view the permissions for this policy, see AWSEC2SpotServiceRolePolicy in the *AWS Managed Policy Reference*.

AWS managed policy: AWSEC2VssSnapshotPolicy

You can attach this managed policy to the IAM instance profile role that you use for your Amazon EC2 Windows instances. The policy grants permissions to allow Amazon EC2 to create and manage VSS snapshots on your behalf.

To view the permissions for this policy, see AWSEC2VssSnapshotPolicy in the *AWS Managed Policy Reference*.

AWS managed policy: DeclarativePoliciesEC2Report

This policy is attached to the service-linked role named `AWSServiceRoleForDeclarativePoliciesEC2Report` to provide access to read-only APIs needed to generate the account status report for declarative policies.

To view the permissions for this policy, see DeclarativePoliciesEC2Report in the *AWS Managed Policy Reference*.

AWS managed policy: EC2FastLaunchFullAccess

You can attach the `EC2FastLaunchFullAccess` policy to your instance profile or other IAM role. This policy grants full access to EC2 Fast Launch actions, and targeted permissions as follows.

Permissions details

- **EC2 Fast Launch** – Administrative access is granted, so that the role can enable or disable EC2 Fast Launch, and describe EC2 Fast Launch images.
- **Amazon EC2** – Access is granted for Amazon EC2 RunInstances, CreateTags, Describe, and Create and Modify Launch Template operations. Access is also granted to create network and security group resources, authorize ingress rules, and delete resources that EC2 Fast Launch created.
- **IAM** – Access is granted to get and use instance profiles whose name contains `ec2fastlaunch` to create the EC2FastLaunchServiceRolePolicy service-linked role.
- **AWS CloudFormation** – Access is granted for EC2 Fast Launch to describe and create CloudFormation stacks, and to delete stacks that it created.

To view the permissions for this policy, see EC2FastLaunchFullAccess in the *AWS Managed Policy Reference*.

AWS managed policy: EC2FastLaunchServiceRolePolicy

This policy is attached to the service-linked role named **AWSServiceRoleForEC2FastLaunch** to allow Amazon EC2 to create and manage a set of pre-provisioned snapshots that reduce the time it takes to launch instances from your EC2 Fast Launch-enabled AMI. For more information, see the section called "Service-linked role".

To view the permissions for this policy, see EC2FastLaunchServiceRolePolicy in the *AWS Managed Policy Reference*.

AWS managed policy: Ec2InstanceConnect

You can attach the `Ec2InstanceConnect` policy to your IAM identities. This policy grants permissions that allows customers to call EC2 Instance Connect to publish ephemeral keys to their EC2 instances and connect via ssh or the EC2 Instance Connect CLI.

To view the permissions for this policy, see Ec2InstanceConnect in the *AWS Managed Policy Reference*.

AWS managed policy: Ec2InstanceConnectEndpoint

This policy is attached to a service-linked role named **AWSServiceRoleForEC2InstanceConnect** to allow EC2 Instance Connect Endpoint to perform actions on your behalf. For more information, see Service-linked role for EC2 Instance Connect Endpoint.

To view the permissions for this policy, see [Ec2InstanceConnectEndpoint]() in the *AWS Managed Policy Reference*. For a description of the updates to this policy, see [Amazon EC2 updates to AWS managed policies]().

Amazon EC2 updates to AWS managed policies

View details about updates to AWS managed policies for Amazon EC2 since this service began tracking these changes.

Change	Description	Date
[AmazonEC2ImageReferencesAccessPolicy]() – New policy	Amazon EC2 added this policy to provide permission to scan all resources types supported by the EC2 DescribeImageReferences API.	August 26, 2025
[Ec2InstanceConnectEndpoint]() – Updated policy	To support the modification of existing Instance Connect Endpoints, Amazon EC2 updated this policy to add permissions to assign and unassign IPv6 addresses and modify security groups on network interfaces created by EC2 Instance Connect Endpoint. Amazon EC2 also updated this policy to replace the `Null` condition operator with the `StringLike` condition operator.	July 31, 2025
[EC2FastLaunchServiceRolePolicy]() – Updated policy	To help prevent orphaned resources, Amazon EC2 updated this policy to add permission to describe volumes, volume attributes and network interfaces,	July 17, 2025

Change	Description	Date
	and to delete volumes and network interfaces that EC2 Fast Launch created.	
[EC2FastLaunchFullAccess]() – Updated policy	Amazon EC2 updated this policy to include Create and Modify Launch Template operations, to create network and security group resources , authorize ingress rules, and delete resources that EC2 Fast Launch created. It can additionally describe and create CloudFormation stacks, and delete stacks that EC2 Fast Launch created.	May 14, 2025
[EC2FastLaunchServiceRolePolicy]() – Updated policy	Amazon EC2 updated this policy to add Amazon EventBridge access to create and manage event rules for EC2 Fast Launch. Additionally, EC2 Fast Launch can now describe AWS CloudFormation stacks, launch an instance from an AMI that's associated with AWS License Manager, get a list of AWS KMS grants it created that can be retired, and delete launch templates that it created.	May 14, 2025

Change	Description	Date
AWSEC2CapacityReservationFleetRolePolicy – Updated permissions	Amazon EC2 updated the `AWSEC2CapacityReservationFleetRolePolicy` managed policy to use the `ArnLike` condition operator instead of the `StringLike` condition operator.	March 03, 2025
AmazonEC2ReadOnlyAccess – Added permissions	Amazon EC2 added a permission that allows you to retrieve security groups using the `GetSecurityGroupsForVpc` operation.	December 27, 2024
EC2FastLaunchFullAccess – New policy	Amazon EC2 added this policy to perform API actions related to the EC2 Fast Launch feature from an instance. The policy can be attached to the instance profile for an instance that's launched from a EC2 Fast Launch enabled AMI.	May 14, 2024
AWSEC2VssSnapshotPolicy – New policy	Amazon EC2 added the `AWSEC2VssSnapshotPolicy` policy that contains permissions to create and add tags to Amazon Machine Images (AMIs) and EBS snapshots.	March 28, 2024

Change	Description	Date
Ec2InstanceConnectEndpoint – New policy	Amazon EC2 added the `Ec2InstanceConnectEndpoint` policy. This policy is attached to the **AWSServiceRoleForEC2InstanceConnect** service-linked role, to allow Amazon EC2 to perform actions on your behalf when you create an EC2 Instance Connect Endpoint.	January 24, 2023
EC2FastLaunchServiceRolePolicy – New policy	Amazon EC2 added the EC2 Fast Launch feature to enable Windows AMIs to launch instances faster by creating a set of pre-provisioned snapshots.	November 26, 2021
Amazon EC2 started tracking changes	Amazon EC2 started tracking changes to its AWS managed policies	March 1, 2021

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 users. If you change a role, the change is propagated to all instances.

> **Note**
>
> Amazon EC2 IAM role credentials are not subject to maximum session durations configured in the role. For more information, see Methods to assume a role in the *IAM User Guide*.

When creating IAM roles, associate least privilege IAM policies that restrict access to the specific API calls the application requires. For Windows-to-Windows communication, use well-defined and well-documented Windows groups and roles to grant application-level access between Windows instances. Groups and roles allow customers to define least privilege application and NTFS folder-level permissions to limit access to application-specific requirements.

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 and the following example: Example: Work with IAM roles.

Contents

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. You can include an IAM role in multiple instance profiles.

To update permissions for an instance, replace its instance profile. We do not recommend removing a role from an instance profile, because there is a delay of up to one hour before this change takes effect.

For more information, see Use instance profiles in the *IAM User Guide*.

Permissions for your use case

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. For more information, see IAM Access Analyzer policy generation 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.

For more information about instance metadata, see Use instance metadata to manage your EC2 instance.

> **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.

For your Amazon EC2 workloads, we recommend that you retrieve session credentials using the method described below. These credentials should enable your workload to make AWS API requests, without needing to use `sts:AssumeRole` to assume the same role that is already associated with the instance. Unless you need to pass session tags for attribute-based access control (ABAC) or pass a session policy to further restrict permissions of the role, such role assumption calls are unnecessary as they create a new set of the same temporary role session credentials.

If your workload uses a role to assume itself, you must create a trust policy that explicitly allows that role to assume itself. If you do not create the trust policy, you get an `AccessDenied` error. For more information, see Update a role trust policy in the *IAM User Guide*.

IMDSv2

Linux

Run the following command from your Linux instance to retrieve the security credentials for an IAM role.

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

Run the following cmdlet from your Windows instance to retrieve the security credentials for an IAM role.

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

IMDSv1

Linux

Run the following command from your Linux instance to retrieve the security credentials for an IAM role.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

Run the following cmdlet from your Windows instance to retrieve the security credentials for an IAM role.

```
Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

The following is example output. If you are not able to retrieve the security credentials, see I can't access the temporary security credentials on my EC2 instance in the *IAM User Guide*.

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

For applications, AWS CLI, and Tools for Windows PowerShell commands that run on the instance, you do not have to explicitly get the temporary security credentials—the AWS SDKs, AWS CLI, and Tools for Windows PowerShell automatically get the credentials from the EC2 instance metadata service and use them. To make a call outside of the instance using temporary security credentials (for example, to test IAM policies), you must provide the access key, secret key, and the session token. For more information, see Using Temporary Security Credentials to Request Access to AWS Resources in the *IAM User Guide*.

Grant permissions to attach an IAM role to an instance

The identities in your AWS account, such as IAM users, must have specific permissions to launch an Amazon EC2 instance with an IAM role, attach an IAM role to an instance, replace the IAM role for an instance, or detach an IAM role from an instance. You must grant permission to use the following API actions as required:

- `iam:PassRole`
- `ec2:AssociateIamInstanceProfile`
- `ec2:DisassociateIamInstanceProfile`
- `ec2:ReplaceIamInstanceProfileAssociation`

Note

If you specify the resource for `iam:PassRole` as `*`, this would grant access to pass any of your IAM roles to an instance. To follow the best practice of least privilege, specify the ARNs of specific IAM roles with `iam:PassRole`, as shown in the example policy below.

Example policy for programmatic access

The following IAM policy grants permissions to launch instances with an IAM role, attach an IAM role to an instance, or replace the IAM role for an instance using the AWS CLI or the Amazon EC2 API.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances",
         "ec2:AssociateIamInstanceProfile",
         "ec2:DisassociateIamInstanceProfile",
         "ec2:ReplaceIamInstanceProfileAssociation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/DevTeam*"
    }
  ]
}
```

Additional requirement for console access

To grant permissions to complete the same tasks using the Amazon EC2 console, you must also include the `iam:ListInstanceProfiles` API action.

Attach an IAM role to an instance

You can create an IAM role and attach it to an instance during or after launch. You can also replace or detach IAM roles.

To attach an IAM role to an instance at launch using the Amazon EC2 console, expand **Advanced details**. For **IAM instance profile**, select the IAM role.

> **Note**
>
> If you created your IAM role using the IAM 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 might have given your instance profile a different name than the role.

You can attach an IAM role to an instance that is running or stopped. If the instance already has an IAM role attached, you must replace it with the new IAM role.

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.
4. Choose **Actions**, **Security**, **Modify IAM role**.
5. For **IAM role**, select the IAM instance profile.
6. Choose **Update IAM role**.

AWS CLI

To attach an IAM role to an instance

Use the associate-iam-instance-profile command to attach the IAM role to the instance. When you specify the instance profile, you can use either 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"
```

PowerShell

To attach an IAM role to an instance

Use the Register-EC2IamInstanceProfile cmdlet.

```
Register-EC2IamInstanceProfile `
    -InstanceId i-1234567890abcdef0 `
    -IamInstanceProfile_Name TestRole-1
```

To replace the IAM role on an instance that already has an attached IAM role, the instance must be running. 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.

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.
4. Choose **Actions**, **Security**, **Modify IAM role**.
5. For **IAM role**, select the IAM instance profile.
6. Choose **Update IAM role**.

AWS CLI

To replace an IAM role for an instance

1. If required, use the describe-iam-instance-profile-associations command to get the association ID.

   ```
   aws ec2 describe-iam-instance-profile-associations \
       --filters Name=instance-id,Values=i-1234567890abcdef0 \
       --query IamInstanceProfileAssociations.AssociationId
   ```

2. Use the replace-iam-instance-profile-association command. Specify the association ID for the existing instance profile and the ARN or name of the new instance profile.

   ```
   aws ec2 replace-iam-instance-profile-association \
       --association-id iip-assoc-0044d817db6c0a4ba \
       --iam-instance-profile Name="TestRole-2"
   ```

PowerShell

To replace an IAM role for an instance

1. If required, use the Get-EC2IamInstanceProfileAssociation cmdlet to get the association ID.

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. Use the Set-EC2IamInstanceProfileAssociation cmdlet. Specify the association ID for the existing instance profile and the ARN or name of the new instance profile.

```
Set-EC2IamInstanceProfileAssociation `
    -AssociationId iip-assoc-0044d817db6c0a4ba `
    -IamInstanceProfile_Name TestRole-2
```

You can detach an IAM role from an instance that is running or stopped.

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.
4. Choose **Actions**, **Security**, **Modify IAM role**.
5. For **IAM role**, choose **No IAM Role**.
6. Choose **Update IAM role**.
7. When promoted for confirmation, enter **Detach**, and then choose **Detach**.

AWS CLI

To detach an IAM role from an instance

1. If required, use describe-iam-instance-profile-associations to get the association ID for the IAM instance profile to detach.

```
aws ec2 describe-iam-instance-profile-associations \
```

```
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. Use the disassociate-iam-instance-profile command.

```
aws ec2 disassociate-iam-instance-profile --association-id iip-
assoc-0044d817db6c0a4ba
```

PowerShell

To detach an IAM role from an instance

1. If required, use Get-EC2IamInstanceProfileAssociation to get the association ID for the IAM instance profile to detach.

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. Use the Unregister-EC2IamInstanceProfile cmdlet.

```
Unregister-EC2IamInstanceProfile -AssociationId iip-assoc-0044d817db6c0a4ba
```

Instance identity roles for Amazon EC2 instances

Each Amazon EC2 instance that you launch has an *instance identity role* that represents its identity. An instance identity role is a type of IAM role. AWS services and features that are integrated to use the instance identity role can use it to identify the instance to the service.

The instance identity role credentials are accessible from the Instance Metadata Service (IMDS) at `/identity-credentials/ec2/security-credentials/ec2-instance`. The credentials consist of an AWS temporary access key pair and a session token. They are used to sign AWS Sigv4 requests to the AWS services that use the instance identity role. The credentials are present in the instance metadata regardless of whether a service or feature that makes use of instance identity roles is enabled on the instance.

Instance identity roles are automatically created when an instance is launched, have no role-trust policy document, and are not subject to any identity or resource policy.

Supported services

The following AWS services use the instance identity role:

- **Amazon EC2 –** EC2 Instance Connect uses the instance identity role to update the host keys for a Linux instance.
- **Amazon GuardDuty –** GuardDuty Runtime Monitoring uses the instance identity role to allow the runtime agent to send security telemetry to the GuardDuty VPC endpoint.
- **AWS Security Token Service (AWS STS) –** Instance identity role credentials can be used with the AWS STS `GetCallerIdentity` action.
- **AWS Systems Manager –** When using Default Host Management Configuration, AWS Systems Manager uses the identity provided by the instance identity role to register EC2 instances. After identifying your instance, Systems Manager can pass your `AWSSystemsManagerDefaultEC2InstanceManagementRole` IAM role to your instance.

Instance identity roles can't be used with other AWS services or features because they do not have an integration with instance identity roles.

Instance identity role ARN

The instance identity role ARN takes the following format:

```
arn:aws-partition:iam::account-number:assumed-role/aws:ec2-instance/instance-id
```

For example:

```
arn:aws:iam::0123456789012:assumed-role/aws:ec2-instance/i-1234567890abcdef0
```

For more information about ARNs, see Amazon Resource Names (ARNs) in the *IAM User Guide*.

Update management for Amazon EC2 instances

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.

For EC2 instances in an Auto Scaling group, you can use the AWS-PatchAsgInstance runbook to help avoid instances that are undergoing patching from being replaced. Alternatively, you can use any automatic update services or recommended processes for installing updates that are provided by the application vendor.

Resources

- **AL2023 –** Updating AL2023 in the *Amazon Linux 2023 User Guide*
- **AL2 –** Manage software on your Amazon Linux 2 instance in the *Amazon Linux 2 User Guide*
- **Windows instances –** the section called "Update management"

Security best practices for Windows instances

We recommend that you follow these security best practices for your Windows instances.

Contents

High-level security best practices

You should adhere to the following high-level security best practices for your Windows instances:

- **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,and more. 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.
- **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.
- **Patch management –** Implement processes that regularly patch, update, and secure the operating system and applications on your EC2 instances.
- **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.

Update management

To ensure the best results when you run Windows Server on Amazon EC2, we recommend that you implement the following best practices:

- Configure Windows Update
- Update drivers
- Use the latest Windows AMIs
- Test performance before migration
- Update launch agents
- Reboot your Windows instance after you install updates. For more information, see Reboot your Amazon EC2 instance.

For information about how to upgrade or migrate a Windows instance to a newer version of Windows Server, see Upgrade an EC2 Windows instance to a newer version of Windows Server.

Configure Windows Update

By default, instances that are launched from AWS Windows Server AMIs don't receive updates through Windows Update.

Update Windows drivers

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

- Use SNS topics to receive updates for new driver releases.
- Use the AWS Systems Manager Automation runbook AWSSupport-UpgradeWindowsAWSDrivers to easily apply the updates across your instances.

Launch instances using 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 view updates to each release of the AWS Windows AMIs, see AWS Windows AMI version history.
- To build with the latest available AMIs, see Query for the Latest Windows AMI Using Systems Manager Parameter Store.
- For more information about specialized Windows AMIs that you can use to launch instances for your database and compliance hardening use cases, see Specialized Windows AMIs in the *AWS Windows AMI Reference*.

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, such as dedicated tenancy, placement groups, instance store volumes, and bare metal.

Install the latest version of EC2Launch v2

Install the latest EC2Launch v2 agent to ensure that the latest enhancements are applied across your fleet. For more information, see Install EC2Launch v2.

If you have a mixed fleet, or if you want to continue to use the EC2Launch (Windows Server 2016 and 2019) or EC2 Config (legacy OS only) agents, update to the latest versions of the respective agents.

Automatic updates are supported on the following combinations of Windows Server version and launch agents. You can opt in to automatic updates in the SSM Quick Setup Host Management console under **Amazon EC2 Launch Agents**.

Windows Version	EC2Launch v1	EC2Launch v2
2016	✓	✓
2019	✓	✓
2022		✓

- For more information about updating to EC2Launch v2, see the section called “Install EC2Launch v2”.
- For information to manually update EC2Config, see the section called “Install EC2Config”.
- For information to manually update EC2Launch, see the section called “Install EC2Launch”.

Configuration management

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 that 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.

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 from the Amazon EC2 console or by 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.

Change management

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, and management packs for Systems

Center Operations Manager and System Center Virtual Machine Manager. Note that Microsoft releases Windows patches the second Tuesday of each month (or as needed) and AWS updates all Windows AMIs managed by AWS 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.

Audit and accountability for Amazon EC2 Windows instances

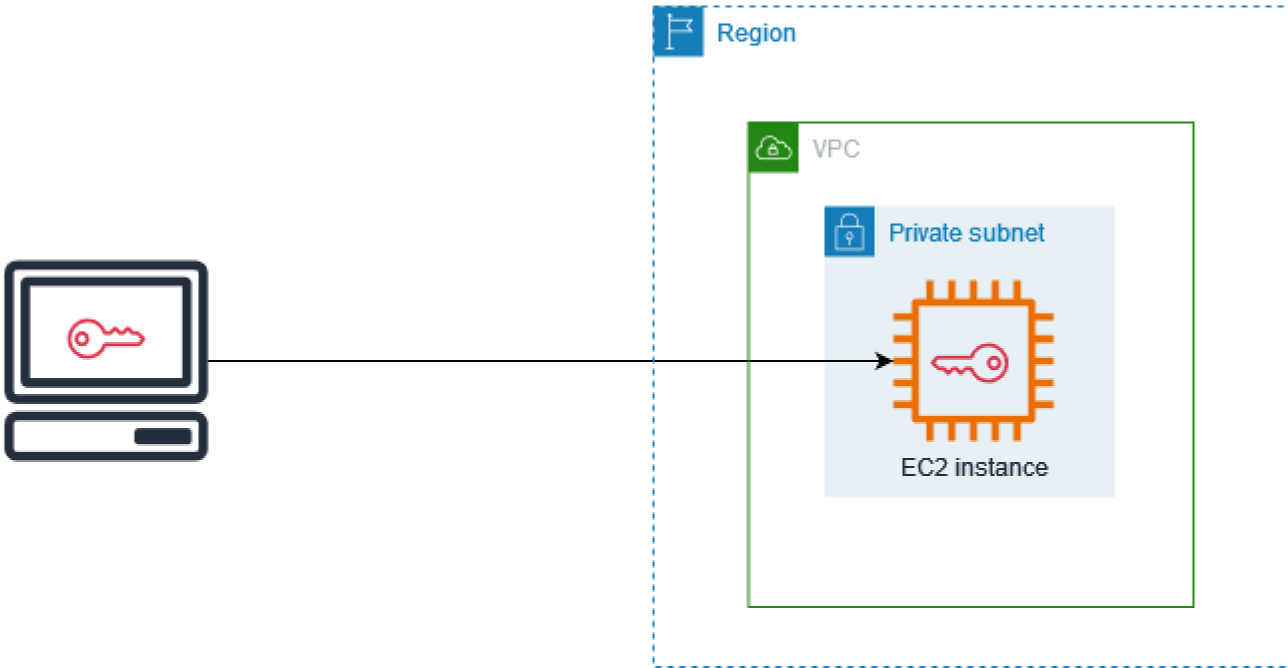
AWS CloudTrail, AWS Config, and AWS Config Rules provide audit and change tracking features for auditing AWS resource changes. Configure Windows event logs to send local log files to a centralized log management system to preserve log data for security and operational behavior analysis. Microsoft System 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.

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, EC2Launch, and EC2Launch v2 launch agents 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.

Amazon EC2 key pairs and Amazon EC2 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. For Linux instances, the private key allows you to securely SSH into your instance. For Windows instances, the private key is required to decrypt the administrator password, which you then use to connect to your instance.

Amazon EC2 stores the public key on your instance, and you store the private key, as shown in the following diagram. It's important that you store your private key in a secure place because anyone who possesses your private key can connect to your instances that use the key pair.



When you launch an instance, you can specify a key pair, so that you can connect to your instance using a method that requires a key pair. Depending on how you manage your security, you can specify the same key pair for all your instances or you can specify different key pairs.

For Linux instances, when your instance boots for the first time, the public key that you specified at launch is placed on your Linux instance in an entry within `~/.ssh/authorized_keys`. When you connect to your Linux instance using SSH, to log in you must specify the private key that corresponds to the public key.

For more information about connecting to your EC2 instance, see Connect to your EC2 instance.

Important

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 I've lost my private key. How can I connect to my instance?

As an alternative to key pairs, you can use AWS Systems Manager Session Manager to connect to your instance with an interactive one-click browser-based shell or the AWS Command Line Interface (AWS CLI).

Contents

Create a key pair for your Amazon EC2 instance

You can use Amazon EC2 to create your key pairs, or you can use a third-party tool to create your key pairs, and then import them to Amazon EC2.

Amazon EC2 supports 2048-bit SSH-2 RSA keys for Linux and Windows instances. Amazon EC2 also supports ED25519 keys for Linux instances.

For instructions on how to connect to your instance after you have created a key pair, see the section called “Connect to your Linux instance using SSH” and the section called “Connect to your Windows instance using RDP”.

Contents

Create a key pair using Amazon EC2

When you create a key pair using Amazon EC2, the public key is stored in Amazon EC2, and you store the private key.

You can create up to 5,000 key pairs per Region. To request an increase, create a support case. For more information, see Creating a support case in the *Support User Guide*.

Console

To create a key pair using Amazon EC2

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. Select a key pair type appropriate for your operating system:

   (Linux instances) For **Key pair type**, choose either **RSA** or **ED25519**.

   (Windows instances) For **Key pair type**, choose **RSA**. **ED25519** keys are not supported for Windows instances.
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**.
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.

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

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

    If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see Error: Unprotected private key file.

AWS CLI

To create a key pair using Amazon EC2

1. Use the create-key-pair command as follows to generate the key pair and to save the private key to a `.pem` file. The `--query` option prints the private key material to the output. The `--output` option saves the private key material in the specified file. The extension should be either `.pem` or `.ppk`, depending on the key format. The private key name can be 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 \
        --key-format pem \
        --query "KeyMaterial" \
        --output text > my-key-pair.pem
    ```

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

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

    If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see Error: Unprotected private key file.

PowerShell

To create a key pair using Amazon EC2

Use the New-EC2KeyPair cmdlet as follows to generate the key and save it to a .pem or .ppk file. The **Out-File** cmdlet saves the private key material in a file with the specified extension. The extension should be either .pem or .ppk, depending on the key format. The private key name can be different from the public key name, but for ease of use, use the same name.

```
(New-EC2KeyPair `
    -KeyName "my-key-pair" `
    -KeyType "rsa" `
    -KeyFormat "pem").KeyMaterial | Out-File -Encoding ascii -FilePath C:\path\my-
key-pair.pem
```

Create a key pair using AWS CloudFormation

When you create a new key pair using AWS CloudFormation, the private key is saved to AWS Systems Manager Parameter Store. The parameter name has the following format:

```
/ec2/keypair/key_pair_id
```

For more information, see AWS Systems Manager Parameter Store in the *AWS Systems Manager User Guide*.

To create a key pair using AWS CloudFormation

1. Specify the AWS::EC2::KeyPair resource in your template.

```
Resources:
  NewKeyPair:
    Type: 'AWS::EC2::KeyPair'
    Properties:
      KeyName: new-key-pair
```

2. Use the describe-key-pairs command as follows to get the ID of the key pair.

```
aws ec2 describe-key-pairs --filters Name=key-name,Values=new-key-pair --query
 KeyPairs[*].KeyPairId --output text
```

The following is example output.

```
key-05abb699beEXAMPLE
```

3. Use the get-parameter command as follows to get the parameter for your key and save the key material in a `.pem` file.

```
aws ssm get-parameter --name /ec2/keypair/key-05abb699beEXAMPLE --with-decryption
 --query Parameter.Value --output text > new-key-pair.pem
```

Required IAM permissions

To enable AWS CloudFormation to manage Parameter Store parameters on your behalf, the IAM role assumed by AWS CloudFormation or your user must have the following permissions:

- `ssm:PutParameter` – Grants permission to create a parameter for the private key material.
- `ssm:DeleteParameter` – Grants permission to delete the parameter that stored the private key material. This permission is required whether the key pair was imported or created by AWS CloudFormation.

When AWS CloudFormation deletes a key pair that was created or imported by a stack, it performs a permissions check to determine whether you have permission to delete parameters, even though AWS CloudFormation creates a parameter only when it creates a key pair, not when it imports a key pair. AWS CloudFormation tests for the required permission using a fabricated parameter name that does not match any parameter in your account. Therefore, you might see a fabricated parameter name in the `AccessDeniedException` error message.

Create a key pair using a third-party tool and import the public key to Amazon EC2

Instead of using Amazon EC2 to create a 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:
  - (Linux and Windows) RSA
  - (Linux only) ED25519

> **Note**
>
> ED25519 keys are not supported for Windows instances.

  - Amazon EC2 does not accept DSA keys.
- Supported formats:
  - OpenSSH public key format (for Linux, the format in `~/.ssh/authorized_keys`)
  - (Linux only) If you connect using SSH while using the EC2 Instance Connect API, the SSH2 format is also supported.
  - SSH private key file format must be PEM or PPK
  - (RSA only) Base64 encoded DER format
  - (RSA only) SSH public key file format as specified in RFC 4716
- Supported lengths:
  - 1024, 2048, and 4096.
  - (Linux only) If you connect using SSH while using the EC2 Instance Connect API, the supported lengths are 2048 and 4096.

To create a key pair using a third-party tool

1. Generate a key pair with a third-party tool of your choice. 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 a key pair.

   > **Important**
   >
   > The private key must be in the PEM or PPK 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, `~/.ssh/my-key-pair.pub` (Linux, macOS) or `C:\keys\my-key-pair.pub` (Windows). The file name extension for this file is not important.
3. Save the private key to a local file that has the `.pem` or `.ppk` extension. For example, `~/.ssh/my-key-pair.pem` or `~/.ssh/my-key-pair.ppk` (Linux, macOS) or `C:\keys\my-key-`

`pair.pem` or `C:\keys\my-key-pair.ppk` (Windows). The file extension is important because, depending on the tool you use to connect to your instance, you'll need a specific file format. OpenSSH requires a `.pem` file, while PuTTY requires a `.ppk` file.

> **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 to Amazon EC2

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 to Amazon EC2

Use the import-key-pair command.

```
aws ec2 import-key-pair \
    --key-name my-key-pair \
    --public-key-material fileb://path/my-key-pair.pub
```

To verify that the key pair was imported successfully

Use the describe-key-pairs command.

```
aws ec2 describe-key-pairs --key-names my-key-pair
```

PowerShell

To import the public key to Amazon EC2

Use the Import-EC2KeyPair cmdlet.

```
$publickey=[Io.File]::ReadAllText("C:\Users\TestUser\.ssh\id_rsa.pub")
Import-EC2KeyPair `
    -KeyName my-key-pair `
    -PublicKey $publickey
```

To verify that the key pair was imported successfully

Use the Get-EC2KeyPair cmdlet.

```
Get-EC2KeyPair -KeyName my-key-pair
```

Describe your key pairs

You can describe the key pairs that you stored in Amazon EC2. You can also retrieve the public key material and identify the public key that was specified at launch.

Tasks

- Describe your key pairs
- Retrieve the public key material
- Identify the public key specified at launch

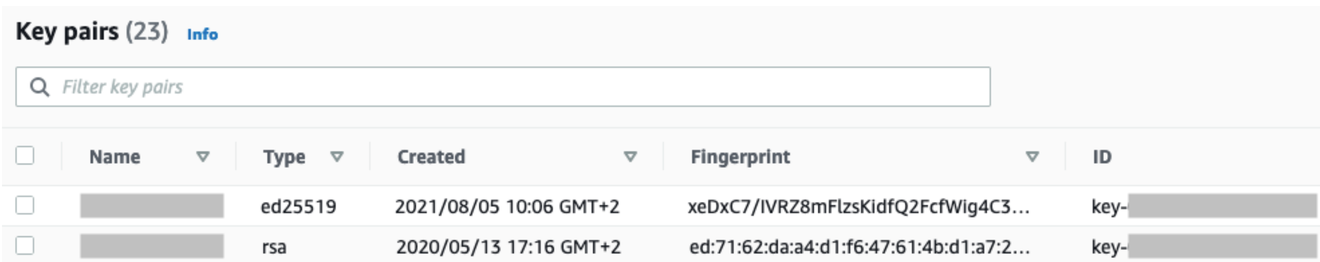
Describe your key pairs

You can view the following information about your public keys that are stored in Amazon EC2: public key name, ID, key type, fingerprint, public key material, the date and time (in the UTC time zone) the key was created by Amazon EC2 (if the key was created by a third-party tool, then it's the date and time the key was imported to Amazon EC2), and any tags that are associated with the public key.

Console

To view information about your key pairs

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigator, choose **Key Pairs**.
3. You can view the information about each public key in the **Key pairs** table.



4. To view a public key's tags, select the checkbox next to the key, and then choose **Actions**, **Manage tags**.

AWS CLI

To view information about a key pair

Use the describe-key-pairs command.

```
aws ec2 describe-key-pairs --key-names key-pair-name
```

PowerShell

To view information about a key pair

Use the Get-EC2KeyPair cmdlet.

```
Get-EC2KeyPair -KeyName key-pair-name
```

Retrieve the public key material

You can get the public key material for your key pairs. The following is an example public key. Note that there are line breaks added for readability.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Private key

To retrieve the public key material using ssh-keygen (Linux)

On your local Linux or macOS computer, use the **ssh-keygen** command. Specify the path where you downloaded your private key (the `.pem` file).

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

If this **ssh-keygen** command fails, run the following **chmod** command to ensure that the private key file has the required permissions.

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

To retrieve the public key material using PuTTYgen (Windows)

On your local Windows computer, start PuTTYgen. 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.

AWS CLI

To retrieve the public key material

Use the following describe-key-pairs command and specify the `--include-public-key` option.

```
aws ec2 describe-key-pairs \
    --key-names key-pair-name \
    --include-public-key \
```

```
    --query "KeyPairs[].PublicKey"
```

PowerShell

To retrieve the public key material

Use the Get-EC2KeyPair cmdlet.

```
(Get-EC2KeyPair -KeyName key-pair-name -IncludePublicKey $true).PublicKey
```

IMDSv2

Linux

Run the following commands from your Linux instance.

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Windows

Run the following cmdlets from your Windows instance.

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

IMDSv1

Linux

Run the following command from your Linux instance.

```
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Windows

Run the following cmdlet from your Windows instance.

```
Invoke-RestMethod -uri  http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

Identify the public key specified at launch

If you specify a public key when you launch an instance, the public key name is recorded by the instance. The public key name reported for an instance does not change, even if you change the public key on the instance or add public keys.

Console

To identify the public key specified at instance launch

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 **Key pair assigned at launch**.

AWS CLI

To identify the public key specified at instance launch

Use the following describe-instances command.

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].KeyName" \
    --output text
```

The following is example output.

```
key-pair-name
```

PowerShell

To identify the public key specified at instance launch

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances | Select KeyName
```

The following is example output.

```
KeyName
-------
key-pair-name
```

Delete your key pair

You can delete a key pair, which removes the public key that is stored in Amazon EC2. Deleting a key pair does not delete the matching private key.

When you delete a public key using the following methods, you're only deleting the public key that you stored in Amazon EC2 when you created or imported the key pair. Deleting a public key doesn't remove the public key from any instances to which you've added it, either when you launched the instance or later. It also doesn't delete the private key on your local computer. You can continue to connect to instances that you launched using a public key that you've deleted from Amazon EC2 as long as you still have the private key (`.pem`) file.

Important

If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the public key 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 public key cannot be found. For more information, see Launch templates in the *Amazon EC2 Auto Scaling User Guide*.

Console

To delete your public key on Amazon EC2

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 **Actions**, **Delete**.

4. In the confirmation field, enter `Delete` and then choose **Delete**.

AWS CLI

To delete your public key on Amazon EC2

Use the delete-key-pair command.

```
aws ec2 delete-key-pair --key-name my-key-pair
```

PowerShell

To delete your public key on Amazon EC2

Use the Remove-EC2KeyPair cmdlet.

```
Remove-EC2KeyPair -KeyName my-key-pair
```

Add or replace a public key on your Linux instance

If you lose a private key, you lose access to any instances that use the key pair. For more information about connecting to an instance using a different key pair than the one that you specified at launch, see I've lost my private key.

When you launch an instance, you can specify a key pair. If you specify a key pair at launch, when your instance boots for the first time, the public key material is placed on your Linux instance in an entry within `~/.ssh/authorized_keys`.

You can change the key pair that is used to access the default system account of your instance by adding a new public key on the instance, or by replacing the public key (deleting the existing public key and adding a new one) on the instance. You can also remove all public keys from an instance. To add or replace a key pair, you must be able to connect to your instance.

You might add or replace a key pair for the following reasons:

- If a user in your organization requires access to the system user using a separate key pair, you can add the public key to your instance.

- If someone has a copy of the private key (`.pem` file) and you want to prevent them from connecting to your instance (for example, if they've left your organization), you can delete the public key on the instance and replace it with a new one.
- If you create a Linux AMI from an instance, the public key material is copied from the instance to the AMI. If you launch an instance from the AMI, the new instance includes the public key from the original instance. To prevent someone who has the private key from connecting to the new instance, you can remove the public key from the original instance *before* creating the AMI.

Use the following procedures to modify the key pair for the default user, such as `ec2-user`. For information about adding users to your instance, see the documentation for the operating system on your instance.

To add or replace a key pair

1. Create a new key pair using the Amazon EC2 console or a third-party tool.
2. Retrieve the public key from your new key pair. For more information, see Retrieve the public key material.
3. Connect to your instance using your existing private key.
4. Using a text editor of your choice, open the `.ssh/authorized_keys` file on the instance. Paste the public key information from your new key pair underneath the existing public key information. Save the file.
5. Disconnect from your instance, and test that you can connect to your instance using the new private key file.
6. (Optional) If you're replacing an existing key pair, connect to your instance and delete the public key information for the original key pair from the `.ssh/authorized_keys` file.

Important

If you're using an Auto Scaling group, ensure that the key pair you're replacing is not specified in your launch template or launch configuration. 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*.

To remove a public key from an instance

1. Connect to your instance.
2. Using a text editor of your choice, open the `.ssh/authorized_keys` file on the instance. Delete the public key information, and then save the file.

> **Warning**
>
> After you remove all the public keys from an instance and disconnect from the instance, you can't connect to it again unless the AMI provides another way of logging in.

Verify the fingerprint of your key pair

To verify the fingerprint of your key pair, compare the fingerprint displayed on the **Key pairs** page in the Amazon EC2 console, or returned by the describe-key-pairs command, with the fingerprint that you generate using the private key on your local computer. These fingerprints should match.

When Amazon EC2 calculates a fingerprint, Amazon EC2 might append padding to the fingerprint with = characters. Other tools, such as **ssh-keygen**, might omit this padding.

If you're trying to verify the fingerprint of your Linux EC2 instance, not the fingerprint of your key pair, see Get the instance fingerprint.

How the fingerprints are calculated

Amazon EC2 uses different hash functions to calculate the fingerprints for RSA and ED25519 key pairs. Furthermore, for RSA key pairs, Amazon EC2 calculates the fingerprints differently using different hash functions depending on whether the key pair was created by Amazon EC2 or imported to Amazon EC2.

The following table lists the hash functions that are used to calculate the fingerprints for RSA and ED25519 key pairs that are created by Amazon EC2 and imported to Amazon EC2.

(Linux instances) Hash functions used to calculate fingerprints

Key pair source	RSA key pairs (Windows and Linux)	ED25519 key pairs (Linux)
Created by Amazon EC2	SHA-1	SHA-256
Imported to Amazon EC2	MD5[1]	SHA-256

[1] If you import a public RSA key to Amazon EC2, the fingerprint is calculated using an MD5 hash function. This is true regardless of how you created the key pair, for example, by using a third-party tool or by generating a new public key from an existing private key created using Amazon EC2.

When using the same key pair in different Regions

If you plan to use the same key pair to connect to instances in different AWS Regions, you must import the public key to all of the Regions in which you'll use it. If you use Amazon EC2 to create the key pair, you can Retrieve the public key material so that you can import the public key to the other Regions.

Note

- If you create an RSA key pair using Amazon EC2, and then you generate a public key from the Amazon EC2 private key, the imported public keys will have a different fingerprint than the original public key. This is because the fingerprint of the original RSA key created using Amazon EC2 is calculated using a SHA-1 hash function, while the fingerprint of the imported RSA keys is calculated using an MD5 hash function.
- For ED25519 key pairs, the fingerprints will be same regardless of whether they’re created by Amazon EC2 or imported to Amazon EC2, because the same SHA-256 hash function is used to calculate the fingerprint.

Generate a fingerprint from the private key

Use one of the following commands to generate a fingerprint from the private key on your local machine.

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 [How to install Linux on Windows with WSL](). 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 the key pair using Amazon EC2**

  Use the OpenSSL tools to generate a fingerprint as shown in the following examples.

  For RSA key pairs:

  ```
  openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt |
   openssl sha1 -c
  ```

  (Linux instances) For ED25519 key pairs:

  ```
  ssh-keygen -l -f path_to_private_key
  ```

- **(RSA key pairs only) If you imported the public key to Amazon EC2**

  You can follow this procedure regardless of how you created the key pair, for example, by using a third-party tool or by generating a new public key from an existing private key created using Amazon EC2

  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 imported the public key to Amazon EC2**

  Use **ssh-keygen** to generate the fingerprint as shown in the following examples.

  For RSA key pairs:

  ```
  ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER
   | openssl md5 -c
  ```

  (Linux instances) For ED25519 key pairs:

```
ssh-keygen -l -f path_to_private_key
```

Amazon EC2 security groups for your EC2 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 for the VPC. After you launch an instance, you can change its security groups.

Security is a shared responsibility between AWS and you. For more information, see Security in Amazon EC2. 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.

Pricing

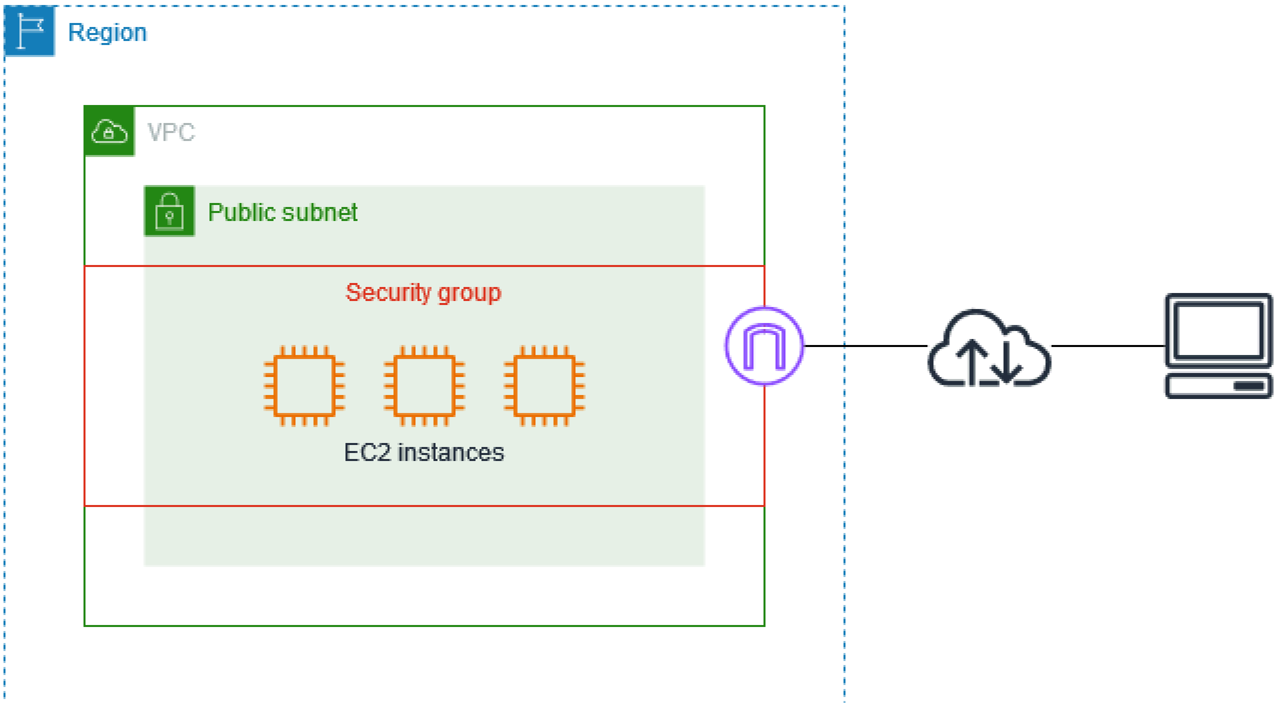
There is no additional charge for using security groups.

Contents

Overview

You can associate each instance with multiple security groups, and you can associate each security group with multiple instances. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. 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 rules from all security groups that are associated with the instance. For more information, see Security group rules in the *Amazon VPC User Guide*.

The following diagram shows a VPC with a subnet, an internet gateway, and a security group. The subnet contains EC2 instances. The security group is associated with the instances. The only traffic that reaches the instance is the traffic allowed by the security group rules. For example, if the security group contains a rule that allows SSH traffic from your network, then you can connect to your instance from your computer using SSH. If the security group contains a rule that allows all traffic from the resources associated with it, then each instance can receive any traffic sent from the other instances.



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. Also, responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. For more information, see Connection tracking.

Create a security group for your Amazon EC2 instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You can add rules to a security group that enable you to connect to your instance using SSH (Linux instances) or RDP (Windows instances). You can also add rules that allow client traffic, for example, HTTP and HTTPS traffic destined to a web server.

You can associate a security group with an instance when you launch the instance. When you add or remove rules from associated security groups, those changes are automatically applied to all instances to which you've associated the security group.

After you launch an instance, you can associate additional security groups. For more information, see Change the security groups for your Amazon EC2 instance.

You can add inbound and outbound security group rules when you create a security group or you can add them later on. For more information, see Configure security group rules. For examples of rules that you can add to a security group, see Security group rules for different use cases.

Considerations

- New security groups start with only an outbound rule that allows all traffic to leave the resource. You must add rules to enable any inbound traffic or to restrict the outbound traffic.
- When configuring a source for a rule that allows SSH or RDP access to your instances, do not allow access from anywhere, because it would allow this 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.
- If there is more than one rule for a specific port, Amazon EC2 applies the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address 203.0.113.1, and another rule that allows access to TCP port 22 from anywhere, then everyone has access to TCP port 22.
- You can associate multiple security groups with 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 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.
- Amazon EC2 blocks traffic on port 25 by default. For more information, see Restriction on email sent using port 25.

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. Enter a descriptive name and brief description for the security group. You can't change the name and description of a security group after it is created.
5. For **VPC**, choose the VPC in which you'll run your Amazon EC2 instances.
6. (Optional) To add inbound rules, choose **Inbound rules**. For each rule, choose **Add rule** and specify the protocol, port, and source. For example, to allow SSH traffic, choose **SSH** for **Type** and specify the public IPv4 address of your computer or network for **Source**.
7. (Optional) To add outbound rules, choose **Outbound rules**. For each rule, choose **Add rule** and specify the protocol, port, and destination. Otherwise, you can keep the default rule, which allows all outbound traffic.
8. (Optional) To add a tag, choose **Add new tag** and enter the tag key and value.
9. Choose **Create security group**.

AWS CLI

To create a security group

Use the following create-security-group command.

```
aws ec2 create-security-group \
    --group-name my-security-group \
    --description "my security group" \
    --vpc-id vpc-1234567890abcdef0
```

For examples that add rules, see the section called “Configure security group rules”.

PowerShell

To create a security group

Use the New-EC2SecurityGroup cmdlet.

```
New-EC2SecurityGroup `
```

```
    -GroupName my-security-group `
    -Description "my security group" `
    -VpcId vpc-1234567890abcdef0
```

For examples that add rules, see the section called “Configure security group rules”.

Change the security groups for your Amazon EC2 instance

You can specify security groups for your Amazon EC2 instances when you launch them. After you launch an instance, you can add or remove security groups. You can also add, remove, or edit security group rules for associated security groups at any time.

Security groups are associated with network interfaces. Adding or removing security groups changes the security groups associated with the primary network interface. You can also change the security groups associated with any secondary network interfaces. For more information, see Modify network interface attributes.

Tasks

- Add or remove security groups
- Configure security group rules

Add or remove security groups

After you launch an instance, you can add or remove security groups from the list of associated security groups. 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 traffic.

Requirements

- The instance must be in the `running` or `stopped` state.

Console

To change the security groups 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 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**.

AWS CLI

To change the security groups for an instance

Use the following modify-instance-attribute command.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

PowerShell

To change the security groups for an instance

Use the Edit-EC2InstanceAttribute cmdlet.

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

Configure security group rules

After you create a security group, you can add, update, and delete its security group rules. When you add, update, or delete a rule, the change is automatically applied to the resources that are associated with the security group.

For examples of rules that you can add to a security group, see Security group rules for different use cases.

Sources and destinations

You can specify the following as sources for inbound rules or destinations for outbound rules.

- **Custom –** A IPv4 CIDR block, and IPv6 CIDR block, another security group, or a prefix list.
- **Anywhere-IPv4** – The 0.0.0.0/0 IPv4 CIDR block.
- **Anywhere-IPv6** – The ::/0 IPv6 CIDR block.
- **My IP –** The public IPv4 address of your local computer.

> **Warning**
>
> If you add inbound rules for ports 22 (SSH) or 3389 (RDP), we strongly recommend that you authorize only the specific IP address or range of addresses that need access to your instances. If you choose **Anywhere-IPv4**, you allow traffic from all IPv4 addresses to access your instances using the specified protocol. If you choose **Anywhere-IPv6**, you allow traffic from all IPv6 addresses to access your instances using the specified protocol.

Console

To configure security group rules

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. To edit the inbound rules, choose **Edit inbound rules** from **Actions** or the **Inbound rules** tab.

    a. To add a rule, choose **Add rule** and enter the type, protocol, port, and source for the rule.

       If the type is 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 for you.

    b. To update a rule, change its protocol, description, and source as needed. However, you can't change the source type. For example, if the source is an IPv4 CIDR block, you can't specify an IPv6 CIDR block, a prefix list, or a security group.

    c. To delete a rule, choose its **Delete** button.

5. To edit the outbound rules, choose **Edit outbound rules** from **Actions** or the **Outbound rules** tab.

    a. To add a rule, choose **Add rule** and enter the type, protocol, port, and destination for the rule. You can also enter an optional description.

       If the type is 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 for you.

    b. To update a rule, change its protocol, description, and source as needed. However, you can't change the source type. For example, if the source is an IPv4 CIDR block, you can't specify an IPv6 CIDR block, a prefix list, or a security group.

    c. To delete a rule, choose its **Delete** button.

6. Choose **Save rules**.

AWS CLI

To add security group rules

Use the authorize-security-group-ingress command to add inbound rules. The following example allows inbound SSH traffic from the CIDR blocks in the specified prefix list.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds=[{PrefixListId=pl-
f8a6439156EXAMPLE}]'
```

Use the authorize-security-group-egress command to add outbound rules. The following example allows outbound TCP traffic on port 80 to instances with the specified security group.

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bb6EXAMPLE}]'
```

To remove security group rules

Use the following [revoke-security-group-ingress] command to remove an inbound rule.

```
aws ec2 revoke-security-group-egress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-09ed298024EXAMPLE
```

Use the following [revoke-security-group-egress] command to remove an outbound rule.

```
aws ec2 revoke-security-group-ingress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-0352250c1aEXAMPLE
```

To modify security group rules

Use the [modify-security-group-rules] command. The following example changes the IPv4 CIDR block of the specified security group rule.

```
aws ec2 modify-security-group-rules \
    --group id sg-1234567890abcdef0 \
    --security-group-rules
 'SecurityGroupRuleId=sgr-09ed298024EXAMPLE,SecurityGroupRule={IpProtocol=tcp,FromPort=80,To
```

PowerShell

To add security group rules

Use the [Grant-EC2SecurityGroupIngress] cmdlet to add inbound rules. The following example allows inbound SSH traffic from the CIDR blocks in the specified prefix list.

```
$plid = New-Object -TypeName Amazon.EC2.Model.PrefixListId
$plid.Id = "pl-f8a6439156EXAMPLE"
Grant-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=22; ToPort=22; PrefixListIds=$plid}
```

Use the [Grant-EC2SecurityGroupEgress] cmdlet to add outbound rules. The following example allows outbound TCP traffic on port 80 to instances with the specified security group.

```
$uigp = New-Object -TypeName Amazon.EC2.Model.UserIdGroupPair
$uigp.GroupId = "sg-0aad1c26bb6EXAMPLE"
```

```
Grant-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=80; ToPort=80; UserIdGroupPairs=
$uigp}
```

To remove security group rules

Use the Revoke-EC2SecurityGroupIngress cmdlet to remove inbound rules.

```
Revoke-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-09ed298024EXAMPLE
```

Use the Revoke-EC2SecurityGroupEgress cmdlet to remove outbound rules.

```
Revoke-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-0352250c1aEXAMPLE
```

To modify security group rules

Use the Edit-EC2SecurityGroupRule cmdlet. The following example changes the IPv4 CIDR block of the specified security group rule.

```
$sgrr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleRequest
$sgrr.IpProtocol = "tcp"
$sgrr.FromPort = 80
$sgrr.ToPort = 80
$sgrr.CidrIpv4 = "0.0.0.0/0"
$sgr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleUpdate
$sgr.SecurityGroupRuleId = "sgr-09ed298024EXAMPLE"
$sgr.SecurityGroupRule = $sgrr
Edit-EC2SecurityGroupRule  `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRule $sgr
```

Delete an Amazon EC2 security group

When you are finished with a security group that you created for use with your Amazon EC2 instances, you can delete it.

Requirements

- The security group can't be associated with an instance or network interface.
- The security group can't be referenced by a rule in another security group.

Console

To delete a security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. (Optional) To verify that your security group is not associated with an instance, do the following:
   a. In the navigation pane, choose **Security Groups**.
   b. Copy the ID of the security group to delete.
   c. In the navigation pane, choose **Instances**.
   d. In the search bar, add **Security group IDs equals** filter and paste the ID of the security group. If there are no results, then the security group is not associated with an instance. Otherwise, you must disassociate the security group before you can delete it.
3. In the navigation pane, choose **Security Groups**.
4. Select the security group and choose **Actions**, **Delete security groups**.
5. If you selected more than one security group, you are prompted for confirmation. If some of the security groups can't be deleted, we display the status of each security group, which indicates whether it will be deleted. To confirm deletion, enter **Delete**.
6. Choose **Delete**.

AWS CLI

To delete a security group

Use the following delete-security-group command.

```
aws ec2 delete-security-group --group-id sg-1234567890abcdef0
```

PowerShell

To delete a security group

Use the Remove-EC2SecurityGroup cmdlet.

```
Remove-EC2SecurityGroup -GroupId sg-1234567890abcdef0
```

Amazon EC2 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 a command such as netcat or similar 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 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, and the host sends the same type of traffic to your instance within 600 seconds, the security group for your instance accepts it regardless of inbound security group rules. The security group accepts it because it’s regarded as response traffic for the original traffic.

When you change a security group rule, its tracked connections are not immediately interrupted. The security group continues to allow packets until existing connections time out. To ensure that traffic is immediately interrupted, or that all traffic is subject to firewall rules regardless of the tracking state, you can use a network ACL for your subnet. Network ACLs are stateless and therefore do not automatically allow response traffic. Adding a network ACL that blocks traffic in either direction breaks existing connections. For more information, see Network ACLs in the *Amazon VPC User Guide*.

Note

Security groups have no effect on DNS traffic to or from the Route 53 Resolver, sometimes referred to as the 'VPC+2 IP address' (see What is Amazon Route 53 Resolver? in the

Amazon Route 53 Developer Guide), or the 'AmazonProvidedDNS' (see Work with DHCP option sets in the *Amazon Virtual Private Cloud User Guide*). If you wish to filter DNS requests through the Route 53 Resolver, you can enable Route 53 Resolver DNS Firewall (see Route 53 Resolver DNS Firewall in the *Amazon Route 53 Developer 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 any port (0-65535), then that flow of traffic is not tracked, unless it is part of an automatically tracked connection. The response traffic for an untracked flow is allowed based on the inbound or outbound rule that permits the response traffic, not based 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 an 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 SSH connection is not interrupted because it is tracked.

Automatically tracked connections

Connections made through the following are automatically tracked, even if the security group configuration does not otherwise require tracking:

- Egress-only internet gateways
- Global Accelerator accelerators
- NAT gateways
- Network Firewall firewall endpoints
- Network Load Balancers
- AWS PrivateLink (interface VPC endpoints)

- AWS Lambda (Hyperplane elastic network interfaces)

Connection tracking allowances

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. Use the `conntrack_allowance_available` network performance metric to determine the number of tracked connections still available for that instance type.

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 ENA settings on your EC2 instance](#).

With Elastic Load Balancing, 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.

Connection tracking performance considerations

Asymmetric routing, where traffic comes into an instance through one network interface and leaves through a different network interface, can reduce the peak performance that an instance can achieve if flows are tracked.

To maintain peak performance when connection tracking is enabled for your security groups, we recommend the following configuration:

- Avoid asymmetric routing topologies, if possible.
- Instead of using security groups for filtering, use network ACLs.
- If you must use security groups with connection tracking, configure the shortest idle connection tracking timeout possible. For more details on idle connection tracking timeout, see the following section.

For more information about performance tuning on the Nitro system, see [Nitro system considerations for performance tuning](#).

Idle connection tracking timeout

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. Connections that remain idle can lead to connection tracking exhaustion and cause connections not to be tracked and packets to be dropped. You can set the timeout for idle connection tracking on an Elastic network interface.

> **Note**
>
> This feature is available only with Nitro-based instances.

There are three configurable timeouts:

- **TCP established timeout**: Timeout (in seconds) for idle TCP connections in an established state.
  - Min: `60` seconds
  - Max: `432000` seconds
  - Default: `350` seconds for Nitrov6 instance types, excluding P6e-GB200. And `432000` seconds for other instance types, including P6e-GB200.
  - Recommended: Less than `432000` seconds
- **UDP timeout**: Timeout (in seconds) for idle UDP flows that have seen traffic only in a single direction or a single request-response transaction.
  - Min: `30` seconds
  - Max: `60` seconds
  - Default: `30` seconds
- **UDP stream timeout**: Timeout (in seconds) for idle UDP flows classified as streams which have seen more than one request-response transaction.
  - Min: `60` seconds
  - Max: `180` seconds
  - Default: `180` seconds

You may want to modify the default timeouts for any of the following cases:

- If you are monitoring tracked connections using Amazon EC2 network performance metrics, the *conntrack_allowance_exceeded* and *conntrack_allowance_available* metrics enable you to monitor dropped packets and tracked connection utilization to proactively manage EC2 instance capacity with scale up or out actions to help meet network connections demand before dropping packets. If you are observing *conntrack_allowance_exceeded* drops on your EC2 instances, you may benefit from setting a lower TCP established timeout to account for stale TCP/UDP sessions resulting from improper clients or network middle boxes.
- Typically, load balancers or firewalls have TCP Established idle timeout in the range of 60 to 90 minutes. If you are running workloads that are expected to handle a very high number of connections (greater than 100k) from appliances like network firewalls, configuring a similar timeout on an EC2 network interface is advised.
- If you are running a workload that utilizes an asymmetric routing topology, we recommend that you configure a TCP Established idle timeout of 60 seconds.
- If you are running workloads with high numbers of connections like DNS, SIP, SNMP, Syslog, Radius and other services that primarily use UDP to serve requests, setting the 'UDP-stream' timeout to 60s provides higher scale/performance for existing capacity and to prevent gray failures.
- For TCP/UDP connections through network load balancers (NLBs) and elastic load balancers (ELB), all connections are tracked. Idle timeout value for TCP flows is 350secs and UDP flows is 120 secs, and varies from interface level timeout values. You may want to configure timeouts at the network interface level to allow for more flexibility for timeout than the defaults for ELB/NLB.

You have the option to configure the connection tracking timeouts when you do the following:

- Create a network interface
- Modify network interface attributes
- Launch an EC2 instance
- Create an EC2 instance launch template

Example

In the following example, the security group has inbound rules that allow TCP and ICMP traffic, and outbound rules that allow all outbound traffic.

Inbound

Protocol type	Port number	Source
TCP	22 (SSH)	203.0.113.1/32
TCP	80 (HTTP)	0.0.0.0/0
TCP	80 (HTTP)	::/0
ICMP	All	0.0.0.0/0

Outbound

Protocol type	Port number	Destination
All	All	0.0.0.0/0
All	All	::/0

With a direct network connection to the instance or network interface, the tracking behavior is as follows:

- Inbound and outbound TCP traffic on port 22 (SSH) 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).
- Inbound and outbound TCP traffic on port 80 (HTTP) is not tracked, because the inbound and outbound rules allow traffic from all IP addresses.
- ICMP traffic is always tracked.

If you remove the outbound rule for IPv4 traffic, all inbound and outbound IPv4 traffic is tracked, including traffic on port 80 (HTTP). The same applies for IPv6 traffic if you remove the outbound rule for IPv6 traffic.

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
- Database server rules
- Rules to connect to instances from your computer
- Rules to connect to instances from an instance with the same security group
- Rules for ping/ICMP
- DNS server rules
- Amazon EFS rules
- Elastic Load Balancing rules

For instructions, see Create a security group and the section called "Configure security group rules".

Web server rules

The following inbound rules allow HTTP and HTTPS access from any IP address. If your VPC is enabled for IPv6, you can add rules to control inbound HTTP and HTTPS traffic from IPv6 addresses.

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows inbound HTTP access from any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows inbound HTTPS access from any IPv4 address
TCP	6	80 (HTTP)	::/0	Allows inbound HTTP access from any IPv6 address

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	443 (HTTPS)	::/0	Allows inbound HTTPS access from any IPv6 address

Database server rules

The following inbound rules are examples of rules you might add for database access, depending on what type of database you're running on your instance. For more information about Amazon RDS instances, see the Amazon RDS User Guide.

For the source IP, specify one of the following:

- A specific IP address or range of IP addresses (in CIDR block notation) in your local network
- A security group ID for a group of instances that access the database

Protocol type	Protocol number	Port	Notes
TCP	6	1433 (MS SQL)	The default port to access a Microsoft SQL Server database, for example, on an Amazon RDS instance
TCP	6	3306 (MYSQL/Aurora)	The default port to access a MySQL or Aurora database, for example, on an Amazon RDS instance
TCP	6	5439 (Redshift)	The default port to access an Amazon Redshift cluster database.
TCP	6	5432 (PostgreSQL)	The default port to access a PostgreSQL database, for example, on an Amazon RDS instance

Protocol type	Protocol number	Port	Notes
TCP	6	1521 (Oracle)	The default port to access an Oracle database, for example, on an Amazon RDS instance

You can optionally restrict outbound traffic from your database servers. For example, you might want to allow access to the internet for software updates, but restrict all other kinds of traffic. You must first remove the default outbound rule that allows all outbound traffic.

Protocol type	Protocol number	Port	Destination IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows outbound HTTP access to any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows outbound HTTPS access to any IPv4 address
TCP	6	80 (HTTP)	::/0	(IPv6-enabled VPC only) Allows outbound HTTP access to any IPv6 address
TCP	6	443 (HTTPS)	::/0	(IPv6-enabled VPC only) Allows outbound HTTPS access to any IPv6 address

Rules to connect to instances from your computer

To connect to your instance, your security group must have inbound rules that allow SSH access (for Linux instances) or RDP access (for Windows instances).

Protocol type	Protocol number	Port	Source IP
TCP	6	22 (SSH)	The public IPv4 address of your computer, or a range of IP addresses in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.
TCP	6	3389 (RDP)	The public IPv4 address of your computer, or a range of IP addresses in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.

Rules to connect to instances from an instance with the same security group

To allow instances that are associated with the same security group to communicate with each other, you must explicitly add rules for this.

Note

If you configure routes to forward the traffic between two instances in different subnets through a middlebox appliance, you must ensure that the security groups for both instances allow traffic to flow between the instances. The security group for each instance must reference the private IP address of the other instance, or the CIDR range of the subnet that contains the other instance, as the source. If you reference the security group of the other instance as the source, this does not allow traffic to flow between the instances.

The following table describes the inbound rule for a security group that enables associated instances to communicate with each other. The rule allows all types of traffic.

Protocol type	Protocol number	Ports	Source IP
-1 (All)	-1 (All)	-1 (All)	The ID of the security group, or the CIDR range of the subnet that contains the other instance (see note).

Rules for ping/ICMP

The **ping** command is a type of ICMP traffic. To ping your instance, you must add one of the following inbound ICMP rules.

Type	Protocol	Source		
Custom ICMP - IPv4	Echo request	The public IPv4 address of your computer, a specific IPv4 address, or an IPv4 or IPv6 address from anywhere.		
All ICMP - IPv4	IPv4 ICMP (1)	The public IPv4 address of your computer, a specific IPv4 address, or an IPv4 or IPv6 address from anywhere.		

To use the **ping6** command to ping the IPv6 address for your instance, you must add the following inbound ICMPv6 rule.

Type	Protocol	Source		
All ICMP - IPv6	IPv6 ICMP (58)	The IPv6 address of your computer, a specific IPv4 address, or an IPv4 or IPv6 address from anywhere.		

DNS server rules

If you've set up your EC2 instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

For the source IP, specify one of the following:

- An IP address or range of IP addresses (in CIDR block notation) in a network
- The ID of a security group for the set of instances in your network that require access to the DNS server

Protocol type	Protocol number	Port
TCP	6	53
UDP	17	53

Amazon EFS rules

If you're using an Amazon EFS file system with your Amazon EC2 instances, the security group that you associate with your Amazon EFS mount targets must allow traffic over the NFS protocol.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	2049 (NFS)	The ID of the security group	Allows inbound NFS access from resources (including the mount tar get) associated with this security group

To mount an Amazon EFS file system on your Amazon EC2 instance, you must connect to your instance. Therefore, the security group associated with your instance must have rules that allow inbound SSH from your local computer or local network.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	22 (SSH)	The IP address range of your local computer, or the range of IP addresses (in CIDR block notation) for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing rules

If you register your EC2 instances with a load balancer, the security group associated with your load balancer must allow communication with the instances. For more information, see the following in the Elastic Load Balancing documentation.

- Security groups for your Application Load Balancer
- Security groups for your Network Load Balancer
- Configure security groups for your Classic Load Balancer

NitroTPM for Amazon EC2 instances

Nitro Trusted Platform Module (NitroTPM) is a virtual device that is provided by the AWS Nitro System and conforms to the TPM 2.0 specification. It securely stores artifacts (such as passwords, certificates, or encryption keys) that are used to authenticate the instance. NitroTPM can generate keys and use them for cryptographic functions (such as hashing, signing, encryption, and decryption).

NitroTPM provides *measured boot*, a process where the bootloader and operating system create cryptographic hashes of every boot binary and combine them with the previous values in NitroTPM internal Platform Configuration Registers (PCRs). With measured boot, you can obtain signed PCR values from NitroTPM and use them to prove to remote entities the integrity of the instance's boot software. This is known as remote *attestation*.

With NitroTPM, keys and secrets can be tagged with a specific PCR value so that they can never be accessed if the value of the PCR, and thus the instance integrity, changes. This special form of conditional access is referred to as *sealing and unsealing*. Operating system technologies, like BitLocker, can use NitroTPM to seal a drive decryption key so that the drive can only be decrypted when the operating system has booted correctly and is in a known good state.

To use NitroTPM, you must select an Amazon Machine Image (AMI) that has been configured for NitroTPM support, and then use the AMI to launch Nitro-based instances. You can select one of Amazon's prebuilt AMIs or create one yourself.

Pricing

There is no additional cost for using NitroTPM. You pay only for the underlying resources that you use.

Contents

Requirements for using NitroTPM with Amazon EC2 instances

To launch an instance with NitroTPM enabled, you must meet the following requirements.

Topics

- AMIs
- Instance types
- Considerations

AMIs

The AMI must have NitroTPM enabled.

Linux AMIs

There are no preconfigured AMIs. You must configure your own AMI. For more information, see Enable a Linux AMI for NitroTPM.

Windows AMIs

To find an AWS Windows AMI that's preconfigured for NitroTPM and UEFI Secure Boot with Microsoft keys, see Find Windows Server AMIs configured with NitroTPM and UEFI Secure Boot in the *AWS Windows AMIs Reference*.

> **Note**
>
> **Operating system** — The AMI must include an operating system with a TPM 2.0 Command Response Buffer (CRB) driver. Most current operating systems include a TPM 2.0 CRB driver.
> **UEFI boot mode** — The AMI must be configured for UEFI boot mode. For more information, see UEFI Secure Boot for Amazon EC2 instances.

Instance types

You must use one of the following virtualized instance types:

- **General purpose**: M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6a, M6g, M6gd, M6i, M6id, M6idn, M6in, M7a, M7g, M7gd, M7i, M7i-flex, M8g, M8gd, M8i, M8i-flex, T3, T3a, T4g

- **Compute optimized**: C5, C5a, C5ad, C5d, C5n, C6a, C6g, C6gd, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i, C7i-flex, C8g, C8gd, C8gn
- **Memory optimized**: R5, R5a, R5ad, R5b, R5d, R5dn, R5n, R6a, R6g, R6gd, R6i, R6idn, R6in, R6id, R7a, R7g, R7gd, R7i, R7iz, R8g, R8gd, R8i, R8i-flex, U7i-6tb, U7i-8tb, U7i-12tb, U7in-16tb, U7in-24tb, U7in-32tb, X2idn, X2iedn, X2iezn, X8g, z1d
- **Storage optimized**: D3, D3en, I3en, I4i, I7i, I7ie, I8g, I8ge
- **Accelerated computing**: F2, G4dn, G5, G6, G6e, G6f, Gr6, Gr6f, Inf1, Inf2, P5, P5e, P5en, P6-B200, Trn2, Trn2u
- **High-performance computing**: Hpc6a, Hpc6id

Considerations

The following considerations apply when using NitroTPM:

- After you launch an instance using an AMI with NitroTPM enabled, if you want to change the instance type, the new instance type that you choose must also support NitroTPM.
- BitLocker volumes that are encrypted with NitroTPM-based keys can only be used on the original instance.
- The NitroTPM state is not displayed in the Amazon EC2 console.
- The NitroTPM state is not included in Amazon EBS snapshots.
- The NitroTPM state is not included in VM Import/Export images.
- NitroTPM is not supported on AWS Outposts., Local Zones, or Wavelength Zones.

Enable a Linux AMI for NitroTPM

To enable NitroTPM for an instance, you must launch the instance using an AMI with NitroTPM enabled. You must configure your Linux AMI with NitroTPM support when you register it. You can't configure NitroTPM support later on.

For the list of Windows AMIs that are preconfigured for NitroTPM support, see Requirements for using NitroTPM with Amazon EC2 instances.

You must create an AMI with NitroTPM configured by using the RegisterImage API. You can't use the Amazon EC2 console or VM Import/Export.

To enable a Linux AMI for NitroTPM

1. Launch a temporary instance with your required Linux AMI. Note the ID of its root volume, which you can find in the console on the **Storage** tab for the instance.
2. After the instance reaches the `running` state, create a snapshot of the instance's root volume. For more information, see Create a snapshot of an EBS volume.
3. Register the snapshot you created as an AMI. In the block device mapping, specify the snapshot that you created for the root volume.

   The following is an example register-image command. For `--tpm-support`, specify `v2.0`. For `--boot-mode`, specify `uefi`.

```
aws ec2 register-image \
    --name my-image \
    --boot-mode uefi \
    --architecture x86_64 \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0abcdef1234567890} \
    --tpm-support v2.0
```

   The following is an example for the Register-EC2Image cmdlet.

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
    -Name my-image `
    -Architecture "x86_64" `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
    -BootMode Uefi `
    -TpmSupport V20
```

4. Terminate the temporary instance that you launched in step 1.

Verify that an AMI is enabled for NitroTPM

To enable NitroTPM for an instance, you must launch the instance using an AMI with NitroTPM enabled. You can describe an image to verify that it is enabled for NitroTPM. If you are the AMI owner, you can describe the `tpmSupport` image attribute.

The Amazon EC2 console does not display `TpmSupport`.

AWS CLI

To verify that NitroTPM is enabled

Use the describe-images command.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].TpmSupport
```

If NitroTPM is enabled for the AMI, the output is as follows. If TPM is not enabled, the output is empty.

```
[
    "v2.0"
]
```

Alternatively, if you are the AMI owner, you can use the describe-image-attribute command with the `tpmSupport` attribute.

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

The following is example output.

```
{
    "ImageId": "ami-0abcdef1234567890",
    "TpmSupport": {
        "Value": "v2.0"
    }
}
```

To find AMIs with NitroTPM enabled

The following example lists the IDs of the AMIs that you own with NitroTPM enabled.

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

```
    --filters Name=tpm-support,Values=v2.0 \
    --query Images[].ImageId
```

PowerShell

To verify that NitroTPM is enabled

Use the Get-EC2Image cmdlet.

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | Select TpmSupport
```

If NitroTPM is enabled for the AMI, the output is as follows. If TPM is not enabled, the output is empty.

```
TpmSupport
----------
v2.0
```

Alternatively, if you are the AMI owner, you can use the Get-EC2ImageAttribute cmdlet with the `tpmSupport` attribute.

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute tpmSupport
```

To find AMIs with NitroTPM enabled

The following example lists the IDs of the AMIs that you own with NitroTPM enabled.

```
Get-EC2Image `
    -Owner self `
    -Filter @{Name="tpm-support; Values="v2.0"} | Select ImageId
```

Enable or stop using NitroTPM on an Amazon EC2 instance

You can enable an Amazon EC2 instance for NitroTPM only at launch. Once an instance is enabled for NitroTPM, you can't disable it. If you no longer need to use NitroTPM, you must configure the operating system to stop using it.

Topics

Launch an instance with NitroTPM enabled

When you launch an instance with the prerequisites, NitroTPM is automatically enabled on the instance. You can enable NitroTPM on an instance only at launch. For information about launching an instance, see Launch an Amazon EC2 instance.

Stop using NitroTPM on an instance

After launching an instance with NitroTPM enabled, you can't disable NitroTPM for the instance. However, you can configure the operating system to stop using NitroTPM by disabling the TPM 2.0 device driver on the instance by using the following tools:

- For **Linux instances**, use tpm-tools.
- For **Windows instances**, use the TPM management console (tpm.msc).

For more information about disabling the device driver, see the documentation for your operating system.

Verify that an Amazon EC2 instance is enabled for NitroTPM

You can verify whether an Amazon EC2 instance is enabled for NitroTPM. If NitroTPM support is enabled on the instance, the command returns `"v2.0"`. Otherwise, the `TpmSupport` field is not present in the output.

The Amazon EC2 console does not display the `TpmSupport` field.

AWS CLI

To verify whether an instance is enabled for NitroTPM

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].TpmSupport
```

PowerShell

To verify whether an instance is enabled for NitroTPM

Use the Get-EC2Instance cmdlet.

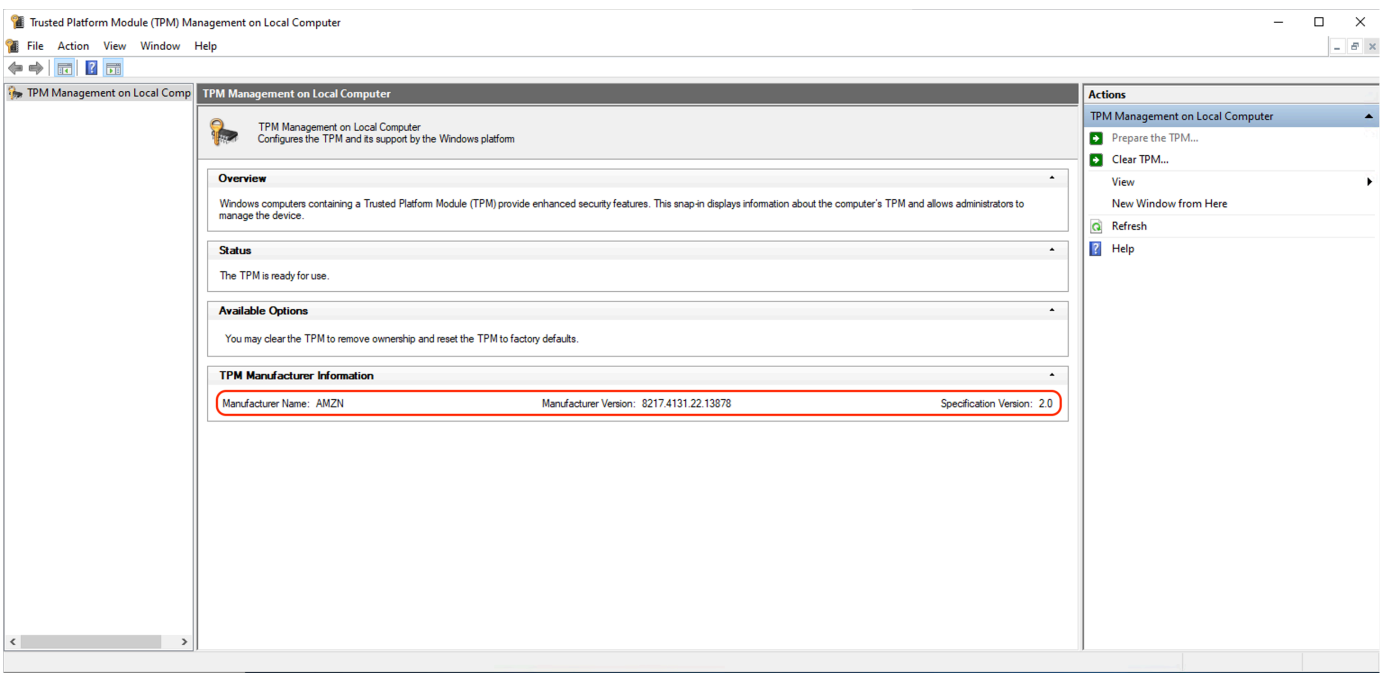
```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.TpmSupport
```

Verify NitroTPM access on your Windows instance

(Windows instances only) To verify whether the NitroTPM is accessible to Windows

1. Connect to your EC2 Windows instance.
2. On the instance, run the tpm.msc program.

   The **TPM Management on Local Computer** window opens.
3. Check the **TPM Manufacturer Information** field. It contains the manufacturer's name and the version of the NitroTPM on the instance.



Retrieve the public endorsement key for an EC2 instance

You can securely retrieve the public endorsement key for an instance at any time.

AWS CLI

To retrieve the public endorsement key for an instance

Use the get-instance-tpm-ek-pub command.

Example 1

The following example gets the `rsa-2048` public endorsement key in `tpmt` format for the specified instance.

```
aws ec2 get-instance-tpm-ek-pub \
    --instance-id i-1234567890abcdef0 \
    --key-format tpmt \
    --key-type rsa-2048
```

The following is the example output.

```
{
    "InstanceId": "i-01234567890abcdef",
    "KeyFormat": "tpmt",
    "KeyType": "rsa-2048",
    "KeyValue": "AAEACwADALIAIINxl2dEhLEXAMPLEUal1yT9UtduBlILZPKh2hszFGmqAAYAgABDA
    EXAMPLEAAABAOiRd7WmgtdGNoV1h/AxmW+CXExblG8pEUfNm0LOLiYnEXAMPLERqApiFa/UhvEYqN4
    Z7jKMD/usbhsQaAB1gKA5RmzuhSazHQkax7EXAMPLEzDthlS7HNGuYn5eG7qnJndRcakS+iNxT8Hvf
    0S1ZtNuItMs+Yp4SO6aU28MT/JZkOKsXIdMerY3GdWbNQz9AvYbMEXAMPLEPyHfzgVO0QTTJVGdDxh
    vxtXCOu9GYf0crbjEXAMPLEd4YTbWdDdgOKWF9fjzDytJSDhrLAOUctNzHPCd/92l5zEXAMPLEOIFA
    Ss50C0/802c17W2pMSVHvCCa9lYCiAfxH/vYKovAAE="
}
```

Example 2

The following example gets the `rsa-2048` public endorsement key in `der` format for the specified instance.

```
aws ec2 get-instance-tpm-ek-pub \
    --instance-id i-1234567890abcdef0 \
    --key-format der \
    --key-type rsa-2048
```

The following is the example output.

```
{
```

```
    "InstanceId": "i-1234567890abcdef0",
    "KeyFormat": "der",
    "KeyType": "rsa-2048",
    "KeyValue": "MIIBIjANBgEXAMPLEw0BAQEFAAOCAQ8AMIIBCgKCAQEA6JF3taEXAMPLEXWH8DGZb4
    JcTFuUbykRR82bQs4uJifaKSOv5NGoEXAMPLEG8Rio3hnuMowP+6xuGxBoAHWAoDlGbO6FJrMdEXAMP
    LEnYUHvMO2GVLsc0a5ifl4buqcmd1FxqRL6I3FPwe9/REXAMPLE0yz5inhI7ppTbwxP8lmQ4qxch0x6
    tjcZ1Zs1DP0EXAMPLERUYLQ/Id/OBU7RBNMlUZ0PGG/G1cI670Zh/RytuOdx9iEXAMPLEtZ0N2A4pYX
    1+PMPK0lIOGssA5Ry03Mc8J3/3aXnOD2/ASRQ4gUBKznQLT/zTZEXAMPLEJUe8IJr2VgKIB/Ef+9gqi
    8AAQIDAQAB"
}
```

PowerShell

To retrieve the public endorsement key for an instance

Use the Get-EC2InstanceTpmEkPub cmdlet.

Example 1

The following example gets the `rsa-2048` public endorsement key in `tpmt` format for the specified instance.

```
Get-EC2InstanceTpmEkPub `
    -InstanceId i-1234567890abcdef0 `
    -KeyFormat tpmt `
    -KeyType rsa-2048
```

Example 2

The following example gets the `rsa-2048` public endorsement key in `der` format for the specified instance.

```
Get-EC2InstanceTpmEkPub `
    -InstanceId i-1234567890abcdef0 `
    -KeyFormat der `
    -KeyType rsa-2048
```

Credential Guard for Windows instances

The AWS Nitro System supports Credential Guard for Amazon Elastic Compute Cloud (Amazon EC2) Windows instances. Credential Guard is a Windows virtualization-based security (VBS) feature that enables the creation of isolated environments to protect security assets, such as Windows user credentials and code integrity enforcement, beyond Windows kernel protections. When you run EC2 Windows instances, Credential Guard uses the AWS Nitro System to protect Windows login credentials from being extracted from the operating system's memory.

Contents

Prerequisites

Your Windows instance must meet the following prerequisites to utilize Credential Guard.

Amazon Machine Images (AMIs)

The AMI must be preconfigured to enable NitroTPM and UEFI Secure Boot. For more information on supported AMIs, see the section called “Requirements”.

Memory integrity

Memory integrity, also known as *hypervisor-protected code integrity (HVCI)* or *hypervisor enforced code integrity*, isn't supported. Before you turn on Credential Guard, you must ensure this feature is disabled. For more information, see Disable memory integrity.

Instance types

The following instance types support Credential Guard across all sizes unless noted otherwise: `C5`, `C5d`, `C5n`, `C6i`, `C6id`, `C6in`, `C7i`, `C7i-flex`, `M5`, `M5d`, `M5dn`, `M5n`, `M5zn`, `M6i`, `M6id`, `M6idn`, `M6in`, `M7i`, `M7i-flex`, `R5`, `R5b`, `R5d`, `R5dn`, `R5n`, `R6i`, `R6id`, `R6idn`, `R6in` `R7i`, `R7iz`, `T3`.

> **Note**
>
> - Though NitroTPM has some required instance types in common, the instance type must be one of the preceding instance types to support Credential Guard.
> - Credential Guard isn't supported for:
>   - Bare metal instances.

- The following instance types: C7i.48xlarge, M7i.48xlarge, and R7i.48xlarge.

For more information about instance types, see the Amazon EC2 Instance Types Guide.

Launch a supported instance

You can use the Amazon EC2 console or AWS Command Line Interface (AWS CLI) to launch an instance which can support Credential Guard. You will need a compatible AMI ID for launching your instance which is unique for each AWS Region.

> **Tip**
>
> You can use the following link to discover and launch instances with compatible Amazon provided AMIs in the Amazon EC2 console: https://console.aws.amazon.com/ec2/v2/home?#Images:visibility=public-images;v=3;search=:TPM-Windows_Server;ownerAlias=amazon

Console

To launch an instance

Follow the steps to launch an instance, specifying a supported instance type and a preconfigured Windows AMI.

AWS CLI

To launch an instance

Use the run-instances command to launch an instance using a supported instance type and preconfigured Windows AMI.

```
aws ec2 run-instances \
    --image-id resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-
English-Full-Base \
    --instance-type c6i.large \
    --region us-east-1 \
    --subnet-id subnet-0abcdef1234567890
```

```
    --key-name key-name
```

PowerShell

To launch an instance

Use the New-EC2Instance command to launch an instance using a supported instance type and preconfigured Windows AMI.

```
New-EC2Instance `
    -ImageId resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-
English-Full-Base `
    -InstanceType c6i.large `
    -Region us-east-1 `
    -SubnetId subnet-0abcdef1234567890 `
    -KeyName key-name
```

Disable memory integrity

You can use the Local Group Policy Editor to disable memory integrity in supported scenarios. The following guidance can be applied for each configuration setting under **Virtualization Based Protection of Code Integrity**:

- **Enabled without lock** – Modify the setting to **Disabled** to disable memory integrity.
- **Enabled with UEFI lock** – Memory integrity has been enabled with UEFI lock. Memory integrity can't be disabled once it has been enabled with UEFI lock. We recommend creating a new instance with memory integrity disabled and terminating the unsupported instance if it's not in use.

To disable memory integrity with the Local Group Policy Editor

1. Connect to your instance as a user account with administrator privileges using the Remote Desktop Protocol (RDP). For more information, see the section called "Connect using an RDP client".
2. Open the Start menu and search for **cmd** to start a command prompt.
3. Run the following command to open the Local Group Policy Editor: `gpedit.msc`
4. In the Local Group Policy Editor, choose **Computer Configuration**, **Administrative Templates**, **System**, **Device Guard**.

5. Select **Turn On Virtualization Based Security**, then select **Edit policy setting**.
6. Open the settings drop-down for **Virtualization Based Protection of Code Integrity**, choose **Disabled**, then choose **Apply**.
7. Reboot the instance to apply the changes.

Turn on Credential Guard

After you have launched a Windows instance with a supported instance type and compatible AMI, and confirmed that memory integrity is disabled, you can turn on Credential Guard.

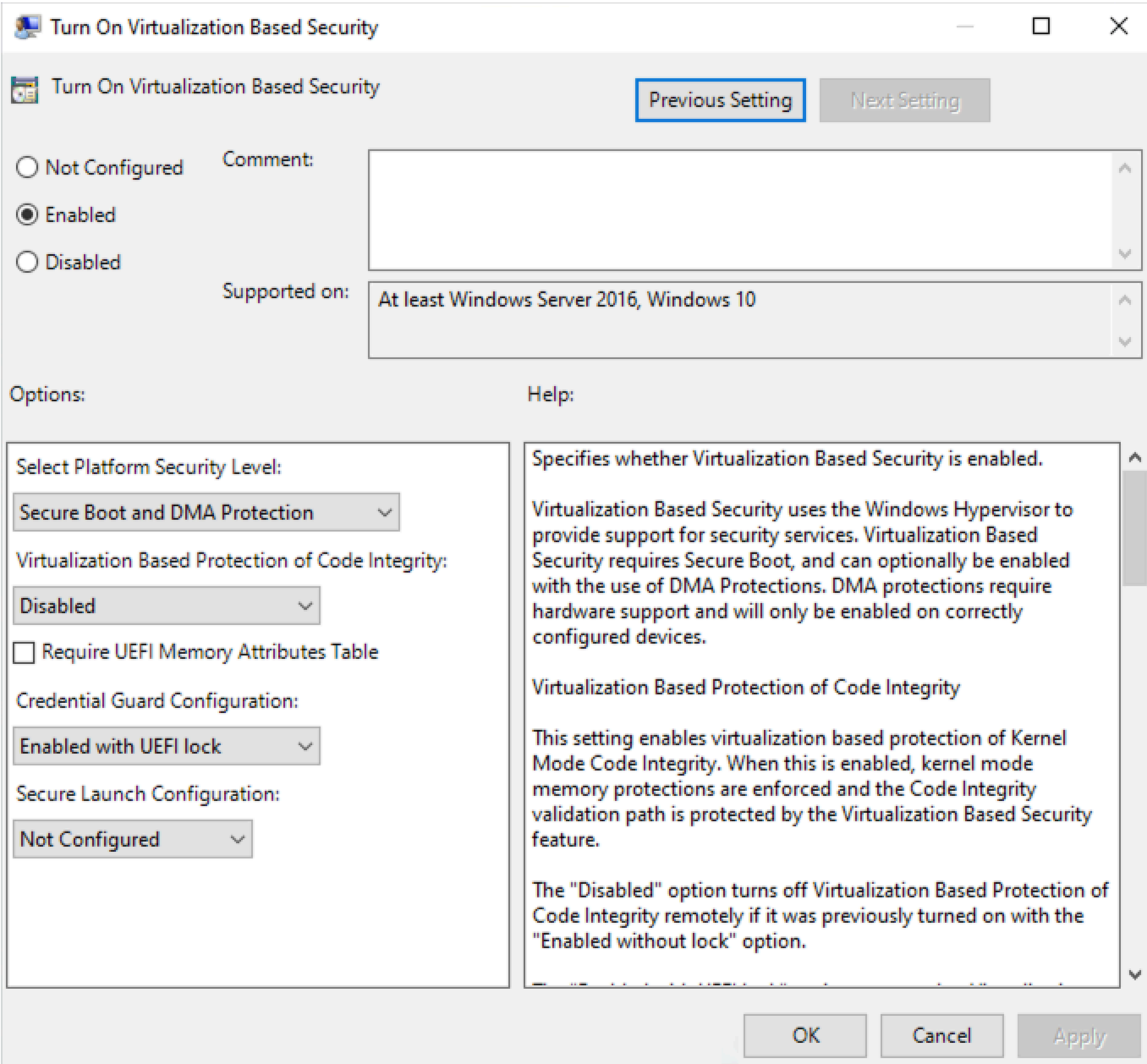
> **Important**
>
> Administrator privileges are required to perform the following steps to turn on Credential Guard.

To turn on Credential Guard

1. Connect to your instance as a user account with administrator privileges using the Remote Desktop Protocol (RDP). For more information, see the section called “Connect using an RDP client”.
2. Open the Start menu and search for **cmd** to start a command prompt.
3. Run the following command to open the Local Group Policy Editor: `gpedit.msc`
4. In the Local Group Policy Editor, choose **Computer Configuration**, **Administrative Templates**, **System**, **Device Guard**.
5. Select **Turn On Virtualization Based Security**, then select **Edit policy setting**.
6. Choose **Enabled** within the **Turn On Virtualization Based Security** menu.
7. For **Select Platform Security Level**, choose **Secure Boot and DMA Protection**.
8. For **Credential Guard Configuration**, choose **Enabled with UEFI lock**.

   > **Note**
   >
   > The remaining policy settings are not required to enable Credential Guard and can be left as **Not Configured**.

The following image displays the VBS settings configured as described previously:



9. Reboot the instance to apply the settings.

Verify that Credential Guard is running

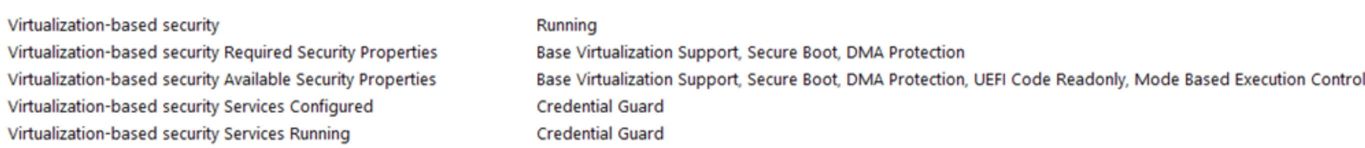
You can use the Microsoft System Information (`Msinfo32.exe`) tool to confirm that Credential Guard is running.

> **Important**
>
> You must first reboot the instance to finish applying the policy settings required to enable Credential Guard.

To verify Credential Guard is running

1. Connect to your instance using the Remote Desktop Protocol (RDP). For more information, see the section called “Connect using an RDP client”.
2. Within the RDP session to your instance, open the Start menu and search for **cmd** to start a command prompt.
3. Open System Information by running the following command: `msinfo32.exe`
4. The Microsoft System Information tool lists the details for VBS configuration. Next to Virtualization-based security Services, confirm that **Credential Guard** appears as **Running**.

   The following image displays VBS is running as described previously:



Virtualization-based security	Running
Virtualization-based security Required Security Properties	Base Virtualization Support, Secure Boot, DMA Protection
Virtualization-based security Available Security Properties	Base Virtualization Support, Secure Boot, DMA Protection, UEFI Code Readonly, Mode Based Execution Control
Virtualization-based security Services Configured	Credential Guard
Virtualization-based security Services Running	Credential Guard

Access Amazon EC2 using an interface VPC endpoint

You can improve the security posture of your VPC by creating a private connection between resources in your VPC and the Amazon EC2 API. You can access the Amazon EC2 API as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. EC2 instances in your VPC don't need public IP addresses to access the Amazon EC2 API.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

Contents

Create an interface VPC endpoint

Create an interface 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 Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

Create an endpoint policy

An endpoint policy is an IAM resource that you can attach to your interface endpoint. The default endpoint policy allows full access to the Amazon EC2 API through the interface endpoint. To control the access allowed to the Amazon EC2 API from your VPC, attach a custom endpoint policy to the interface endpoint.

An endpoint policy specifies the following information:

- The principals 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 Control access to services using endpoint policies in the *AWS PrivateLink 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": "*"
    },
    {
        "Action": [
            "ec2:CreateVolume"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    },
    {
        "Action": [
            "ec2:RunInstances"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    }]
}
```

Storage options for your Amazon EC2 instances

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.

Block storage

- Amazon EBS — Amazon EBS provides durable, block-level storage volumes that you can attach and detach from your instances. You can attach multiple EBS volumes to an instance. An EBS volume persists independently from the life of its associated instance. You can encrypt your EBS volumes. To keep a backup copy of your data, you can create snapshots from your EBS volumes. Snapshots are stored in Amazon S3. You can create an EBS volume from a snapshot.
- Instance store temporary block storage for EC2 instances — Instance store provides temporary block-level storage for instances. The number, size, and type of instance store volumes are determined by the instance type and instance size. 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.

Object storage

- 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.

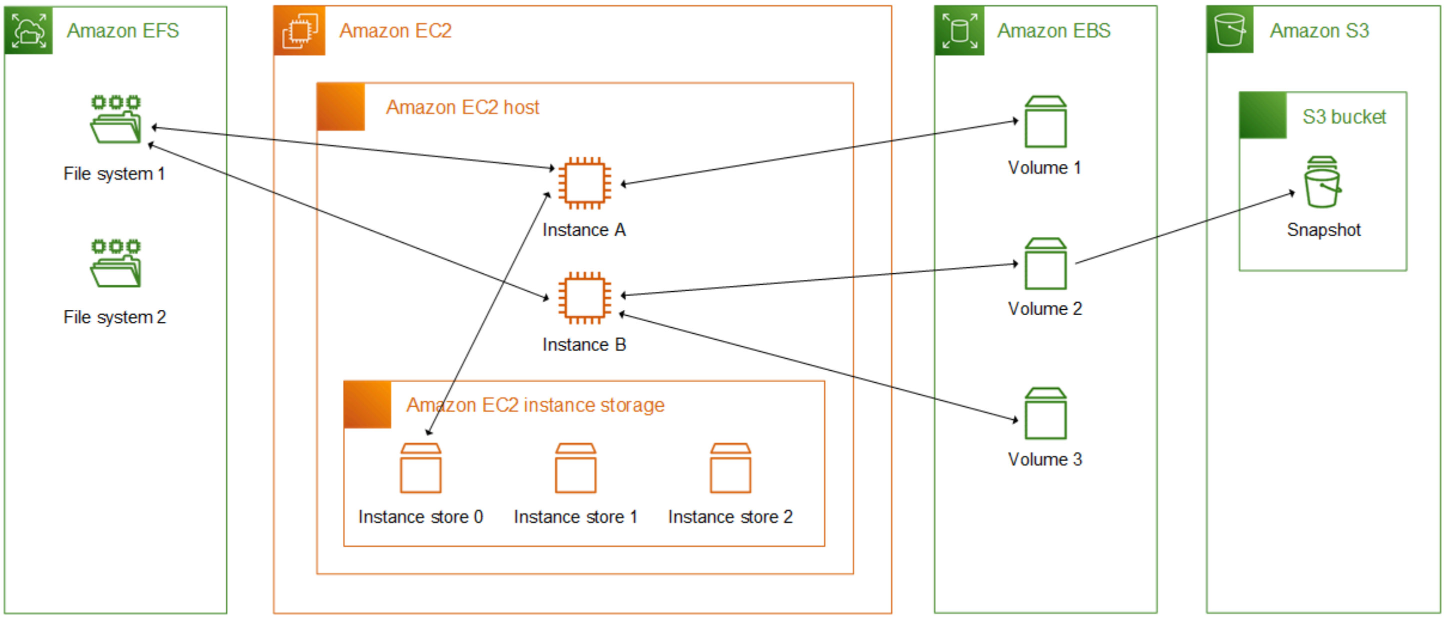
File storage

- Amazon EFS (Linux instances only) — Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances.
- Amazon FSx — With Amazon FSx, you can launch, run, and scale feature-rich, high-performance file systems in the cloud. Amazon FSx is a fully-managed service that supports a wide range of workloads. You can choose between these widely-used file systems: Lustre, NetApp ONTAP, OpenZFS, and Windows File Server.

File caching

- Use Amazon File Cache with Amazon EC2 instances — Amazon File Cache provides temporary, high-performance cache on AWS for processing file data. The cache provides read and write data access to compute workloads on Amazon EC2 with sub-millisecond latencies, up to hundreds of GB/s of throughput, and up to millions of IOPS.

The following figure shows the relationship between these storage options and your instance.



AWS Storage pricing

Open AWS Pricing, scroll to **Pricing for AWS products** and select **Storage**. Choose the storage product to open its pricing page.

Amazon EBS persistent block storage for Amazon EC2 instances

Amazon Elastic Block Store (Amazon EBS) provides scalable, high-performance block storage resources that can be used with Amazon EC2 instances. With Amazon EBS, you can create and manage the following block storage resources:

- **Amazon EBS volumes** — These are storage volumes that you attach to Amazon EC2 instances. After you attach a volume to an instance, you can use it in the same way you would use block storage. The instance can interact with the volume just as it would with a local drive.

- **Amazon EBS snapshots** — These are point-in-time backups of Amazon EBS volumes that persist independently from the volume itself. You can create snapshots to back up the data on your Amazon EBS volumes. You can then restore new volumes from those snapshots at any time.

You can create and attach EBS volumes to an instance during launch, and you can create and attach EBS volumes to an instance at any time after launch. You can also increase the size or performance of your EBS volumes without detaching the volume or restarting your instance.

You can create EBS snapshots from an EBS volume at any time after creation. You can use EBS snapshots to back up the data stored on your volumes. You can then use those snapshots to instantly restore volumes, or to migrate data across AWS accounts, AWS Regions, or Availability Zones. You can use Amazon Data Lifecycle Manager or AWS Backup to automate the creation, retention, and deletion of your EBS snapshots.

A *managed EBS volume* is managed by a service provider, such as Amazon EKS Auto Mode. You can't directly modify the settings of a managed EBS volume. Managed EBS volumes are identified by a **true** value in the **Managed** field. For more information, see Amazon EC2 managed instances.

For more information about working with volumes and snapshots, see the Amazon EBS User Guide.

Amazon EBS volume limits for Amazon EC2 instances

The maximum number of Amazon EBS volumes that you can attach to an instance depends on the instance type and instance size. If you exceed the volume attachment limit for an instance, the attachment request fails with the `AttachmentLimitExceeded` error.

When considering how many volumes to attach to your instance, you should consider whether you need increased I/O bandwidth or increased storage capacity.

Bandwidth versus capacity

For consistent and predictable bandwidth use cases, use Amazon EBS-optimized instances with General Purpose SSD volumes or Provisioned IOPS SSD volumes. For maximum performance, match the IOPS you have provisioned for your volumes with the bandwidth available for your instance type.

For RAID configurations, you might 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.

Contents

Volume limits for instances built on the Nitro System

The volume limits for instances built on the Nitro System depend on the instance type. Some Nitro instance types have a **dedicated EBS volume limit**, while most have a **shared volume limit**. To view the volume attachement limits for each instance type, see the following:

- Amazon EBS specifications – General purpose
- Amazon EBS specifications – Compute optimized
- Amazon EBS specifications – Memory optimized
- Amazon EBS specifications – Storage optimized
- Amazon EBS specifications – Accelerated computing
- Amazon EBS specifications – High-performance computing
- Amazon EBS specifications – Previous generation

Dedicated EBS volume limit

The following Nitro instance types have a dedicated EBS volume limit that varies depending on instance size. The limit is not shared with other device attachments. In other words, you can attach any number of EBS volumes up to the volume attachment limit, regardless of the number of attached devices, such as NVMe instance store volumes and network interfaces.

- **General purpose:** M7a | M7i | M7i-flex | M8g | M8gd | M8i | M8i-flex
- **Compute optimized:** C7a | C7i | C7i-flex | C8g | C8gd | C8gn
- **Memory optimized:** R7a | R7i | R7iz | R8g | R8gd | R8i | R8i-flex | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X8g
- **Storage optimized:** I7i | I7ie | I8g | I8ge

- **Accelerated computing:** F2 | G6 | G6e | G6f | Gr6 | Gr6f | P4d | P4de | P5 | P5e | P5en | P6-B200 | P6e-GB200 | Trn2 | Trn2u
- **High performance computing:** Hpc7a

Shared EBS volume limit

All other Nitro instance types (not listed in Dedicated EBS volume limit) have a volume attachment limit that is shared between Amazon EBS volumes, network interfaces, and NVMe instance store volumes. You can attach any number of Amazon EBS volumes up to that limit, less the number of attached network interfaces and NVMe instance store volumes. Keep in mind that every instance must have at least one network interface, and that NVMe instance store volumes are automatically attached at launch.

Most Nitro instances support a maximum of 28 attachments. The following examples demonstrate how to calculate how many EBS volumes you can attach.

Examples

- With an `m5.xlarge` instance with only the primary network interface, you can attach 27 EBS volumes.

  28 volumes - 1 network interface = 27
- With an `m5.xlarge` instance with two additional network interfaces, you can attach 25 EBS volumes.

  28 volumes - 3 network interfaces = 25
- With an `m5d.xlarge` instance with two additional network interfaces, you can attach 24 EBS volumes.

  28 volumes - 3 network interfaces - 1 NVMe instance store volume = 24

Volume limits for Xen-based instances

The volume limits for Xen-based instances, such as T2, depend on the operating system.

For more information, see Xen-based instances.

Linux instances

Attaching more than 40 volumes to a Xen-based Linux instance can cause boot failures. This number includes the root volume, plus any attached instance store volumes and Amazon EBS volumes.

If you experience boot problems on an instance with a large number of volumes, stop the instance, detach any volumes that are not essential to the boot process, start the instance, and then reattach the volumes after the instance is running.

> **Important**
>
> Attaching more than 40 volumes to a Xen-based Linux instance is supported on a best effort basis only and is not guaranteed.

Windows instances

The following table shows the volume limits for Xen-based Windows instances based on the driver used. That these numbers include the root volume, plus any attached instance store volumes and Amazon EBS volumes.

Driver	Volume Limit
AWS PV	26
Citrix PV	26
Red Hat PV	17

We recommend that you do not that attach more than 26 volumes to a Xen-based Windows instance 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 the section called “Upgrade PV drivers”.

> **Important**
>
> Attaching more than the following number of volumes to a Xen-based Windows instance is supported on a best effort basis only and is not guaranteed.

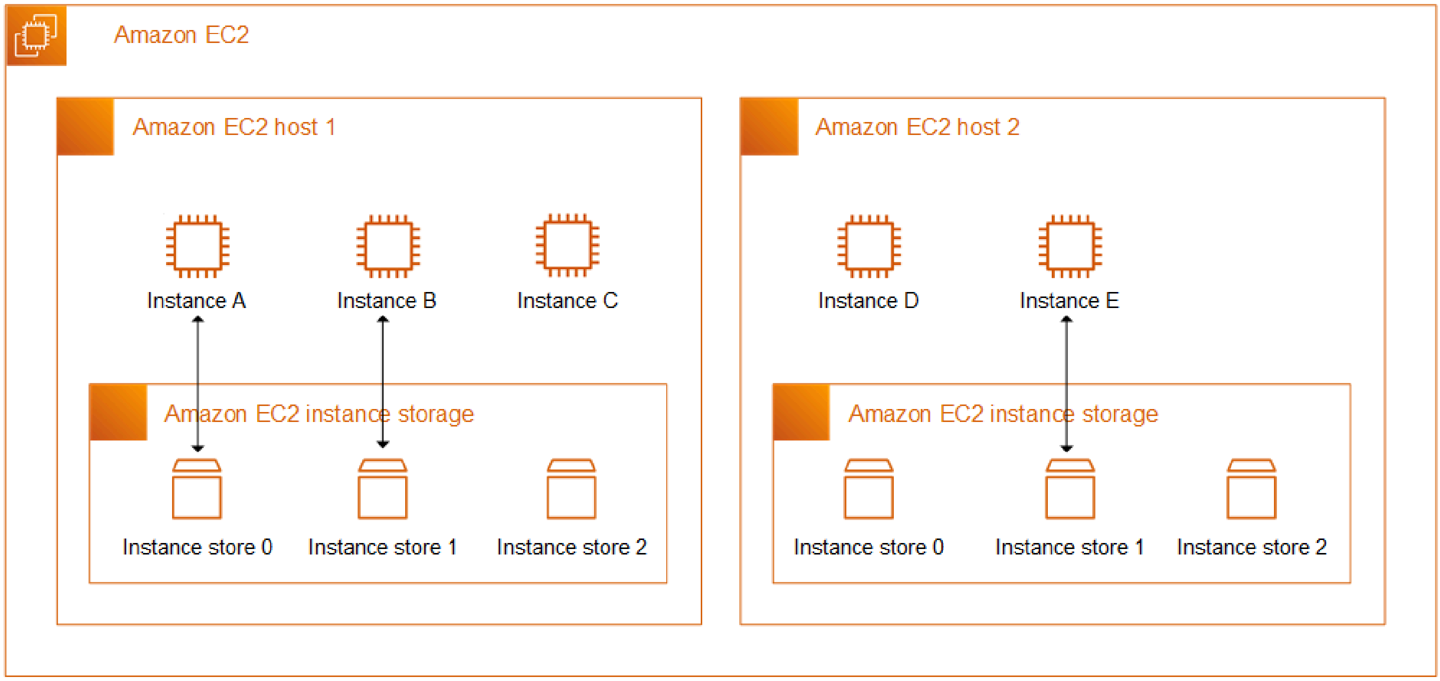
For more information about how device names are related to volumes, see How volumes are attached and mapped for Amazon EC2 Windows instances.

Instance store temporary block storage for EC2 instances

An *instance store* provides temporary block-level storage for your EC2 instance. This storage is provided by 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. It can also be used to store temporary data that you replicate 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 and instance size. For example, not every instance type provides instance store volumes. For more information, see Instance store volume limits for EC2 instances.

The virtual devices for instance store volumes are given virtual device names in order from `ephemeral0` to `ephemeral23`. For example, with an instance type that supports one instance store volume, the virtual device name of the one volume is `ephemeral0`. With an instance type that supports four instance store volumes, the virtual device names of the four volumes are as follows: `ephemeral0`, `ephemeral1`, `ephemeral2` and `ephemeral3`.



Instance store pricing

There is no additional charge to use the instance store volumes provided for your instance. Instance store volumes are included as part of the usage cost of the instance.

Contents

Data persistence for Amazon EC2 instance store volumes

Instance store volumes are attached only at instance launch. You can't attach instance store volumes after launch. You can’t detach an instance store volume from one instance and attach it to a different instance.

An instance store volume exists only during the lifetime of the instance to which it is attached. You can’t configure an instance store volume to persist beyond the lifetime of its associated instance.

The data on an instance store volume persists even if the instance is rebooted. However, the data does not persist if the instance is stopped, hibernated, or terminated. When the instance is stopped, hibernated, or terminated, every block of the instance store volume is cryptographically erased.

Therefore, do not rely on instance store volumes for valuable, long-term data. If you need to retain the data stored on an instance store volume beyond the lifetime of the instance, you need to manually copy that data to more persistent storage, such as an Amazon EBS volume, an Amazon S3 bucket, or an Amazon EFS file system.

There are some events that can result in your data not persisting throughout the lifetime of the instance. The following table indicates whether data on instance store volumes is persisted during specific events, for both virtualized and bare metal instances.

Event	What happens to your data?
User-initiated instance lifecycle events	
The instance is rebooted	The data persists
The instance is stopped	The data does not persist
The instance is hibernated	The data does not persist
The instance is terminated	The data does not persist
The instance type is changed	The data does not persist *
An EBS-backed AMI is created from the instance	The data does not persist in the created AMI **
An instance store-backed AMI is created from the instance (Linux instances)	The data persists in the AMI bundle uploaded to Amazon S3 ***
User-initiated OS events	
A shutdown is initiated	The data does not persist †
A restart is initiated	The data persists
AWS scheduled events	

Event	What happens to your data?
Instance stop	The data does not persist
Instance reboot	The data persists
System reboot	The data persists
Instance retirement	The data does not persist
Unplanned events	
Simplified automatic recovery	The data does not persist
CloudWatch action based recovery	The data does not persist
The underlying disk fails	The data on the failed disk does not persist
Power failure	The data persists upon reboot

* If the new instance type supports instance store, the instance gets the number of instance store volumes supported by the new instance type, but the data does not transfer to the new instance. If the new instance type does not support instance store, the instance does not get the instance store volumes.

** The data is not included in the EBS-backed AMI, and it is not included on instance store volumes attached to instances launched from that AMI.

*** The data is included in the AMI bundle that is uploaded to Amazon S3. When you launch an instance from that AMI, the instance gets the instance store volumes bundled in the AMI with the data they contained at the time the AMI was created.

† Termination protection and stop protection do not protect instances against instance stops or terminations as a result of shutdowns initiated through the operating system on the instance. Data stored on instance store volumes does not persist in both instance stop and termination events.

Instance store volume limits for EC2 instances

The number, size, and type of instance store volumes are determined by the instance type. Some instance types, such as M6, C6, and R6, do not support instance store volumes, while other instance types, such as M5d, C6gd, and R6gd, do support instance store volumes. You can’t attach more instance store volumes to an instance than is supported by its instance type. For the instance types that do support instance store volumes, the number and size of the instance store volumes vary by instance size. For example, `m5d.large` supports 1 x 75 GB instance store volume, while `m5d.24xlarge` supports 4 x 900 GB instance store volumes.

For instance types with **NVMe instance store volumes**, all of the supported instance store volumes are automatically attached to the instance at launch. For instance types with **non-NVMe instance store volumes**, such as C1, C3, M1, M2, M3, R3, D2, H1, I2, X1, and X1e, you must manually specify the block device mappings for the instance store volumes that you want to attach at launch. Then, after the instance has launched, you must format and mount the attached instance store volumes before you can use them. You can't attach an instance store volume after you launch the instance.

Some instance types use NVMe or SATA-based solid state drives (SSD), while others use SATA-based hard disk drives (HDD). SSDs deliver high random I/O performance 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 for EC2 instances.

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.

Available instance store volumes

The *Amazon EC2 Instance Types Guide* provides the quantity, size, type, and performance optimizations of instance store volumes available on each supported instance type. For more information, see the following:

- Instance store specifications – General purpose
- Instance store specifications – Compute optimized
- Instance store specifications – Memory optimized
- Instance store specifications – Storage optimized
- Instance store specifications – Accelerated computing
- Instance store specifications – High-performance computing
- Instance store specifications – Previous generation

Console

To retrieve instance store volume information

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instance Types**.
3. Add the filter **Local instance storage = true**. The **Storage** column indicates the total size of the instance storage for the instance type.
4. (Optional) Click the **Preferences** icon and then turn on **Storage disk count**. This column indicates the number of instance store volumes.
5. (Optional) Add filters to further scope to specific instance types of interest.

AWS CLI

To retrieve instance store volume information

Use the describe-instance-types command. The following example displays the total size of the instance storage for each instance type in the R6i instance families with instance store volumes.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r6i*" "Name=instance-storage-
supported,Values=true" \
    --query "InstanceTypes[].[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
    --output table
```

The following is example output.

```
----------------------------
|  DescribeInstanceTypes   |
+----------------+---------+
|  r6id.16xlarge  |  3800   |
|  r6idn.16xlarge |  3800   |
|  r6idn.8xlarge  |  1900   |
|  r6id.2xlarge   |  474    |
|  r6idn.xlarge   |  237    |
|  r6id.12xlarge  |  2850   |
|  r6idn.2xlarge  |  474    |
|  r6id.xlarge    |  237    |
|  r6idn.24xlarge |  5700   |
```

```
|  r6id.4xlarge   |  950   |
|  r6id.32xlarge  |  7600  |
|  r6id.24xlarge  |  5700  |
|  r6idn.large    |  118   |
|  r6idn.4xlarge  |  950   |
|  r6id.large     |  118   |
|  r6id.8xlarge   |  1900  |
|  r6idn.32xlarge |  7600  |
|  r6idn.metal    |  7600  |
|  r6id.metal     |  7600  |
|  r6idn.12xlarge |  2850  |
+----------------+--------+
```

To get complete instance storage details for an instance type

Use the describe-instance-types command.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r6id.16xlarge" \
    --query "InstanceTypes[].InstanceStorageInfo"
```

The example output shows that this instance type has two 1900 GB NVMe SSD volumes, for a total of 3800 GB of instance storage.

```
[
    {
        "TotalSizeInGB": 3800,
        "Disks": [
            {
                "SizeInGB": 1900,
                "Count": 2,
                "Type": "ssd"
            }
        ],
        "NvmeSupport": "required",
        "EncryptionSupport": "required"
    }
]
```

PowerShell

To retrieve instance store volume information

Use the Get-EC2InstanceType cmdlet. The following example displays the total size of the instance storage for each instance type in the R6i instance families with instance store volumes.

```
(Get-EC2InstanceType -Filter @{Name="instance-type"; 
 Values="r6i*"},@{Name="instance-storage-supported"; Values="true"})
 | Format-Table @{Name="InstanceType";Expression={$_.InstanceType}}, 
 @{Name="TotalSize";Expression={$_.InstanceStorageInfo.TotalSizeInGB}}
```

The following is example output.

```
InstanceType   TotalSize
------------   ---------
r6idn.16xlarge      3800
r6id.16xlarge       3800
r6id.xlarge          237
r6idn.8xlarge       1900
r6idn.2xlarge        474
r6id.12xlarge       2850
r6idn.xlarge         237
r6id.2xlarge         474
r6id.4xlarge         950
r6idn.24xlarge      5700
r6id.32xlarge       7600
r6id.24xlarge       5700
r6idn.large          118
r6id.large           118
r6idn.4xlarge        950
r6id.8xlarge        1900
r6id.metal          7600
r6idn.32xlarge      7600
r6idn.metal         7600
r6idn.12xlarge      2850
```

To get complete instance storage details for an instance type

Use the Get-EC2InstanceType cmdlet. The output is converted to JSON format.

```
(Get-EC2InstanceType -Filter @{Name="instance-type"; 
 Values="r6id.16xlarge"}).InstanceStorageInfo | ConvertTo-Json
```

The example output shows that this instance type has two 1900 GB NVMe SSD volumes, for a total of 3800 GB of instance storage.

```
{
  "Disks": [
    {
      "Count": 2,
      "SizeInGB": 1900,
      "Type": "ssd"
    }
  ],
  "EncryptionSupport": {
    "Value": "required"
  },
  "NvmeSupport": {
    "Value": "required"
  },
  "TotalSizeInGB": 3800
}
```

SSD instance store volumes for EC2 instances

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 an EC2 instance.

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 volume limits for EC2 instances.

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.

Linux instances

To access NVMe volumes, the NVMe drivers must be installed. The following AMIs meet this requirement:

- AL2023

- Amazon Linux 2
- Amazon Linux AMI 2018.03 and later
- Ubuntu 14.04 or later with `linux-aws` kernel

> **Note**
>
> AWS Graviton-based instance types require Ubuntu 18.04 or later with `linux-aws` kernel

- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later
- Bottlerocket

After you connect to your instance, you can list the NVMe devices using the **lspci** command. The following is example output for an `i3.8xlarge` instance, which supports four NVMe devices.

```
[ec2-user ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Device 1d0f:ec20
00:17.0 Non-Volatile memory controller: Device 1d0f:cd01
00:18.0 Non-Volatile memory controller: Device 1d0f:cd01
00:19.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1a.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1f.0 Unassigned class [ff80]: XenSource, Inc. Xen Platform Device (rev 01)
```

If you are using a supported operating system but you do not see the NVMe devices, verify that the NVMe module is loaded using the following command.

- Amazon Linux, Amazon Linux 2, Ubuntu 14/16, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS 7

```
$ lsmod | grep nvme
nvme          48813  0
```

- Ubuntu 18

```
$ cat /lib/modules/$(uname -r)/modules.builtin | grep nvme
s/nvme/host/nvme-core.ko
kernel/drivers/nvme/host/nvme.ko
kernel/drivers/nvmem/nvmem_core.ko
```

The NVMe volumes are compliant with the NVMe 1.0e specification. You can use the NVMe commands with your NVMe volumes. With Amazon Linux, you can install the `nvme-cli` package from the repo using the **yum install** command. With other supported versions of Linux, you can download the `nvme-cli` package if it's not available in the image.

Windows instances

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 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 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**.

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.

Non-NVMe SSD volumes

The following instances support instance store volumes that use non-NVMe SSDs to deliver high random I/O performance: C3, I2, M3, R3, and X1. For more information about the instance store volumes supported by each instance type, see Instance store volume limits for EC2 instances.

SSD-based instance store volume I/O 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 do not have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10 percent 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 have 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.

Instance store volume TRIM support

Some instance types support SSD volumes with TRIM. For more information, see Instance store volume limits for EC2 instances.

Note

(Windows instances only) 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.

(Windows instances) 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 **Linux instances**, use the `fstrim` command to enable periodic TRIM. On **Windows instances**, use the `fsutil behavior set DisableDeleteNotify 0` command to ensure TRIM support is enabled during normal operation.

Add instance store volumes to an EC2 instance

For instance types with **NVMe instance store volumes**, all of the supported instance store volumes are automatically attached to the instance at launch. They are automatically enumerated and assigned a device name on instance launch.

For instance types with **non-NVMe instance store volumes**, such as C1, C3, M1, M2, M3, R3, D2, H1, I2, X1, and X1e, you must manually specify the block device mappings for the instance store volumes that you want to attach at launch. Block device mappings can be specified in the instance launch request or in the AMI used to launch the instance. The block device mapping includes a device name and the volume that it maps to. For more information, see [Block device mappings for volumes on Amazon EC2 instances]

> **Important**
>
> Instance store volumes can be attached to an instance only when you launch it. You can't attach instance store volumes to an instance after you've launched it.

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.

Consideration for root volumes

A block device mapping always specifies the root volume for the instance. The root volume is always mounted automatically.

Linux instances – The root volume is either an Amazon EBS volume or an instance store volume. For instances with an instance store volume for the root volume, the size of this volume varies by AMI, but the maximum size is 10 GB. For more information, see Root device type.

Windows instances – The root volume must be an Amazon EBS volume. Instance store is not supported for the root volume.

Contents

Add instance store volumes to an Amazon EC2 AMI

You can create an AMI with a block device mapping that includes instance store volumes.

If you launch an instance that supports **non-NVMe instance store volumes** using an AMI that specifies instance store volume block device mappings, the instance includes the instance store volumes. If the number of instance store volume block device mappings in the AMI exceeds the number of instance store volumes available to the instance, the additional instance store volume block device mappings are ignored.

If you launch an instance that supports **NVMe instance store volumes** using an AMI that specifies instance store volume block device mappings, the instance store volume block device mappings are ignored. Instances that support NVMe instance store volumes get all of their supported instance store volumes, regardless of the block device mappings specified in the instance launch request and the AMI. The device mapping of these volumes depends on the order in which the operating system enumerates the volumes.

Considerations

- The number of available instance store volumes depends on the instance type. For more information, see the section called "Available instance store volumes".
- You must specify a device name for each block device. For more information, see Device names for volumes on Amazon EC2 instances.
- 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.
- For M3 instances, specify instance store volumes in the block device mapping of the instance, not the AMI. Amazon EC2 might ignore instance store volume block device mappings in the AMI.

Console

To add instance store volumes to an Amazon EBS-backed AMI

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.
6. Choose **Create image**.

AWS CLI

To add instance store volumes to an AMI

Use the create-image command with the `--block-device-mappings` option to specify a block device mapping for an EBS-backed AMI. Use the register-image command with the `--block-device-mappings` option to specify a block device mapping for an instance store-backed AMI.

```
--block-device-mappings file://mapping.json
```

The following block device mapping adds two instance store volumes.

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

To add instance store volumes to an AMI

Use the New-EC2Image cmdlet with the `-BlockDeviceMapping` parameter to specify a block device mapping for an EBS-backed AMI. Use the Register-EC2Image cmdlet with the `-BlockDeviceMapping` parameter to specify a block device mapping for an instance store-backed AMI.

```
-BlockDeviceMapping $bdm
```

The following block device mapping adds two instance store volumes.

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

Add instance store volumes to an EC2 instance during launch

When you launch an instance type with **non-NVMe instance store volumes**, such as C1, C3, M1, M2, M3, R3, D2, H1, I2, X1, and X1e, you must specify the block device mappings for the instance store volumes that you want to attach at launch. The block device mappings must be specified in the instance launch request or in the AMI used to launch the instance.

If the AMI includes block device mappings for the instance store volumes, you do not need to specify block device mappings in the instance launch request, unless you need more instance store volumes than is included in the AMI.

If the AMI does not include block device mappings for instance store volumes, then you must specify the block device mappings in the instance launch request.

For instance types with NVMe instance store volumes, all of the supported instance store volumes are automatically attached to the instance at launch.

Considerations

- The number of available instance store volumes depends on the instance type. For more information, see the section called “Available instance store volumes”.
- You must specify a device name for each block device. For more information, see Device names for volumes on Amazon EC2 instances.
- For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.

Console

To specify a block device mapping in an instance launch request

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the dashboard, choose **Launch instance**.
3. In the **Application and OS Images** section, select the AMI to use.
4. In the **Configure storage** section, the **Instance store volumes** section lists the instance store volumes that can be attached to the instance.
5. For each instance store volume to attach, for **Device name**, select the device name to use.
6. Configure the remaining instance settings as needed, and then choose **Launch instance**.

AWS CLI

To specify a block device mapping in an instance launch request

Use the [run-instances](#) command with the `--block-device-mappings` option.

```
--block-device-mappings file://mapping.json
```

The following block device mapping adds two instance store volumes.

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

To specify a block device mapping in an instance launch request

Use the [New-EC2Instance](#) cmdlet with the `-BlockDeviceMapping` option.

```
-BlockDeviceMapping $bdm
```

The following block device mapping adds two instance store volumes.

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

Make instance store volume available for use on an EC2 instance

After you launch an instance with attached instance store volumes, you must mount the volumes before you can access them.

Linux instances

You can format volumes with the file system of your choice after you launch your instance.

To make an instance store volume available on Linux

1. Connect to the instance using an SSH client. For more information, see Connect to your Linux instance using SSH.
2. Use the `df -h` command to view the volumes that are formatted and mounted.

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   72K  3.8G   1% /dev
tmpfs           3.8G     0  3.8G   0% /dev/shm
/dev/nvme0n1p1  7.9G  1.2G  6.6G  15% /
```

3. Use the `lsblk` to view any volumes that were mapped at launch but not formatted and mounted.

```
$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1    0    8G  0 disk
##nvme0n1p1   259:2    0    8G  0 part /
##nvme0n1p128 259:3    0    1M  0 part
nvme1n1       259:0    0 69.9G  0 disk
```

4. To format and mount an instance store volume that was mapped only, do the following:

   a. Create a file system on the device using the `mkfs` command.

   ```
   $ sudo mkfs -t xfs /dev/nvme1n1
   ```

   b. Create a directory on which to mount the device using the `mkdir` command.

   ```
   $ sudo mkdir /data
   ```

   c. Mount the device on the newly created directory using the `mount` command.

```
$ sudo mount /dev/nvme1n1 /data
```

Windows instances

For Windows instances, we reformat the instance store volumes with the NTFS file system.

You can view the instance store volumes using Windows Disk Management. For more information, see List non-NVMe disks.

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.
9. Choose **Next**, **Finish**.

Enable instance store swap volume for M1 and C1 EC2 instances

Note

This topic applies to `c1.medium` and `m1.small` Linux instances only.

The `c1.medium` and `m1.small` instance types have a limited amount of physical memory. Therefore, they are given a 900 MiB swap volume at launch time to act as virtual memory, or *swap space*, for the Linux system. Swap space in Linux can be used when a system requires more memory than it has been physically allocated. When swap space is enabled, Linux systems can swap infrequently used memory pages from physical memory to swap space (either a dedicated partition or a swap file in an existing file system) and free up that space for memory pages that require high-speed access.

Note

- Using swap space for memory paging is not as fast or efficient as using RAM. If your workload is regularly paging memory into swap space, you should consider migrating to a larger instance type with more RAM. For more information, see Amazon EC2 instance type changes.
- Although the Linux kernel sees this swap space as a partition on the root device, it is actually a separate instance store volume, regardless of your root device type.

Amazon Linux automatically enables and uses this swap space, but your AMI may require some additional steps to recognize and use this swap space. To see if your instance is using swap space, you can use the **swapon -s** command.

```
[ec2-user ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/xvda3                              partition       917500  0       -1
```

The above instance has a 900 MiB swap volume attached and enabled. If you don't see a swap volume listed with this command, you may need to enable swap space for the device. Check your available disks using the **lsblk** command.

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

```
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk
```

Here, the swap volume `xvda3` is available to the instance, but it is not enabled (notice that the `MOUNTPOINT` field is empty). You can enable the swap volume with the **swapon** command.

> **Note**
>
> You must prepend `/dev/` to the device name listed by **lsblk**. Your device may be named differently, such as `sda3`, `sde3`, or `xvde3`. Use the device name for your system in the command below.

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

Now the swap space should show up in **lsblk** and **swapon -s** output.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk [SWAP]
[ec2-user ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/xvda3                              partition       917500  0       -1
```

You also need to edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

Append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvda3       none    swap    sw  0       0
```

To use an instance store volume as swap space

Any instance store volume can be used as swap space. For example, the `m3.medium` instance type includes a 4 GB SSD instance store volume that is appropriate for swap space. If your instance store volume is much larger (for example, 350 GB), you may consider partitioning the volume with a smaller swap partition of 4-8 GB and the rest for a data volume.

> **Note**
>
> This procedure applies only to instance types that support instance storage. For a list of supported instance types, see Instance store volume limits for EC2 instances.

1. List the block devices attached to your instance to get the device name for your instance store volume.

```
[ec2-user ~]$ lsblk -p
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/xvdb   202:16   0   4G  0 disk /media/ephemeral0
/dev/xvda1 202:1    0   8G  0 disk /
```

   In this example, the instance store volume is `/dev/xvdb`. Because this is an Amazon Linux instance, the instance store volume is formatted and mounted at `/media/ephemeral0`; not all Linux operating systems do this automatically.

2. (Optional) If your instance store volume is mounted (it lists a `MOUNTPOINT` in the **lsblk** command output), unmount it with the following command.

```
[ec2-user ~]$ sudo umount /dev/xvdb
```

3. Set up a Linux swap area on the device with the **mkswap** command.

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swapspace version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. Enable the new swap space.

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. Verify that the new swap space is being used.

```
[ec2-user ~]$ swapon -s
```

```
Filename     Type  Size Used Priority
/dev/xvdb                              partition 4188668 0 -1
```

6. Edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

If your `/etc/fstab` file has an entry for `/dev/xvdb` (or `/dev/sdb`) change it to match the line below; if it does not have an entry for this device, append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvdb       none     swap     sw   0        0
```

Important

Instance store volume data is lost when an instance is stopped or hibernated; this includes the instance store swap space formatting created in Step 3. If you stop and restart an instance that has been configured to use instance store swap space, you must repeat Step 1 through Step 5 on the new instance store volume.

Initialize instance store volumes on EC2 instances

Because of the way that Amazon EC2 virtualizes disks, the first write to any location on some instance store volumes performs more slowly than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you initialize your drives by writing once to every drive location before production use.

Note

Instance types with direct-attached solid state drives (SSD) and TRIM support provide maximum performance at launch time, without initialization. For information about the instance store for each instance type, see Instance store volume limits for EC2 instances.

If you require greater flexibility in latency or throughput, we recommend using Amazon EBS.

To initialize the instance store volumes, use the following dd commands, depending on the store to initialize (for example, `/dev/sdb` or `/dev/nvme1n1`).

> **Note**
>
> Make sure to unmount the drive before performing this command.
> Initialization can take a long time (about 8 hours for an extra large instance).

To initialize the instance store volumes, use the following commands on the `m1.large`, `m1.xlarge`, `c1.xlarge`, `m2.xlarge`, `m2.2xlarge`, and `m2.4xlarge` instance types:

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

To perform initialization on all instance store volumes at the same time, use the following command:

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

Configuring drives for RAID initializes them by writing to every drive location. When configuring software-based RAID, make sure to change the minimum reconstruction speed:

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

Root volumes for your Amazon EC2 instances

When you launch an instance, we create a *root volume* for the instance. The root volume contains the image used to boot the instance. Each instance has a single root volume. You can add storage volumes to your instances during or after launch.

The AMI that you use to launch an instance determines the type of root volume. You can launch an instance from either an *Amazon EBS-backed AMI* (Linux and Windows instances) or an *instance store-backed AMI* (Linux instances only). There are significant differences between what you can do with each type of AMI. For more information about these differences, see Root device type.

We recommend that you use AMIs backed by Amazon EBS, because these instances launch faster and use persistent storage.

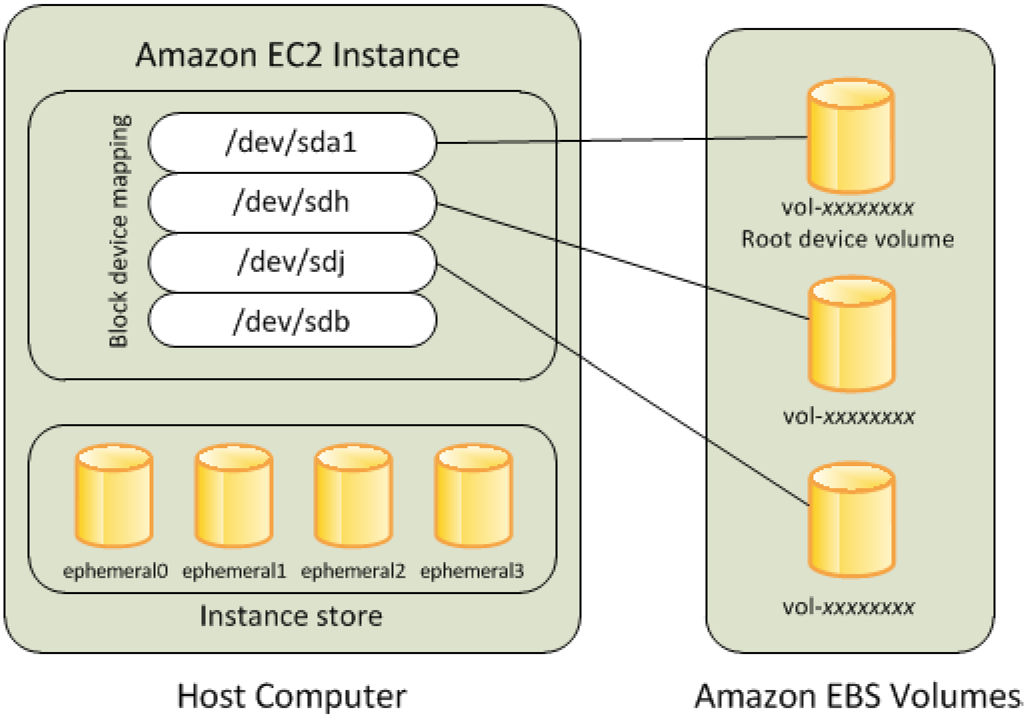
We reserve specific device names for root volumes. For more information, see Device names for volumes on Amazon EC2 instances.

Contents

Amazon EBS-backed instances

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

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



Limitation

You can't use `st1` or `sc1` EBS volumes as root volumes.

Instance failure

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

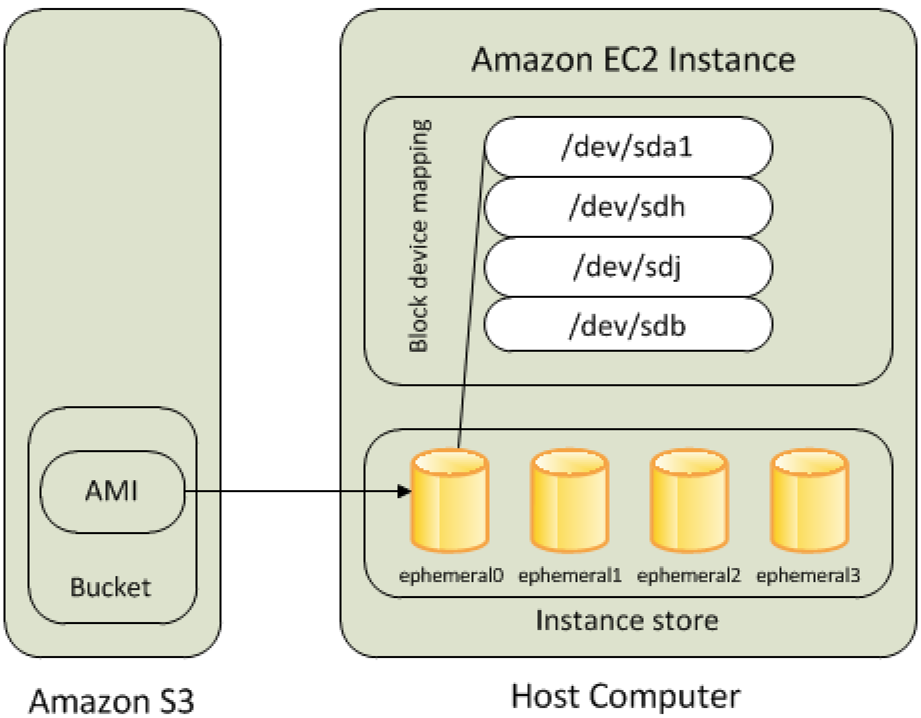
- Stop and then start again (try this method first).
- Automatically snapshot all relevant volumes and create a new AMI. For more information, see Create an Amazon EBS-backed AMI.
- Attach the volume to the new instance by following these steps:
  1. Create a snapshot of the root volume.
  2. Register a new AMI using the snapshot.
  3. Launch a new instance from the new AMI.
  4. Detach the remaining Amazon EBS volumes from the old instance.
  5. Reattach the Amazon EBS volumes to the new instance.

Instance store-backed instances (Linux instances only)

> **Note**
>
> Windows instances do not support instance-store backed root volumes.

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

Any data on the instance store volumes persists as long as the instance is running, but this data is deleted when the instance is terminated (instance store-backed instances do not support the **Stop** action) or if it fails (such as if an underlying drive has issues). For more information, see [Instance store temporary block storage for EC2 instances](#).



Supported instance types

Only the following instance types support an instance store volume as the root volume: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

Instance failure

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

Keep an Amazon EBS root volume after an Amazon EC2 instance terminates

By default, the Amazon EBS root volume for an instance is deleted when the instance terminates. You can change the default behavior to ensure that an Amazon EBS root volume persists after the instance terminates. To change the default behavior, set the `DeleteOnTermination` attribute to `false`. You can do so either at instance launch or later on.

Tasks

- Configure the root volume to persist during instance launch
- Configure the root volume to persist for an existing instance
- Confirm that a root volume is configured to persist

Configure the root volume to persist during instance launch

You can configure the root volume to persist when you launch an instance.

Console

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

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. Choose an Amazon Machine Image (AMI), choose and instance type, choose a key pair, and configure your network settings.
4. For **Configure storage**, choose **Advanced**.
5. Expand the root volume.
6. For **Delete on termination**, choose **No**.
7. When you are finished configuring your instance, choose **Launch instance**.

AWS CLI

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

Use the run-instances command and include the following option.

```
--block-device-mappings file://mapping.json
```

In `mapping.json`, specify a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

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

PowerShell

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

Use the New-EC2Instance cmdlet and include the following parameter.

```
-BlockDeviceMapping $bdm
```

Create a block device mapping that sets the `DeleteOnTermination` attribute to `$false`.

```
$ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs.DeleteOnTermination = $false
$bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "dev/xvda"
$bdm.Ebs = $ebs
```

Configure the root volume to persist for an existing instance

You can configure the root volume to persist for a running instance. Note that you can't complete this task using the Amazon EC2 console.

AWS CLI

To configure the root volume to persist for an existing instance

Use the modify-instance-attribute command with a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`.

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

PowerShell

To configure the root volume to persist for an existing instance

Use the Edit-EC2InstanceAttribute cmdlet with a block device mapping that sets the `DeleteOnTermination` attribute to `$false`.

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

Confirm that a root volume is configured to persist

You can confirm that a root volume is configured to persist.

Console

To confirm that a root volume is configured to persist

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.

AWS CLI

To confirm that a root volume is configured to persist

Use the describe-instances command and verify that the `DeleteOnTermination` attribute is set to `false`.

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].BlockDeviceMappings"
```

The following is example output.

```
[
    [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "AttachTime": "2024-07-12T04:05:33.000Z",
                "DeleteOnTermination": false,
                "Status": "attached",
                "VolumeId": "vol-1234567890abcdef0"

        }
    ]
]
```

PowerShell

To confirm that a root volume is configured to persist

Use the [Get-EC2Instance](#) cmdlet and verify that the `DeleteOnTermination` attribute is set to `False`.

```
(Get-EC2Instance -InstanceId i-
i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

The following is example output.

```
AssociatedResource  :
AttachTime          : 7/12/2024 4:05:33 AM
DeleteOnTermination : False
Operator            :
Status              : attached
VolumeId            : vol-1234567890abcdef0
```

Replace the root volume for an Amazon EC2 instance without stopping it

Amazon EC2 enables you to replace the root Amazon EBS volume for a running instance while retaining the following:

- Data stored on instance store volumes — Instance store volumes remain attached to the instance after the root volume has been restored.
- Data stored on data (non-root) Amazon EBS volumes — Non-root Amazon EBS volumes remain attached to the instance after the root volume has been restored.
- 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.

Contents

- Replace a root volume

How root volume replacement works

When you replace the root volume for an instance, we create *root volume replacement task*. The original root volume is detached from the instance, and the new root volume is attached to the instance in its place. The instance's block device mapping is updated to reflect the ID of the replacement root volume.

When you replace the root volume for an instance, you must specify the source of the snapshot for the new volume. The following are the possible options.

Restore a root volume to its original state

This option replaces the current root volume with a volume that is based on the snapshot that was used to create it.

Considerations for using the launch state

The replacement root volume gets the same type, size, and delete on termination attributes as the original root volume.

Replace the root volume using a snapshot

This option replaces the current root volume with a replacement volume that is based on the snapshot that you specify. For example, a specific snapshot that you previously created from this root volume. This is useful if you need to recover from issues caused by corruption of the root volume or network configuration errors in the guest operating system.

The replacement root volume gets the same type, size, and delete on termination attributes as the original root volume.

Considerations for using a snapshot

- You can only use snapshots that were created directly from the instance's current or previous root volumes.
- You can't use snapshot copies created from snapshots that were taken from the root volume.
- After successfully replacing the root volume, you can still use snapshots taken from the original root volume to replace the new (replacement) root volume.

Replace the root volume using an AMI

This option replaces the current root volume using an AMI that you specify. This is useful if you need to perform operating system and application patching or upgrades. The AMI must have the same product code, billing information, architecture type, and virtualization type as the instance.

If the instance is enabled for ENA or sriov-net, then you must use an AMI that supports those features. If the instance is not enabled for ENA or sriov-net, then you can either select an AMI that doesn't include support for those features, or you can automatically add support if you select an AMI that supports ENA or sriov-net.

If the instance is enabled for NitroTPM, then you must use an AMI that has NitroTPM enabled. NitroTPM support is not enabled if the instance was not configured for it, regardless of the AMI that you select.

You can select an AMI with a different boot mode than that of the instance, as long as the instance supports the boot mode of the AMI. If the instance does not support the boot mode, the request fails. If the instance supports the boot mode, the new boot mode is propagated to the instance and its UEFI data is updated accordingly. If you manually modified the boot order or added a private UEFI Secure Boot key to load private kernel modules, the changes are lost during root volume replacement.

The replacement root volume gets the same volume type and delete on termination attribute as the original root volume, and it gets the size of the AMI root volume block device mapping.

> **Note**
>
> The size of the AMI root volume block device mapping must be equal to or greater than the size of the original root volume. If the size of the AMI root volume block device mapping is smaller than the size of the original root volume, the request fails.

After the root volume replacement task completes, the following new and updated information is reflected when you describe the instance using the console, AWS CLI or AWS SDKs:

- New AMI ID
- New volume ID for the root volume
- Updated boot mode configuration (if changed by the AMI)
- Updated NitroTPM configuration (if enabled by the AMI)

- Updated ENA configuration (if enabled by the AMI)
- Updated sriov-net configuration (if enabled by the AMI)

The new AMI ID is also reflected in the instance metadata.

Considerations for using an AMI:

- If you use an AMI that has multiple block device mappings, only the root volume of the AMI is used. The other (non-root) volumes are ignored.
- You can only use this feature if you have permissions to the AMI and its associated root volume snapshot. You cannot use this feature with AWS Marketplace AMIs.
- You can only use an AMI without a product code only if the instance does not have a product code.
- The size of the AMI root volume block device mapping must be equal to or greater than the size of the original root volume. If the size of the AMI root volume block device mapping is smaller than the size of the original root volume, the request fails.
- The instance identity documents for the instance are automatically updated.
- If the instance supports NitroTPM, the NitroTPM data for the instance is reset and new keys are generated.

You can choose whether to keep the original root volume after the root volume replacement process has completed. If you choose delete the original root volume after the replacement process completes, the original root volume is automatically deleted and becomes unrecoverable. If you choose to keep the original root volume after the process completes, the volume remains provisioned in your account; you must manually delete the volume when you no longer need it.

The root volume replacement task transitions through 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 root volume is still attached.

- `failing-detached` — The replacement task is in the process of failing and the instance might not have a root volume attached.
- `failed-detached` — The replacement task has failed and the instance doesn't have a root volume attached.

If the root volume replacement task fails, the instance is rebooted and the original root volume remains attached to the instance.

Considerations

Before you begin, consider the following.

Requirements

- The instance must be in the `running` state.
- The instance is automatically rebooted during the process. The contents of the memory (RAM) is erased during the reboot. No manual reboots are required.
- You can't replace the root volume if it is an instance store volume. Only instances with Amazon EBS root volumes are supported.
- You can replace the root volume for all virtualized instance types and EC2 Mac bare metal instances. No other bare metal instance types are supported.
- You can only use snapshots that were created directly from the instance's current or previous root volumes.
- If your account is enabled for Amazon EBS encryption by default in the current Region, the replacement root volume created by the root volume replacement task is always encrypted, regardless of the encryption status of the specified snapshot or the root volume of the specified AMI.

Encryption outcomes

The following table summarizes the possible encryption outcomes.

	Original root volume	Specified snapshot or AMI	Encryption by default	Replacement root volume	Encryption key used for replacement root volume
Restore replacement root volume to initial launch state	Encrypted	Not applicable	Not considered	Encrypted	Same KMS key as original root volume
	Unencrypted	Not applicable	Disabled	Unencrypted	Not applicable
	Unencrypted	Not applicable	Enabled	Encrypted	Account's default KMS key for Amazon EBS encryption
Restore replacement root volume from snapshot or AMI	Encrypted	Unencrypted	Not considered	Encrypted	Same KMS key as original root volume
	Encrypted	Encrypted	Not considered	Encrypted	Same KMS key as original root volume
	Unencrypted	Unencrypted	Disabled	Unencrypted	Not applicable
	Unencrypted	Unencrypted	Enabled	Encrypted	Account's default KMS key for Amazon EBS encryption

	Original root volume	Specified snapshot or AMI	Encryption by default	Replacement root volume	Encryption key used for replacement root volume
	Unencrypted	Encrypted	Not considered	Encrypted	If the AMI or snapshot is owned by the account, the replacement volume is encrypted with the AMI or snapshot's KMS key. If AMI or snapshot is shared with the account, replacement volume is encrypted with the account's default KMS key for Amazon EBS encryption.

Replace a root volume

When you replace the root volume for an instance, a *root volume replacement task* is created. You can use the root volume replacement task to monitor the progress and outcome of the replacement process.

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**.

   > **Note**
   >
   > The **Replace root volume** action is disabled if the selected instance is not in the `running` state.

4. For **Restore**, choose one of the following options:
   - **Launch state** – Restore the replacement root volume from the snapshot that was used to create the current root volume.
   - **Snapshot** – Restore the replacement root volume to the snapshot that you specify. For **Snapshot**, select the snapshot to use.
   - **Image** – Restore the replacement root volume using the AMI that you specify. For **Image**, select the AMI to use.
5. (Optional) For **Volume initialization rate**, you can specify the Amazon EBS Provisioned Rate for Volume Initialization (volume initialization rate), in MiB/s, at which the snapshot blocks are to be downloaded from Amazon S3 to the volume. For more information, see Use an Amazon EBS Provisioned Rate for Volume Initialization. To use the default initialization rate or fast snapshot restore (if it is enabled for the selected snapshot), don't specify a rate.
6. (Optional) To delete the root volume that you are replacing, select **Delete replaced root volume**.
7. Choose **Create replacement task**.
8. To monitor the replacement task, choose the **Storage** tab for the instance and expand **Recent root volume replacement tasks**.

AWS CLI

To restore the replacement root volume to the launch state

Use the create-replace-root-volume-task command. For `--instance-id`, specify the ID of the instance for which to replace the root volume. Omit the `--snapshot-id` and `--image-id` parameters. To delete the original root volume after it has been replaced, include `--delete-replaced-root-volume` and specify `true`. To specify the volume initialization rate at which the snapshot blocks are downloaded from Amazon S3 to the volume, for `--volume-initialization-rate`, specify a value between `100` and `300` MiB/s.

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

To restore the replacement root volume to a specific snapshot

Use the create-replace-root-volume-task command. For `--instance-id`, specify the ID of the instance for which to replace the root volume. For `--snapshot-id`, specify the ID of the snapshot to use. To delete the original root volume after it has been replaced, include `--delete-replaced-root-volume` and specify `true`. To specify the volume initialization rate at which the snapshot blocks are downloaded from Amazon S3 to the volume, for `--volume-initialization-rate`, specify a value between `100` and `300` MiB/s.

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--snapshot-id snap-9876543210abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

To restore the replacement root volume using an AMI

Use the create-replace-root-volume-task command. For `--instance-id`, specify the ID of the instance for which to replace the root volume. For `--image-id`, specify the ID of the AMI to use. To delete the original root volume after it has been replaced, include `--delete-replaced-root-volume` and specify `true`. To specify the volume initialization rate at which the snapshot blocks are downloaded from Amazon S3 to the volume, for `--volume-initialization-rate`, specify a value between `100` and `300` MiB/s.

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--image-id ami-09876543210abcdef \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

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.

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-1234567890abcdef0 \
    --query ReplaceRootVolumeTasks[].TaskState
```

The following is example output.

```
[
    "succeeded"
]
```

Alternatively, specify the `instance-id` filter to filter the results by instance.

```
$ aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-1234567890abcdef0
```

PowerShell

To restore the replacement root volume to the launch state

Use the New-EC2ReplaceRootVolumeTask command. For `-InstanceId`, specify the ID of the instance for which to replace the root volume. Omit the `-SnapshotId` and `-ImageId` parameters. To delete the original root volume after it has been replaced, include `-DeleteReplacedRootVolume` and specify `$true`. To specify the volume initialization rate at which the snapshot blocks are downloaded from Amazon S3 to the volume, for `-VolumeInitializationRate`, specify a value between `100` and `300` MiB/s.

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -VolumeInitializationRate 150 `
```

```
    -DeleteReplacedRootVolume $true
```

To restore the replacement root volume to a specific snapshot

Use the New-EC2ReplaceRootVolumeTask command. For `--InstanceId`, specify the ID of the instance for which to replace the root volume. For `-SnapshotId`, specify the ID of the snapshot to use. To delete the original root volume after it has been replaced, include `-DeleteReplacedRootVolume` and specify `$true`. To specify the volume initialization rate at which the snapshot blocks are downloaded from Amazon S3 to the volume, for `-VolumeInitializationRate`, specify a value between `100` and `300` MiB/s.

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -SnapshotId snap-9876543210abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

To restore the replacement root volume using an AMI

Use the New-EC2ReplaceRootVolumeTask command. For `-InstanceId`, specify the ID of the instance for which to replace the root volume. For `-ImageId`, specify the ID of the AMI to use. To delete the original root volume after it has been replaced, include `-DeleteReplacedRootVolume` and specify `$true`. To specify the volume initialization rate at which the snapshot blocks are downloaded from Amazon S3 to the volume, for `-VolumeInitializationRate`, specify a value between `100` and `300` MiB/s.

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -ImageId ami-0abcdef1234567890 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

To view the status of a root volume replacement task

Use the Get-EC2ReplaceRootVolumeTask command and specify the IDs of the root volume replacement tasks to view.

```
(Get-EC2ReplaceRootVolumeTask `
    -ReplaceRootVolumeTaskIds replacevol-1234567890abcdef0).TaskState.Value
```

The following is example output.

```
Succeeded
```

Alternatively, specify the `instance-id` filter to filter the results by instance.

```
PS C:\> Get-EC2ReplaceRootVolumeTask -Filters @{Name = 'instance-id'; Values =
 'i-1234567890abcdef0'} | Format-Table
```

Device names for volumes on Amazon EC2 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 Amazon EBS volume limits for Amazon EC2 instances.

Contents

Available device names

Linux instances

There are two types of virtualization available for Linux instances: paravirtual (PV) and hardware virtual machine (HVM). The virtualization type of an instance is determined by the AMI used to launch the instance. All instance types support HVM AMIs. Some previous generation instance types support PV AMIs. Be sure to note the virtualization type of your AMI because the recommended and available device names that you can use depend on the virtualization type of your instance. For more information, see Virtualization types.

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume.

Virtualization type	Available	Reserved for root volume	Recommended for EBS data volumes	Instance store volumes
Paravirtual	/dev/sd[a-z] /dev/sd[a-z][1-15] /dev/hd[a-z] /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[f-p] /dev/sd[f-p][1-6]	/dev/sd[b-e]
HVM	/dev/sd[a-z] /dev/xvd[a-c][a-z] /dev/xvdd[a-x]	Differs by AMI /dev/sda1 or /dev/xvda	/dev/sd[b-z] /dev/xvdb[b-z] *	/dev/sd[b-e] /dev/sd[b-h] (h1.16xlarge) /dev/sd[b-y] (d2.8xlarge) /dev/sd[b-i] (i2.8xlarge) **

* The device names that you specify for NVMe EBS volumes in a block device mapping are renamed using NVMe device names (`/dev/nvme[0-26]n1`). The block device driver can assign NVMe device names in a different order than you specified for the volumes in the block device mapping.

** NVMe instance store volumes are automatically enumerated and assigned an NVMe device name.

Windows instances

AWS Windows AMIs use one of the following sets of drivers to permit access to virtualized hardware:

- AWS PV: Paravirtual drivers for Windows instances
- AWS NVMe: AWS NVMe drivers

Device names for Nitro based instances

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume to a Nitro based instance.

Driver type	Available	Reserved for root volume	Recommended for EBS volumes	Instance store volumes
AWS NVMe	xvd[a-z] xvd[a-c][a-z] xvdd[a-x] /dev/sda1	/dev/sda1	xvd[b-z] xvdb[b-z]	*

* NVMe instance store volumes are automatically enumerated and are assigned a Windows drive letter.

Device names for Xen based instances

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume to a Xen based instance.

Driver type	Available	Reserved for root volume	Recommended for EBS volumes	Instance store volumes
AWS PV	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]
Citrix PV (no longer supported)	xvd[b-z] xvd[b-c][a-z] /dev/sda1	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]

Driver type	Available	Reserved for root volume	Recommended for EBS volumes	Instance store volumes
	/dev/sd[b-e]			
Red Hat PV (no longer supported)	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-p]	xvdc[a-x] xvd[a-e]

For more information about instance store volumes, see Instance store temporary block storage for EC2 instances. For more information about NVMe EBS volumes (Nitro-based instances), including how to identify the EBS device, see Amazon EBS and NVMe in the *Amazon EBS User Guide*.

Device name considerations

Keep the following in mind when selecting a device name:

- The ending portion of device names that you use shouldn't overlap as it can cause issues when you start your instance. For example, avoid using combinations such as `/dev/xvdf` and `xvdf` for volumes attached to the same instance.
- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- The number of NVMe instance store volumes for an instance depends on the size of the instance. NVMe instance store volumes are automatically enumerated and assigned an NVMe device name (Linux instances) or a Windows drive letter (Windows instances).
- (Windows instances) 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 Windows launch agents on Amazon EC2 Windows instances, and How volumes are attached and mapped for Amazon EC2 Windows instances.

- (Linux instances) Depending on the block device driver of the kernel, the device could be attached with a different name than you specified. For example, if you specify a device name of `/dev/sdh`, your device could be renamed `/dev/xvdh` or `/dev/hdh`. In most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), the trailing letter could change (`/dev/sda` could become `/dev/xvde`). In these cases, the trailing letter of each device name is incremented the same number of times. For example, if `/dev/sdb` is renamed `/dev/xvdf`, then `/dev/sdc` is renamed `/dev/xvdg`. Amazon Linux creates a symbolic link for the name you specified to the renamed device. Other operating systems could behave differently.
- (Linux instances) HVM AMIs do not support the use of trailing numbers on device names, except for `/dev/sda1`, which is reserved for the root device, and `/dev/sda2`. While using `/dev/sda2` is possible, we do not recommend using this device mapping with HVM instances.
- (Linux instances) When using PV AMIs, you cannot attach volumes that share the same device letters both with and without trailing digits. For example, if you attach a volume as `/dev/sdc` and another volume as `/dev/sdc1`, only `/dev/sdc` is visible to the instance. To use trailing digits in device names, you must use trailing digits on all device names that share the same base letters (such as `/dev/sdc1`, `/dev/sdc2`, `/dev/sdc3`).
- (Linux instances) Some custom kernels might have restrictions that limit use to `/dev/sd[f-p]` or `/dev/sd[f-p][1-6]`. If you're having trouble using `/dev/sd[q-z]` or `/dev/sd[q-z][1-6]`, try switching to `/dev/sd[f-p]` or `/dev/sd[f-p][1-6]`.

Before you specify the device name that you've selected, verify that it is available. Otherwise, you'll get an error that the device name is already in use. To view the disk devices and their mount points, use the **lsblk** command (Linux instances), or the Disk Management utility or the **diskpart** command (Windows instances).

Block device mappings for volumes on Amazon EC2 instances

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 mappings 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. However, the only way to attach instance store volumes to an instance is to use block device mappings to attach the volumes as the instance is launched.

Contents

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

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 for volumes on Amazon EC2 instances.

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. You can't specify the ID of an archived snapshot.
- 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 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.

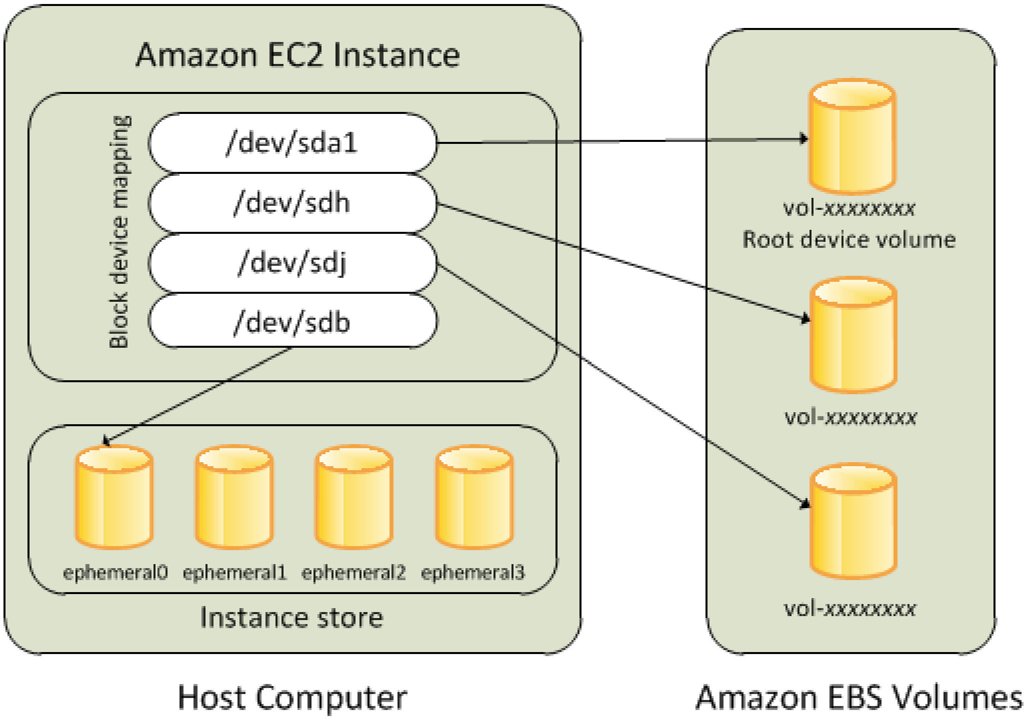
- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store 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 and Update the block device mapping when launching an instance.

How devices are made available in the operating system

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

Linux instances – The device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots. The instance type determines which instance store volumes are formatted and mounted by default. You can mount additional instance store volumes at launch, as long as you don't exceed the number of instance store volumes available for your instance type. For more information, see Instance store temporary block storage for EC2 instances. The block device driver for the instance determines which devices are used when the volumes are formatted and mounted.

Windows instances – 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 drive letters are assigned to EBS volumes; for more information, see the section called "Windows launch agents".

Add block device mappings to an AMI

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. To add more block devices to an AMI, you must create your own AMI.

Contents

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.

Console

To add volumes to an AMI

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**.

AWS CLI

To add volumes to an AMI

Use the create-image command to specify a block device mapping for an EBS-backed AMI. Use the register-image command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the `--block-device-mappings` parameter. You can specified arguments encoded in JSON directly on the command line or by referencing a JSON file, as shown here.

```
--block-device-mappings file://mapping.json
```

To add an instance store volume, use the following mapping. Note that NVMe instance store volumes are added automatically.

```
{
    "DeviceName": "device_name",
    "VirtualName": "ephemeral0"
}
```

To add an empty 100 GiB volume, use the following mapping.

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "VolumeSize": 100
    }
}
```

To add an EBS volume based on a snapshot, use the following mapping.

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "SnapshotId": "snap-1234567890abcdef0"
    }
}
```

To omit a mapping for a device, use the following mapping.

```
{
    "DeviceName": "device_name",
    "NoDevice": ""
}
```

PowerShell

Use the New-EC2Image cmdlet to specify a block device mapping for an EBS-backed AMI. Use the Register-EC2Image cmdlet to specify a block device mapping for an instance store-backed AMI.

Add the `-BlockDeviceMapping` option, specifying the updates in `bdm`:

```
-BlockDeviceMapping $bdm
```

The following mapping adds a volume based on a snapshot.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.SnapshotId = "snap-1234567890abcdef0"
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

The following mapping adds an empty 100 GB volume.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.VolumeSize = 100
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

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.

Console

To view the EBS volumes for an AMI

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.)

AWS CLI

To view the EBS volumes for an AMI

Use the describe-images command.

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
```

```
    --query Image[0].BlockDeviceMappings
```

PowerShell

To view the EBS volumes for an AMI

Use the Get-EC2Image cmdlet.

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).BlockDeviceMappings
```

Add block device mappings to Amazon EC2 instance

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.

Tasks

- Update the block device mapping when launching an instance
- Update the block device mapping of a running instance
- View the EBS volumes in an instance block device mapping
- View the instance block device mapping for instance store volumes

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.

Console

To update the volumes for an instance at launch

1. Follow the procedure to launch an instance, but don't launch the instance until you've completed the following steps to update the volumes.
2. (Optional) To add a volume, choose **Configure storage**, **Add new volume**. Select the volume size and volume type.
3. (Optional) To suppress a volume that was specified by the block device mapping of the AMI, choose **Configure storage**, **Remove**.
4. (Optional) To modify the configuration of an EBS volume, on the **Configure storage** pane, choose **Advanced**. Expand the information for the volume, and make whatever changes you need.
5. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

AWS CLI

To update the volumes for an instance at launch

Use the run-instances command with the `--block-device-mappings` option.

```
--block-device-mappings file://mapping.json
```

For example, suppose that an AMI block device mapping specifies the following:

- `/dev/xvda` - EBS root volume
- `/dev/sdh` - EBS volume created from `snap-1234567890abcdef0`
- `/dev/sdj` - Empty EBS volume with a size of `100`
- `/dev/sdb` - Instance store volume `ephemeral0`

Suppose that the following is the instance block device mapping in `mapping.json`.

```
[
    {
```

```
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "VolumeSize": 100
            }
        },
        {
            "DeviceName": "/dev/sdj",
            "NoDevice": ""
        },
        {
            "DeviceName": "/dev/sdh",
            "Ebs": {
                "VolumeSize": 300
            }
        },
        {
            "DeviceName": "/dev/sdc",
            "VirtualName": "ephemeral1"
        }
]
```

The instance block device mapping does the following:

- Overrides the size of the root volume, `/dev/xvda`, increasing it to 100 GiB.
- Prevents `/dev/sdj` from attaching to the instance.
- Overrides the size of `/dev/sdh`, increasing it to 300 GiB. Notice that you don't need to specify the snapshot ID again.
- Adds an ephemeral volume, `/dev/sdc`. If the instance type doesn't support multiple instance store volumes, this has no effect. If the instance type supports NVMe instance store volumes, they are automatically enumerated and included in the instance block device mapping and can't be overridden.

PowerShell

To update the volumes for an instance at launch

Use the `-BlockDeviceMapping` parameter with the New-EC2Instance cmdlet with the `-BlockDeviceMapping` parameter.

```
-BlockDeviceMapping $bdm
```

Suppose that the following is the instance block device mapping in `$bdm`.

```
$bdm = @()

$root = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$root.DeviceName = "/dev/xvda"
$ebs1 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs1.VolumeSize = 100
$root.Ebs = $ebs1
$bdm += $root

$sdj = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdj.DeviceName = "/dev/sdj"
$sdj.NoDevice = ""
$bdm += $sdj

$sdh = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdh.DeviceName = "/dev/sdh"
$ebs2 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs2.VolumeSize = 300
$sdh.Ebs = $ebs2
$bdm += $sdh

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral1"
$bdm += $sdc
```

The instance block device mapping does the following:

- Overrides the size of the root volume, `/dev/xvda`, increasing it to 100 GiB.
- Prevents `/dev/sdj` from attaching to the instance.
- Overrides the size of `/dev/sdh`, increasing it to 300 GiB. Notice that you don't need to specify the snapshot ID again.
- Adds an ephemeral volume, `/dev/sdc`. If the instance type doesn't support multiple instance store volumes, this has no effect. If the instance type supports NVMe instance store volumes, they are automatically enumerated and included in the instance block device mapping and can't be overridden.

Update the block device mapping of a running instance

You do not need to stop the instance before changing this attribute.

AWS CLI

To update the block device mapping of a running instance

Use the modify-instance-attribute command.

Add the `--block-device-mappings` option:

```
--block-device-mappings file://mapping.json
```

In `mapping.json`, specify the updates. For example, the following update changes the root device volume to persist.

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

PowerShell

To update the block device mapping of a running instance

Use the Edit-EC2InstanceAttribute cmdlet.

Add the `-BlockDeviceMapping` option:

```
-BlockDeviceMapping $bdm
```

In bdm, specify the updates. For example, the following update changes the root device volume to persist.

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
```

```
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

View the EBS volumes in an instance block device mapping

You can easily enumerate the EBS volumes mapped to an instance.

Console

To view the EBS volumes for an instance

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select the instance and look at the details displayed in the **Storage** tab. At a minimum, the following information is available for the root device:

   - **Root device type** (for example, **EBS**)
   - **Root device name** (for example, `/dev/xvda`)
   - **Block devices** (for example, `/dev/xvda`, `/dev/sdf`, and `/dev/sdj`)

   If the instance was launched with additional EBS volumes using a block device mapping, they appear under **Block devices**. Any instance store volumes do not appear on this tab.
4. To display additional information about an EBS volume, choose its volume ID to go to the volume page.

AWS CLI

To view the EBS volumes for an instance

Use the describe-instances command.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[*].Instances[0].BlockDeviceMappings
```

PowerShell

To view the EBS volumes for an instance

Use the Get-EC2Instance cmdlet.

```
(Get-EC2Instance -InstanceId i-0bac57d7472c89bac).Instances.BlockDeviceMappings
```

View the instance block device mapping for instance store volumes

The instance type determines the number and type of instance store volumes that are available to the instance. If the number of instance store volumes in a block device mapping exceeds the number of instance store volumes available to an instance, the additional volumes are ignored. To view the instance store volumes for your instance, run the **lsblk** command (Linux instances) or open **Windows Disk Management** (Windows instances). To learn how many instance store volumes are supported by each instance type, see Amazon EC2 instance type specifications.

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

Linux instances

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.

```
Node             SN                   Model
 Namespace
---------------- -------------------- ----------------------------------------
 ---------
/dev/nvme0n1     vol06afc3f8715b7a597 Amazon Elastic Block Store               1

/dev/nvme1n1     AWS2C1436F5159EB6614 Amazon EC2 NVMe Instance Storage         1
```

```
/dev/nvme2n1     AWSB1F4FF0C0A6C281EA Amazon EC2 NVMe Instance Storage         1
  ...
```

Windows instances

You can use Disk Management or PowerShell to list both EBS and instance store NVMe volumes. For more information, see the section called “Map NVME disks to volumes”.

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 Use instance metadata to manage your EC2 instance.

Linux instances

First, connect to your running instance. From the instance, use this query to get its block device mapping.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/block-device-mapping/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store–backed `m1.small` instance looks like this.

```
ami
ephemeral0
root
swap
```

The ami device is the root device as seen by the instance. The instance store volumes are named ephemeral[0-23]. The swap device is for the page file. If you've also mapped EBS volumes, they appear as ebs1, ebs2, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

Windows instances

First, connect to your running instance. From the instance, use this query to get its block device mapping.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store–backed m1.small instance looks like this.

```
ami
ephemeral0
root
swap
```

The ami device is the root device as seen by the instance. The instance store volumes are named ephemeral[0-23]. The swap device is for the page file. If you've also mapped EBS volumes, they appear as ebs1, ebs2, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

How volumes are attached and mapped for Amazon EC2 Windows instances

> **Note**
>
> This topic applies to Windows instances only.

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 Amazon EBS volume limits for Amazon EC2 instances.

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.

When you add a volume to your instance, you specify the device name that Amazon EC2 uses. For more information, see Device names for volumes on Amazon EC2 instances. 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.

Methods to map disks to EBS volumes

- Map NVMe disks on Amazon EC2 Windows instance to volumes
- Map non-NVMe disks on Amazon EC2 Windows instance to volumes

Map NVMe disks on Amazon EC2 Windows instance to volumes

With Nitro-based instances, EBS volumes are exposed as NVMe devices. This topic explains how to view the **NVMe disks** that are available to the Windows operating system on your instance. It also shows how to map those NVMe disks to the underlying Amazon EBS volumes and the device names specified for the block device mappings used by Amazon EC2.

Topics

List NVMe disks

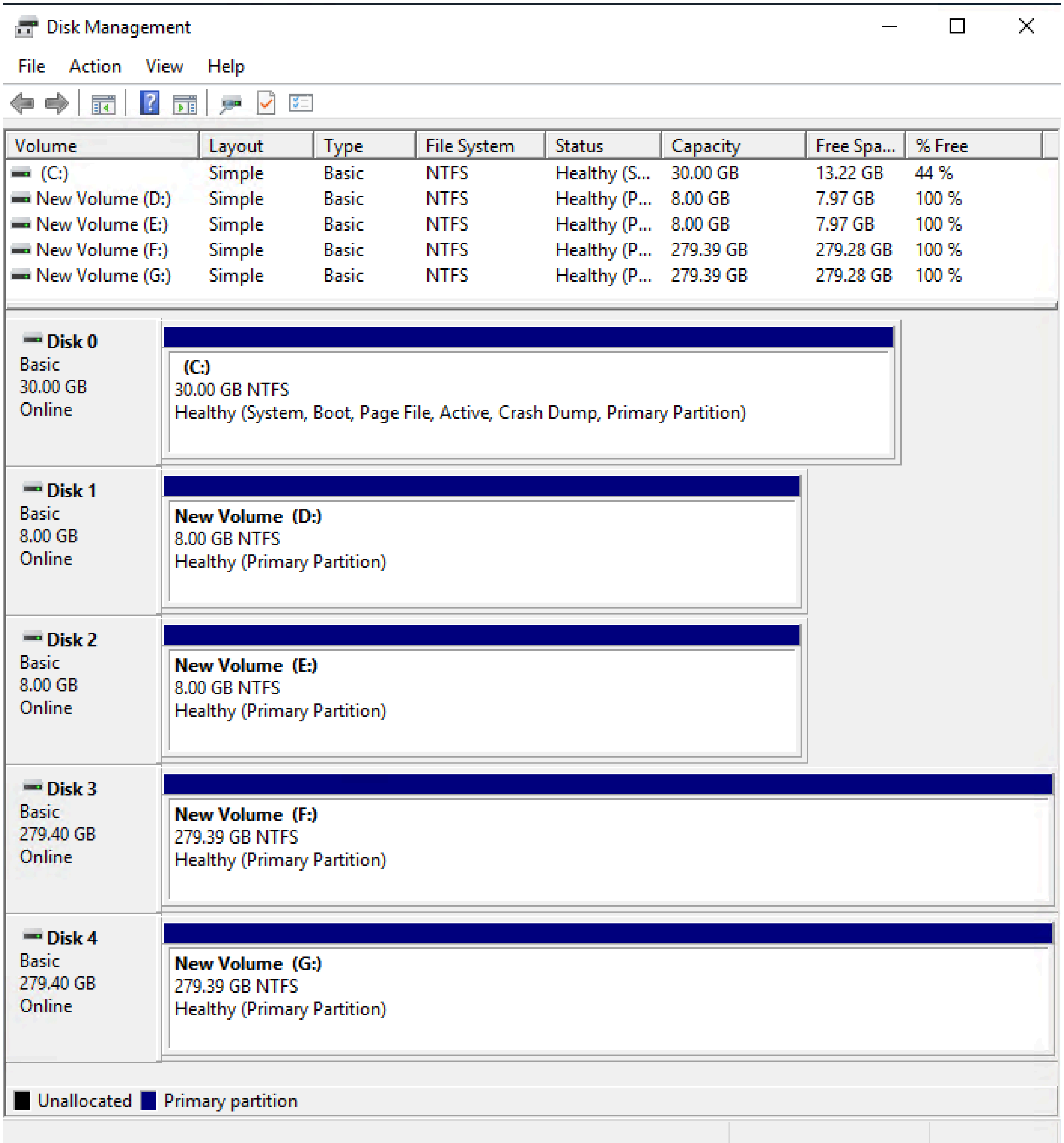
You can find the disks on your Windows instance using Disk Management or Powershell.

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 using RDP.
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.



PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume. It is intended for use with [Nitro-based instances](), 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.

```
# 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 -clike '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 -clike 'vol*'){

        $Device  = ((Get-EC2Volume -VolumeId $EbsVolumeId ).Attachment).Device
        $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]
        }
        $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-0a4064b39e5f534a2 xvdc      N/A
   3          1 F           AWS-6AAD8C2AEEE1193F0 Ephemeral N/A           Temporary
 Storage
   4          1 G           AWS-13E7299C2BD031A28 Ephemeral N/A           Temporary
 Storage
```

If you did not configure your credentials for Tools for Windows PowerShell on the Windows instance, the script cannot get the EBS volume ID and uses N/A in the `EbsVolumeId` column.

Map NVMe disks to volumes

You can use the Get-Disk command to map Windows disk numbers to EBS volume IDs.

```
PS C:\> Get-Disk
Number Friendly Name Serial Number                    HealthStatus
 OperationalStatus      Total Size Partition


                      Style
------ ------------- -------------                    ------------
 -----------------      ---------- ----------
3      NVMe Amazo... AWS6AAD8C2AEEE1193F0_00000001.   Healthy              Online
             279.4 GB MBR
4      NVMe Amazo... AWS13E7299C2BD031A28_00000001.   Healthy              Online
             279.4 GB MBR
2      NVMe Amazo... vol0a4064b39e5f534a2_00000001.   Healthy              Online
                 8 GB MBR
0      NVMe Amazo... vol03683f1d861744bc7_00000001.   Healthy              Online
                30 GB MBR
1      NVMe Amazo... vol082b07051043174b9_00000001.   Healthy              Online
                 8 GB MBR
```

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-03683f1d861744bc7
Device Name: sda1

Disk Number: 1
Volume ID: vol-082b07051043174b9
Device Name: xvdb

Disk Number: 2
Volume ID: vol-0a4064b39e5f534a2
Device Name: xvdc
```

Map non-NVMe disks on Amazon EC2 Windows instance to volumes

For instances launched 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. This topic explains how to view the **non-NVMe disks** that are available to the Windows operating system on your instance. It also shows how to map those non-NVMe disks to the underlying Amazon EBS volumes and the device names specified for the block device mappings used by Amazon EC2.

Note

If you launch an instance If your Windows AMI uses Red Hat PV drivers, you can update your instance to use the Citrix drivers. For more information, see the section called "Upgrade PV drivers".

Topics

List non-NVMe disks

You can find the disks on your Windows instance using Disk Management or PowerShell.

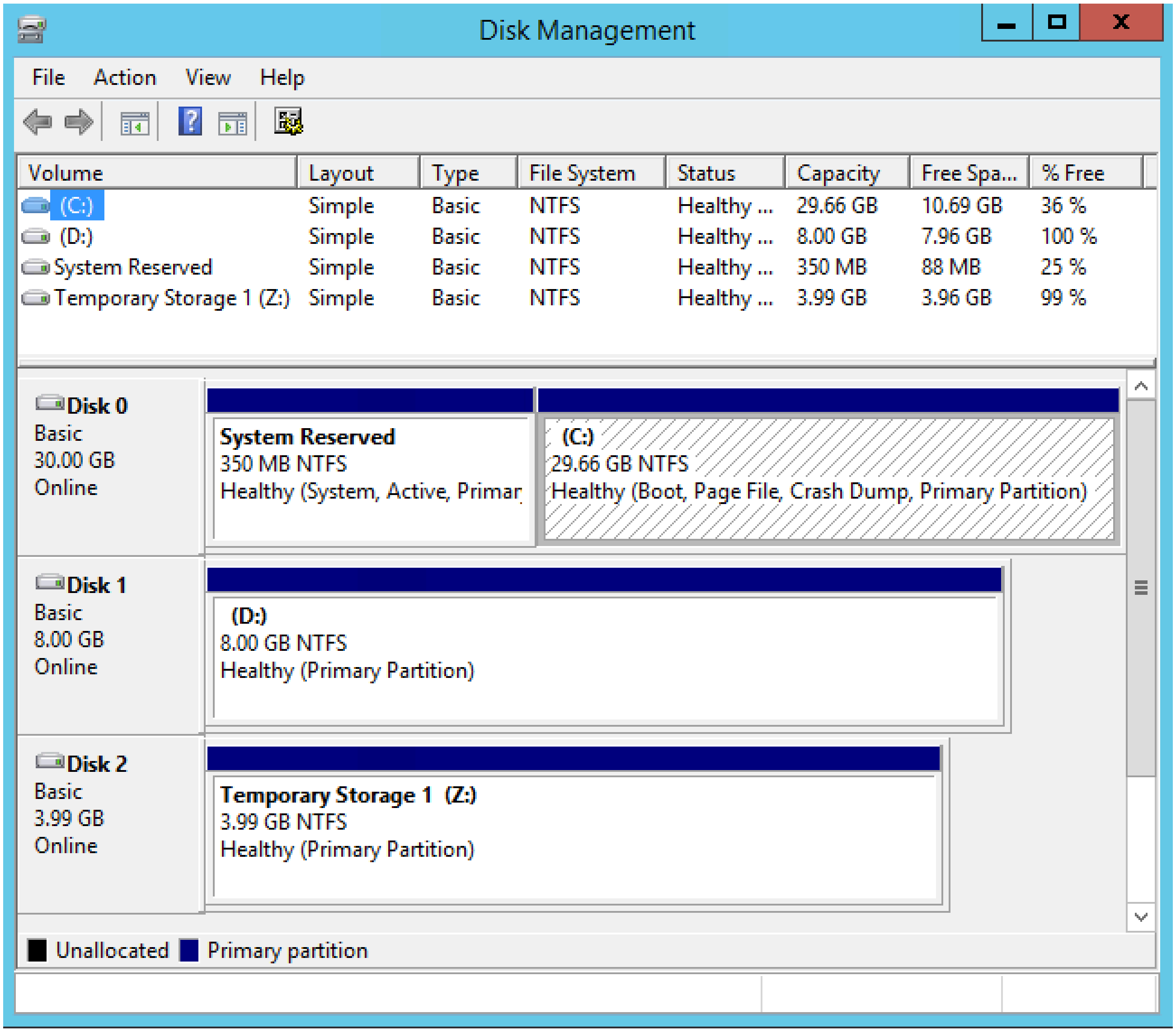
Disk Management

To find the disks on your Windows instance

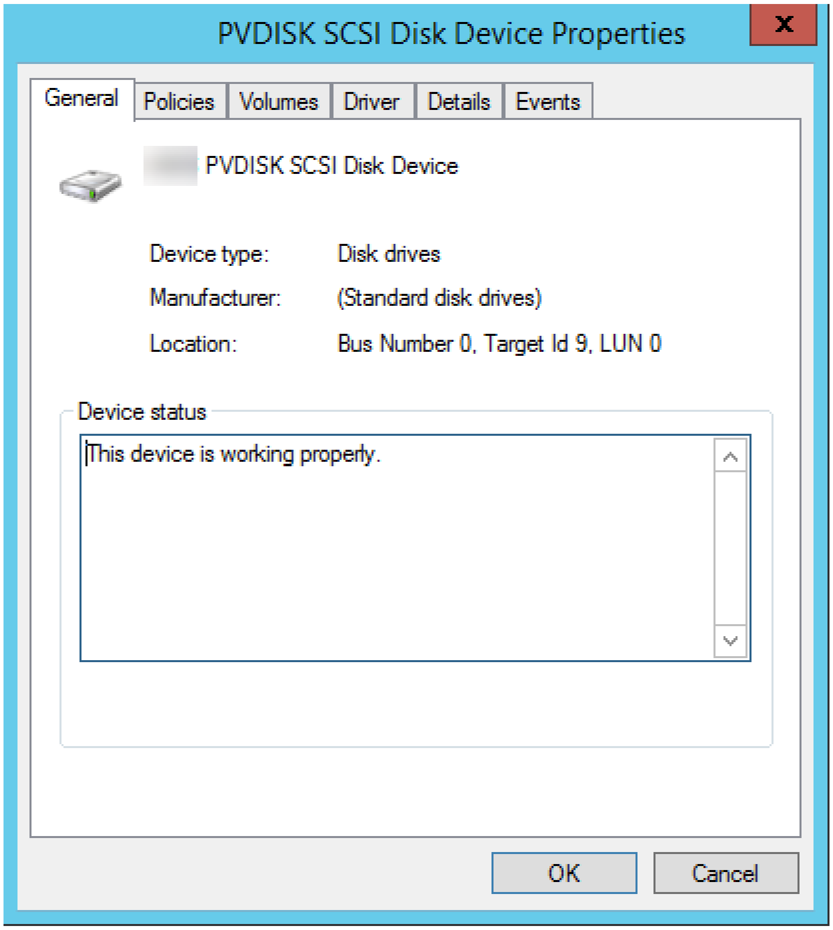
1. Log in to your Windows instance using Remote Desktop. For more information, see Connect to your Windows instance using RDP.
2. Start the Disk Management utility.

   On the taskbar, right-click the Windows logo, and then choose **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 non-NVMe disks to volumes](). 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`.



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 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
}

[string[]]$array1 = @()
[string[]]$array2 = @()
[string[]]$array3 = @()
[string[]]$array4 = @()

Get-WmiObject Win32_Volume | Select-Object Name, DeviceID | ForEach-Object {
  $array1 += $_.Name
  $array2 += $_.DeviceID
}

$i = 0
While ($i -ne ($array2.Count)) {
  $array3 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).SerialNumber) -replace "_[^ ]*$" -replace "vol", "vol-"
  $array4 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).FriendlyName)
  $i ++
}

[array[]]$array = $array1, $array2, $array3, $array4

Try {
  $InstanceId = Get-EC2InstanceMetadata -Category "InstanceId"
  $Region = Get-EC2InstanceMetadata -Category "Region" | Select-Object -
ExpandProperty SystemName
}
Catch {
```

```
  Write-Host "Could not access the instance Metadata using AWS Get-
EC2InstanceMetadata CMDLet.
Verify you have AWSPowershell SDK version '3.1.73.0' or greater installed and
 Metadata is enabled for this instance." -ForegroundColor Yellow
}
Try {
  $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
 $InstanceId).Instances.BlockDeviceMappings
  $VirtualDeviceMap = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping").GetEnumerator() | Where-Object { $_.Key -ne "ami" }
}
Catch {
  Write-Host "Could not access the AWS API, therefore, VolumeId is not available.
Verify that you provided your access keys or assigned an IAM role with adequate
 permissions." -ForegroundColor Yellow
}

Get-disk | ForEach-Object {
  $DriveLetter = $null
  $VolumeName = $null
  $VirtualDevice = $null
  $DeviceName = $_.FriendlyName

  $DiskDrive = $_
  $Disk = $_.Number
  $Partitions = $_.NumberOfPartitions
  $EbsVolumeID = $_.SerialNumber -replace "_[^ ]*$" -replace "vol", "vol-"
  if ($Partitions -ge 1) {
    $PartitionsData = Get-Partition -DiskId $_.Path
    $DriveLetter = $PartitionsData.DriveLetter | Where-object { $_ -notin @("",
 $null) }
    $VolumeName = (Get-PSDrive | Where-Object { $_.Name -in
 @($DriveLetter) }).Description | Where-object { $_ -notin @("", $null) }
  }
  If ($DiskDrive.path -like "*PROD_PVDISK*") {
    $BlockDeviceName = Convert-SCSITargetIdToDeviceName((Get-WmiObject -Class
 Win32_Diskdrive | Where-Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" +
 $DiskDrive.Number) }).SCSITargetId)
    $BlockDeviceName = "/dev/" + $BlockDeviceName
    $BlockDevice = $BlockDeviceMappings | Where-Object { $BlockDeviceName -like "*"
 + $_.DeviceName + "*" }
    $EbsVolumeID = $BlockDevice.Ebs.VolumeId
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
```

```
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON_EC2_NVME*") {
    $BlockDeviceName = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping")."ephemeral$((Get-WmiObject -Class Win32_Diskdrive | Where-
Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" + $DiskDrive.Number) }).SCSIPort - 2)"
    $BlockDevice = $null
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  ElseIf ($DiskDrive.path -like "*NETAPP*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $EbsVolumeID = "FSxN Volume"
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  Else {
    $BlockDeviceName = $null
    $BlockDevice = $null
  }
```

```
    New-Object PSObject -Property @{
      Disk          = $Disk;
      Partitions    = $Partitions;
      DriveLetter   = If ($DriveLetter -eq $null) { "N/A" } Else { $DriveLetter };
      EbsVolumeId   = If ($EbsVolumeID -eq $null) { "N/A" } Else { $EbsVolumeID };
      Device        = If ($BlockDeviceName -eq $null) { "N/A" } Else
 { $BlockDeviceName };
      VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
      VolumeName    = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
      DeviceName    = If ($DeviceName -eq $null) { "N/A" } Else { $DeviceName };
    }
} | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, DeviceName, VolumeName
```

Run the script as follows:

```
PS C:\> .\mapping.ps1
```

The following is example output.

```
Disk  Partitions  DriveLetter   EbsVolumeId            Device      VirtualDevice
 DeviceName             VolumeName
----  ----------  -----------   -----------            ------      -------------
 ----------             ----------
   0           1            C   vol-0561f1783298efedd  /dev/sda1   N/A
 NVMe Amazon Elastic B   N/A
   1           1            D   vol-002a9488504c5e35a  xvdb        N/A
 NVMe Amazon Elastic B   N/A
   2           1            E   vol-0de9d46fcc907925d  xvdc        N/A
 NVMe Amazon Elastic B   N/A
```

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 non-NVMe disks to volumes

The block device driver for the instance assigns the actual volume names when mounting volumes.

Mappings

- Instance store volumes

- EBS volumes

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 volume limits for EC2 instances.

Location	Device name
Bus Number 0, Target ID 78, LUN 0	xvdca
Bus Number 0, Target ID 79, LUN 0	xvdcb
Bus Number 0, Target ID 80, LUN 0	xvdcc
Bus Number 0, Target ID 81, LUN 0	xvdcd
Bus Number 0, Target ID 82, LUN 0	xvdce
Bus Number 0, Target ID 83, LUN 0	xvdcf
Bus Number 0, Target ID 84, LUN 0	xvdcg
Bus Number 0, Target ID 85, LUN 0	xvdch
Bus Number 0, Target ID 86, LUN 0	xvdci
Bus Number 0, Target ID 87, LUN 0	xvdcj
Bus Number 0, Target ID 88, LUN 0	xvdck
Bus Number 0, Target ID 89, LUN 0	xvdcl

EBS volumes

The following table describes how the Citrix PV and AWS PV drivers map non-NVME EBS volumes to Windows volumes.

Location	Device name
Bus Number 0, Target ID 0, LUN 0	/dev/sda1
Bus Number 0, Target ID 1, LUN 0	xvdb
Bus Number 0, Target ID 2, LUN 0	xvdc
Bus Number 0, Target ID 3, LUN 0	xvdd
Bus Number 0, Target ID 4, LUN 0	xvde
Bus Number 0, Target ID 5, LUN 0	xvdf
Bus Number 0, Target ID 6, LUN 0	xvdg
Bus Number 0, Target ID 7, LUN 0	xvdh
Bus Number 0, Target ID 8, LUN 0	xvdi
Bus Number 0, Target ID 9, LUN 0	xvdj
Bus Number 0, Target ID 10, LUN 0	xvdk
Bus Number 0, Target ID 11, LUN 0	xvdl
Bus Number 0, Target ID 12, LUN 0	xvdm
Bus Number 0, Target ID 13, LUN 0	xvdn
Bus Number 0, Target ID 14, LUN 0	xvdo
Bus Number 0, Target ID 15, LUN 0	xvdp
Bus Number 0, Target ID 16, LUN 0	xvdq
Bus Number 0, Target ID 17, LUN 0	xvdr
Bus Number 0, Target ID 18, LUN 0	xvds
Bus Number 0, Target ID 19, LUN 0	xvdt

Location	Device name
Bus Number 0, Target ID 20, LUN 0	xvdu
Bus Number 0, Target ID 21, LUN 0	xvdv
Bus Number 0, Target ID 22, LUN 0	xvdw
Bus Number 0, Target ID 23, LUN 0	xvdx
Bus Number 0, Target ID 24, LUN 0	xvdy
Bus Number 0, Target ID 25, LUN 0	xvdz

Torn write prevention on Amazon EC2 Linux instances

Note

Torn write prevention is supported with Linux instances only.

Torn write prevention is a block storage feature designed by AWS to improve the performance of your I/O-intensive relational database workloads and reduce latency without negatively impacting data resiliency. Relational databases that use InnoDB or XtraDB as the database engine, such as MySQL and MariaDB, will benefit from torn write prevention.

Typically, relational databases that use pages larger than the *power fail atomicity* of the storage device use *data logging* mechanisms to protect against torn writes. MariaDB and MySQL use a *doublewrite buffer* file to log data before writing it to data tables. In the event of incomplete or torn writes, as a result of operating system crashes or power loss during write transactions, the database can recover the data from the doublewrite buffer. The additional I/O overhead associated with writing to the doublewrite buffer impacts database performance and application latency, and it reduces the number transactions that can be processed per second. For more information about doublewrite buffer, see the MariaDB and MySQL documentation.

With *torn write prevention*, data is written to storage in *all-or-nothing* write transactions, which eliminates the need for using the doublewrite buffer. This prevents partial, or torn, data from being written to storage in the event of operating system crashes or power loss during write transactions.

The number of transactions processed per second can be increased by up to 30 percent, and write latency can be decreased by up to 50 percent, without compromising the resiliency of your workloads.

Pricing

There are no additional costs for using torn write prevention.

Contents

Block sizes for torn write prevention on Amazon EC2

Torn write prevention supports write operations for 4 KiB, 8 KiB, and 16 KiB blocks of data. The data block start logical block address (LBA) must be aligned to the respective block boundary size of 4 KiB, 8 KiB, or 16 KiB. For example, for 16 KiB write operations, the data block start LBA must be aligned to a block boundary size of 16 KiB.

The following table shows support across storage and instance types.

	4 KiB blocks	8 KiB blocks	16 KiB blocks
Instance store volumes	All NVMe instance store volumes attached to current generation I-family instances.	I4i, Im4gn, Is4gen, I7i, I7ie, I8g, and I8ge instances supported by AWS Nitro SSD.	
Amazon EBS volumes	All Amazon EBS volumes attached to Nitro-based instances.		

To confirm whether your instance and volume support torn write prevention, query to check if the instance supports torn write prevention and other details, like supported block and boundary sizes. For more information, see Check Amazon EC2 instance support for torn write prevention.

Requirements for using torn write prevention on Amazon EC2

For torn write prevention to work properly, an I/O operation must meet size, alignment, and boundary requirements, as specified in the `NTWPU`, `NTWGU`, `NTWBU` fields. You must configure your operating system to ensure that the specific storage subsystem (file system, LVM, RAID, etc) does not modify I/O properties down the storage stack, including block merges, splits, or block address relocation, before being submitted to the device.

Torn write prevent has been tested with the following configuration:

- An instance type and storage type that supports the required block size.
- Amazon Linux 2 with kernel version 5.10 or later.
- ext4 with `bigalloc` enabled and a cluster size of 16 KiB, and the most recent ext4 utilities (e2fsprogs 1.46.5 or later).
- `O_DIRECT` file access mode to bypass Linux kernel buffer cache.

> **Note**
>
> You do not need to disable I/O merging for MySQL and MariaDB workloads.

Check Amazon EC2 instance support for torn write prevention

To confirm whether your instance and volume supports torn write prevention, and to view the NVMe namespace vendor specific data that contains torn write prevention information, use the following command.

```
$ sudo nvme id-ns -v device_name
```

> **Note**
>
> The command returns the vendor-specific information in hex with ASCII interpretation. You might need to build a tool, similar to `ebsnvme-id`, into your applications that can read and parse the output.

For example, the following command returns the NVMe namespace vendor specific data that contains torn write prevention information for `/dev/nvme1n1`.

```
$ sudo nvme id-ns -v /dev/nvme1n1
```

If your instance and volume support torn write prevention, it returns the following AWS torn write prevention information in the NVMe namespace vendor specific data.

> **Note**
>
> The bytes in the following table represent the offset in bytes from the beginning of the NVMe namespace vendor specific data.

Bytes	Description
`0:31`	The name of the device attachment mount point, for example `/dev/xvda` . You provide this during volume attachment request and it can be used by the Amazon EC2 instance to create a symlink to the NVMe block device(`nvmeXn1`).
`32:63`	The volume ID. For example, `vol01234567890abcdef` . This field can be used to map the NVMe device to the attached volume.
`64:255`	Reserved for future use.
`256:257`	Namespace Torn Write Prevention Unit size (NTWPU). This field indicates the namespace specific size of the write operation guarantee d to be written atomically to the NVM during a power failure or error condition. This field is specified in logical blocks represented in zero based values.
`258:259`	Namespace Torn Write Prevention Granularity size (NTWPG). This field indicates the namespace specific size increments below `NTWPU` of the write operation guaranteed to be written atomically to the NVM during a power failure or error condition. That is, size should be `NTWPG * n <= NTWPU` where `n` is positive integer. The write operation LBA offset

Bytes	Description
	also must be aligned to this field. This field is specified in logical blocks represented in zero based values.
`260:263`	Namespace Torn Write Prevention Boundary size (NTWPB). This field indicates the atomic boundary size for this namespace for the `NTWPU` value. Writes to this namespace that cross atomic boundaries are not guaranteed to be written atomically to the NVM during a power failure or error condition. A value of `0h` indicates that there are no atomic boundaries for power fail or error conditions. All other values specify a size in terms of logical blocks using the same encoding as the `NTWPU` field.

Configure your workload on Amazon EC2 for torn write prevention

Torn write prevention is enabled by default on supported instance types with supported volumes. You do not need to enabled any additional settings to enable your volume or instance for torn write prevention.

> **Note**
>
> There is no performance impact on workloads that do not support torn write prevention. You do not need to make any changes for these workloads.
>
> Workloads that do support torn write prevention, but are not configured to use it, continue to use the doublewrite buffer and do not receive any performance benefits.

To configure your MySQL or MariaDB software stack to disable the doublewrite buffer and use torn write prevention, complete the following steps:

1. Configure your volume to use ext4 file system with the BigAlloc option and set the cluster size to 4 KiB, 8 KiB, or 16 KiB. Using BigAlloc with a cluster size of 4 KiB, 8 KiB, or 16 KiB ensures that the file system allocates files that align with the respective boundary.

   ```
   $ mkfs.ext4 -O bigalloc -C 4096|8192|16384 device_name
   ```

> **Note**
>
> For MySQL and MariaDB, you must use `-C 16384` to match the database page size. Setting allocation granularity to a value other than a multiple of the page size can result in allocations that might be mismatched with torn write prevention boundaries of the storage device.

For example:

```
$  mkfs.ext4 -O bigalloc -C 16384 /dev/nvme1n1
```

2. Configure InnoDB to use the `O_DIRECT` flushing method and turn off InnoDB doublewrite. Use your preferred text editor to open `/etc/my.cnf`, and update the `innodb_flush_method` and `innodb_doublewrite` parameters as follows:

```
innodb_flush_method=O_DIRECT
innodb_doublewrite=0
```

> **Important**
>
> If you are using Logical Volume Manager (LVM) or other storage virtualization layer, make sure that the starting offsets of the volumes are aligned on 16 KiB multiples. This is relative to the underlying NVMe storage to account for the metadata headers and superblocks used by the storage virtualization layer. If you add an offset to the LVM physical volume, it can cause misalignment between the file system allocations and the NVMe device's offsets, which would invalidate torn write prevention. For more information, see `--dataalignmentoffset` in the Linux manual page.

Application consistent Windows VSS based Amazon EBS snapshots

You can take application-consistent snapshots of all Amazon EBS volumes attached to your Amazon EC2 Windows instances using AWS Systems Manager Run Command. The snapshot process uses the Windows Volume Shadow Copy Service (VSS) to take EBS volume 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 based EBS snapshots. You only pay for EBS snapshots created by the backup process. For more information, see How am I billed for my Amazon EBS snapshots?

Note

Application consistent Windows VSS based snapshots are supported with Windows instances only.

Contents

What is VSS?

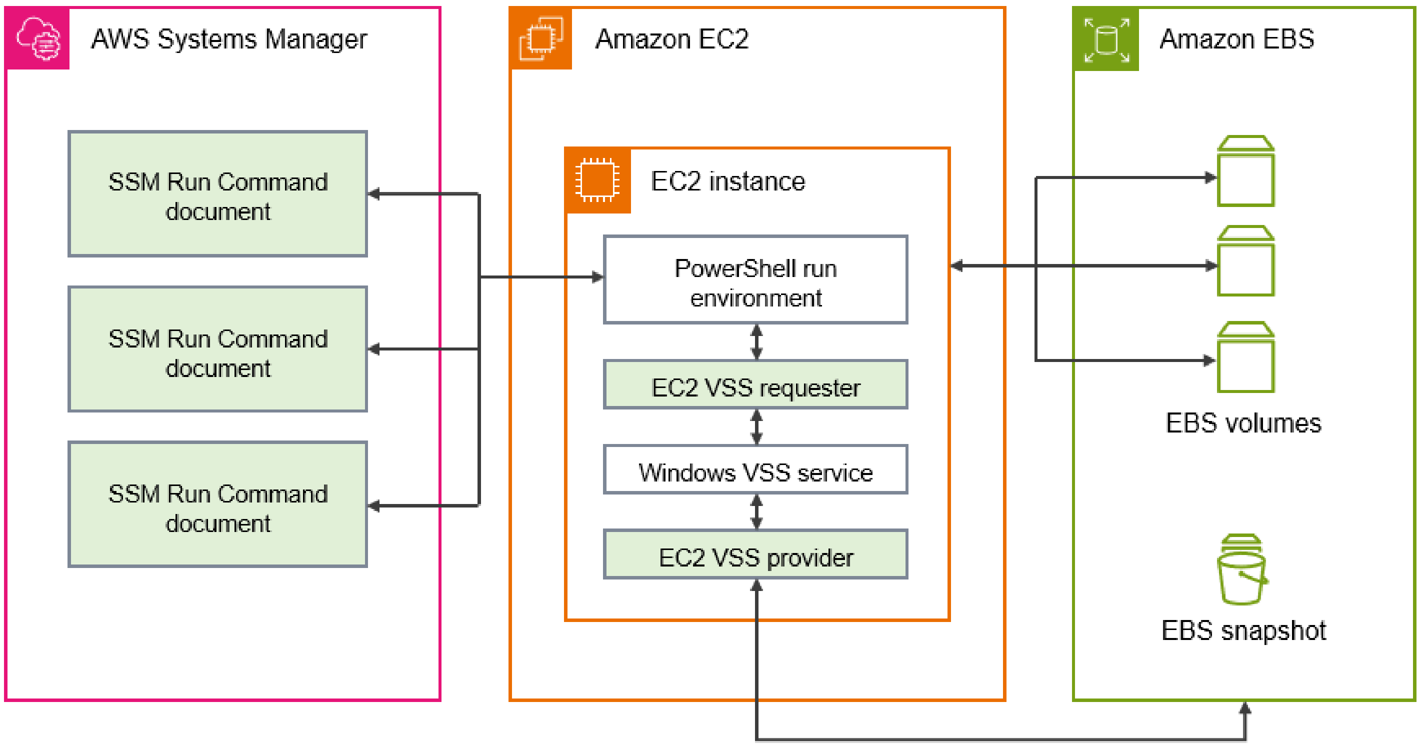
Volume Shadow Copy Service (VSS) is a backup and recovery technology included in Microsoft Windows. It can create backup copies, or snapshots, of computer files or volumes while they are in use. For more information, see Volume Shadow Copy Service.

To create an application-consistent snapshot, the following software components are involved.

- *VSS service* — Part of the Windows operating system
- *VSS requester* — The software that requests the creation of shadow copies
- *VSS writer* — Typically provided as part of an application, such as SQL Server, to ensure a consistent data set to back up

- *VSS provider* — The component that creates the shadow copies of the underlying volumes

The Windows VSS based Amazon EBS snapshot solution consists of multiple Systems Manager (SSM) Run Command documents that facilitate backup creation, and a Systems Manager Distributor package, called `AwsVssComponents`, that includes an *EC2 VSS requester* and an *EC2 VSS provider*. The `AwsVssComponents` package must be installed on EC2 Windows instances to take application-consistent snapshots of EBS volumes. The following diagram illustrates the relationship between these software components.



How the VSS based Amazon EBS snapshot solution works

The process for taking application-consistent, VSS based EBS snapshot scripts consists of the following steps.

1. Complete the Prerequisites to create Windows VSS based EBS snapshots.
2. Enter parameters for the `AWSEC2-VssInstallAndSnapshot` SSM document and run this document by using Run Command. For more information, see Run the AWSEC2-VssInstallAndSnapshot command document (recommended).
3. The Windows VSS service 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 based EBS snapshots successfully created by this process with **AppConsistent:true**.
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, as described in Use the AWS VSS solution to restore data for your instance.

Prerequisites to create Windows VSS based EBS snapshots

You can create VSS based EBS snapshots with Systems Manager Run Command, AWS Backup, or Amazon Data Lifecycle Manager. The following prerequisites apply for all solutions.

System requirements

Ensure that your EC2 Windows instance meets all of the system requirements to create VSS based snapshots, including supported versions of the Windows operating system, .NET framework, PowerShell, AWS Tools for Windows PowerShell, and the AWS Systems Manager Agent.

IAM permissions

The IAM role that's attached to your Amazon EC2 Windows instance must have permission to create application-consistent snapshots with VSS. To grant the necessary permissions, you can attach the `AWSEC2VssSnapshotPolicy` managed policy to your instance profile.

VSS components

To create application-consistent snapshots on Windows operating systems, the `AwsVssComponents` package must be installed on the instance. The package contains an on-instance EC2 VSS Agent that functions as the VSS requester, and an EC2 VSS provider for EBS volumes.

System requirements

Install the Systems Manager Agent

VSS is orchestrated by the Systems Manager Agent using PowerShell. Ensure that you have installed SSM Agent version `3.0.502.0` or later on your EC2 instance. If you are already using an older version of the SSM Agent, update it using Run Command. For more information, see [Setting up Systems Manager for Amazon EC2 instances] and [Working with SSM Agent on Amazon EC2 instances for Windows Server] in the *AWS Systems Manager User Guide*.

Amazon EC2 Windows instance requirements

VSS based EBS snapshots are supported for instances running Windows Server 2016 and later.

.NET Framework version

The `AwsVssComponents` package requires .NET Framework version `4.6` or later. Windows operating system versions prior to Windows Server 2016 default to an earlier version of the .NET Framework. If your instance uses an earlier version of the .NET Framework, you must install version `4.6` or later using Windows Update.

AWS Tools for Windows PowerShell version

Ensure that your instance is running AWS Tools for Windows PowerShell version `3.3.48.0` or later. To check your version, run the following command in the PowerShell terminal on the instance.

```
C:\> Get-AWSPowerShellVersion
```

If you need to update AWS Tools for Windows PowerShell on your instance, see [Installing the AWS Tools for Windows PowerShell] in the *AWS Tools for PowerShell User Guide*.

Windows PowerShell version

Ensure that your instance is running Windows PowerShell major version 3, 4, or 5. To check your version, run the following command in a PowerShell terminal on the instance.

```
C:\> $PSVersionTable.PSVersion
```

PowerShell language mode

Ensure that your instance has the PowerShell language mode set to `FullLanguage`. For more information, see [about_Language_Modes] in the Microsoft documentation.

Use an IAM managed policy to grant permissions for VSS based snapshots

The AWSEC2VssSnapshotPolicy managed policy enables Systems Manager to perform the following actions on your Windows instance:

- Create and tag EBS snapshots
- Create and tag Amazon Machine Images (AMIs)
- Attach metadata, such as the device ID, to the default snapshot tags that VSS creates.

This topic covers permission details for the VSS managed policy, and how to attach it to your EC2 instance profile IAM role.

Contents

AWSEC2VssSnapshotPolicy managed policy details

An AWS managed policy is a standalone policy that Amazon provides for AWS customers. AWS managed policies are designed to grant permissions for common use cases. You can't change the permissions that are defined in AWS managed policies. However, you can copy the policy and use it as a baseline for a customer managed policy that is specific to your use case.

For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

To use the **AWSEC2VssSnapshotPolicy** managed policy, you can attach it to the IAM role that's attached to your EC2 Windows Instances. This policy enables the EC2 VSS solution to create and add tags to Amazon Machine Images (AMIs) and EBS Snapshots. To attach the policy, see Attach the VSS snapshot managed policy to your instance profile role.

Permissions granted by AWSEC2VssSnapshotPolicy

The **AWSEC2VssSnapshotPolicy** policy includes the following Amazon EC2 permissions to allow Amazon EC2 to create and manage VSS snapshots on your behalf. You can attach this managed policy to the IAM instance profile role that you use for your EC2 Windows instances.

- **ec2:CreateTags** – Add tags to EBS snapshots and AMIs to help identify and categorize the resources.

- **ec2:DescribeInstanceAttribute** – Retrieve the EBS volumes and corresponding block device mappings that are attached to the target instance.
- **ec2:CreateSnapshots** – Create snapshots of EBS volumes.
- **ec2:CreateImage** – Create an AMI from a running EC2 instance.
- **ec2:DescribeImages** – Retrieve the information for EC2 AMIs and snapshots.
- **ec2:DescribeSnapshots** – Determine the create time and status of snapshots to verify application consistency.

> **Note**
>
> To view permission details for this policy, see AWSEC2VssSnapshotPolicy in the *AWS Managed Policy Reference*.

Streamline permissions for specific use cases - advanced

The `AWSEC2VssSnapshotPolicy` managed policy includes permissions for all of the ways that you can create VSS based snapshots. You can create a custom policy that includes only the permissions that you need.

Use case: Create AMI, Use case: Use AWS Backup service

If you exclusively use the `CreateAmi` option, or if you create VSS based snapshots only through the AWS Backup service, then you can streamline the policy statements as follows.

- Omit policy statements identified by the following statement IDs (SIDs):
  - `CreateSnapshotsWithTag`
  - `CreateSnapshotsAccessInstance`
  - `CreateSnapshotsAccessVolume`
- Adjust the `CreateTagsOnResourceCreation` statement as follows:
  - Remove `arn:aws:ec2:*:*:snapshot/*` from the resources.
  - Remove `CreateSnapshots` from the `ec2:CreateAction` condition.
- Adjust the `CreateTagsAfterResourceCreation` statement to remove `arn:aws:ec2:*:*:snapshot/*` from the resources.

- Adjust the `DescribeImagesAndSnapshots` statement to remove `ec2:DescribeSnapshots` from the statement action.

Use case: Snapshot only

If you don't use the `CreateAmi` option, then you can streamline the policy statements as follows.

- Omit policy statements identified by the following statement IDs (SIDs):
  - `CreateImageAccessInstance`
  - `CreateImageWithTag`
- Adjust the `CreateTagsOnResourceCreation` statement as follows:
  - Remove `arn:aws:ec2:*:*:image/*` from the resources.
  - Remove `CreateImage` from the `ec2:CreateAction` condition.
- Adjust the `CreateTagsAfterResourceCreation` statement to remove `arn:aws:ec2:*:*:image/*` from the resources.
- Adjust the `DescribeImagesAndSnapshots` statement to remove `ec2:DescribeImages` from the statement action.

> **Note**
>
> To ensure that your customized policy performs as expected, we recommend that you regularly review and incorporate updates to the managed policy.

Attach the VSS snapshot managed policy to your instance profile role

To grant permissions for VSS based snapshots for your EC2 Windows instance, you can attach the **AWSEC2VssSnapshotPolicy** managed policy to your instance profile role as follows. It's important to ensure that your instance meets all System requirements.

> **Note**
>
> To use the managed policy, your instance must have the `AwsVssComponents` package version `2.3.1` or later installed. For version history, see AwsVssComponents package versions.

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles** to see a list of IAM roles that you have access to.
3. Select the **Role name** link for the role that's attached to your instance. This opens the role detail page.
4. To attach the managed policy, choose **Add permissions**, located in the upper right corner of the list panel. Then select **Attach policies** from the dropdown list.
5. To streamline results, enter the policy name in the search bar (`AWSEC2VssSnapshotPolicy`).
6. Select the checkbox next to the name of the policy to attach, and choose **Add permissions**.

Manage VSS components package for Windows VSS based EBS snapshots

Before you create VSS based EBS snapshots, ensure that you have the latest version of the VSS components package installed on your Windows instance. There are several ways that you can install the `AwsVssComponents` package onto an existing instance, as follows:

- (Recommended) Run the AWSEC2-VssInstallAndSnapshot command document (recommended). This automatically installs or updates if needed every time it runs.
- Manually install the VSS components on an EC2 Windows instance.
- Update the VSS components package on your EC2 Windows instance.

You can also create an AMI with EC2 Image Builder that uses the `aws-vss-components-windows` managed component to install the `AwsVssComponents` package for the image. The managed component uses AWS Systems Manager Distributor to install the package. After Image Builder creates the image, every instance that you launch from the associated AMI will have the VSS package installed on it. For more information about how you can create an AMI with the VSS package installed, see Distributor package managed components for Windows in the *EC2 Image Builder User Guide*.

Contents

Manually install the VSS components on an EC2 Windows instance

Your EC2 Windows instance must have VSS components installed before you can create application-consistent snapshots with Systems Manager. If you don't run the `AWSEC2-VssInstallAndSnapshot` command document to automatically install or update the package every time you create application-consistent snapshots, you must manually install the package.

You must also install manually if you plan to use one of the following methods to create application-consistent snapshots from your EC2 instance.

- Create VSS snapshots using AWS Backup
- Create VSS snapshots using Amazon Data Lifecycle Manager

If you need to perform a manual install, we recommend that you use the latest AWS VSS component package to improve the reliability and performance of application-consistent snapshots on your EC2 Windows instances.

> **Note**
>
> To automatically install or update the `AwsVssComponents` package whenever you create application-consistent snapshots, we recommend that you use Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` document. For more information, see Run the AWSEC2-VssInstallAndSnapshot command document (recommended).

To install the VSS components on an Amazon EC2 Windows instance, follow the steps for your preferred environment.

Console

To install the VSS components using SSM Distributor

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
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**, enter a version or 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 user performing this task. For more information, see [Configure EC2 instance permissions](#) 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](#).

11. Choose **Run**.

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 requester 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

Run the following command to download and install the required VSS components for Systems Manager.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"action":["Install"],"name":["AwsVssComponents"]}'
```

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 requester 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

Run the following command to download and install the required VSS components for Systems Manager.

```
Send-SSMCommand `
    -DocumentName "AWS-ConfigureAWSPackage" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'action'='Install';'name'='AwsVssComponents'}
```

Verify the signature on AWS VSS components

Use the following procedure to verify the signature on the `AwsVssComponents` package.

1. Connect to your Windows instance. For more information, see Connect to your Windows instance using RDP.
2. Navigate to C:\Program Files\Amazon\AwsVssComponents.
3. Open the context (right-click) menu for `ec2-vss-agent.exe`, and then choose **Properties**.
4. Navigate to the **Digital Signatures** tab and verify that the name of the signer is Amazon Web Services Inc.
5. Use the preceding steps to verify the signature on `Ec2VssInstaller` and `Ec2VssProvider.dll`.

Update the VSS components package on your EC2 Windows instance

We recommend that you keep the VSS components updated with the latest recommended version. There are several different ways that you can update components when a new version of the `AwsVssComponents` package is released.

Update methods

- You can repeat the steps described in Manually install the VSS components on an EC2 Windows instance when a new version of the AWS VSS components is released.
- You can configure a Systems Manager State Manager association to automatically download and install new or updated VSS components when the `AwsVssComponents` package becomes available.
- You can automatically install or update the `AwsVssComponents` package whenever you create application-consistent snapshots, when you use Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` document.

> **Note**
>
> We recommend that you use Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` command document, which automatically installs or updates the `AwsVssComponents` package before it creates the application-consistent snapshots. For more information, see Run the AWSEC2-VssInstallAndSnapshot command document (recommended).

To create a Systems Manager State Manager association, follow the steps for your preferred environment.

Console

When you create a Systems Manager State Manager association, there are two options for updating the `AwsVssComponents` package, as follows:

Uninstall and reinstall

This method downloads and installs the package with no additional prerequisites.

In-place update

This performs an in-place update for the package, and has the following prerequisites:

- The SSM Agent version that's installed on the instance must be version `3.3.808.0` or later. For more information, see Working with SSM Agent on EC2 instances for Windows Server in the *AWS Systems Manager User Guide*.
- If specified, the `AwsVssComponents` package version must be version `2.5.0` or later. Earlier versions don't support in-place update.

> **Note**
>
> if your instance doesn't meet these prerequisites, in-place update will fail. Use the **Uninstall and reinstall** option instead.

To create a State Manager association

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.

2. In the navigation pane, choose **State Manager**.

   Or, if the Systems Manager homepage opens first, open the navigation pane and then choose **State Manager**.

3. Choose **Create association**.
4. In the **Name** field, enter a descriptive name.
5. In the **Document** list, choose **AWS-ConfigureAWSPackage**.
6. In the **Parameters** section, choose **Install** from the **Action** list.
7. For **Installation type**, choose either **Uninstall and reinstall** or **In-place update**.
8. In the **Name** field, enter `AwsVssComponents`. You can keep the **Version** and **Additional Arguments** fields empty.
9. In the **Targets** section, choose an option.

   > **Note**
   >
   > If you choose to target instances by using tags, and you specify tags that map to Linux instances, the association succeeds on the Windows instance but fails on the Linux instances. The overall status of the association shows **Failed**.

10. In the **Specify schedule** section, choose an option.
11. In the **Advanced options** section, for **Compliance severity**, choose a severity level for the association. For more information, see Learn about association compliance. For **Change Calendars**, select a preconfigured change calendar. For more information, see about AWS Systems Manager Change Calendar.
12. For **Rate control**, do the following:

    - For **Concurrency**, specify either a number or a percentage of managed nodes on which to run the command at the same time.
    - For **Error threshold**, specify when to stop running the command on other managed nodes after it fails on either a number or a percentage of nodes.

13. (Optional) For **Output options**, to save the command output to a file, select **Enable writing output to S3**. Enter the bucket and prefix (folder) names in the boxes.
14. Choose **Create association**, and then choose **Close**. The system attempts to create the association on the instances and immediately apply the state.

> **Note**
>
> If EC2 instances for Windows Server show a status of **Failed**, verify that the SSM Agent is running on the instance, and verify that the instance is configured with an AWS Identity and Access Management (IAM) role for Systems Manager. For more information, see [Setting up AWS Systems Manager](#).

AWS CLI

Use the [create-association](#) command to update a Distributor package on a schedule without taking the associated application offline. Only new or updated files in the package are replaced.

To create a State Manager association

Run the following command to create an association. The value of `--name`, the document name, is always `AWS-ConfigureAWSPackage`. The following command uses the key `InstanceIds` to specify target instances.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters action=Install,installationType="Uninstall and
 reinstall",name=AwsVssComponents \
    --targets Key=InstanceIds,Values=i-1234567890abcdef0,i-000011112222abcde
```

PowerShell

To create a State Manager association

Use the [New-SSMAssociation](#) cmdlet.

```
New-SSMAssociation `
    -Name "AWS-ConfigureAWSPackage" `
    -Parameter  @{
        "action" = "Install"
        "installationType" = "Uninstall and reinstall"
        "name" = "AwsVssComponents"
    } `
    -Target @{
        "Key" = "InstanceIds"
        "Values" = @("i-1234567890abcdef0", "i-000011112222abcde")
```

```
    }
```

Create VSS based EBS snapshots for your EC2 Windows instance

After you've met all of the Prerequisites to create Windows VSS based EBS snapshots, you can use any of the following methods to create VSS based snapshots from your EC2 instance.

AWS Systems Manager command documents

Use Systems Manager command documents to create VSS based snapshots.

To automate backups, you can create an AWS Systems Manager maintenance window task that uses the `AWSEC2-VssInstallAndSnapshot` command document. For more information, see Working with Maintenance Windows (Console) in the *AWS Systems Manager User Guide*.

AWS Backup

You can create a VSS backup when using AWS Backup by enabling VSS in the console or CLI. For more information, see Creating Windows VSS backups in the *AWS Backup Developer Guide*.

> **Note**
>
> AWS Backup doesn't automatically install the `AwsVssComponents` package on your instance. You must perform a manual install on your instance. For more information, see Manually install the VSS components on an EC2 Windows instance.

Amazon Data Lifecycle Manager

You can create VSS snapshots using Amazon Data Lifecycle Manager by enabling pre and post scripts in your snapshot lifecycle policies. For more information, see Automating application-consistent snapshots in the *Amazon EBS User Guide*.

> **Note**
>
> Amazon Data Lifecycle Manager doesn't automatically install the `AwsVssComponents` package on your instance. You must perform a manual install on your instance. For more information, see Manually install the VSS components on an EC2 Windows instance.

Use Systems Manager command documents to create VSS based snapshots

You can use AWS Systems Manager command documents to create VSS based snapshots. The following content introduces the command documents that are available, and the runtime parameters that the documents use to create your snapshots.

Before you use any of the Systems Manager command documents, ensure that you've met all Prerequisites to create Windows VSS based EBS snapshots.

Topics

Parameters for Systems Manager VSS snapshot documents

The Systems Manager documents that create VSS snapshots all use the following parameters, except where noted:

AmiName (string, optional)

If the **CreateAmi** option is set to `True`, specify the name of the AMI that the backup creates.

description (string, optional)

Specify a description for the snapshots or image that this process creates.

CollectDiagnosticLogs (string, optional)

To collect more information during snapshot and AMI creation steps, set this parameter to "`True`". The default value for this parameter is "`False`". Consolidated diagnostic logs are saved as a `.zip` format archive at the following location on your instance:

`C:\ProgramData\Amazon\AwsVss\Logs\timestamp.zip`

CopyOnly (string, optional)

If you use the native SQL Server backup in addition to AWS VSS, performing a Copy-only backup prevents AWS VSS from breaking the native differential backup chain. To perform a Copy-only backup operation, set this parameter to `True`.

The default value for this parameter is `False`, which causes AWS VSS to perform a full backup operation.

CreateAmi (string, optional)

To create a VSS based Amazon Machine Image (AMI) to back up your instance, set this parameter to `True`. The default value for this parameter is `False`, which backs up your instance with an EBS snapshot instead.

For more information about creating an AMI from an instance, see Create an Amazon EBS-backed AMI.

executionTimeout (string, optional)

Specify the maximum time in seconds to run the snapshot creation process on the instance, or to create an AMI from the instance. Increasing this timeout allows the command to wait longer for VSS to start its freeze and complete tagging of the resources it creates. This timeout only applies to the snapshot or AMI creation steps. The initial step to install or update the `AwsVssComponents` package is not included in the timeout.

ExcludeBootVolume (string, optional)

This setting excludes boot volumes from the backup process if you create snapshots. To exclude boot volumes from your snapshots, set **ExcludeBootVolume** to `True`, and **CreateAmi** to `False`.

If you create an AMI for your backup, this parameter should be set to `False`. The default value for this parameter is `False`.

NoWriters (string, optional)

To exclude application VSS writers from the snapshot process, set this parameter to `True`. Excluding application VSS writers can help you resolve conflicts with third-party VSS backup components. The default value for this parameter is `False`.

If `SaveVssMetadata` is `True`, this parameter must be set to `False`.

SaveVssMetadata (string, optional)

To save VSS metadata files during every snapshot, set this parameter to `True`. The default value is `False`. VSS Metadata files help provide insights into which components or writers were included in a backup operation, and the associated files for each component.

Metadata files have the associated snapshot set id in their names. You can find them at the following location on your instance:

```
C:\ProgramData\Amazon\AwsVss\VssMetadata\
```

Warning

- Saving VSS metadata files requires `AwsVssComponents` package version 2.4.0 or later. If your instance has an earlier version installed, setting `SaveVssMetadata` to `True` causes the snapshot creation to fail.
- The `NoWriters` and `SaveVssMetadata` parameters are mutually exclusive. If both are set to `True` then snapshot creation fails.

tags (string, optional)

We recommend that you tag your snapshots and images to help you locate and manage your resources, for example, to restore volumes from a list of snapshots. The system adds the `Name` key, with a blank value where you can specify the name that you want to apply to your output snapshots or images.

If you want to specify additional tags, separate tags with a semicolon in between. For example, `Key=Environment,Value=Test;Key=User,Value=TestUser1`.

Note

Tag keys and values must only contain alphanumeric characters and the following special characters: `() ./\-"'@_+:={}`.

By default, the system adds the following reserved tags for VSS based snapshots and images.

- **Device** – For VSS based snapshots, this is the device name of the EBS volume that the snapshot captures.
- **AppConsistent** – This tag indicates the successful creation of a VSS based snapshot or AMI.
- **AwsVssConfig** – This identifies snapshots and AMIs that are created with VSS enabled. The tag includes meta information such as the `AwsVssComponents` version, and the Snapshot Set ID.

Warning

Specifying any of these reserved tags in your parameter list will cause an error.

VssVersion (string, optional)

For the `AWSEC2-VssInstallAndSnapshot` document only, you can specify the `VssVersion` parameter to install a specific version of `AwsVssComponents` package on your instance. Leave this parameter blank to install the recommended default version.

If the specified version of the `AwsVssComponents` package is already installed, the script skips the install step and moves on to the backup step. For a list of `AwsVssComponents` package versions and operating support, see AWS VSS solution version history.

Run Systems Manager VSS snapshot command documents

You can create VSS based EBS snapshots with AWS Systems Manager command documents as follows.

Run the AWSEC2-VssInstallAndSnapshot command document (recommended)

When you use AWS Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` document, the script runs the following steps.

1. The script first installs or updates the `AwsVssComponents` package on your instance, depending on whether it's already installed.
2. The script creates the application-consistent snapshots after the first step completes.

To run the `AWSEC2-VssInstallAndSnapshot` document, follow the steps for your preferred environment.

Console

Create VSS based EBS snapshots from the console

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. Select **Run Command** from the navigation pane. This shows a list of commands that are currently running in your account, if applicable.
3. Choose **Run command**. This opens a list of command documents that you have access to.
4. Select `AWSEC2-VssInstallAndSnapshot` from the list of command documents. To streamline results, you can enter all or part of the document name. You can also filter by the owner, by platform types, or by tags.

When you select a command document, details populate below the list.

5. Select `Default version at runtime` from the **Document version** list.
6. Configure the **Command parameters** to define how `AWSEC2-VssInstallAndSnapshot` will install the `AwsVssComponents` package and back up with VSS snapshots or an AMI. For parameter details, see Parameters for Systems Manager VSS snapshot documents.
7. For **Target selection**, specify tags or select instances manually to identify the instances on which to run this operation.

> **Note**
>
> If you select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? for troubleshooting tips.

8. For additional parameters to define Systems Manager Run Command behavior such as **Rate control**, enter values as described in Running commands from the console.
9. 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.

AWS CLI

You can run the following commands in the AWS CLI to create VSS based EBS snapshots and get the status of your snapshot creation.

Create VSS based EBS snapshots

Run the following command to create VSS based EBS snapshots. To create the snapshots, you must identify the instances with the `--instance-ids` parameter. For more information about other parameters that you can use, see Parameters for Systems Manager VSS snapshot documents.

```
aws ssm send-command \
```

```
 --document-name "AWSEC2-VssInstallAndSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"],"VssVersion":[""]}'
```

If successful, the command document 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.

Get command status

To get the current status of the snapshots, run the following command using the command ID returned from **send-command**.

```
aws ssm get-command-invocation
 --instance-ids "i-01234567890abcdef" \
 --command-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
 --plugin-name "CreateVssSnapshot"
```

PowerShell

Run the following commands with AWS Tools for Windows PowerShell to create VSS based EBS snapshots and get the current runtime status for the creation of your output. Specify parameters described in the prior list to modify the behavior of the snapshot process.

Create VSS based EBS snapshots with Tools for Windows PowerShell

Run the following command to create VSS based EBS snapshots or AMIs.

```
Send-SSMCommand -DocumentName "AWSEC2-VssInstallAndSnapshot" -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value';'VssVersion'=''}
```

Get command status

To get the current status of the snapshots, run the following command using the command ID returned from **Send-SSMCommand**.

```
Get-SSMCommandInvocationDetail -InstanceId "i-01234567890abcdef" -CommandId
 "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" -PluginName "CreateVssSnapshot"
```

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.

Run the AWSEC2-CreateVssSnapshot command document

To run the `AWSEC2-CreateVssSnapshot` document, follow the steps for your preferred environment.

Console

Create VSS based EBS snapshots from the console

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. Select **Run Command** from the navigation pane. This shows a list of commands that are currently running in your account, if applicable.
3. Choose **Run command**. This opens a list of command documents that you have access to.
4. Select `AWSEC2-CreateVssSnapshot` from the list of command documents. To streamline results, you can enter all or part of the document name. You can also filter by the owner, by platform types, or by tags.

   When you select a command document, details populate below the list.
5. Select `Default version at runtime` from the **Document version** list.
6. Configure the **Command parameters** to define how `AWSEC2-CreateVssSnapshot` will back up with VSS snapshots or an AMI. For parameter details, see Parameters for Systems Manager VSS snapshot documents.
7. For **Target selection**, specify tags or select instances manually to identify the instances on which to run this operation.

   > **Note**
   >
   > If you select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? for troubleshooting tips.

8. For additional parameters to define Systems Manager Run Command behavior such as **Rate control**, enter values as described in [Running commands from the console](#).
9. 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.

AWS CLI

You can run the following command in the AWS CLI to create VSS based EBS snapshots.

Create VSS based EBS snapshots

Run the following command to create VSS based EBS snapshots. To create the snapshots, you must identify the instances with the `--instance-ids` parameter. For more information about other parameters that you can use, see [Parameters for Systems Manager VSS snapshot documents](#).

```
aws ssm send-command \
 --document-name "AWSEC2-CreateVssSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"]}'
```

If successful, the command document 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.

PowerShell

Run the following command with AWS Tools for Windows PowerShell to create VSS based EBS snapshots.

Create VSS based EBS snapshots with Tools for Windows PowerShell

Run the following command to create VSS based EBS snapshots. To create the snapshots, you must identify the instances with the `InstanceId` parameter. You can specify more than one instance to create snapshots for. For more information about other parameters that you can use, see [Parameters for Systems Manager VSS snapshot documents](#).

```
Send-SSMCommand -DocumentName AWSEC2-CreateVssSnapshot -InstanceId
 "i-01234567890abcdef" -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.

Run command documents for a Windows Failover Cluster with shared EBS storage

You can use any of the command line procedures described in the previous section to create a VSS based snapshot. The command document (`AWSEC2-VssInstallAndSnapshot` or `AWSEC2-CreateVssSnapshot`) must run on the primary node in your cluster. The document will fail on the secondary nodes as they don't have access to the shared disks. If your primary and secondary change dynamically, you can run the AWS Systems Manager Run Command document on multiple nodes with the expectation that the command will succeed on the primary node and fail on secondary nodes.

> **Note**
>
> To automate backups, you can create an AWS Systems Manager maintenance window task that uses the `AWSEC2-VssInstallAndSnapshot` document. For more information, see [Working with Maintenance Windows (Console)](#) in the *AWS Systems Manager User Guide*.

Troubleshoot Windows VSS based EBS snapshots

Before you try any other troubleshooting steps, we recommend that you verify the following details.

- Ensure that you've met all Prerequisites to create Windows VSS based EBS snapshots.
- Verify that you're using the latest Windows OS version support of the `AwsVssComponents` package for your operating system. The issue that you've observed might have been addressed in newer versions.

Topics

Check log files

If you experience problems or receive error messages when you create VSS based EBS snapshots, you can view the command output in the Systems Manager console.

For Systems Manager documents that create VSS snapshots, you can set the `CollectDiagnosticLogs` parameter to "`True`" at runtime. When the `CollectDiagnosticLogs` parameter is set to "`True`", VSS collects additional logs to aid in debugging. For more information, see Collect additional diagnostic logs.

If you collect diagnostic logs, the Systems Manager document stores them on your instance at the following location: `C:\ProgramData\Amazon\AwsVss\Logs\timestamp.zip`. The default for the `CollectDiagnosticLogs` parameter is "`False`".

> **Note**
>
> For additional debugging help, you can send the `.zip` file to Support.

The following additional logs are available, whether you collect diagnostic logs or not:

- `%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**.

If you're using Systems Manager with VPC endpoints, and the Systems Manager send-command API action (**Run Command** in the console) failed, verify that you configured the following endpoint correctly: **com.amazonaws.*region*.ec2**.

Without the Amazon 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*.

Collect additional diagnostic logs

To collect additional diagnostic logs when you use the Systems Manager send command to run the VSS snapshot document, set the `CollectDiagnosticLogs` input parameter to "`True`" at runtime. We recommend that you set this parameter to "`True`" when you troubleshoot.

To see a command line example, select one of the following tabs.

AWS CLI

The following example runs the `AWSEC2-CreateVssSnapshot` Systems Manager document in the AWS CLI:

```
aws ssm send-command \
    --document-name "AWSEC2-CreateVssSnapshot" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"description":["Example - create diagnostic logs at runtime."],"tags":["Key=tag_name,Value=tag_value"],"CollectDiagnosticLogs":["True"]}'
```

PowerShell

The following example runs the `AWSEC2-CreateVssSnapshot` Systems Manager document in PowerShell:

```
Send-SSMCommand `
    -DocumentName "AWSEC2-CreateVssSnapshot" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'description'='Example - create diagnostic logs at
 runtime.';'tags'='Key=tag_name,Value=tag_value';'CollectDiagnosticLogs'='True'}
```

Use VSS on instances with proxy configured

If you experience problems when creating VSS based EBS snapshots on instances that use a proxy to reach EC2 endpoints, verify the following settings on your instance:

- Verify that the proxy is configured so that the EC2 service endpoints in the instance's Region and IMDS are reachable by AWS Tools for Windows PowerShell running as SYSTEM.
- To support using the system configured WinHTTP proxy, make sure that you've installed the latest `AwsVssComponents` version on your instance. For more information about configuring WinHTTP proxy, see Netsh Commands for Windows Hypertext Transfer Protocol (WINHTTP) on the Microsoft website.

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 based 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, `AWSEC2-CreateVssSnapshot` snapshots your volumes.

The following issues 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

Most timeout issues that occur when you run the `AWSEC2-CreateVssSnapshot` command document are related to the workload on the instance being too high at the time of the backup. The following actions can help you take a successful snapshot:

- Retry the `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 `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 is a high volume of Amazon EC2 API calls in your account within the same Region where you're running a snapshot, API throttling might delay snapshot operations. To reduce throttling impact, use the latest `AwsVssComponents` package. This package utilizes the EC2 `CreateSnapshots` API action to reduce the number of mutating actions like per-volume snapshot creation and tagging.
- If you have multiple `AWSEC2-CreateVssSnapshot` command scripts running at the same time, you can take the following steps to reduce concurrency issues.
  - Consider scheduling snapshots during periods of lower API activity.
  - If you use **Run Command** in the Systems Manager console (or **SendCommand** in the API) to run the command script, you can use Systems Manager rate controls to reduce concurrency.

    You can also use Systems Manager rate controls to reduce concurrency for services like AWS Backup that use Systems Manager to run the command script.
- Run the command `vssadmin list writers` in a shell and see if it reports any errors in the **Last error** field for any writers on the system. If any writers report a **time out** error, consider retrying snapshots when the instance is under less load.
- When you use smaller instance types like *`t2 | t3 | t3a`*.nano or *`t2 | t3 | t3a`*.micro, timeouts due to memory and CPU constraints can occur. The following actions might help reduce timeout issues.
  - Try closing memory or CPU intensive applications before taking snapshots.
  - Try taking snapshots during periods of lower instance activity.

Error: Cannot invoke method. Method invocation is supported only on core types in this language mode

You will encounter this error when the PowerShell language mode is not set to `FullLanguage`. The `AWSEC2-CreateVssSnapshot` SSM document requires PowerShell to be configured to `FullLanguage` mode.

To verify the language mode, run the following command on the instance in a PowerShell console:

```
$ExecutionContext.SessionState.LanguageMode
```

For more information about language modes, see about_Language_Modes in the Microsoft documentation.

Use the AWS VSS solution to restore data for your instance

You can restore EBS volumes for a Windows instance from VSS based snapshots created by the AWS VSS solution. If your AWS VSS solution snapshots contain backups of a Microsoft SQL Server database, you can restore the database using the `AWSEC2-RestoreSqlServerDatabaseWithVss` AWS Systems Manager Automation runbook.

The database restore runbook automates the entire restore process, including creating volumes from the snapshots and attaching them to the instance. The automation leverages VSS technology to restore the database, allowing you to restore without stopping your SQL Server application or disconnecting any active connections.

For detailed instructions on how to use the Microsoft SQL Server database runbook, see Restore from VSS based snapshots in the *Microsoft SQL Server on Amazon EC2 User Guide*.

Customize a script to restore EBS volumes from AWS VSS solution snapshots

You can use the `RestoreVssSnapshotSampleScript.ps1` script as a model to create your own custom script that restores EBS volumes from AWS VSS solution snapshots. The sample 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 based EBS snapshots

1. Download the RestoreVssSnapshotSampleScript.zip file and extract the file contents.
2. Open `RestoreVssSnapshotSampleScript.ps1` 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 solution version history

This page includes release notes by version for the AWS VSS component package, as well as component and script version requirements for each supported version of Windows Server.

Topics

AwsVssComponents package versions

The following table describes the released versions of the AWS VSS component package.

Version	Details	Release date	Downloadable
2.5.1	Fixed a case where SQL database restoration could fail when the target database parameter is specified.	March 13, 2025	Yes

Version	Details	Release date	Downloadable
2.5.0	• Added the capability to read VSS metadata files and restore a Microsoft SQL Server database on the instance. For more information, see [Restore from VSS based snapshots]() in the *Microsoft SQL Server on Amazon EC2 User Guide*. • Added support for the in-place update option when installing or upgrading the `AwsVssComponents` package.	January 17, 2025	Yes
2.4.0	Added the capability to save VSS metadata files on snapshot creation. To enable this feature, see SaveVssMetadata in [Parameters for Systems Manager VSS snapshot documents]().	October 7, 2024	Yes
2.3.3	Updated the VSS agent to ensure that the `Ec2VssProvider` is used during snapshot creation.	June 25, 2024	Yes
2.3.2	Fixed a case where VSS provider registration is not removed on uninstallation.	May 9, 2024	Yes
2.3.1	Added new default tag `AwsVssConfig` to identify snapshots and AMIs created by AWS VSS.	March 7, 2024	Yes

Version	Details	Release date	Downloadable
2.2.1	- Added support for using the `DescribeInstanceAttribute` API. - Bug fixes and reliability improvements. - Deprecated support for Windows Server 2012 and 2012 R2. AWS VSS components version 2.2.1 installation on Windows Server 2012 and 2012 R2 will fail. AWS VSS components version 2.1.0 is the last version to support Windows Server 2012 and 2012 R2.	January 18, 2024	Yes
2.1.0	Added support for using the `CreateSnapshots` API.	November 6, 2023	Yes
2.0.1	Added support for using the WinHTTP proxy settings.	October 26, 2023	No
2.0.0	Added capability to the AWS VSS component to create snapshots and AMIs, which enables compatibility with PowerShell module logging, script block logging, and transcription features.	April 28, 2023	No
1.3.2.0	Fixed a case where installation failure is not reported correctly.	May 10, 2022	No

Version	Details	Release date	Downloadable
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.	February 6, 2020	Yes
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.	March 19, 2019	No
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.	November 15, 2018	No
1.1	Fixed AWS VSS components that were being used incorrectly as the default Windows Backup and Restore provider.	December 12, 2017	No
1.0	Initial release.	November 20, 2017	No

Windows OS version support

The following table shows which AWS VSS solution versions you should run on each version of Windows Server on Amazon EC2.

Windows Server version	AwsVssComponents version	AWSEC2-VssInstallAndSnapshot version name	AWSEC2-CreateVssSnapshot version name
Windows Server 2025	default	default	default
Windows Server 2022	default	default	default
Windows Server 2019	default	default	default
Windows Server 2016	default	default	default
Windows Server 2012 R2	2.1.0	not supported	2012R2
Windows Server 2012	2.1.0	not supported	2012R2

Windows Server version	AwsVssComponents version	AWSEC2-VssInstallAndSnapshot version name	AWSEC2-CreateVssSnapshot version name
Windows Server 2008 R2	1.3.1.0	not supported	2008R2

Object storage, file storage, and file caching on Amazon EC2

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 Amazon EC2 instances.

Contents

Use Amazon S3 with Amazon EC2 instances

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. You can use Amazon S3 to store and retrieve any amount of data for a range of use cases, such as data lakes, websites, backups, and big data analytics, from an Amazon EC2 instance or from anywhere over the internet. For more information, see What is Amazon S3?

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 `amzn-s3-demo-bucket1` bucket, then it is addressable using the URL `https://amzn-s3-demo-bucket1.s3.amazonaws.com/photos/mygarden.jpg`. For more information, see [How Amazon S3 works].

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.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

wget

> **Note**
>
> This method works for public objects only. If the object is not public, you receive an `ERROR 403: Forbidden` message. If you receive this error, you must use either the Amazon S3 console, AWS CLI, AWS API, AWS SDK, or AWS Tools for Windows PowerShell, and you must have the required permissions. For more information, see Identity and access management for Amazon S3 and Downloading an object in the *Amazon S3 User Guide*.

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions, and available for download on Windows. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

```
[ec2-user ~]$ wget https://amzn-s3-demo-bucket.s3.amazonaws.com/path-to-file
```

PowerShell

You can use the AWS Tools for Windows PowerShell to move objects to and from Amazon S3.

Use the Copy-S3Object cmdlet to copy an Amazon S3 object to your Windows instance as follows.

```
Copy-S3Object `
    -BucketName amzn-s3-demo-bucket `
    -Key path-to-file `
    -LocalFile my_copied_file.ext
```

Alternatively, you can open the Amazon S3 console by using a web browser on the Windows instance.

AWS CLI

You can use the AWS Command Line Interface (AWS CLI) to download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the AWS Command Line Interface detail page.

The aws s3 cp command is similar to the Unix **cp** command. You can copy files from Amazon S3 to your instance, copy files from your instance to Amazon S3, and copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
aws s3 cp s3://amzn-s3-demo-bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
aws s3 cp my_copied_file.ext s3://amzn-s3-demo-bucket/my_folder/my_file.ext
```

The aws s3 sync command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
aws s3 sync s3://amzn-s3-demo-source-bucket local_directory
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. You can use this API to help develop your application and integrate it with other APIs and SDKs. For more information, see Code examples for Amazon S3 using AWS SDKs in the *Amazon Simple Storage Service API Reference*.

Use Amazon EFS with Amazon EC2 Linux instances

> **Note**
>
> Amazon EFS is not supported on Windows instances.

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.

This tutorial shows you how to create and attach an Amazon EFS file system using the Amazon EFS Quick Create wizard during instance launch. For a tutorial on how to create a file system using the Amazon EFS console, see Getting started with Amazon Elastic File System in the *Amazon Elastic File System User Guide*.

> **Note**
>
> When you create an EFS file system using EFS Quick Create, the file system is created with the following service recommended settings:
>
> - Automatic backups enabled.
> - Manage mount targets in the selected VPC.
> - General Purpose performance mode.
> - Bursting throughput mode.
> - Encryption of data at rest enabled using your default key for Amazon EFS (`aws/elasticfilesystem`).

- Amazon EFS lifecycle management enabled with a 30-day policy.

Tasks

- Create an EFS file system using Amazon EFS Quick Create
- Test the EFS file system
- Delete the EFS file system

Create an EFS file system using Amazon EFS Quick Create

You can create an EFS file system and mount it to your instance when you launch your instance using the Amazon EFS Quick Create feature of the Amazon EC2 launch instance wizard.

To create an EFS file system using Amazon EFS Quick Create

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch instance**.
3. (Optional) Under **Name and tags**, for **Name**, enter a name to identify your instance.
4. Under **Application and OS Images (Amazon Machine Image)**, choose a Linux operating system, and then for **Amazon Machine Image (AMI)**, select a Linux AMI.
5. Under **Instance type**, for **Instance type**, select an instance type or keep the default.
6. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or create a new one.
7. Under **Network settings**, choose **Edit** (at right), and then for **Subnet**, select a subnet.

   > **Note**
   >
   > You must select a subnet before you can add an EFS file system.

8. Under **Configure storage**, choose **Edit** (at bottom right), and then do the following:

   a. For **File systems**, ensure that **EFS** is selected, and then choose **Create new shared file system**.

   b. For **File system name** enter a name for the Amazon EFS file system, and then choose **Create file system**.

   c. For **Mount point**, specify a custom mount point or keep the default.

d. To enable access to the file system, select **Automatically create and attach security groups**. By selecting this checkbox, the following security groups will be automatically created and attached to the instance and the mount targets of the file system:

   - Instance security group – Includes an outbound rule that allows traffic over the NFS 2049port, but includes no inbound rules.
   - File system mount targets security group – Includes an inbound rule that allows traffic over the NFS 2049 port from the instance security group (described above), and an outbound rule that allows traffic over the NFS 2049 port.

   > **Note**
   >
   > Alternatively, you can manually create and attach the security groups. If you want to manually create and attach the security groups, clear **Automatically create and attach the required security groups**.

e. To automatically mount the shared file system when the instance launches, select **Automatically mount shared file system by attaching required user data script**. To view the user data that is automatically generated, expand **Advanced details**, and scroll down to **User data**.

   > **Note**
   >
   > If you added user data before selecting this checkbox, the original user data is overwritten by the automatically generated user data.

9. Configure any other instance configuration settings as needed.
10. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

Test the EFS file system

You can connect to your instance and verify that the file system is mounted to the directory that you specified (for example, /mnt/efs).

To verify that the file system is mounted

1. Connect to your instance. For more information, see [Connect to your Linux instance using SSH]().
2. From the terminal window for the instance, run the **df -T** command to verify that the EFS file system is mounted.

```
$ df -T
Filesystem     Type              1K-blocks    Used          Available Use% Mounted
 on
/dev/xvda1     ext4                8123812 1949800            6073764  25% /
devtmpfs       devtmpfs            4078468      56            4078412   1% /dev
tmpfs          tmpfs               4089312       0            4089312   0% /dev/shm
efs-dns        nfs4       9007199254740992       0   9007199254740992   0% /mnt/efs
```

Note that the name of the file system, shown in the example output as *efs-dns*, has the following form.

```
file-system-id.efs.aws-region.amazonaws.com:/
```

3. (Optional) Create a file in the file system from the instance, and then verify that you can view the file from another instance.

   a. From the instance, run the following command to create the file.

   ```
   $ sudo touch /mnt/efs/test-file.txt
   ```

   b. From the other instance, run the following command to view the file.

   ```
   $ ls /mnt/efs
   test-file.txt
   ```

Delete the EFS file system

If you no longer need your file system, you can delete it.

To delete the file system

1. Open the Amazon Elastic File System console at https://console.aws.amazon.com/efs/.
2. Select the file system to delete.

3. Choose **Actions**, **Delete file system**.
4. When prompted for confirmation, enter the file system ID and choose **Delete file system**.

Use Amazon FSx with Amazon EC2 instances

The Amazon FSx family of services makes it easy to launch, run, and scale shared storage powered by popular commercial and open-source file systems. You can use the *new launch instance wizard* to automatically attach the following types of Amazon FSx file systems to your Amazon EC2 instances at launch:

- Amazon FSx for NetApp ONTAP provides fully managed shared storage in the AWS Cloud with the popular data access and management capabilities of NetApp ONTAP.
- Amazon FSx for OpenZFS provides fully managed cost-effective shared storage powered by the popular OpenZFS file system.

> **Note**
>
> - This functionality is available in the new launch instance wizard only. For more information, see Launch an EC2 instance using the launch instance wizard in the console
> - Amazon FSx for Windows File Server and Amazon FSx for Lustre file systems can't be mounted at launch. You must mount these file systems manually after launch.

You can choose to mount an existing file system that you created previously, or you can create a new file system to mount to an instance at launch.

Topics

- Security groups and user data script
- Mount an Amazon FSx file system at launch

Security groups and user data script

When you mount an Amazon FSx file system to an instance using the launch instance wizard, you can choose whether to automatically create and attach the security groups needed to enable access to the file system, and whether to automatically include the user data scripts needed to mount the file system and make it available for use.

Topics

- Security groups
- User data script

Security groups

If you choose to automatically create the security groups that are needed to enable access to the file system, the launch instance wizard creates and attaches two security groups - one security group is attached to the instance, and the other is attached to the file system. For more information about the security group requirements, see FSx for ONTAP file system access control with Amazon VPC and FSx for OpenZFS file system access control with Amazon VPC.

We add the tag `Name=instance-sg-1` to the security group that is created and attached to the instance. The value in the tag is automatically incremented each time the launch instance wizard creates a security group for Amazon FSx file systems.

The security group includes the following output rules, but no inbound rules.

Outbound rules

Protocol type	Port number	Destination
UDP	111	*file system security group*
UDP	20001 - 20003	*file system security group*
UDP	4049	*file system security group*
UDP	2049	*file system security group*
UDP	635	*file system security group*
UDP	4045 - 4046	*file system security group*
TCP	4049	*file system security group*
TCP	635	*file system security group*

Protocol type	Port number	Destination
TCP	2049	*file system security group*
TCP	111	*file system security group*
TCP	4045 - 4046	*file system security group*
TCP	20001 - 20003	*file system security group*
All	All	*file system security group*

The security group that is created and attached to the file system is tagged with `Name=fsx-sg-1`. The value in the tag is automatically incremented each time the launch instance wizard creates a security group for Amazon FSx file systems.

The security group includes the following rules.

Inbound rules

Protocol type	Port number	Source
UDP	2049	*instance security group*
UDP	20001 - 20003	*instance security group*
UDP	4049	*instance security group*
UDP	111	*instance security group*
UDP	635	*instance security group*
UDP	4045 - 4046	*instance security group*
TCP	4045 - 4046	*instance security group*
TCP	635	*instance security group*
TCP	2049	*instance security group*
TCP	4049	*instance security group*

Protocol type	Port number	Source
TCP	20001 - 20003	*instance security group*
TCP	111	*instance security group*

Outbound rules

Protocol type	Port number	Destination
All	All	0.0.0.0/0

User data script

If you choose to automatically attach user data scripts, the launch instance wizard adds the following user data to the instance. This script installs the necessary packages, mounts the file system, and updates your instance settings so that the file system will automatically re-mount whenever the instance restarts.

```
#cloud-config
package_update: true
package_upgrade: true
runcmd:
- yum install -y nfs-utils
- apt-get -y install nfs-common
- svm_id_1=svm_id
- file_system_id_1=file_system_id
- vol_path_1=/vol1
- fsx_mount_point_1=/mnt/fsx/fs1
- mkdir -p "${fsx_mount_point_1}"
- if [ -z "$svm_id_1" ]; then printf "\n${file_system_id_1}.fsx.eu-north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0 0\n" >> /etc/fstab; else printf "\n${svm_id_1}.${file_system_id_1}.fsx.eu-north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0 0\n" >> /etc/fstab; fi
- retryCnt=15; waitTime=30; while true; do mount -a -t nfs4 defaults; if [ $? = 0 ] || [ $retryCnt -lt 1 ]; then echo File system mounted successfully; break; fi; echo File system not available, retrying to mount.; ((retryCnt--)); sleep $waitTime; done;
```

Mount an Amazon FSx file system at launch

To mount a new or existing Amazon FSx file system at launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then choose **Launch instance** to open the launch instance wizard.
3. In the **Application and OS Images** section, select the AMI to use.
4. In the **Instance type** section, select the instance type.
5. In the **Key pair** section, select an existing key pair or create a new one.
6. In the **Network settings** section, do the following:

   a. Choose **Edit**.

   b. If you want to **mount an existing file system**, for **Subnet**, choose the file system's preferred subnet. We recommend that you launch the instance into the same Availability Zone as the file system's preferred subnet to optimize performance.

      If you want to **create a new file system** to mount to an instance, for **Subnet**, choose the subnet into which to launch the instance.

      > **Important**
      >
      > You must select a subnet to enable the Amazon FSx functionality in the new launch instance wizard. If you do not select a subnet, you will not be able to mount an existing file system or create a new one.

7. In the **Storage** section, do the following:

   a. Configure the volumes as needed.

   b. Expand the **File systems** section and select **FSx**.

   c. Choose **Add shared file system**.

   d. For **File system**, select the file system to mount.

> **Note**
>
> The list displays all Amazon FSx for NetApp ONTAP and Amazon FSx for OpenZFS file systems in your account in the selected Region.

e. To automatically create and attach the security groups needed to enable access to the file system, select **Automatically create and attach security groups**. If you prefer to create the security groups manually, clear the checkbox. For more information, see Security groups.

f. To automatically attach the user data scripts needed to mount the file system, select **Automatically mount shared file system by attaching required user data script**. If you prefer to provide the user data scripts manually, clear the checkbox. For more information, see User data script.

8. In the **Advanced** section, configure the additional instance settings as needed.
9. Choose **Launch**.

Use Amazon File Cache with Amazon EC2 instances

Amazon File Cache provides a fully managed, high-speed cache on AWS that makes it easier to process file data, regardless of where the data is stored. Amazon File Cache serves as a temporary, high-performance storage location for data that's stored in on-premises file systems, AWS file systems, and Amazon Simple Storage Service (Amazon S3) buckets. You can use this capability to make dispersed datasets available to file-based applications on AWS with a unified view, and at high speeds—sub-millisecond latencies and high throughput. For more information, see the Amazon File Cache User Guide.

Amazon File Cache works with the most popular Linux AMIs, and is compatible with x86-based instance types and Graviton instance types. You can access your cache from your Amazon EC2 instances using the open-source Lustre client. You can mount your cache and then work with the files and directories in your cache using standard Linux commands. Amazon EC2 instances can access your cache from other Availability Zones within the same virtual private cloud (VPC), provided that your network configuration allows access across subnets within the VPC. You can also create a cache in a shared VPC.

To get started, see Getting started with Amazon File Cache in the *Amazon File Cache User Guide*.

Manage your Amazon EC2 resources

A *resource* is an entity that you can work with. Amazon EC2 creates resources as you use the features of the service. For example, Amazon EC2 resources include images, instances, fleets, key pairs, and security groups. All Amazon EC2 resource types include attributes that describe the resources. For example, names, descriptions, resource identifiers, and Amazon Resource Names (ARN).

Amazon EC2 resources are specific to the AWS Region or zone in which they reside. For example, an Amazon Machine Image (AMI) is specific to an AWS Region, but the instance that you launch from an AMI is specific to the zone in which you launch it. You can specify an Amazon EC2 resource in a permissions policy using its ARN.

Your AWS account has default quotas for Amazon EC2. These quotas define the maximum number of resources that you can create. For example, there are quotas for the maximum number of vCPUs across your running instances. If launching an instance or starting a stopped instance would cause you to exceed your quota, the operation fails.

You can search for specific resources in your AWS account by Region, using resource IDs or tags. To search for specific resources or resource types across multiple Regions, use Amazon EC2 Global View.

Contents

Select a Region for your Amazon EC2 resources

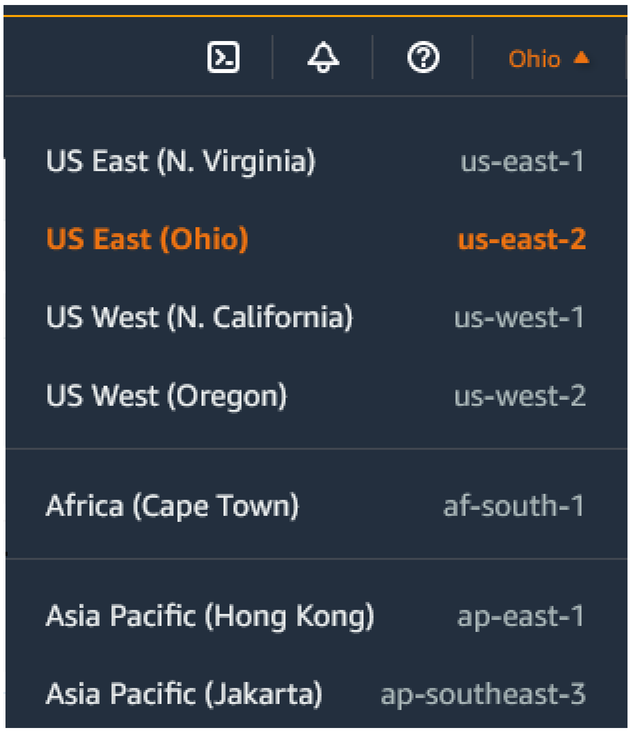
Amazon EC2 resources are specific to the AWS Region or zone in which they reside. When you create an Amazon EC2 resource, you select the Region for the resource.

Considerations

- Some AWS resources might not be available in all Regions. Ensure that you can create all the AWS resources that you need in a Region before you start creating resources in a Region.
- Some Regions are disabled by default. You must enable these Regions before you can use them. For more information , see AWS Regions.

To select a Region for a resource using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, choose the **Regions** selector and then choose the Region.



3. The Regions selector includes all resources that are available for use in your AWS account. Choose the underlined text near the bottom of the list to view the Regions that are not enabled for your account.

To select a Region for a resource using the AWS CLI

You can configure the AWS CLI to use a default Region. If you don't specify a Region in the command, the default Region is used. To use a different Region for a specific command, add the following option.

```
--region us-east-1
```

To select a Region for a resource using the Tools for PowerShell

You can configure the Tools for Windows PowerShell to use a default Region. If you don't specify a Region in a command, the default Region is used. To use a different Region for a specific command, add the following parameter.

```
-Region us-east-1
```

Find your Amazon EC2 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

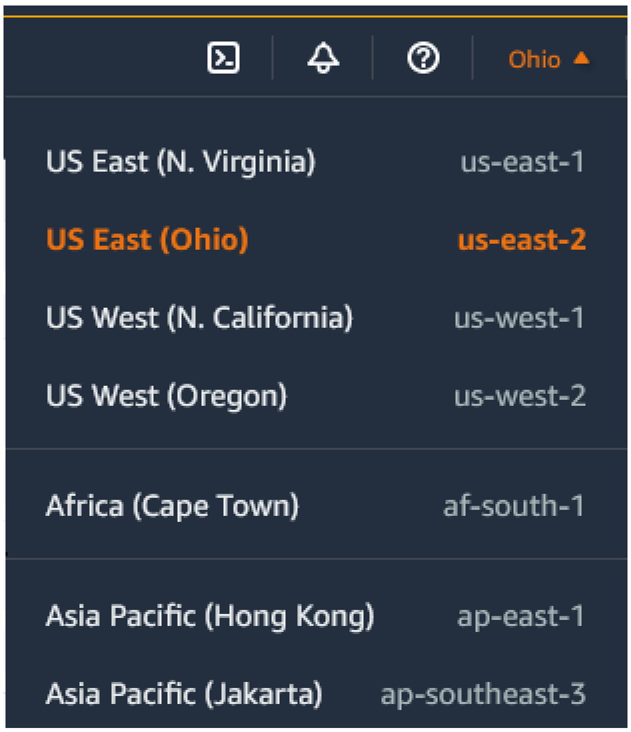
Contents

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. Amazon EC2 resources are specific to an AWS Region. From the navigation bar, choose a Region from the **Regions** selector.



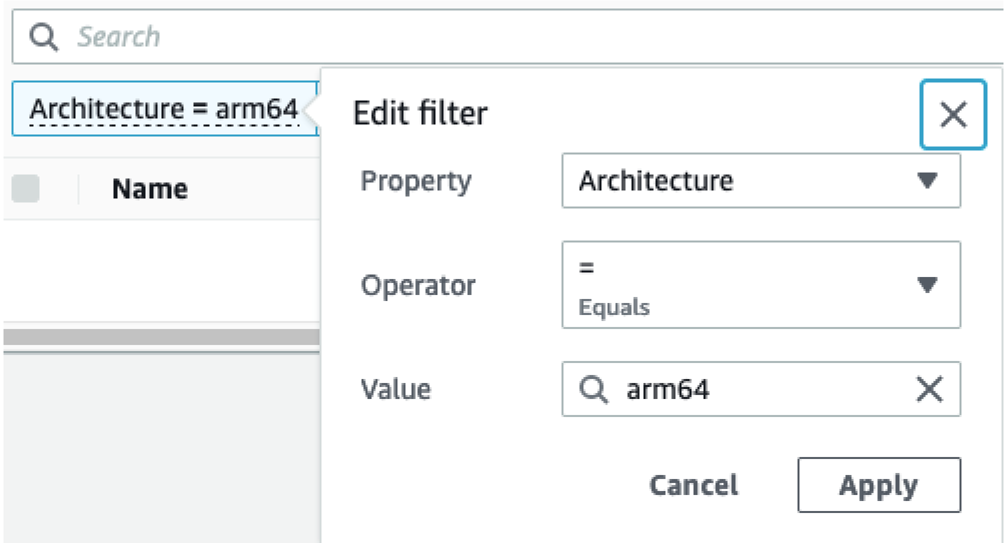
3. In the navigation pane, choose the option that corresponds to the resource type. For example, to list all your instances, choose **Instances**.

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. Select the filter from the list.

4. Select an operator, for example, **= (Equals)**. Some attributes have more available operators to select. Note that not all screens support selecting an operator.
5. Select a filter value.
6. To edit a selected filter, choose the filter token (blue box), make the required edits, and then choose **Apply**. Note that not all screens support editing the selected filter.



7. When you are finished, remove the filter.

Supported filters

The Amazon EC2 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. API filtering supports **=** (equals) and **:** (contains) operators, and is always case sensitive.
- *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. In addition to **=** (equals) and **:** (contains) operators, client filtering can also support range operators, such as **>=** (greater than or equal), and negation (inverse) operators, such as **!=** (does not equal).

The 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 or tags, without specifying an attribute or tag key 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, or in any of their tags, such as the Name. If your free text search returns unexpected matches, apply additional filters.

Search by attribute

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 operators for the selected attribute appears.
3. Choose the **=** (Equals) operator.

   A list of possible values for the selected attribute and operator appears.
4. Choose **stopped** from the list.

Search by tag

Searching by a tag lets you filter the resources in the currently displayed table by a tag key or a tag value.

Tag searches use either *API filtering* or *client filtering*, depending on the settings in the Preferences window.

To ensure API filtering for tags

1. Open the **Preferences** window.
2. Clear the **Use regular expression matching** checkbox. If this checkbox is selected, client filtering is performed.
3. Select the **Use case sensitive matching** checkbox. If this checkbox is cleared, client filtering is performed.
4. Choose **Confirm**.

When searching by tag, you can use the following values:

- **(empty)** – Find all resources with the specified tag key, but there must be no tag value.
- **All values** – Find all resources with the specified tag key and any tag value.
- **Not tagged** – Find all resources that do not have the specified tag key.
- The displayed value – Find all resources with the specified tag key and the specified tag value.

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. In the **Instances** and **AMIs** screens, inverse searches are performed by selecting the **!=** (Does not equal) or **!:** (Does not contain) operator and then selecting a value. In other screens, 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. Under **Client filters**, choose **Instance state (client)**. Inverse search is only supported on *client* filters.

   A list of possible operators for the selected attribute appears.
3. Choose **!=** (Does not equal), and then choose **terminated**.

To filter instances based on an instance state attribute, you can also use the search icons ( ) in the **Instance 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`, under **Client filters**, search by the **Security group name** attribute, choose **!=**, and in the search bar, 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. On the **Instances** and **AMIs** screens, partial searches can only be performed with the **:** (Contains) operator. On other screens, you can select the client filter attribute and immediately enter only a part of the keyword that you want to search for. For example, on the **Instance type** screen, 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

To use regular expression searches, you must select the **Use regular expression matching** checkbox in the **Preferences** window.

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, to search for a value that starts with `s`, search for `^s`. To search for a value that ends with `xyz`, search for `xyz$`. Or to search for a value that starts with a number that is followed by one or more characters, search for `[0-9]+.*`.

> **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.

Case-sensitive search

To use case-sensitive searches, you must select the **Use case sensitive matching** checkbox in the **Preferences** window. The case-sensitive preference only applies to client and tag filters.

> **Note**
>
> API filters are always case-sensitive.

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 values `prod`, `prods`, and `production`, a search of `prod*` matches all values, whereas `prod?` matches only `prod` and `prods`. To use the literal values, escape them with a backslash (\). For example, "`prod\*`" would match `prod*`.

> **Note**
>
> Wildcard search is supported with attribute and tag searches on API filters only. It is not supported with keyword searches, and with attribute and tag searches on client filters.

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 running state.

Save filter sets using the console

A *saved filter set* is a customized group of filters that you can create and reuse to efficiently view your Amazon EC2 resources. This feature helps streamline your workflow, enabling quick access to specific resource views.

Saved filter sets are only supported in the Amazon EC2 console and are currently only available for the **Volumes** page.

Key features

- **Customization:** Create filter sets tailored to your needs. For example, you can create a filter set to display only your gp3 volumes that were created after a specified date.
- **Default filter:** Set a default filter set for a page, and the default filters are automatically applied when you navigate to the page. If no default is set, no filters are applied.
- **Easy application:** Select a saved filter set to apply it instantly. Amazon EC2 then displays the relevant resources, with the active filters indicated by blue tokens.
- **Flexibility:** Create, modify, and delete filter sets as needed.

Create a filter set

To create a new filter set

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Volumes**.

   > **Note**
   >
   > Saved filter sets are currently only available for **Volumes**.

3. In the search field, select filters for your filter set.
4. Choose the arrow next to the **Clear filters** button, and choose **Save new filter set**.
5. In the **Save filter set** window, do the following:

   a. For **Filter set name**, enter a name for the filter set.

   b. (Optional) For **Filter set description**, enter a description for the filter set.

   c. (Optional) To set the filter set as the default filter, select the **Set as default** checkbox.

> **Note**
>
> The default filter is automatically applied every time you open the console page.

d. Choose **Save**.

Modify a filter set

To modify a filter set

1. From the **Saved filter sets** list, select the filter to modify.
2. To add a filter, in the search field, select a filter to add to your filter set. To delete a filter in the set, choose the **X** on the filter token.
3. Choose the arrow next to the **Clear filters** button, and choose **Modify filter set**.
4. In the **Modify filter set** window, do the following:

   a. (Optional) To set the filter set as the default filter, select the **Set as default** checkbox.

   > **Note**
   >
   > The default filter is automatically applied every time you open the console page.

   b. Choose **Modify**.

Delete a filter set

To delete a filter set

1. From the **Saved filter sets** list, select the filter to delete.
2. Choose the arrow next to the **Clear filters** button, and choose **Delete filter set**.
3. In the **Delete filter set** window, review the filter to confirm this is the filter you want to delete, and then choose **Delete**.

List and filter using the command line

Each resource type has a corresponding API actions that you use to describe, list, or get 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 up to 50 filters and up to 200 values per filter in a single request.
- Filter strings can be up to 255 characters in length.
- 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?\`.
- You can't specify a filter value of null. Instead, use client-side filtering. For example, the following command uses the **--query** option and returns the IDs of the instances that were launched without a key pair.

```
aws ec2 describe-instances \
    --query 'Reservations[*].Instances[?!not_null(KeyName)].InstanceId' \
    --output text
```

AWS CLI

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 option to include only the running instances in the output.

```
--filters Name=instance-state-name,Values=running
```

To list only the instance IDs for your running instances, add the `--query` option as follows.

```
--query "Reservations[*].Instances[*].InstanceId"
```

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 output.

You can use the following option to list all instances whose type is either `m5.large` or `m5d.large`.

```
--filters Name=instance-type,Values=m5.large,m5d.large
```

You can use the following option to list all stopped instances whose type is `t2.micro`.

```
--filters Name=instance-state-name,Values=stopped Name=instance-type,Values=t2.micro
```

Example Example: Use wildcards in a filter value

You can use the following option with [describe-snapshots]() to return only the snapshots whose description is "database".

```
--filters Name=description,Values=database
```

The * wildcard matches zero or more characters. You can use the following option to return only the snapshots whose description includes the word database.

```
--filters Name=description,Values=*database*
```

The ? wildcard matches exactly 1 character. You can use the following option to return only the snapshots whose description is "database" or "database" followed by one character.

```
--filters Name=description,Values=database?
```

You can use the following option to return only the snapshots whose description is "database" followed by up to four characters. This excludes descriptions with "database" followed by five or more characters.

```
--filters Name=description,Values=database????
```

Example Example: Filter based on date

With the AWS CLI, you can use JMESPath to filter results using expressions. For example, the following describe-snapshots command displays the IDs of all snapshots created by the specified AWS account before the specified date. 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<='2024-03-31')].[SnapshotId]" \
    --output text
```

The following example 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>='2024-01-01') && (StartTime<='2024-12-31')].
[SnapshotId]" \
    --output text
```

Example: Filter based on tags

For examples of how to filter a list of resources according to their tags, see Filter Amazon EC2 resources by tag.

PowerShell

Example Example: Specify a single filter

You can list your Amazon EC2 instances using Get-EC2Instance. Without filters, the response contains information for all of your resources. You can use the following parameter to include only the running instances in the output.

```
-Filter @{Name="instance-state-name"; Values="running"}
```

The following example lists only the instance IDs for your running instances.

```
(Get-EC2Instance -Filter @{Name="instance-state-name"; Values="stopped"}).Instances
 | Select InstanceId
```

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 option to list all instances whose type is either `m5.large` or `m5d.large`.

```
-Filter @{Name="instance-type"; Values="m5.large", "m5d.large"}
```

You can use the following option to list all stopped instances whose type is `t2.micro`.

```
-Filter @{Name="instance-state-name"; Values="stopped"}, @{Name="instance-type";
 Values="t2.micro"}
```

View resources across Regions using Amazon EC2 Global View

Amazon EC2 Global View enables you to view and search for Amazon EC2 and Amazon VPC resources in a single AWS Region, or across multiple Regions simultaneously in a single console. For more information, see View resources across Regions using AWS Global View.

View resources across Regions using AWS Global View

AWS Global View enables you to view some of your Amazon EC2 and Amazon VPC resources across a single AWS Region, or across multiple Regions in a single console. AWS Global View also provides *global search* functionality that lets you search for specific resources or specific resource types across multiple Regions simultaneously.

AWS Global View does not let you modify resources in any way.

Supported resources

Using AWS Global View, you can view a global summary of the following resources across all of the Regions for which your AWS account is enabled.

- Auto Scaling groups
- Capacity Reservations and Capacity Blocks

- DHCP option set
- Egress-only internet gateways
- Elastic IPs
- Endpoint services
- Instances
- Internet gateways
- Managed prefix lists
- NAT gateways
- Network ACLs
- Network interfaces
- Route tables
- Security groups
- Subnets
- Volumes
- VPCs
- VPC endpoints
- VPC peering connections

Required permissions

A user must have the following permissions to use AWS Global View.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
    "autoscaling:DescribeAutoScalingGroups",
    "ec2:DescribeRegions",
    "ec2:DescribeCapacityReservations",
```

```
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeEgressOnlyInternetGateways",
        "ec2:DescribeAddresses",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribePrefixLists",
        "ec2:DescribeNatGateways",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcPeeringConnections"
      ],
      "Resource": "*"
      }]
    }
```

To use AWS Global View

Sign in to the [AWS Global View console](https://console.aws.amazon.com/ec2globalview/home).

> **Important**
>
> You cannot use a private window in Firefox to access AWS Global View.

The console consists of the following:

- **Resource explorer** – This page includes the following sections:
  - **Summary** – Provides a high-level overview of your resources across all Regions.

    Expand **Show all resource summary** indicates the number of Regions for which your AWS account is enabled. The remaining fields indicate the number of resources that you currently have in those Regions. Choose any of the links to view the resources of that type across all Regions. For example, if the link below the **Instances** label is **29 in 10 Regions**, it indicates that you currently have 29 instances across 10 Regions. Choose the link to view a list of all 29 instances.

- **Resource explorer –** Lists all of the AWS Regions (including those for which your account is not enabled) and provides totals for each resource type for each Region.

  Choose a Region name to view all resources of all types for that specific Region. For example, choose **Africa (Cape Town) af-south-1** to view all VPCs, subnets, instances, security groups, volumes, and Auto Scaling groups in that Region. Alternatively, select a Region and choose **View resources for selected Region**.

  Choose the value for a specific resource type in a specific Region to view only resources of that type in that Region. For example, choose the value for Instances for **Africa (Cape Town) af-south-1** to view only the instances in that Region.

- **Global search –** This page enables you to search for specific resources or specific resource types across a single Region or across multiple Regions. It also enables you to view details for a specific resource.

  To search for resources, enter the search criteria in the field preceding the grid. You can search by Region, by resource type, and by the tags assigned to resources.

  To view the details for a specific resource, select it in the grid. You can also choose the resource ID of a resource to open it in its respective console. For example, choose an instance ID to open the instance in the Amazon EC2 console, or choose a subnet ID to open the subnet in the Amazon VPC console.

- **Regions and Zones –** This page allows you to view and manage all available Regions, Availability Zones, Local Zones, and Wavelength Zones.

  From the **Regions** tab, you can view all the AWS Regions. The **Status** column shows the Regions that are enabled for your AWS account. From this page, you can select a Region to:

  - View details of the Region such as the Region code, Geography, and number of each type of Zone.

    You can also view the list of Availability Zones, Local Zones, and Wavelength Zones and the list of your resources in the Region.

  - Enable or disable the Region.

  From each zone tab, you can view the list of that zone type. From the **Local Zones** tab, you can opt-in to a Local Zone.

Tip

If you only use specific Regions or resource types, you can customize AWS Global View to display only those Regions and resource types. To customize the displayed Regions and resource types, in the navigation panel, choose **Settings**, and then on the **Resources** and **Regions** tabs, select the Regions and resource types that you do not want to be displayed in AWS Global View.

Tag your Amazon EC2 resources

To help you manage your instances, images, and other Amazon EC2 resources, you can assign your own metadata to each resource in the form of *tags*. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it. This topic describes tags and shows you how to create them.

Warning

Tag keys and their values are returned by many different API calls. Denying access to `DescribeTags` doesn’t automatically deny access to tags returned by other APIs. As a best practice, we recommend that you do not include sensitive data in your tags.

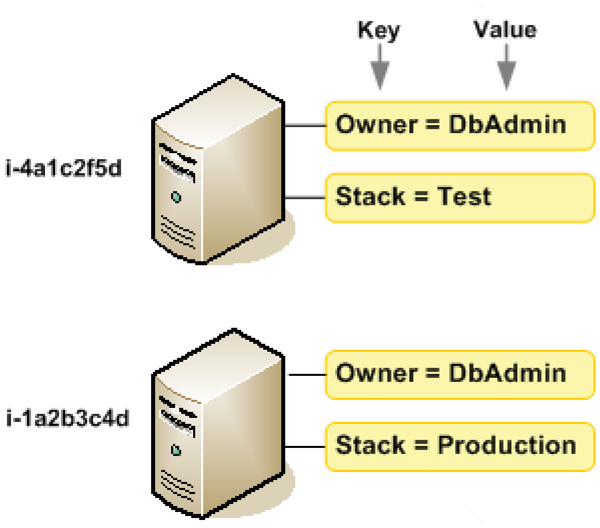
Contents

Tag basics

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 Tagging Best Practices AWS Whitepaper.

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.

> **Note**
>
> After you delete a resource, its tags might remain visible in the console, API, and CLI output for a short period. These tags will be gradually disassociated from the resource and be permanently deleted.

Tag your resources

When you use 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 Editor** in the AWS Resource Groups console. 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 might 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 Amazon EC2 resources during creation].

You can apply tag-based resource-level permissions in your IAM policies to the Amazon EC2 API actions that support tagging on creation to implement granular control over the users and groups that can tag resources on creation. Your resources are properly secured from creation—tags are applied immediately to your resources, therefore any tag-based resource-level permissions controlling the use of resources are immediately effective. Your resources can be tracked and reported on more accurately. You can enforce the use of tagging on new resources, and control which tag keys and values are set on your resources.

You can also apply resource-level permissions to the `CreateTags` and `DeleteTags` Amazon EC2 API actions in your IAM policies to control which tag keys and values are set on your existing resources. For more information, see [Example: Tag resources].

For more information about tagging your resources for billing, see Using cost allocation tags in the *AWS Billing User Guide*.

Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8
- Maximum value length – 256 Unicode characters in UTF-8
- Allowed characters
  - Although EC2 allows for any character in its tags, other AWS services are more restrictive. The allowed characters across all AWS services are: letters (`a-z`, `A-Z`), numbers (`0-9`), and spaces representable in UTF-8, and the following characters: `+ - = . _ : / @`.
  - If you enable instance tags in instance metadata, instance tag *keys* can only use letters (`a-z`, `A-Z`), numbers (`0-9`), and the following characters: `+ - = . , _ : @`. Instance tag *keys* can't contain spaces or `/`, and can't comprise only `.` (one period), `..` (two periods), or `_index`. For more information, see View tags for your EC2 instances using instance metadata.
- 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.

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 Amazon EC2 resources during creation.

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 using attribute-based access.

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 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 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 User Guide*.

> **Note**
>
> Elastic IP addresses that are tagged do not appear on your cost allocation report.

Grant permission to tag Amazon EC2 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 using attribute-based access.

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `ec2:RunInstances` or `ec2:CreateVolume`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `ec2:CreateTags` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `ec2:CreateTags` action.

In the IAM policy definition for the `ec2:CreateTags` action, use the `Condition` element with the `ec2:CreateAction` condition key to give tagging permissions to the action that creates the resource.

The following example demonstrates a policy that allows users to launch instances and apply any tags to instances and volumes during launch. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "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:us-east-1:111122223333:*/*",
      "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.

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 to control access the Amazon EC2 API. For more information, see Conditions with multiple context keys or values in the *IAM User Guide*.

Add and remove tags for Amazon EC2 resources

When you create an Amazon EC2 resource, such as an Amazon EC2 instance, you can specify the tags to add to the resource. You can also use the Amazon EC2 console to display the tags for a specific Amazon EC2 resource. You can also add or remove tags from an existing Amazon EC2 resource.

You can use the **Tag Editor** in the AWS Resource Groups console to view, add, or remove tags across of all of your AWS resources across all Regions. You can apply or remove tags from multiple types of resources at the same time. For more information, see the Tagging AWS Resources User Guide.

Tasks

- Add tags using the console
- Add tags using the AWS CLI
- Add tags using PowerShell
- Add tags using CloudFormation

Add tags using the console

You can add tags to an existing resource directly from the page for a resource.

To add tags to an existing resource

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region where the resource is located.
3. In the navigation pane, select a resource type (for example, **Instances**).
4. Select the resource from the list.

5. From the **Tags** tab, choose **Manage tags**.
6. Choose **Add new tag** and enter a key and a value for the tag.
7. Choose **Save**.

Add tags using the AWS CLI

You can add tags when you create a resource or to an existing resource.

To add a tag on resource creation

Use the `-tag-specifications` option to tag a resource on creation. A tag specification requires the type of resource to be tagged, the tag key, and the tag value. The following example creates a tag and adds it to a tag specification.

```
--tag-specifications 'ResourceType=instance,Tags=[{Key=stack,Value=production}]'
```

To add a tag to an existing resource

The following examples demonstrate how to add tags to existing resources 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-0abcdef1234567890 \
    --tags Key=stack,Value=production
```

Example Example: Add tags to multiple resources

This example adds (or overwrites) two tags for an AMI and an instance. One of the tags contains just a key (webserver), with no value (we set the value to an empty string). The other tag consists of a key (stack) and value (**`Production`**). If the command succeeds, no output is returned.

```
aws ec2 create-tags \
    --resources ami-0abcdef1234567890 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'
```

Add tags using PowerShell

You can add tags when you create a resource or to an existing resource.

To add a tag on resource creation

Use the `-TagSpecification` parameter to tag a resource on creation. A tag specification requires the type of resource to be tagged, the tag key, and the tag value. The following example creates a tag and adds it to a tag specification.

```
$tag = @{Key="stack"; Value="production"}
$tagspec = new-object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance"
$tagspec.Tags.Add($tag)
```

The following example specifies this tag in the `-TagSpecification` parameter.

```
-TagSpecification $tagspec
```

To add a tag to an existing resource

Use the New-EC2Tag cmdlet. You must specify the resource, the tag key, and the tag value.

```
New-EC2Tag `
    -Resource i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="production"}
```

Add tags 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"
            }
        ]
    }
]
```

Filter Amazon EC2 resources by tag

After you add tags, you can filter your Amazon EC2 resources based tag keys and tag values.

Console

To filter resources by tag

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, select a resource type (for example, **Instances**).
3. Choose the search field.
4. In the list, under **Tags**, choose the tag key.
5. Choose the corresponding tag value from the list.
6. When you are finished, remove the filter.

For more information about using filters in the Amazon EC2 console, see Find your Amazon EC2 resources.

AWS CLI

To describe resources of a single type with the specified tag key

Add the following filter to a `describe` command to describe the resources of that type with a Stack tag, regardless of the value of the tag.

```
--filters Name=tag-key,Values=Stack
```

To describe resources of a single type with the specified tag

Add the following filter to a `describe` command to describe the resources of that type with the tag Stack=production.

```
--filters Name=tag:Stack,Values=production
```

To describe resources of a single type with the specified tag value

Add the following filter to a `describe` command to describe the resources of that type with a tag with the value production, regardless of the tag key.

```
--filters Name=tag-value,Values=production
```

To describe all EC2 resources with the specified tag

Add the following filter to the describe-tags command to describe all EC2 resources with the tag Stack=test.

```
--filters Name=key,Values=Stack Name=value,Values=test
```

PowerShell

To filter resources of a single type by tag key

Add the following filter to a `Get` cmdlet to describe the resources of that type with a Stack tag, regardless of the value of the tag.

```
-Filter @{Name="tag-key"; Values="Stack"}
```

To filter resources of a single type by tag

Add the following filter to a `Get` cmdlet to describe the resources of that type with the tag Stack=production.

```
-Filter @{Name="tag:Stack"; Values="production"}
```

To filter resources of a single type by tag value

Add the following filter to a `Get` cmdlet to describe the resources of that type with a tag with the value production, regardless of the value of the tag key.

```
-Filter @{Name="tag-value"; Values="production"}
```

To filter all EC2 resources by tag

Add the following filter to the [Get-EC2Tag](#) cmdlet to describe all EC2 resources with the tag Stack=test.

```
-Filter @{Name="tag:Stack"; Values="test"}
```

View tags for your EC2 instances using instance metadata

You can access an instance's tags from the instance metadata. By accessing tags from the instance metadata, you no longer need to use the `DescribeInstances` or `DescribeTags` API calls to retrieve tag information, which reduces your API transactions per second, and lets your tag retrievals scale with the number of instances that you control. Furthermore, local processes that are running on an instance can view the instance's tag information directly from the instance metadata.

By default, tags are not available from the instance metadata; you must explicitly allow access. You can allow access at instance launch, or after launch on a running or stopped instance. You can also allow access to tags by specifying this in a launch template. Instances that are launched by using the template allow access to tags in the instance metadata.

If you add or remove an instance tag, the instance metadata is updated while the instance is running, without needing to stop and then start the instance.

Tasks

- Enable access to tags in instance metadata
- Retrieve tags from instance metadata
- Disable access to tags in instance metadata

Enable access to tags in instance metadata

By default, there is no access to instance tags in the instance metadata. For each instance, you must explicitly enable access.

> **Note**
>
> If you allow access to tags in instance metadata, instance tag *keys* are subject to specific restrictions. Non-compliance will result in failed launches for new instances or an error for existing instances. The restrictions are:
>
> - Can only include letters (`a-z`, `A-Z`), numbers (`0-9`), and the following characters: `+ - = . , _ : @`.
> - Can't contain spaces or `/`.
> - Can't consist only of `.` (one period), `..` (two periods), or `_index`.
>
> For more information, see Tag restrictions.

Console

To enable access to tags in instance metadata during instance launch

1. Follow the procedure to launch an instance.
2. Expand **Advanced details**, and for **Allow tags in metadata**, choose **Enable**.
3. In the **Summary panel**, review your instance configuration, and then choose **Launch instance**. For more information, see Launch an EC2 instance using the launch instance wizard in the console.

To enable access to tags in instance metadata after instance 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 then choose **Actions**, **Instance settings**, **Allow tags in instance metadata**.
4. To allow access to tags in instance metadata, select the **Allow** checkbox.
5. Choose **Save**.

AWS CLI

To enable access to tags in instance metadata during instance launch

Use the run-instances command and add the following `--metadata-options` option.

```
--metadata-options "InstanceMetadataTags=enabled"
```

To enable access to tags in instance metadata after instance launch

Use the following modify-instance-metadata-options command.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags enabled
```

To verify that access to tags in instance metadata is enabled

Use the describe-instances command and check the value of `InstanceMetadataTags`.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].MetadataOptions[].InstanceMetadataTags"
```

The following is example output. The value is either `enabled` or `disabled`.

```
[
    "enabled"
]
```

PowerShell

To enable access to tags in instance metadata during instance launch

Use the New-EC2Instance cmdlet and add the following -MetadataOptions_InstanceMetadataTags parameter.

```
-MetadataOptions_InstanceMetadataTags enabled
```

To enable access to tags in instance metadata after instance launch

Use the Edit-EC2InstanceMetadataOption cmdlet.

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTags enabled
```

To verify that access to tags in instance metadata is enabled

Use the Get-EC2Instance cmdlet and check the value of `InstanceMetadataTags`.

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.MetadataOptions.InstanceMetadataTags.Value
```

The following is example output. The value is either `enabled` or `disabled`.

```
enabled
```

Retrieve tags from instance metadata

After you allow access to instance tags in the instance metadata, you can access the `tags/instance` category from the instance metadata. For more information, see Access instance metadata for an EC2 instance.

IMDSv2

Linux

Run the following command from your Linux instance to list all the tag keys for the instance.

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance
```

This example gets the value of a key obtained in the previous example. The IMDSv2 request uses the stored token that was created using the command in the previous example. The token must not be expired.

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

Run the following cmdlet from your Windows instance to list all the tag keys for the instance.

```
$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

This example gets the value of a key obtained in the previous example. The IMDSv2 request uses the stored token that was created using the command in the previous example. The token must not be expired.

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

IMDSv1

Linux

Run the following command from your Linux instance to list all the tag keys for the instance.

```
curl http://169.254.169.254/latest/meta-data/tags/instance
```

This example gets the value of a key obtained in the previous example.

```
curl http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

Run the following cmdlet from your Windows instance to list all the tag keys for the instance.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

This example gets the value of a key obtained in the previous example.

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Disable access to tags in instance metadata

You can disable access to instance tags in the instance metadata. You don't need to disable access to instance tags on instance metadata at launch because it's turned off by default.

Console

To disable access to tags in instance metadata

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 then choose **Actions**, **Instance settings**, **Allow tags in instance metadata**.
4. To turn off access to tags in instance metadata, clear the **Allow** checkbox.
5. Choose **Save**.

AWS CLI

To disable access to tags in instance metadata

Use the following modify-instance-metadata-options command.

```
aws ec2 modify-instance-metadata-options \
```

```
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags disabled
```

PowerShell

To disable access to tags in instance metadata

Use the Edit-EC2InstanceMetadataOption cmdlet.

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTag disabled
```

Amazon EC2 service quotas

When you create your AWS account, we set default *quotas* (also referred to as limits) on your AWS resources on a per-Region basis. If you attempt to exceed the quota for a resource, the request fails. For example, there is a maximum number of Amazon EC2 vCPUs that you can provision for On-Demand Instances in a Region. If you attempt to launch an instance in a Region and this request would cause your usage to exceed this quota, the request fails. If this happens, you can reduce your resource usage or request a quota increase.

The Service Quotas console is a central location where you can view and manage your quotas for AWS services, and request a quota increase for many of the resources that you use. Use the quota information that we provide to manage your AWS infrastructure. Plan to request any quota increases in advance of the time that you'll need them.

Related quota documentation

- Amazon EC2 endpoints and quotas
- Amazon EC2 instance type quotas
- Quotas for Amazon EBS
- Amazon VPC quotas

View your current quotas

You can view your quotas for each Region using the Service Quotas console.

To view your current quotas using the Service Quotas console

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/.
2. From the navigation bar (at the top of the screen), select a Region.
3. On the **Manage quotas** pane, select a service; for example, **Amazon Elastic Compute Cloud (Amazon EC2)**. Choose **View quotas**.
4. Use the filter field to filter the list by resource name. For example, enter **`On-Demand`** to locate the quotas for On-Demand Instances.
5. To view more information, choose the quota name to open the details page for the quota.

Request an increase

You can request a quota increase for each Region.

To request an increase using the Service Quotas console

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/.
2. From the navigation bar (at the top of the screen), select a Region.
3. Use the filter field to filter the list by resource name. For example, enter **`On-Demand`** to locate the quotas for On-Demand Instances.
4. If the quota is adjustable, choose the quota and then choose **Request quota increase**.
5. For **Change quota value**, enter the new quota value.
6. Choose **Request**.
7. To view any pending or recently resolved requests in the console, choose **Dashboard** from the navigation pane. For pending requests, choose the status of the request to open the request receipt. The initial status of a request is **Pending**. After the status changes to **Quota requested**, you'll see the case number with Support. Choose the case number to open the ticket for your request.

For more information, including how to use the AWS CLI or SDKs to request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Restriction on email sent using port 25

By default, Amazon EC2 allows outbound traffic over port 25 only to private IPv4 addresses. Traffic over port 25 is blocked to public IPv4 addresses and IPv6 addresses.

You can request that this restriction be removed. For more information, see How do I remove the restriction on port 25 from my Amazon EC2 instance or Lambda function?

This restriction does not apply to outbound traffic over port 25 destined for:

- IP addresses in the primary CIDR block of the VPC in which the originating network interface exists.
- IP addresses in the CIDRs defined in RFC 1918, RFC 6598, and RFC 4193.

Monitor Amazon EC2 resources

Monitoring is an important part of maintaining the reliability, availability, and performance of your 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.

AWS provides various tools that you can use to monitor Amazon EC2. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment. In addition, we provide the following:

- **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 Amazon EC2 instances.
- **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 Amazon EC2 instances.
- **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.

- **Amazon EventBridge events** – 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 the section called "Automate using EventBridge".
- **AWS CloudTrail logs** – Capture detailed information about the calls made to the Amazon EC2 API and stores them as log files in Amazon S3. You can use CloudTrail logs to determine which calls were made, the source IP address for the call, who made the call, and when the call was made. For more information, see the section called "Log API calls using CloudTrail".
- **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*.

Monitor the status of your Amazon EC2 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 the 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 Amazon EC2 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. 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 CloudWatch alarms for Amazon EC2 instances that fail status checks.

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 Automatic instance recovery.

Contents

Types of status checks

There are three types of status checks.

- System status checks
- Instance status checks
- Attached EBS status checks

System status checks

System status checks monitor the AWS systems on which your instance runs. These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself. For instances backed by Amazon EBS, you can stop and start the instance yourself, which in most cases results in the instance being migrated to a new host. For instances backed by instance store (supported only for Linux instances), you can terminate and replace the instance. 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

If a system status check fails, we increment the StatusCheckFailed_System metric.

Bare metal instances

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 connectivity 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).

> **Note**
>
> Recent Linux distributions that use `systemd-networkd` for network configuration might report on health checks differently from earlier distributions. During the boot process, this type of network can start earlier and potentially finish before other startup tasks that can

also affect instance health. Status checks that depend on network availability can report a healthy status before other tasks complete.

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel
- During a reboot, an instance status check reports a failure until the instance becomes available again.

If an instance status check fails, we increment the StatusCheckFailed_Instance metric.

Bare metal instances

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.

Attached EBS status checks

Attached EBS status checks monitor if the Amazon EBS volumes attached to an instance are reachable and able to complete I/O operations. The `StatusCheckFailed_AttachedEBS` metric is a binary value that indicates impairment if one or more of the EBS volumes attached to the instance are unable to complete I/O operations. These status checks detect underlying issues with the compute or Amazon EBS infrastructure. When the attached EBS status check metric fails, you can either wait for AWS to resolve the issue, or you can take actions, such as replacing the affected volumes or stopping and restarting the instance.

The following are examples of issues that can cause attached EBS status checks to fail:

- Hardware or software issues on the storage subsystems underlying the EBS volumes
- Hardware issues on the physical host that impact reachability of the EBS volumes
- Connectivity issues between the instance and EBS volumes

You can use the `StatusCheckFailed_AttachedEBS` metric to help improve the resilience of your workload. You can use this metric to create Amazon CloudWatch alarms that are triggered based on the result of the status check. For example, you could fail over to a secondary instance or Availability Zone when you detect a prolonged impact. Alternatively, you can monitor the I/O performance of each attached volume using EBS CloudWatch metrics to detect and replace the impaired volume. If your workload is not driving I/O to any EBS volumes attached to your instance, and the EBS status check indicates an impairment, you can stop and start the instance to move it to a new host. This can resolve underlying host issues that are impacting the reachability of the EBS volumes. For more information, see Amazon CloudWatch metrics for Amazon EBS.

You can also configure your Amazon EC2 Auto Scaling groups to detect attached EBS status check failures, and then replace the affected instance with a new one. For more information, see Monitor and replace Auto Scaling instances with impaired Amazon EBS volumes in the *Amazon EC2 Auto Scaling User Guide*.

> **Note**
>
> The attached EBS status check metric is available only for Nitro instances.

View status checks for Amazon EC2 instances

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). To troubleshoot system or instance status check failures yourself, see Troubleshoot Amazon EC2 Linux instances with failed status checks.

Console

To view status checks

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 and alarms** tab.

5. To review the CloudWatch metrics for status checks, on the **Status and alarms** tab, expand **Metrics** to see the graphs for the following metrics:

   - **Status check failed for system**
   - **Status check failed for instance**
   - **Status check failed for attached EBS**

   For more information, see the section called "Status check metrics".

AWS CLI

To view status checks

Use the describe-instance-status command.

Example: Get the status of all running instances

```
aws ec2 describe-instance-status
```

Example: Get the status of all instances

```
aws ec2 describe-instance-status --include-all-instances
```

Example: Get the status of a single running instance

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Example: Get all instances with a status of `impaired`

```
aws ec2 describe-instance-status \
--filters Name=instance-status.status,Values=impaired
```

PowerShell

To view status checks

Use the Get-EC2InstanceStatus command.

Example: Get the status of all running instances

```
Get-EC2InstanceStatus
```

Example: Get the status of all instances

```
Get-EC2InstanceStatus -IncludeAllInstance $true
```

Example: Get the status of a single running instance

```
Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0
```

Example: Get all instances with a status of `impaired`

```
Get-EC2InstanceStatus \
-Filter @{Name="instance-status.status"; Values="impaired"}
```

Create CloudWatch alarms for Amazon EC2 instances that fail status checks

You can use the [status check metrics](https://docs.aws.amazon.com/) to create CloudWatch alarms to notify you when an instance has a failed status check.

Status checks and status check alarms can temporarily enter an *insufficient data* state if there are missing metric data points. Although rare, this can happen when there is an interruption in the metric reporting systems, even when an instance is healthy. We recommend that you treat this state as missing data instead of a status check failure or alarm breach. This is especially important when taking stop, terminate, reboot, or recover actions on the instance in response.

Console

This example configures an alarm that sends a notification when an instance fails a status check. You can optionally stop, terminate, or recover the instance.

To create a status check alarm

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, 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 create a new topic, Amazon SNS sends a confirmation email to each new address. Each recipient must choose the confirmation link in the email. Only confirmed addresses receive alert notifications.
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**.

If you need to change an instance status alarm, you can edit it.

To edit a status check alarm

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance 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**.

AWS CLI

In the following example, the alarm publishes a notification to an 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

1. Select an existing SNS topic or create a new one. For more information, see Accessing Amazon SNS in the AWS CLI in the *AWS Command Line Interface User Guide*.
2. Use the following list-metrics command to view the available Amazon CloudWatch metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following put-metric-alarm command to create the alarm.

```
aws cloudwatch put-metric-alarm \
    --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 \
    --metric-name StatusCheckFailed \
    --namespace AWS/EC2 \
    --statistic Maximum \
    --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
    --unit Count \
    --period 300 \
    --evaluation-periods 2 \
    --threshold 1 \
    --comparison-operator GreaterThanOrEqualToThreshold \
    --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

The period is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes. The evaluation period is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2. The alarm actions are the actions to perform when this alarm is triggered.

PowerShell

To create a status check alarm

Use the Write-CWMetricAlarm cmdlet as follows to publish notifications to an SNS topic when the instance fails status checks for at least two consecutive periods.

```
Write-CWMetricAlarm `
    -AlarmName "StatusCheckFailed-Alarm-for-i-1234567890abcdef0" `
    -MetricName "StatusCheckFailed" `
    -Namespace "AWS/EC2" `
    -Statistic "Maximum" `
    -Dimension @{Name="InstanceId"; Values="i-1234567890abcdef0"} `
    -Unit "Count" `
    -Period 300 `
    -EvaluationPeriod 2 `
    -Threshold 1 `
    -ComparisonOperator "GreaterThanOrEqualToThreshold" `
    -AlarmAction "arn:aws:sns:us-west-2:111122223333:my-sns-topic"
```

The period is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes. The evaluation period is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2. The alarm actions are the actions to perform when this alarm is triggered.

State change events for Amazon EC2 instances

Amazon EC2 sends an `EC2 Instance State-change Notification` event to Amazon EventBridge when the state of an instance changes.

The following is example data for this event. In this example, the instance entered the `pending` state.

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2021-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
   ],
```

```
    "detail":{
       "instance-id":"i-1234567890abcdef0",
       "state":"pending"
    }
}
```

The possible values for `state` are:

- `pending`
- `running`
- `stopping`
- `stopped`
- `shutting-down`
- `terminated`

When you launch or start an instance, it enters the `pending` state and then the `running` state. When you stop an instance, it enters the `stopping` state and then the `stopped` state. When you terminate an instance, it enters the `shutting-down` state and then the `terminated` state. For more information, see [Amazon EC2 instance state changes](#).

Create an alarm that sends an email when an Amazon EC2 instance changes state

To receive email notifications when your instance changes state, create an Amazon SNS topic and then create an EventBridge rule for the `EC2 Instance State-change Notification` event.

To create an SNS topic

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Topics**.
3. Choose **Create topic**.
4. For **Type**, choose **Standard**.
5. For **Name**, enter a name for your topic.
6. Choose **Create topic**.
7. Choose **Create subscription**.
8. For **Protocol**, choose **Email**.

9. For **Endpoint**, enter the email address that receives the notifications.
10. Choose **Create subscription**.
11. You'll receive an email message with the following subject line: AWS Notification - Subscription Confirmation. Follow the directions to confirm your subscription.

To create an EventBridge rule

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Choose **Create rule**.
3. For **Name**, enter a name for your rule.
4. For **Rule type**, choose **Rule with an event pattern**.
5. Choose **Next**.
6. For **Event pattern**, do the following:

   a. For **Event source**, choose **AWS services**.

   b. For **AWS service**, choose **EC2**.

   c. For **Event type**, choose **EC2 Instance State-change Notification**.

   d. By default, we send notifications for any state change for any instance. If you prefer, you can select specific states or specific instances.
7. Choose **Next**.
8. Specify a target as follows:

   a. For **Target types**, choose **AWS service**.

   b. For **Select a target**, choose **SNS topic**.

   c. For **Topic**, choose the SNS topic that you created in the previous procedure.
9. Choose **Next**.
10. (Optional) Add tags to your rule.
11. Choose **Next**.
12. Choose **Create rule**.
13. To test your rule, initiate a state change. For example, start a stopped instance, stop a running instance, or launch an instance. You'll receive email messages with the following subject line: AWS Notification Message. The body of the email contains the event data.

Scheduled events for Amazon EC2 instances

To ensure infrastructure reliability and performance, AWS can schedule events to reboot, stop, and retire your instances. These events do not occur frequently.

If one of your instances will be affected by a scheduled event, AWS notifies you in advance by email, using the email address that's associated with your AWS account. The email provides details about the event, such as the start and end dates. Depending on the event type, 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 EventBridge. For more information, see Monitoring events in AWS Health with Amazon EventBridge.

Scheduled events are managed by AWS. You can't schedule events for your instances. However, you can:

- View scheduled events for your instances.
- Customize scheduled event notifications to include or remove tags from the email notification.
- Reschedule certain scheduled events.
- Create custom event windows for scheduled events.
- Take action when an instance is scheduled to reboot, stop, or retire.

To ensure that you receive notifications of scheduled events, verify your contact information on the Account page.

Note

When an instance is affected by a scheduled event, and it is part of an Auto Scaling group, Amazon EC2 Auto Scaling eventually replaces it as part of its health checks, with no further action necessary on your part. For more information about the health checks performed by Amazon EC2 Auto Scaling, see Health checks for instances in an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.

Types of scheduled events

Amazon EC2 can create the following types of scheduled events for your instances, where the event occurs at a scheduled time:

Event type	Event code	Event action
Instance stop	`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 with an Amazon EBS root volume.
Instance retirement	`instance-retirement`	At the scheduled time, the instance is stopped if it has an Amazon EBS root volume, or terminated if it has an instance store root volume.
Instance reboot	`instance-reboot`	At the scheduled time, the instance is rebooted. The instance stays on the host, and during the reboot, the host undergoes maintenance. This is known as an in-place reboot.
System reboot	`system-reboot`	At the scheduled time, the instance is rebooted and migrated to a new host. This is known as a reboot migration.
System maintenance	`system-maintenance`	At the scheduled time, the instance might be temporarily affected by network maintenance or power maintenance.

Determine the event type

You can check what type of event is scheduled for your instance.

Console

To determine the event type

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Events**.
3. In the table, the event code appears in the **Event type** column.
4. To filter the table to show only the events for instances, in the search field choose **Resource type: instance** from the filter list.

AWS CLI

To determine the event type for an instance

Use the describe-instance-status command. If the instance has an associated scheduled event, the output provides information about the scheduled event.

```
aws ec2 describe-instance-status \
    --instance-id i-1234567890abcdef0 \
    --query InstanceStatuses[].Events
```

The following is example output. The scheduled event code is `system-reboot`.

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2020-03-14T22:00:00.000Z",
            "NotBefore": "2020-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2020-04-05T11:00:00.000Z"
        }
    ]
]
```

PowerShell

To determine the event type for an instance

Use the Get-EC2InstanceStatus cmdlet. If the instance has an associated scheduled event, the output provides information about the scheduled event.

```
(Get-EC2InstanceStatus `
    -InstanceId i-1234567890abcdef0).Events
```

The following is example output. The scheduled event code is `system-reboot`.

```
Code               : system-reboot
Description        : The instance is scheduled for a reboot
InstanceEventId    : instance-event-0d59937288b749b32
NotAfter           : 2020-03-14T22:00:00.000Z
NotBefore          : 2020-03-14T20:00:00.000Z
NotBeforeDeadline  : 2020-04-05T11:00:00.000Z
```

Contents

Manage Amazon EC2 instances scheduled to stop or retire

When AWS detects irreparable failure of the underlying host for your instance, it schedules the instance to either stop or terminate, depending on the instance's root volume type.

- If the instance has an Amazon EBS root volume, the instance is scheduled to stop.
- If the instance has an instance store root volume, the instance is scheduled to terminate.

For more information, see Instance retirement.

> **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 you can take

Actions you can take for instances with an EBS root volume

When you receive a scheduled `instance-stop` event notification, you can take one of the following actions:

- **Wait for scheduled stop:** You can wait for the instance to stop within its scheduled maintenance window.
- **Perform manual stop and start:** You can stop and start the instance yourself at a time that suits you, which migrates it to a new host. This is not the same as rebooting the instance. For more information, see Stop and start Amazon EC2 instances.
- **Automate stop and start:** You can automate an immediate stop and start in response to a scheduled `instance-stop` event. For more information, see Running operations on EC2 instances automatically in response to events in AWS Health in the *AWS Health User Guide*.

Actions you can take for instances with an instance store root volume

When you receive a scheduled `system-retirement` event notification, and you want to retain your data, you can take the following actions:

1. Launch a replacement instance from your most recent AMI.
2. Migrate all necessary data to the replacement instance before the instance is scheduled to terminate.
3. Terminate the original instance, or wait for it to terminate as scheduled.

For more information about the actions you can take, see Instance retirement.

Manage Amazon EC2 instances scheduled for reboot

When AWS must perform tasks such as installing updates or maintaining the underlying host, it can schedule an instance reboot. During the scheduled reboot, the instance either stays on the same host, or migrates to a different host, depending on the event, as follows:

- `instance-reboot` event
  - During the reboot, the instance remains on the host. This is known as an *in-place reboot*.
  - The current host undergoes maintenance.
  - Typically completes in seconds.
- `system-reboot` event
  - During the reboot, the instance is migrated to a new host. This is known as a *reboot migration*.
  - Typically completes in minutes.

To check what type of event is scheduled for your instance, see Determine the event type.

Actions you can take

When you receive a scheduled `instance-reboot` or `system-reboot` event notification, you can take one of the following actions:

- **Wait for scheduled reboot:** You can wait for the instance reboot to occur within its scheduled maintenance window.
- **Reschedule the reboot:** You can reschedule the instance reboot to a date and time that suits you.
- **Perform a user-initiated reboot:** You can manually reboot the instance yourself at a time that suits you. However, the outcome depends on the event:
  - `instance-reboot` event – Your instance remains on the current hardware (in-place reboot), no host maintenance takes place, and the event stays open.
  - `system-reboot` event
    - If reboot migration is enabled on your instance, a user-initiated reboot attempts to migrate your instance to new hardware. If successful, the event is cleared. If unsuccessful, an in-place reboot occurs and the event remains scheduled.
    - If reboot migration is disabled on your instance, a user-initiated reboot keeps the instance on the same hardware (in-place reboot), no host maintenance takes place, and the event

remains scheduled. When the scheduled event eventually takes place, AWS will move your instance to new hardware (reboot migration).

After AWS reboots your instance

The following applies after AWS reboots your instance:

- The scheduled event is cleared.
- The event description is updated.
- For an `instance-reboot` event:
  - Maintenance of the underlying host is complete.
- For a `system-reboot` event:
  - The instance moves to a new host.
  - The instance retains its IP address and DNS name.
  - Any data on local instance store volumes is preserved.
- You can use your instance after it has fully booted.

Alternative options

If you can't reschedule the reboot event or enable reboot migration for a user-initiated reboot, but want to maintain normal operation during the scheduled maintenance window, you can do the following:

- **For an instance with an EBS root volume**
  - Manually stop and start the instance to migrate it to a new host. This is not the same as manually rebooting the instance, where the instance stays on the same host.
  - Optionally, automate an immediate instance stop and start in response to the scheduled reboot event. For more information, see Running operations on EC2 instances automatically in response to events in AWS Health in the *AWS Health User Guide*.

> **Important**
>
> The data on instance store volumes is lost when an instance is stopped. For more information, see Stop and start Amazon EC2 instances.

- **For an instance with an instance store root volume**

1. Launch a replacement instance from your most recent AMI.
2. Migrate all necessary data to the replacement instance before the scheduled maintenance window.
3. Terminate the original instance.

Enable or disable reboot migration

When an instance is scheduled for a `system-reboot` event, you can reboot it before the event. The outcome of a user-initiated reboot depends on the instance's reboot migration setting:

- Enabled – A user-initiated reboot attempts to migrate your instance to new hardware (reboot migration). If successful, the event is cleared. If unsuccessful, an in-place reboot occurs and the event remains scheduled. Note that even when enabled, reboot migration can only occur if your instance meets the reboot migration requirements.
- Disabled – A user-initiated reboot keeps the instance on the same hardware (in-place reboot), no host maintenance takes place, and the event remains scheduled. When the scheduled event eventually takes place, AWS will move your instance to new hardware (reboot migration).

A reboot with migration takes longer than an in-place reboot:

- In-place reboot: Approximately 30 seconds
- Reboot with migration: Several minutes

> **Note**
>
> Instances that receive a `system-reboot` event notification are enabled for user-initiated reboot migration by default.

Requirements for enabling reboot migration

Reboot migration can be enabled on instances that meet the following criteria:

Instance types

Not all instance types support enabling reboot migration. You can view the instance types that support enabling reboot migration.

Console

To view the instance types that support enabling reboot migration

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instance Types**.
3. In the filter bar, enter **Reboot Migration support: supported**. As you enter the characters and the filter name appears, you can select it.

   The **Instance types** table displays all the instance types that support enabling reboot migration.

AWS CLI

To view the instance types that support enabling reboot migration

Use the describe-instance-types command with the `reboot-migration-support` filter.

```
aws ec2 describe-instance-types \
    --filters Name=reboot-migration-support,Values=supported \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

To view the instance types that support enabling reboot migration

Use the Get-EC2InstanceType cmdlet with the `reboot-migration-support` filter.

```
Get-EC2InstanceType `
    -Filter @{Name="reboot-migration-support";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

Tenancy

- Shared
- Dedicated Instance

For more information, see Amazon EC2 Dedicated Instances.

Limitations

Reboot migration is **not** supported for instances with the following characteristics:

- Platform: Instances running natively on the Xen hypervisor
- Instance size: `metal` instances
- Tenancy: Dedicated Host. For Dedicated Hosts, use Dedicated Host Auto Recovery instead.
- Storage: Instances with instance store volumes
- Networking: Instances using an Elastic Fabric Adapter
- Auto Scaling: Instances that are part of an Auto Scaling group

Steps for enabling or disabling reboot migration

When an instance receives a `system-reboot` event, it is enabled for reboot migration by default. You can disable reboot migration so that during a user-initiated reboot, the instance stays on the same hardware (in-place reboot).

The `default` configuration doesn't enable reboot migration for an unsupported instance. For more information, see Requirements for enabling reboot migration.

You can disable or enable reboot migration on a running or stopped instance.

AWS CLI

To disable reboot migration

Use the modify-instance-maintenance-options command and set the `--reboot-migration` parameter to `disabled`.

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration disabled
```

To enable reboot migration

Use the modify-instance-maintenance-options command and set the `--reboot-migration` parameter to `default`.

```
aws ec2 modify-instance-maintenance-options \
```

```
    --instance-id i-0abcdef1234567890 \

    --reboot-migration default
```

PowerShell

To disable reboot migration

Use the Edit-EC2InstanceMaintenanceOption cmdlet.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId  `
    -RebootMigration Disabled
```

To enable reboot migration

Use the Edit-EC2InstanceMaintenanceOption cmdlet.

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -RebootMigration Enabled
```

Manage Amazon EC2 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 you can take

Actions you can take for instances with an EBS root volume

When you receive a `system-maintenance` event notification, you can take one of the following actions:

- **Wait for scheduled maintenance:** You can wait for the maintenance to occur as scheduled.
- **Perform manual stop and stop:** You can stop and start the instance, which migrates it to a new host. This is not the same as rebooting the instance. For more information, see Stop and start Amazon EC2 instances.
- **Automate stop and start:** You can automate an immediate stop and start in response to a scheduled maintenance event. For more information, see Running operations on EC2 instances automatically in response to events in AWS Health in the *AWS Health User Guide*.

Actions you can take for instances with an instance store root volume

When you receive a `system-maintenance` event notification, you can take one of the following actions:

- **Wait for scheduled maintenance:** You can wait for the maintenance to occur as scheduled.
- **Launch a replacement instance:** If you want to maintain normal operation during the scheduled maintenance window:
  1. Launch a replacement instance from your most recent AMI.
  2. Migrate all necessary data to the replacement instance before the scheduled maintenance window.
  3. Terminate the original instance.

View scheduled events that affect your Amazon EC2 instances

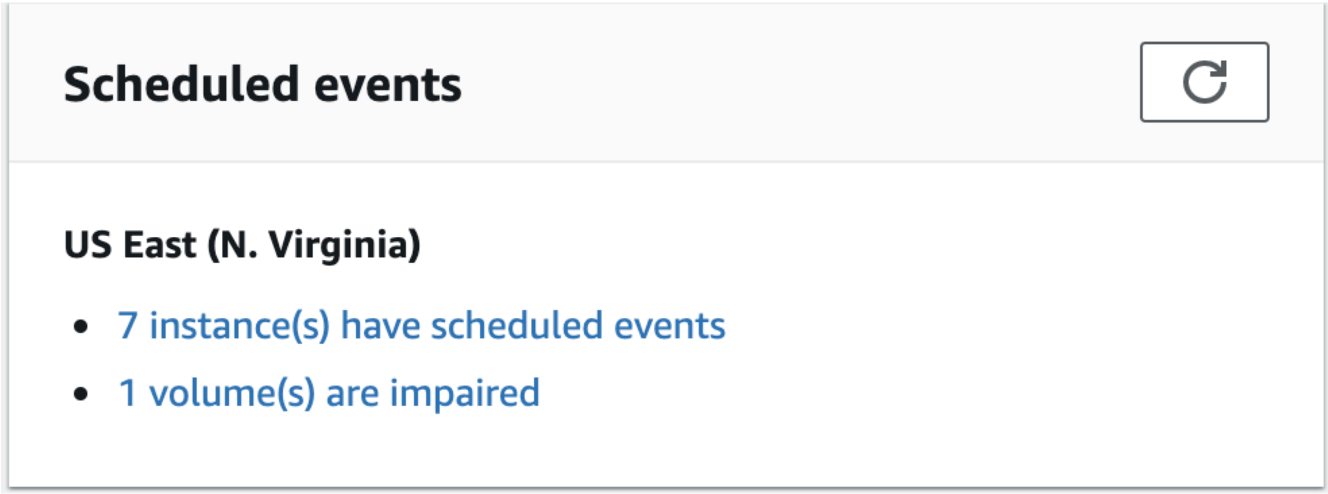
In addition to receiving notification of scheduled events in email, you can check for scheduled events.

Console

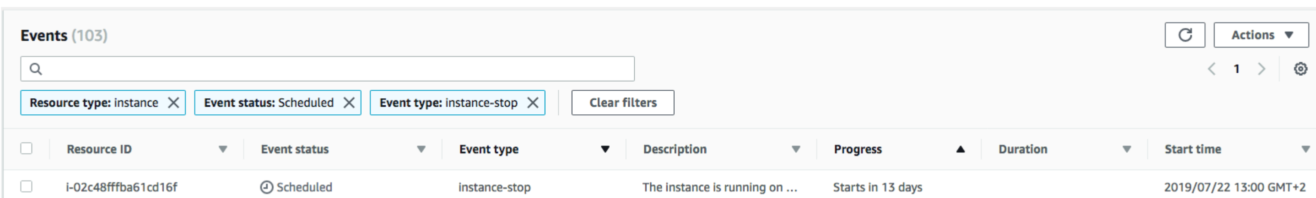
To view scheduled events for your instances

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

2. The dashboard displays any resources with an associated event under **Scheduled events**.



3. For more detail, choose **Events** in the navigation pane. Any resources with an associated event are displayed. You can filter by characteristics such as event type, resource type, and Availability Zone.



AWS CLI

To view scheduled events for your instances

Use the describe-instance-status command.

```
aws ec2 describe-instance-status \
    --instance-id i-1234567890abcdef0 \
    --query "InstanceStatuses[].Events"
```

The following example output shows a reboot event.

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
```

```
                "NotAfter": "2019-03-15T22:00:00.000Z",
                "NotBefore": "2019-03-14T20:00:00.000Z",
                "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
            }

    ]
]
```

The following example output shows an instance retirement event.

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0e439355b779n26",

            "Code": "instance-stop",
            "Description": "The instance is running on degraded hardware",
            "NotBefore": "2015-05-23T00:00:00.000Z"
        }
    ]
]
```

PowerShell

To view scheduled events for your instances

Use the following Get-EC2InstanceStatus command.

```
(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 by using Instance Metadata Service Version 2 or Instance Metadata Service Version 1.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

The following is example output with information about a scheduled system reboot event, in JSON format.

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "active"
  }
]
```

To view event history about completed or canceled events for your instances using instance metadata

You can retrieve information about completed or canceled events for your instances from instance metadata by using Instance Metadata Service Version 2 or Instance Metadata Service Version 1.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/maintenance/history
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/
history
```

The following is example output with information about a system reboot event that was canceled, and a system reboot event that was completed, in JSON format.


```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Canceled] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "canceled"
  },
  {
    "NotBefore" : "29 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Completed] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "29 Jan 2019 09:17:23 GMT",
    "State" : "completed"
  }
]
```


AWS Health

You can use the AWS Health Dashboard to learn about events that can affect your instance. The AWS 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 your AWS Health Dashboard in the *AWS Health User Guide*.

Customize scheduled event notifications for your EC2 instances

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.

Tasks

- Include tags in event notifications
- Remove tags from event notifications
- View the tags to be included in event notifications

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.

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. Turn on **Include 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 tags associated with the affected instance or Dedicated Host, select **Include all tags**.

- To select the tags to include, select **Choose the tags to include** and then select or enter the tag keys.

6. Choose **Save**.

AWS CLI

To include all tags in event notifications

Use the register-instance-event-notification-attributes command and set the `IncludeAllTagsOfInstance` parameter to `true`.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute "IncludeAllTagsOfInstance=true"
```

To include specific tags in event notifications

Use the register-instance-event-notification-attributes 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"]'
```

PowerShell

To include all tags in event notifications

Use the Register-EC2InstanceEventNotificationAttribute cmdlet.

```
Register-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_IncludeAllTagsOfInstance $true
```

To include specific tags in event notifications

Use the Register-EC2InstanceEventNotificationAttribute cmdlet.

```
Register-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_InstanceTagKey tag_key_1, tag_key_2, tag_key_3
```

Remove tags from event notifications

You can remove tags from event notifications.

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. To remove all tags from event notifications, turn off **Include tags in event notifications**.
5. To remove specific tags from event notifications, choose the **X**) for the corresponding tag keys.
6. Choose **Save**.

AWS CLI

To remove all tags from event notifications

Use the deregister-instance-event-notification-attributes command and set the `IncludeAllTagsOfInstance` parameter to `false`.

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute "IncludeAllTagsOfInstance=false"
```

To remove a tag from event notifications

Use the deregister-instance-event-notification-attributes 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_3"]'
```

PowerShell

To remove all tags from event notifications

Use the Unregister-EC2InstanceEventNotificationAttribute cmdlet.

```
Unregister-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_IncludeAllTagsOfInstance $false
```

To remove a tag from event notifications

Use the Unregister-EC2InstanceEventNotificationAttribute cmdlet.

```
Unregister-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_InstanceTagKey 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.

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 to be included in event notifications

Use the describe-instance-event-notification-attributes command.

```
aws ec2 describe-instance-event-notification-attributes
```

PowerShell

To view the tags to be included in event notifications

Use the Get-EC2InstanceEventNotificationAttribute cmdlet.

```
Get-EC2InstanceEventNotificationAttribute
```

Reschedule a scheduled event for an EC2 instance

You can reschedule an event so that it occurs at a specific date and time that suits you. After you reschedule an event, it might take a minute or two before the the updated date is displayed.

Limitations

- Only events with an event deadline date can be rescheduled. The event can be rescheduled up to the event deadline date. The event deadline date is indicated in the **Deadline** column (console) and the `NotBeforeDeadline` field (AWS CLI).
- Only events that have not yet started can be rescheduled. The start time is indicated in the **Start time** column (console) and the `NotBefore` field (AWS CLI). Events that are scheduled to start in the next 5 minutes can't 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.

Console

To reschedule an event

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 two for the updated event start time to be reflected in the console.

AWS CLI

To reschedule an event

Use the modify-instance-event-start-time command.

```
aws ec2 modify-instance-event-start-time \
    --instance-id i-1234567890abcdef0 \
    --instance-event-id instance-event-0d59937288b749b32 \
    --not-before 2020-03-25T10:00:00.000
```

PowerShell

To reschedule an event

Use the Edit-EC2InstanceEventStartTime cmdlet.

```
Edit-EC2InstanceEventStartTime `
    -InstanceId i-1234567890abcdef0 `
    -InstanceEventId instance-event-0d59937288b749b32 `
    -NotBefore 2020-03-25T10:00:00.000
```

Create custom event windows for scheduled events that affect your Amazon EC2 instances

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 automatic instance recovery and unplanned reboots.

Work with event windows

- Considerations
- Create an event window
- Associate a target with an event window
- Disassociate a target from an event window
- Modify an event window
- Delete an event window

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.

- You can stop and start an instance before the scheduled event. This migrates the instance to a new host and clears the event.

Create an event window

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.

Event windows recur weekly until you delete them.

Console

To create an event window

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. Choose **Create instance event window**.
5. For **Event window name**, enter a descriptive name for the event window.
6. 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.
7. (Optional) For **Target details**, associate one or more instances with the event window. Use instance IDs or instance tags to associate instances. Use host IDs to associate Dedicated

Hosts. When these targets are scheduled for maintenance, the event will occur during this event window.

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.

8. (Optional) For **Event window tags**, choose **Add tag**, and enter the key and value for the tag. Repeat for each tag.
9. Choose **Create event window**.

AWS CLI

To create an event window with a time range

Use the create-instance-event-window command.

```
aws ec2 create-instance-event-window \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

The following is example output.

```
{
    "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

Use the create-instance-event-window command.

```
aws ec2 create-instance-event-window \
    --cron-expression "* 21-23 * * 2,3" \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

To create an event window with a time range

Use the New-EC2InstanceEventWindow cmdlet.

```
$timeRange = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange.StartWeekDay = "monday"
$timeRange.EndWeekDay = "wednesday"
$timeRange.StartHour = 2
$timeRange.EndHour = 8
```

```
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec `
    -TimeRange @($timeRange)
```

The following is example output.

```
AssociationTarget     :
CronExpression        :
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {Amazon.EC2.Model.InstanceEventWindowTimeRange}
```

To create an event window with a cron expression

Use the New-EC2InstanceEventWindow cmdlet.

```
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec`
    -CronExpression "* 21-23 * * 2,3"
```

The following is example output.

```
AssociationTarget     :
CronExpression        : * 21-23 * * 2,3
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {}
```

Associate a target with an event window

After you create an event window, you can associate targets with the event window. You can associate only one type of target with an event window. You can specify instance IDs, Dedicated Host IDs, or instance tags.

Console

To associate targets with an event window

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Events**.
3. Select the event window to modify.
4. Choose **Actions**, **Modify instance event window**.

AWS CLI

To associate instance tags with an event window

Use the associate-instance-event-window command.

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [
                {
                    "Key": "k2",
                    "Value": "v2"
                },
                {
                    "Key": "k1",
```

```
                    "Value": "v1"
                }
            ],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

To associate instance IDs with an event window

Use the associate-instance-event-window command.

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

To associate a Dedicated Host with an event window

Use the associate-instance-event-window command.

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
```

```
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
            ]
        },
        "State": "creating"
    }
}
```

PowerShell

To associate instance tags with an event window

Use the Register-EC2InstanceEventWindow cmdlet.

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1,$tag2)
```

To associate instance IDs with an event window

Use the Register-EC2InstanceEventWindow cmdlet.

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

To associate a Dedicated Host with an event window

Use the Register-EC2InstanceEventWindow cmdlet.

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

Disassociate a target from an event window

Console

To disassociate targets with an event window

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Events**.
3. Select the event window to modify.
4. Choose **Actions**, **Modify instance event window**.

AWS CLI

To disassociate instance tags from an event window

Use the disassociate-instance-event-window command.

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
```

```
}
```

To disassociate instance IDs from an event window

Use the disassociate-instance-event-window command.

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

The following is example 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

Use the disassociate-instance-event-window command.

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
```

```
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

PowerShell

To disassociate instance tags from an event window

Use the Unregister-EC2InstanceEventWindow cmdlet.

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1, $tag2)
```

To disassociate instance IDs from an event window

Use the Unregister-EC2InstanceEventWindow cmdlet.

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

To disassociate a Dedicated Host from an event window

Use the Unregister-EC2InstanceEventWindow cmdlet.

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

Modify an event window

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.

You can modify either a time range or a cron expression when modifying the event window, but not both.

Console

To modify an event window

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 the time range of an event window

Use the modify-instance-event-window command.

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
```

```
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

To modify a set of time ranges for an event window

Use the modify-instance-event-window command.

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
 "wednesday", "EndHour": 8},
   {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
 "EndHour": 8}]'
```

The following is example 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-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

To modify the cron expression of an event window

Use the modify-instance-event-window command.

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

The following is example output.

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
```

```
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

To modify the time range of an event window

Use the Edit-EC2InstanceEventWindow cmdlet.

```
$timeRange1 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange1.StartWeekDay = "monday"
$timeRange1.EndWeekDay = "wednesday"
$timeRange1.StartHour = 2
$timeRange1.EndHour = 8
$timeRange2 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange2.StartWeekDay = "thursday"
$timeRange2.EndWeekDay = "friday"
$timeRange2.StartHour = 1
$timeRange2.EndHour = 6
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -TimeRange @($timeRange1, $timeRange2)
```

To modify the cron expression of an event window

Use the Edit-EC2InstanceEventWindow cmdlet.

```
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -CronExpression "* 21-23 * * 2,3"
```

Delete an event window

You can delete one event window at a time.

Console

To delete an event window

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

Use the delete-instance-event-window command and specify the event window to delete.

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890
```

To force-delete an event window

Use the `--force-delete` parameter if the event window is currently associated with targets.

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --force-delete
```

PowerShell

To delete an event window

Use the Remove-EC2InstanceEventWindow cmdlet.

```
Remove-EC2InstanceEventWindow `
```

```
    -InstanceEventWindowId iew-0abcdef1234567890
```

To force-delete an event window

Use the Remove-EC2InstanceEventWindow cmdlet.

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -ForceDelete $true
```

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 Manage detailed monitoring for your EC2 instances.

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.

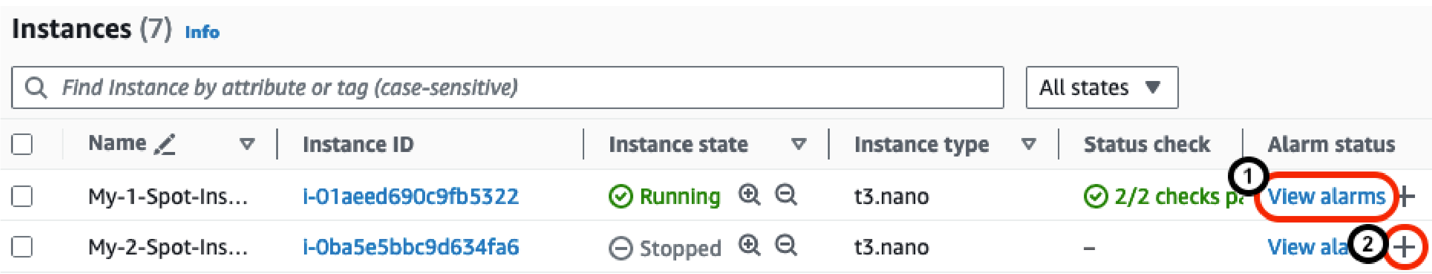
For Amazon CloudWatch billing and cost information, see CloudWatch billing and cost in the *Amazon CloudWatch User Guide*.

Contents

- [Create alarms that stop, terminate, reboot, or recover an instance]

Manage CloudWatch alarms for your EC2 instances in the Amazon EC2 console

From the **Instances** screen in the Amazon EC2 console, you can manage the Amazon CloudWatch alarms for your instances. In the **Instances** table, the **Alarm status** column provides two console controls: a control for viewing alarms, and another for creating or editing them. The following screenshot indicates these console controls, numbered **1** (**View alarms**) and **2** (a **+** sign for creating or editing an alarm).



View alarms from the Instances screen

You can view each instance's alarms from the **Instances** screen.

To view an instance's alarm from the Instances screen

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/].
2. In the navigation pane, choose **Instances**.
3. In the **Instances** table, for your chosen instance, choose **View alarms** (numbered **1** in the preceding screenshot).
4. In the **Alarm details for *i-1234567890abcdef0*** window, choose the alarm name to view the alarm in the CloudWatch console.

Create alarms from the Instances screen

You can create an alarm for each instance from the **Instances** screen.

To create an alarm for an instance from the Instances screen

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

2. In the navigation pane, choose **Instances**.
3. In the **Instances** table, for your chosen instance, choose the plus sign (numbered **2** in the preceding screenshot).
4. In the **Manage CloudWatch alarms** screen, create your alarm. For more information, see [Create a CloudWatch alarm for an instance](#).

Edit alarms from the Instances screen

You can edit the alarm for an instance from the **Instances** screen.

To edit an alarm for an instance from the Instances screen

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. In the **Instances** table, for your chosen instance, choose the plus sign (numbered **2** in the preceding screenshot).
4. In the **Manage CloudWatch alarms** screen, edit your alarm. For more information, see Edit or delete a CloudWatch alarm in the *Amazon CloudWatch User Guide*.

Manage detailed monitoring for your EC2 instances

Amazon CloudWatch provides two categories of monitoring: *basic monitoring* and *detailed monitoring*. By default, your instance is configured for basic monitoring. You can optionally enable detailed monitoring to help you more quickly identify and act on operational issues. You can enable or turn off detailed monitoring at launch or when the instance is running or stopped.

Enabling detailed monitoring on an instance does not affect the monitoring of its attached EBS volumes. For more information, see Amazon CloudWatch metrics for Amazon EBS.

The following table highlights the differences between basic monitoring and detailed monitoring for your instances.

Monitoring type	Description	Charges
Basic monitoring	Status check metrics are available in 1-minute periods. All other metrics are available in 5-minute periods.	No charge.

Monitoring type	Description	Charges
Detailed monitoring	All metrics, including status check metrics, are available in 1-minute periods. To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitorin g, you can also get aggregated data across groups of similar instances.	You are charged per metric that Amazon EC2 sends to CloudWatc h. You are not charged for data storage. For more information, see **Paid tier** and **Example 1 - EC2 Detailed Monitoring** on the Amazon CloudWatch pricing page.

Contents

Required permissions

To enable detailed monitoring for an instance, your user must have permission to use the MonitorInstances API action. To turn off detailed monitoring for an instance, your user must have permission to use the UnmonitorInstances API action.

Enable detailed monitoring at launch

Use the following procedures to enable detailed monitoring at launch. By default, your instance uses basic monitoring.

Console

To enable detailed monitoring when launching an instance

When launching an instance using the Amazon EC2 console, under **Advanced details**, select the **Detailed CloudWatch monitoring** checkbox.

AWS CLI

To enable detailed monitoring when launching an instance

Use the run-instances command with the `--monitoring` option.

```
--monitoring Enabled=true
```

PowerShell

To enable detailed monitoring when launching an instance

Use the New-EC2Instance cmdlet with the `-Monitoring` parameter.

```
-Monitoring $true
```

Manage detailed monitoring

Use the following procedures to manage detailed monitoring for a running or stopped instance.

Console

To manage detailed monitoring

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**, **Monitor and troubleshoot**, **Manage detailed monitoring**.
5. On the **Detailed monitoring** page, for **Detailed monitoring**, do one of the following:
   - Detailed monitoring – Select **Enable**.
   - Basic monitoring – Clear **Enable**.
6. Choose **Confirm**.

AWS CLI

To enable detailed monitoring

Use the following monitor-instances command.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

To disable detailed monitoring

Use the unmonitor-instances command.

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To enable detailed monitoring

Use the Start-EC2InstanceMonitoring cmdlet.

```
Start-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

To disable detailed monitoring

Use the Stop-EC2InstanceMonitoring cmdlet.

```
Stop-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

CloudWatch metrics that are available 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. Note that for the Minimum, Maximum, and Average statistics, the minimum granularity for the metrics that EC2 provides is 1 minute.

For information about how to view the available metrics using the AWS Management Console or the AWS CLI, see View available metrics in the *Amazon CloudWatch User Guide*.

For information about getting the statistics for these metrics, see Statistics for CloudWatch metrics for your instances.

Contents

- Status check metrics
- Traffic mirroring metrics
- Auto Scaling group metrics
- Amazon EC2 metric dimensions
- Amazon EC2 usage metrics

Instance metrics

The `AWS/EC2` namespace includes the following instance metrics.

Metric	Description	Unit	Meaningful statistics
`CPUUtilization`	The percentage of physical CPU time that Amazon EC2 uses to run the EC2 instance, which includes time spent to run both the user code and the Amazon EC2 code. At a very high level, `CPUUtilization` is the sum of guest `CPUUtilization` and hypervisor `CPUUtilization`. Tools in your operating system can show a different percentage than CloudWatch due to factors such as legacy device simulation, configuration of non-legacy devices, interrupt-heavy workloads, live migration, and live update.	Percent	• Average • Minimum • Maximum
`DiskReadOps`	Completed read operations from all instance store volumes available to the instance in a specified period of time. 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.	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
	If there are no instance store volumes, either the value is 0 or the metric is not reported.		
`DiskWriteOps`	Completed write operations to all instance store volumes available to the instance in a specified period of time. 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. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`DiskReadBytes`	Bytes read from all instance store volumes available to the instance. 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. The number reported is the number of bytes received during the period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `DiskReadBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`DiskWriteBytes`	Bytes written to all instance store volumes available to the instance. 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. The number reported is the number of bytes received during the period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `DiskWriteBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
MetadataNoToken	The number of times the Instance Metadata Service (IMDS) 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 (IMDSv1), which does not use a token. If all requests use token-backed sessions, i.e., Instance Metadata Service Version 2 (IMDSv2), the value is 0. For more information, see Transition to using Instance Metadata Service Version 2.	Nitro instances: None Xen instances : Count	• Sum • Percentiles
MetadataNoTokenRejected	The number of times an IMDSv1 call was attempted after IMDSv1 was disabled. If this metric appears, it indicates that an IMDSv1 call was attempted and rejected. You can either re-enable IMDSv1 or make sure all of your calls use IMDSv2. For more information, see Transition to using Instance Metadata Service Version 2.	Nitro instances: None Xen instances : Count	• Sum • Percentiles

Metric	Description	Unit	Meaningful statistics
`NetworkIn`	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 (5-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring and the statistic is Sum, divide it by 60.	Bytes	• Sum • Average • Minimum • Maximum
`NetworkOut`	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 (5-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring and the statistic is Sum, divide it by 60.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
NetworkPa cketsIn	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 monitorin g only (5-minute periods). To calculate the number of packets per second (PPS) your instance received for the 5 minutes, divide the Sum statistic value by 300.	Count	• Sum • Average • Minimum • Maximum
NetworkPa cketsOut	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. This metric is available for basic monitorin g only (5-minute periods). To calculate the number of packets per second (PPS) your instance sent for the 5 minutes, divide the Sum statistic value by 300.	Count	• Sum • Average • Minimum • Maximum

CPU credit metrics

The AWS/EC2 namespace includes the following CPU credit metrics for your burstable performance instances.

Metric	Description	Unit	Meaningful statistics
`CPUCreditUsage`	The number of CPU credits spent by the instance for CPU utilization. One CPU credit equals one vCPU running at 100% utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes). CPU credit metrics are available at a 5-minute frequency only. If you specify a period greater than five minutes, use the `Sum` statistic instead of the `Average` statistic.	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum
`CPUCreditBalance`	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. 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. The credits in the `CPUCreditBalance` are available for the instance to spend to burst beyond its baseline CPU utilization. When an instance is running, credits in the `CPUCreditBalance` do not expire. When	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
	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. CPU credit metrics are available at a 5-minute frequency only.		
`CPUSurplusCreditBalance`	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 5-minute frequency only.	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`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 5-minute frequency only.	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum

Dedicated Host metrics

The `AWS/EC2` namespace includes the following metrics for T3 Dedicated Hosts.

Metric	Description	Unit	Meaningful statistics
`DedicatedHostCPUUtilization`	The percentage of allocated compute capacity that is currently in use by the instances running on the Dedicated Host.	Percent	• Sum • Average • Minimum • Maximum

Amazon EBS metrics for Nitro-based instances

The AWS/EC2 namespace includes additional Amazon EBS metrics for volumes that are attached to Nitro-based instances that are not bare metal instances.

Metric	Description	Unit	Meaningful statistics
EBSReadOps	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 (5-minute) monitoring, you can divide this number by 300 to calculate the Read IOPS. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function DIFF_TIME to find the operations per second. For example, if you have graphed EBSReadOps in CloudWatch as m1, the metric math formula m1/(DIFF_TIME(m1)) returns the metric in operations/second. For more information about DIFF_TIME and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Count	• Sum • Average • Minimum • Maximum
EBSWriteOps	Completed write operations to all EBS volumes attached to the instance in a specified period of time. To calculate the average write I/O operation s per second (Write IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
	are using basic (5-minute) monitoring, you can divide this number by 300 to calculate the Write IOPS. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the operations per second. For example, if you have graphed `EBSWriteOps` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in operations/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.		
`EBSReadBytes`	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 (5-minute) monitoring, you can divide this number by 300 to find Read Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `EBSReadBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`EBSWriteBytes`	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 (5-minute) monitoring, you can divide this number by 300 to find Write Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `EBSWriteBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Bytes	• Sum • Average • Minimum • Maximum
`EBSIOBalance%`	Provides information about the percentage of I/O credits remaining in the burst bucket. This metric is available for basic monitoring only. This metric is available only for some `*.4xlarge` instance sizes and smaller that burst to their maximum performance for only 30 minutes at least once every 24 hours. The `Sum` statistic is not applicable to this metric.	Percent	• Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
EBSByteBalance%	Provides information about the percentage of throughput credits remaining in the burst bucket. This metric is available for basic monitoring only. This metric is available only for some *.4xlarge instance sizes and smaller that burst to their maximum performance for only 30 minutes at least once every 24 hours. The Sum statistic is not applicable to this metric.	Percent	• Minimum • Maximum

For information about the metrics provided for your EBS volumes, see Metrics for Amazon EBS volumes in the *Amazon EBS User Guide*. For information about the metrics provided for your EC2 Fleets and Spot Fleets, see Monitor your EC2 Fleet or Spot Fleet using CloudWatch.

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 Amazon EC2 instances.

The AWS/EC2 namespace includes the following status check metrics.

Metric	Description	Unit	Meaningful statistics
StatusCheckFailed	Reports whether the instance has passed all status checks in the last minute. This metric can be either 0 (passed) or 1 (failed).	Count	• Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
	By default, this metric is available at a 1-minute frequency at no charge.		
`StatusCheckFailed_Instance`	Reports whether the instance has passed the instance status check in the last minute. This metric can be either `0` (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge.	Count	• Average • Minimum • Maximum
`StatusCheckFailed_System`	Reports whether the instance has passed the system status check in the last minute. This metric can be either `0` (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge.	Count	• Average • Minimum • Maximum
`StatusCheckFailed_AttachedEBS`	Reports whether the instance has passed the attached EBS status check in the last minute. This metric can be either `0` (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge.	Count	• Average • Minimum • Maximum

The `AWS/EBS` namespace includes the following status check metric.

Metric	Description	Unit	Meaningful statistics
VolumeStalledIOCheck	**Note:** For Nitro instances only. Not published for volumes attached to Amazon ECS and AWS Fargate tasks. Reports whether a volume has passed or failed a *stalled IO check* in the last minute. This metric can be either `0` (passed) or `1` (failed).	None	• Average • Minimum • Maximum

Traffic mirroring metrics

The `AWS/EC2` namespace includes metrics for mirrored traffic. For more information, see Monitor mirrored traffic using Amazon CloudWatch in the *Amazon VPC Traffic Mirroring Guide*.

Auto Scaling group metrics

The `AWS/AutoScaling` namespace includes metrics for Auto Scaling groups. For more information, see Monitor CloudWatch metrics for your Auto Scaling groups and instances in the *Amazon EC2 Auto Scaling User Guide*.

Amazon EC2 metric dimensions

You can use the following dimensions to refine the metrics listed in the previous tables.

Dimension	Description
`AutoScalingGroupName`	This dimension filters the data you request for all instances in a specified capacity group. An *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.
`ImageId`	This dimension filters the data you request for all instances running this Amazon EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.

Dimension	Description
`InstanceId`	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
`InstanceType`	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

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 AWS usage metrics in the *Amazon CloudWatch User Guide*.

Amazon EC2 publishes the following metrics in the `AWS/Usage` namespace.

Metric	Description
`ResourceCount`	The number of the specified resources running in your account. The resources are defined by the dimensions associated with the metric. The most useful statistic for this metric is `MAXIMUM`, which represents the maximum number of resources used during the 1-minute period.

The following dimensions are used to refine the usage metrics that are published by Amazon EC2.

Dimension	Description
`Service`	The name of the AWS service containing the resource. For Amazon EC2 usage metrics, the value for this dimension is `EC2`.
`Type`	The type of entity that is being reported. Currently, the only valid value for Amazon EC2 usage metrics is `Resource`.
`Resource`	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.
`Class`	The class of resource being tracked. For Amazon EC2 usage metrics with `vCPU` as the value of the `Resource` dimension, the valid values are `Standard/OnDemand` , `F/OnDemand` , `G/OnDemand` , `Inf/OnDemand` , `P/OnDemand` , and `X/OnDemand` . The values for this dimension define the first letter of the instance types that are reported by the metric. For example, `Standard/OnDemand` returns information about all running instances with types that start with A, C, D, H, I, M, R, T, and Z, and `G/OnDemand` returns information about all running instances with types that start with G.

Install and configure the CloudWatch agent using the Amazon EC2 console to add additional metrics

Installing and configuring the CloudWatch agent using the Amazon EC2 console is in beta for Amazon EC2 and is subject to change.

By default, Amazon CloudWatch provides basic metrics, such as `CPUUtilization` and `NetworkIn`, for monitoring your Amazon EC2 instances. To collect additional metrics, you can install the CloudWatch agent on your EC2 instances, and then configure the agent to emit selected metrics. Instead of manually installing and configuring the CloudWatch agent on every EC2 instance, you can use the Amazon EC2 console to do this for you.

This topic explains how you can use the Amazon EC2 console to install the CloudWatch agent on your instances and configure the agent to emit selected metrics.

For the manual steps for this process, see Installing the CloudWatch agent using AWS Systems Manager in the *Amazon CloudWatch User Guide*. For more information about the CloudWatch agent, see Collect metrics, logs, and traces with the CloudWatch agent.

Topics

Prerequisites

To use Amazon EC2 to install and configure the CloudWatch agent, you must meet the user and instance prerequisites described in this section.

User prerequisites

To use this feature, your IAM console user or role must have the permissions required for using Amazon EC2 and the following IAM permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/EC2-Custom-Metrics-*"
        },
        {
```

```
                "Effect": "Allow",
                "Action": [
                    "ssm:SendCommand",
                    "ssm:ListCommandInvocations",
                    "ssm:DescribeInstanceInformation"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Effect": "Allow",
                "Action": [
                    "iam:GetInstanceProfile",
                    "iam:SimulatePrincipalPolicy"
                ],
                "Resource": "*"
            }
        ]
}
```

Instance prerequisites

- Instance state: `running`
- Supported operating system: Linux
- AWS Systems Manager Agent (SSM Agent): Installed. Two notes about SSM Agent:
  - SSM Agent is preinstalled on some Amazon Machine Images (AMIs) provided by AWS and trusted third-parties. For information about the supported AMIs and the instructions for installing SSM Agent, see Amazon Machine Images (AMIs) with SSM Agent preinstalled in the *AWS Systems Manager User Guide*.
  - If you experience issues with the SSM Agent, see Troubleshooting SSM Agent in the *AWS Systems Manager User Guide*.
- IAM permissions for the instance: The following AWS managed policies must be added to an IAM role that is attached to the instance:
  - AmazonSSMManagedInstanceCore – Enables an instance to use Systems Manager to install and configure the CloudWatch agent.
  - CloudWatchAgentServerPolicy – Enables an instance to use the CloudWatch agent to write data to CloudWatch.

For information about how to add IAM permissions to your instance, see [Use instance profiles](#) in the *IAM User Guide*.

How it works

Before you can use the Amazon EC2 console to install and configure the CloudWatch agent, you must make sure that your IAM user or role, and the instances on which you want to add metrics, meet certain prerequisites. Then, you can use the Amazon EC2 console to install and configure the CloudWatch agent on your selected instances.

First meet the prerequisites

- **You need the required IAM permissions –** Before you get started, make sure that your console user or role has the required IAM permissions to use this feature.
- **Instances –** To use the feature, your EC2 instances must be Linux instances, have the SSM Agent installed, have the required IAM permissions, and be running.

Then you can use the feature

1. **Select your instances –** In the Amazon EC2 console, you select the instances on which to install and configure the CloudWatch agent. You then start the process by choosing **Configure CloudWatch agent**.
2. **Validate SSM Agent –** Amazon EC2 checks that the SSM Agent is installed and started on each instance. Any instances that fail this check are excluded from the process. The SSM Agent is used for performing actions on the instance during this process.
3. **Validate IAM permissions –** Amazon EC2 checks that each instance has the required IAM permissions for this process. Any instances that fail this check are excluded from the process. The IAM permissions enable the CloudWatch agent to collect metrics from the instance and integrate with AWS Systems Manager to use the SSM Agent.
4. **Validate CloudWatch agent –** Amazon EC2 checks that the CloudWatch agent is installed and running on each instance. If any instances fail this check, Amazon EC2 offers to install and start the CloudWatch agent for you. The CloudWatch agent will collect the selected metrics on each instance once this process is completed.
5. **Select metric configuration –** You select the metrics for the CloudWatch agent to emit from your instances. Once selected, Amazon EC2 stores a configuration file in Parameter Store, where it remains until the process is completed. Amazon EC2 will delete the configuration file from

Parameter Store unless the process is interrupted. Note that if you don't select a metric, but you previously added it to your instance, it will be removed from your instance when this process is completed.

6. **Update CloudWatch agent configuration** – Amazon EC2 sends the metric configuration to the CloudWatch agent. This is the last step in the process. If it succeeds, your instances can emit data for the selected metrics and Amazon EC2 deletes the configuration file from Parameter Store.

Costs

Additional metrics that you add during this process are billed as custom metrics. For more information about CloudWatch metrics pricing, see Amazon CloudWatch Pricing.

Install and configure the CloudWatch agent

You can use the Amazon EC2 console to install and configure the CloudWatch agent to add additional metrics.

> **Note**
>
> Every time you perform this procedure, you overwrite the existing CloudWatch agent configuration. If you don't select a metric that was selected previously, it will be removed from the instance.

To install and configure the CloudWatch agent 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 instances on which to install and configure the CloudWatch agent.
4. Choose **Actions**, **Monitor and troubleshoot**, **Configure CloudWatch agent**.

   > **Tip**
   >
   > This feature is not available in all AWS Regions. If **Configure CloudWatch agent** is not available, try another Region.

5. For each step in the process, read the console text, and then choose **Next**.
6. To complete the process, in the final step, choose **Complete**.

Statistics for CloudWatch metrics for your instances

You can get statistics for the CloudWatch metrics for your instances. *Statistics* are metric data aggregations over specified periods of time. CloudWatch provides statistics based on the metric data points provided by your custom data or provided by other services in AWS to CloudWatch. Aggregations are made using the namespace, metric name, dimensions, and the data point unit of measure, within the time period you specify. The following table describes the available statistics.

Statistic	Description
`Minimum`	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
`Maximum`	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
`Sum`	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
`Average`	The value of `Sum` / `SampleCount` during the specified period. By comparing this statistic with the `Minimum` and `Maximum`, you can determine the full scope of a metric and how close the average use is to the `Minimum` and `Maximum`. This comparison helps you to know when to increase or decrease your resources as needed.
`SampleCount`	The count (number) of data points used for the statistical calculation.
`pNN.NN`	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45).

Contents

Get statistics for a specific instance

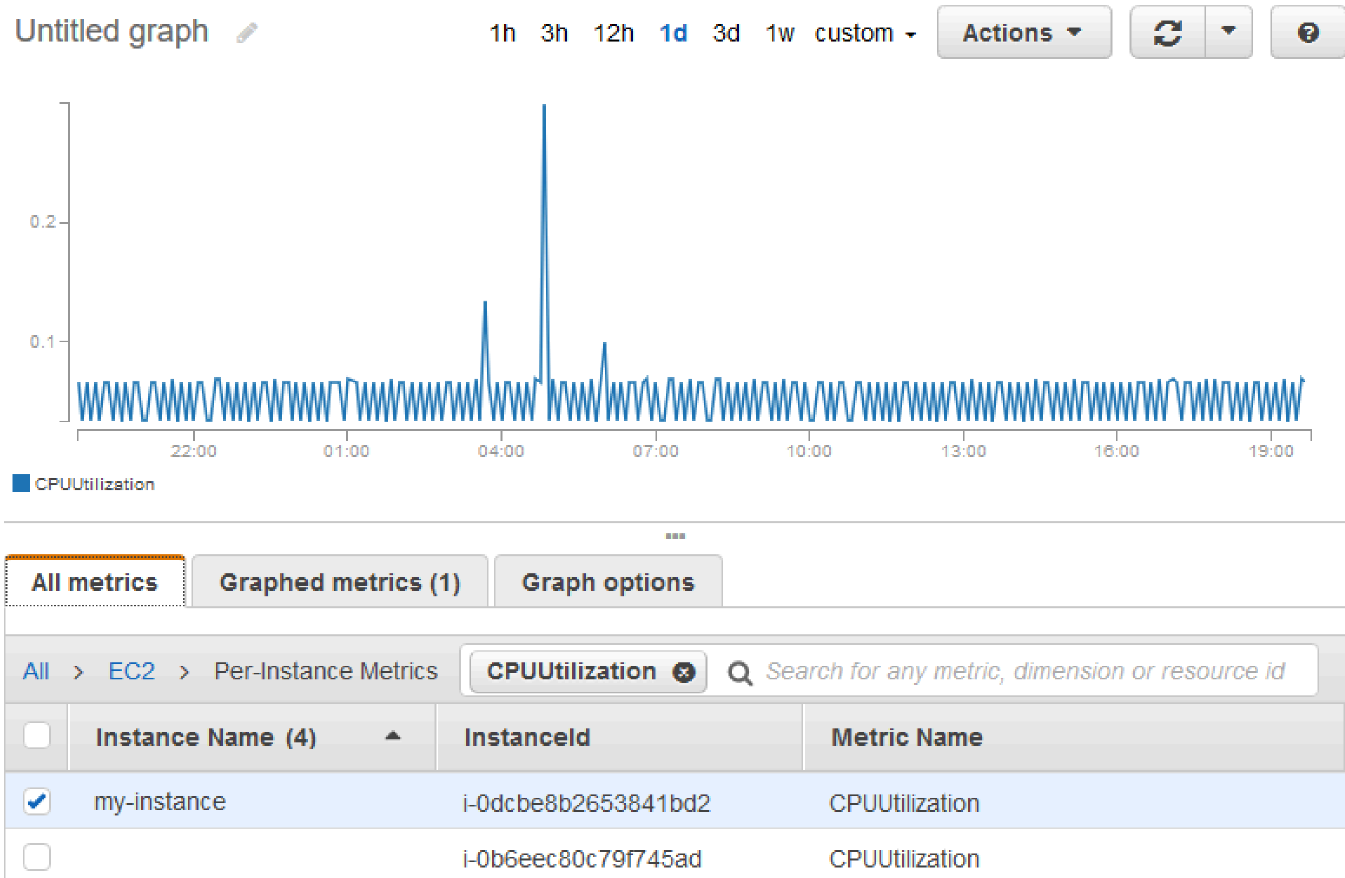
You can use the AWS Management Console or the AWS CLI to get statistics for a specific instance. The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

Requirements

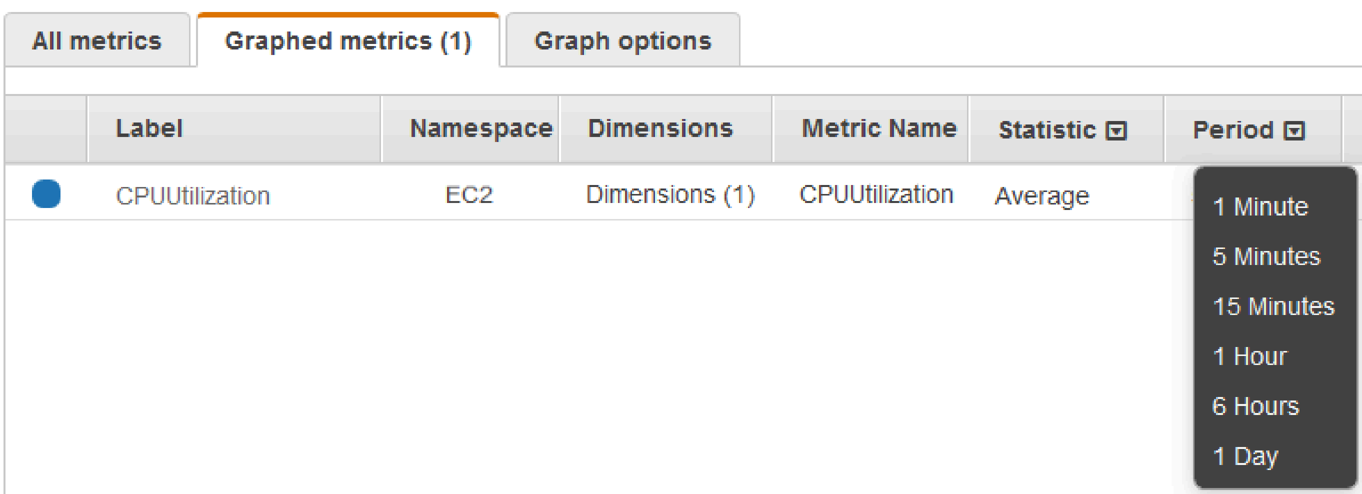
- You must have the ID of the instance. You can get the instance ID using the AWS Management Console or the describe-instances command.
- By default, basic monitoring is enabled, but you can enable detailed monitoring. For more information, see Manage detailed monitoring for your EC2 instances.

To display the CPU utilization for a specific instance (console)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.
4. Choose the **Per-Instance Metrics** dimension.
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**.



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.



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 2022-10-18T23:18:00 --end-time 2022-10-19T23:18:00
```

The following is example output. Each value represents the maximum CPU utilization percentage for a single EC2 instance.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T12:18:00Z",
            "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 (at an additional charge), which provides data in 1-minute periods.

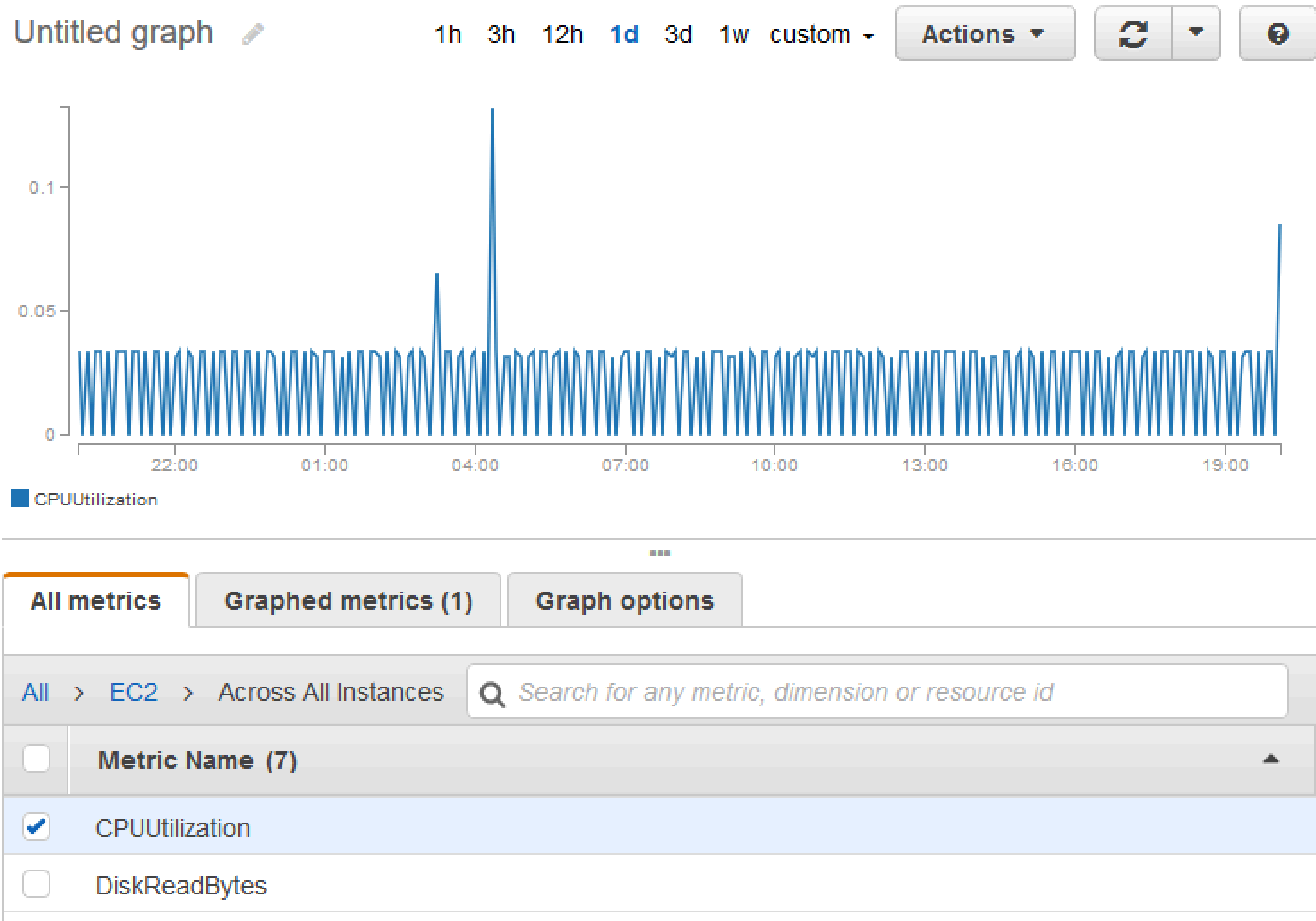
Note that Amazon CloudWatch cannot aggregate data across AWS Regions. Metrics are completely separate between Regions.

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)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **Across All Instances**.
4. Choose the row that contains **CPUUtilization**, which displays a graph for the metric for all your EC2 instances. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get average CPU utilization across your instances (AWS CLI)

Use the get-metric-statistics command as follows to get the average of the **CPUUtilization** metric across your instances.

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/EC2 \
    --metric-name CPUUtilization \
    --period 3600  --statistics "Average" "SampleCount" \
    --start-time 2022-10-11T23:18:00 \
    --end-time 2022-10-12T23:18:00
```

The following is example output:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2022-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

Aggregate statistics by Auto Scaling group

You can aggregate statistics for the EC2 instances in an Auto Scaling group. Note that Amazon CloudWatch cannot aggregate data across 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 1-minute periods for a 24-hour interval across all EC2 instances in the specified Auto Scaling group.

To display DiskWriteBytes for the instances in an Auto Scaling group (console)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Auto Scaling Group**.
4. Choose the row for the **DiskWriteBytes** metric and the specific Auto Scaling group, which displays a graph for the metric for the instances in the Auto Scaling group. To name the graph,

choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.

5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To display DiskWriteBytes for the instances in an Auto Scaling group (AWS CLI)

Use the get-metric-statistics command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
 --period 360 \
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --
start-time 2022-10-16T23:18:00 --end-time 2022-10-18T23:18:00
```

The following is example output:


```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2022-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2022-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```


Aggregate statistics by AMI

You can aggregate statistics by AMI 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 (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)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Image (AMI) Id**.
4. Choose the row for the **CPUUtilization** metric and the specific AMI, which displays a graph for the metric for the specified AMI. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get the average CPU utilization for an image ID (AWS CLI)

Use the get-metric-statistics command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --period 3600 \
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-
time 2022-10-10T00:00:00 --end-time 2022-10-11T00:00:00
```

The following is example output. Each value represents an average CPU utilization percentage for the EC2 instances running the specified AMI.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
```

```
            "Timestamp": "2022-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


View the monitoring graphs 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 CloudWatch metrics that are available for your instances.

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 Graphing metrics in the *Amazon CloudWatch User Guide*.

Create a CloudWatch alarm for an instance

You can create a CloudWatch alarm that monitors CloudWatch metrics for one of your instances. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm using the Amazon EC2 console, or using the more advanced options provided by the CloudWatch console.

To create an alarm using the CloudWatch console

For examples, see Creating Amazon CloudWatch Alarms in the *Amazon CloudWatch User Guide*.

To create an alarm using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance 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 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 specify an action to take when the alarm is triggered. Choose an action from the list.
7. For **Alarm thresholds**, select the metric and criteria for the alarm. For example, to create an alarm that is triggered when CPU utilization reaches 80% for a 5 minute period, do the following:

    a. Keep the default setting for **Group samples by** (**Average**) and **Type of data to sample** (**CPU utilization**).
    b. Choose **>=** for **Alarm when** and enter `0.80` for **Percent**.
    c. Enter **1** for **Consecutive period** and select **5 minutes** for **Period**.
8. (Optional) For **Sample metric data**, choose **Add to dashboard**.

9. Choose **Create**.

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 Edit or delete 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.

> **Note**
>
> For Amazon CloudWatch alarms billing and pricing information, see CloudWatch billing and cost in the *Amazon CloudWatch User Guide*.

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 AWS 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 ID and root volume. However, you cannot start a terminated instance. Instead, you must launch a new instance. When an instance is stopped or terminated, data on instance store volumes is lost.

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.

> **Important**
>
> Status check alarms can temporarily enter the `INSUFFICIENT_DATA` state if there are missing metric data points. Although rare, this can happen when there is an interruption in the metric reporting systems, even when an instance is healthy. We recommend that you treat the `INSUFFICIENT_DATA` state as missing data instead of an alarm breach, especially when configuring the alarm to stop, terminate, reboot, or recover an 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

You must have the `iam:CreateServiceLinkedRole` to create or modify an alarm that performs EC2 alarm actions. A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Create a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Contents

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. Data on instance store volumes is lost when the instance is terminated or stopped.

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**.

   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**.

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. Data on instance store volumes is lost when the instance is terminated. For information about enabling and disabling termination protection for an instance, see [Change instance termination protection](#).

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**.

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. Data on instance store volumes is retained when the instance is rebooted. The instance store volumes must be re-mounted into the filesystem after a reboot. For more information, see Reboot your Amazon EC2 instance.

> **Important**
>
> To avoid a race condition between the reboot and recover actions, avoid setting the same number of evaluation periods for a reboot alarm and a recover alarm. We recommend that you set reboot alarms to three evaluation periods of one minute each. For more information, see Evaluating an alarm in the *Amazon CloudWatch User Guide*.

To create an alarm to reboot an instance (Amazon EC2 console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance 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 **1 Minute**. If **1 Minute is disabled**, you must enable detailed monitoring, or you can choose **5 Minutes** instead.

   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**.

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.

> **Note**
>
> 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 that meet certain characteristics. For more information, see [Automatic instance recovery](#).

If your instance has a public IP address, it retains the public IP address after recovery.

> **Important**
>
> To avoid a race condition between the reboot and recover actions, avoid setting the same number of evaluation periods for a reboot alarm and a recover alarm. We recommend that you set recover alarms to two evaluation periods of one minute each. For more information, see Evaluating an alarm in the *Amazon CloudWatch User Guide*.

To create an alarm to recover an instance (Amazon EC2 console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance 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 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**.

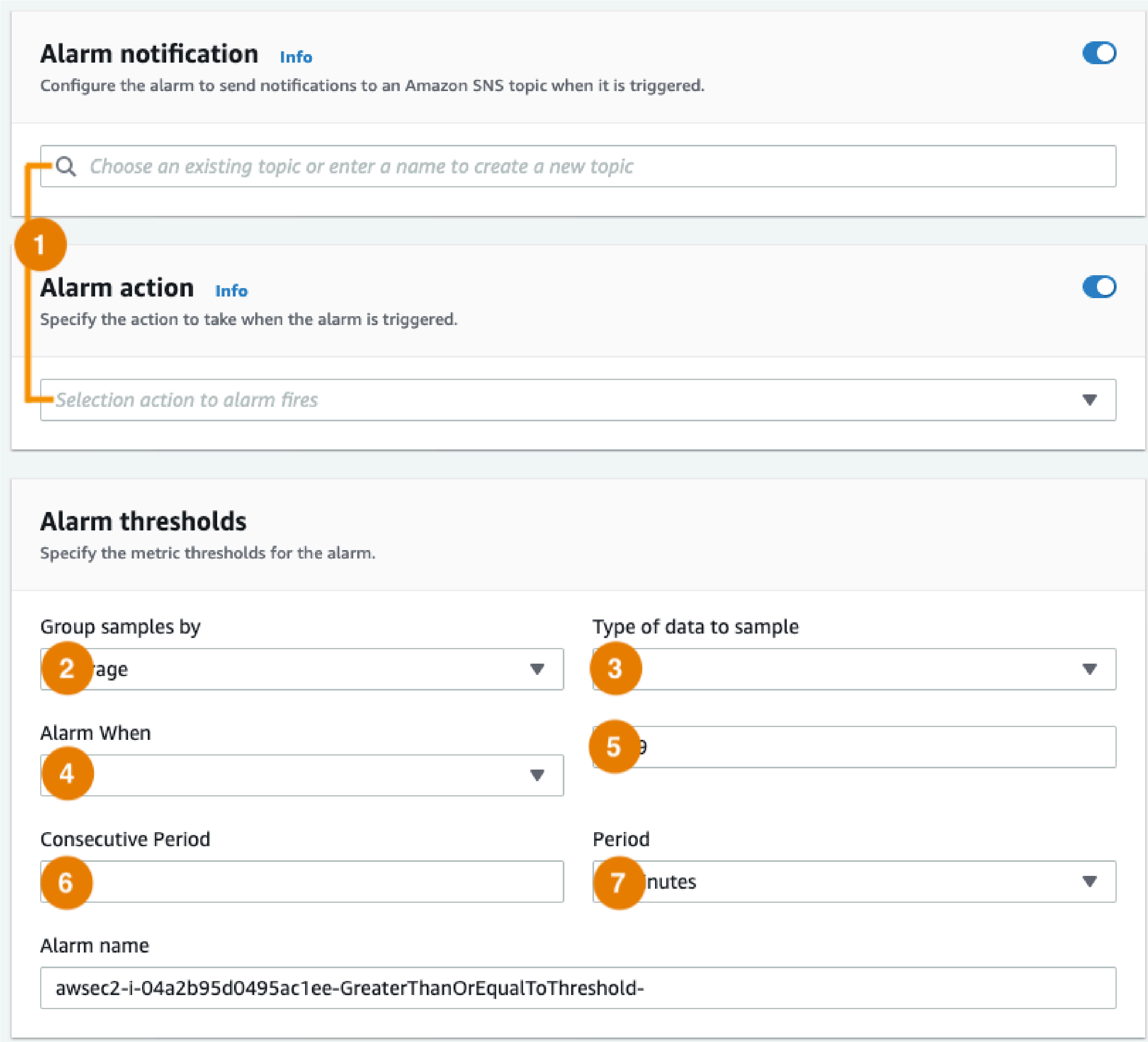
e. For **Consecutive period** and **Period**, specify the evaluation period for the alarm. In this example, enter **2** consecutive periods of **1 Minute**. If **1 Minute is disabled**, you must enable detailed monitoring, or you can choose **5 Minutes** instead.

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**.

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.



Scenario 1: Stop idle development and test instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
1	Stop

Setting	Value
2	Maximum
3	CPU Utilization
4	<=
5	10%
6	1
7	1 Hour

Scenario 2: Stop idle instances

Create an alarm that stops an instance and sends an email when the instance has been idle for 24 hours.

Setting	Value
1	Stop and email
2	Average
3	CPU Utilization
4	<=
5	5%
6	24
7	1 Hour

Scenario 3: Send email about web servers with unusually high traffic

Create an alarm that sends email when an instance exceeds 10 GB of outbound network traffic per day.

Setting	Value
1	Email
2	Sum
3	Network Out
4	>
5	10 GB
6	24
7	1 Hour

Scenario 4: Stop web servers with unusually high traffic

Create an alarm that stops an instance and send a text message (SMS) if outbound traffic exceeds 1 GB per hour.

Setting	Value
1	Stop and send SMS
2	Sum
3	Network Out
4	>
5	1 GB
6	1
7	1 Hour

Scenario 5: Stop an impaired instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

Setting	Value
1	Stop
2	Average
3	Status Check Failed: System
4	-
5	-
6	1
7	15 Minutes

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.

Setting	Value
1	Terminate
2	Maximum
3	Network Out
4	<=
5	100,000 bytes
6	1

Setting	Value
7	5 Minutes

Automate Amazon EC2 using EventBridge

You can use Amazon EventBridge 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 create rules to indicate which events you're interested in, and the actions to take when an event matches a rule. The actions that can be automatically triggered include the following:

- Invoke an AWS Lambda function
- Invoke Amazon EC2 Run Command
- Relay the event to Amazon Kinesis Data Streams
- Activate an AWS Step Functions state machine
- Notify an Amazon SNS topic
- Notify an Amazon SQS queue

The following are examples of how you can use EventBridge with Amazon EC2:

- Activate a Lambda function whenever an instance enters the running state.
- Notify an Amazon SNS topic when an Amazon EBS volume is created or modified.
- Send 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.

Amazon EC2 event types

Amazon EC2 supports the following event types:

- EC2 AMI State Change
- EC2 Fast Launch State-change Notification

- EC2 Fleet Error
- EC2 Fleet Information
- EC2 Fleet Instance Change
- EC2 Fleet Spot Instance Request Change
- EC2 Fleet State Change
- EC2 Instance Rebalance Recommendation
- EC2 Instance State-change Notification
- EC2 Spot Fleet Error
- EC2 Spot Fleet Information
- EC2 Spot Fleet Instance Change
- EC2 Spot Fleet Spot Instance Request Change
- EC2 Spot Fleet State Change
- EC2 Spot Instance Interruption Warning
- EC2 Spot Instance Request Fulfillment
- EC2 ODCR Underutilization Notification

For information about the event types supported by Amazon EBS, see Amazon EventBridge for Amazon EBS.

Log Amazon EC2 API calls using AWS CloudTrail

The Amazon EC2 API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all Amazon EC2 API calls as events. The calls captured include calls made by the console. Using the information collected by CloudTrail, you can determine the request that was made to the Amazon EC2 API, the IP address from which the request was made, and when it was made.

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 user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.
- Whether the request was made with temporary security credentials for a role or federated user.

- Whether the request was made by another AWS service.

CloudTrail is active in your AWS account when you create the account and you automatically have access to the CloudTrail **Event history**. The CloudTrail **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded management events in an AWS Region. For more information, see Working with CloudTrail Event history in the *AWS CloudTrail User Guide*. There are no CloudTrail charges for viewing the **Event history**.

For an ongoing record of events in your AWS account past 90 days, create a trail or a CloudTrail Lake event data store.

CloudTrail trails

A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. All trails created using the AWS Management Console are multi-Region. You can create a single-Region or a multi-Region trail by using the AWS CLI. Creating a multi-Region trail is recommended because you capture activity in all AWS Regions in your account. If you create a single-Region trail, you can view only the events logged in the trail's AWS Region. For more information about trails, see Creating a trail for your AWS account and Creating a trail for an organization in the *AWS CloudTrail User Guide*.

You can deliver one copy of your ongoing management events to your Amazon S3 bucket at no charge from CloudTrail by creating a trail, however, there are Amazon S3 storage charges. For more information about CloudTrail pricing, see AWS CloudTrail Pricing. For information about Amazon S3 pricing, see Amazon S3 Pricing.

CloudTrail Lake event data stores

CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to Apache ORC format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying advanced event selectors. The selectors that you apply to an event data store control which events persist and are available for you to query. For more information about CloudTrail Lake, see Working with AWS CloudTrail Lake in the *AWS CloudTrail User Guide*.

CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the pricing option you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see AWS CloudTrail Pricing.

Amazon EC2 API management events in CloudTrail

Management events provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, CloudTrail logs management events.

All Amazon EC2 API actions are logged as management events. For a list of the API actions that are logged to CloudTrail, see the Amazon EC2 API Reference. For example, calls to the RunInstances, DescribeInstances, and StopInstances actions are logged as management events.

Amazon EC2 API event examples

An event represents a single request from any source and includes information about the requested API operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so events don't appear in any specific order.

The following log file record shows that a user terminated an instance.


```
{
   "Records":[
      {
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"user"
         },
         "eventTime":"2016-05-20T08:27:45Z",
         "eventSource":"ec2.amazonaws.com",
         "eventName":"TerminateInstances",
         "awsRegion":"us-west-2",
         "sourceIPAddress":"198.51.100.1",
         "userAgent":"aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
         "requestParameters":{
```

```
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d"
               }]
            }
         },
         "responseElements":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d",
                  "currentState":{
                     "code":32,
                     "name":"shutting-down"
                  },
                  "previousState":{
                     "code":16,
                     "name":"running"
                  }
               }]
            }
         },
         "requestID":"be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
         "eventID":"6e12345-2a4e-417c-aa78-7594fEXAMPLE",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```


For information about CloudTrail record contents, see CloudTrail record contents in the *AWS CloudTrail User Guide*.

Audit connections made using EC2 Instance Connect

You can use AWS CloudTrail to audit the users that connect to your instances using EC2 Instance Connect.

To audit SSH activity via EC2 Instance Connect using the AWS CloudTrail console

1. Open the CloudTrail console at https://console.aws.amazon.com/cloudtrail/.
2. Verify that you are in the correct Region.
3. In the navigation pane, choose **Event history**.

4. For **Filter**, choose **Event source**, **ec2-instance-connect.amazonaws.com**.
5. (Optional) For **Time range**, select a time range.
6. Choose the **Refresh events** icon.
7. The page displays the events that correspond to the SendSSHPublicKey API calls. Expand an event using the arrow to view additional details, such as the user name and AWS access key that was used to make the SSH connection, and the source IP address.
8. To display the full event information in JSON format, choose **View event**. The **requestParameters** field contains the destination instance ID, OS username, and public key that were used to make the SSH connection.


```
{
    "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": {
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNO01234567890EXAMPLE"
        }
     },
    "responseElements": null,
    "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",
    "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "0987654321"
}
```

If you have configured your AWS account to collect CloudTrail events in an S3 bucket, you can download and audit the information programmatically. For more information, see Getting and viewing your CloudTrail log files in the *AWS CloudTrail User Guide*.

Monitor .NET and SQL Server applications using 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 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.

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*.

Track your Free Tier usage for Amazon EC2

When you create your AWS account, you can get started with Amazon EC2 for free using the AWS Free Tier. Your Free Tier benefits are different depending on whether you created your account before July 15, 2025, or on or after July 15, 2025. For more information, see Explore AWS services with AWS Free Tier in the *AWS Billing User Guide*.

Free Tier benefits before and after July 15, 2025

The following table compares your Free Tier benefits based on your AWS account creation date:

Free Tier benefit	Account created before July 15, 2025	Account created on or after July 15, 2025
Instance types marked **Free tier eligible**	`t2.micro`, `t3.micro`	`t3.micro`, `t3.small`, `t4g.micro` , `t4g.small` , `c7i-flex.large` , `m7i-flex.large`
AMIs marked **Free tier eligible**	Check for AMIs marked **Free tier eligible**.	Check for AMIs marked **Free tier eligible**.
Usage limit	Restricted to usage limits, after which you are billed pay-as-you-go rates.	Receive USD $100 sign-up credit and earn up to $100 in additional credits.
Free Tier duration	Your Free Tier lasts for 12 months from the date you create your account. During	Your Free Tier lasts for 6 months from the date you created your account, or when

Free Tier benefit	Account created before July 15, 2025	Account created on or after July 15, 2025
	this time, if you exceed your Free Tier usage limits, you are billed pay-as-you-go rates.	your credits are used up, whichever happens first. You can't exceed your Free Tier limits.

To list the instance types that are free tier eligible

Use the describe-instance-types command with the `free-tier-eligible` filter.

```
aws ec2 describe-instance-types \
    --filters Name=free-tier-eligible,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

To list the AMIs that are free tier eligible

Use the describe-images command with the `free-tier-eligible` filter.

```
aws ec2 describe-images \
    --filters Name=free-tier-eligible,Values=true \
    --query "Images[*].[ImageId]" \
    --output text | sort
```

Track Free Tier usage for accounts created before July 15, 2025

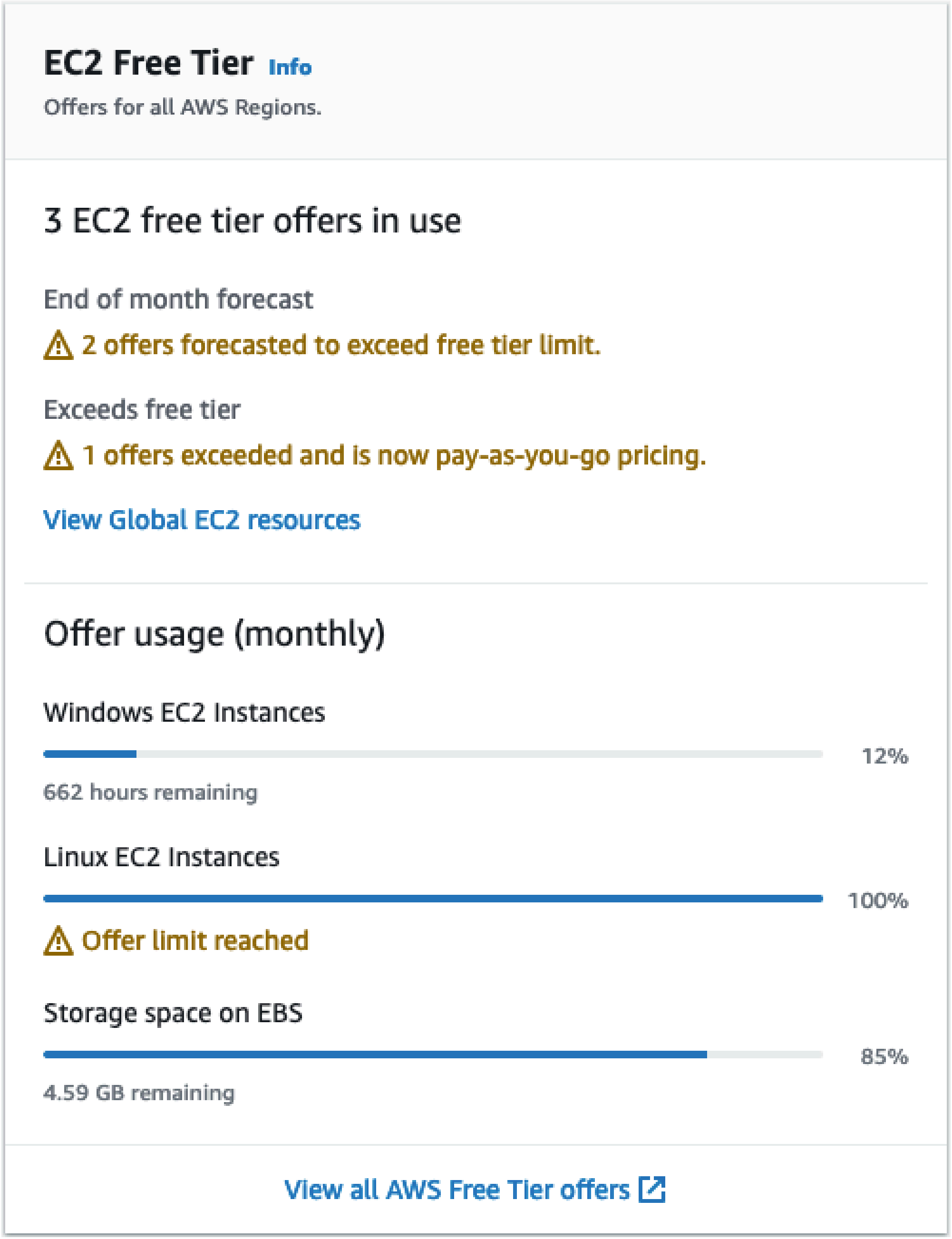
> **Note**
>
> ***This section only applies to Free Tier users who created AWS accounts before July 15, 2025. If you created your account on or after July 15, 2025, see Tracking your AWS Free Tier usage in the AWS Billing User Guide.***

If you created your account before July 15, 2025, you can use Amazon EC2 without incurring charges if you've been an AWS customer for less than 12 months and you stay within the AWS Free Tier usage limits. It's important to track your Free Tier usage to avoid billing surprises. If you exceed the Free Tier limits, you'll incur standard pay-as-go charges. For more information, see AWS Free Tier.

> **Note**
>
> If you've been an AWS customer for more than 12 months, you're no longer eligible for Free Tier usage and you won't see the **EC2 Free Tier** box that is described in the following procedure.

To track your Free Tier usage

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **EC2 Dashboard**.
3. Find the **EC2 Free Tier** box (at top right).



4. In the **EC2 Free Tier** box, check your Free Tier usage, as follows:

   - Under **EC2 Free Tier offers in use**, take note of the warnings:
     - **End of month forecast** – This warns that you will incur charges this month if you continue with your current usage pattern.
     - **Exceeds Free Tier** – This warns that you've exceeded your Free Tier limits and you're already incurring charges.

- Under **Offer usage (monthly)**, take note of your usage of Linux instances, Windows instances, and EBS storage. The percentage indicates how much of your Free Tier limits you've used this month. If you're at 100%, you will incur charges for further use.

> **Note**
>
> This information appears only after you've created an instance. However, usage information is not updated in real time; it's updated three times a day.

5. To avoid incurring further charges, delete any resources that are either incurring charges now, or will incur charges if you exceed your Free Tier limit usage.

   - For the instructions to delete your instance, see Terminate Amazon EC2 instances.
   - To check if you have resources in other Regions that might be incurring charges, in the **EC2 Free Tier** box, choose **View Global EC2 resources** to open the **EC2 Global View**. For more information, see View resources across Regions using AWS Global View.

6. To view your resource usage for all AWS services under the AWS Free Tier, at the bottom of the **EC2 Free Tier** box, choose **View all AWS Free Tier offers**. For more information, see Trying services using AWS Free Tier in the *AWS Billing User Guide*.

Understand codes for Amazon EC2 in billing and usage reports

When you use Amazon EC2, we include related codes in your AWS billing and usage reports. Reviewing these codes helps you understand your costs and usage patterns for Amazon EC2. Tracking and managing your expenses is essential for optimizing your costs.

The following tables describe the codes for Amazon EC2 that appear in your billing and usage reports. For a list of the Region codes used in the billing and usage reports, see AWS Region billing codes.

Billing codes for:

- Instances
- Bare metal instances
- Dedicated Hosts
- Dedicated Instances
- EBS optimization

- Capacity Reservations

Related resources

- the section called “Billing and purchasing options”
- the section called “Understand AMI billing”
- Amazon EC2 pricing

Instances

Code	Description	Units
region-BoxUsage :*instance-type*	The running time for On-Demand Instances.	Hours
region-HostBoxUsage :*instance-type*	The running time for instances on Dedicated Hosts.	Hours
region-SpotUsage :*instance-type*	The running time for Spot Instances.	Hours

Bare metal instances

Code	Description	Units
region-BoxUsage :*instance-family* .metal	The running time for bare metal On-Demand Instances.	Hours
region-HostBoxUsage :*instance-family* .metal	The running time for bare metal instances on Dedicated Hosts.	Hours

Code	Description	Units
region-SpotUsage :*instance-family* .metal	The running time for bare metal Spot Instances.	Hours

Dedicated Hosts

Code	Description	Units
region-HostUsage :*host-type*	The time that Dedicated Hosts are provisioned.	Hours
region-ReservedHostUsage :*host-type*	The time that Dedicated Host Reservations are applied.	Hours

Dedicated Instances

Code	Description	Units
region-DedicatedUsage :*instance-type*	The running time for Dedicated Instances.	Hours + per-Region fee

EBS optimization

Code	Description	Units
region-EBSOptimized :*instance-type*	The time that EBS optimization is enabled.	Hours

Capacity Reservations

Code	Description	Units
region-Reservation :*instance-type*	The reserved instance time for Capacity Reservations.	Hours
region-UnusedBox :*instance-type*	The unused reserved instance time for Capacity Reservations.	Hours
region-DedicatedRes :*instance-type*	The reserved instance time for Dedicated Capacity Reservations.	Hours
region-UnusedDed :*instance-type*	The unused reserved instance time for Dedicated Capacity Reservations.	Hours

Troubleshoot issues with Amazon EC2 instances

The following procedures and tips can help you troubleshoot issues with your Amazon EC2 instances.

Issues

Troubleshoot Amazon EC2 instance launch issues

The following are troubleshooting tips to help you solve issues when launching an Amazon EC2 instance.

Launch Issues

- The requested configuration is currently not supported. Please check the documentation for supported configurations.
- Instance terminates immediately
- Insufficient permissions
- High CPU usage shortly after Windows starts (Windows instances only)

Invalid device name

Description

You get the `Invalid device name` *`device_name`* error when you try to launch a new instance.

Cause

If you get this error when you try to launch an instance, the device name specified for one or more volumes in the request has an invalid device name. Possible causes include:

- The device name might be in use by the selected AMI.
- The device name might be reserved for root volumes.
- The device name might be used for another volume in the request.
- The device name might not be valid for the operating system.

Solution

To resolve the issue:

- Ensure that the device name is not used in the AMI that you selected. Run the following command to view the device names used by the AMI.

```
aws ec2 describe-images --image-id ami-0abcdef1234567890 --query
 'Images[*].BlockDeviceMappings[].DeviceName'
```

- Ensure that you are not using a device name that is reserved for root volumes. For more information, see Available device names.
- Ensure that each volume specified in your request has a unique device name.
- Ensure that the device names that you specified are in the correct format. For more information, see Available device names.

Instance limit exceeded

Description

You get the `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get an `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance, you have reached the limit on the number of instances that you can launch in a Region. When you create your AWS account, we set default limits on the number of instances you can run on a per-Region basis.

Solution

You can request an instance limit increase on a per-region basis. For more information, see Amazon EC2 service quotas.

Insufficient instance capacity

Description

You get the `InsufficientInstanceCapacity` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get 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 Amazon EC2 instance type changes.
- If you are launching instances into a cluster placement group, you can get an insufficient capacity error.

The requested configuration is currently not supported. Please check the documentation for supported configurations.

Description

You get the `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.

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 Amazon EBS volume limits for Amazon EC2 instances.
- 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 CLI

1. Use the describe-instances command and specify the instance ID.

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

2. Review the JSON response returned by the command and note the values in the `StateReason` response element.

   The following code block shows an example of a `StateReason` response element.

```
"StateReason": {
  "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",
  "Code": "Server.InternalError"
},
```

To 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*.

Insufficient permissions

Description

You get the `"errorMessage": "You are not authorized to perform this operation."` error when you try to launch a new instance, and the launch fails.

Cause

If you get this error when you try to launch an instance, you don't have the required IAM permissions to launch the instance.

Possible missing permissions include:

- `ec2:RunInstances`
- `iam:PassRole`

Other permissions might also be missing. For the list of permissions required to launch an instance, see the example IAM policies under Example: Use the EC2 launch instance wizard and Launch instances (RunInstances).

Solution

To resolve the issue:

- If you are making requests as an IAM user, verify that you have the following permissions:
  - `ec2:RunInstances` with a wildcard resource ("*")
  - `iam:PassRole` with the resource matching the role ARN (for example, `arn:aws:iam::999999999999:role/ExampleRoleName`)
- If you don't have the preceding permissions, edit the IAM policy associated with the IAM role or user to add the missing required permissions.

If your issue is not resolved and you continue receiving a launch failure error, you can decode the authorization failure message included in the error. The decoded message includes the permissions that are missing from the IAM policy. For more information, see How do I decode an authorization failure message after I receive an "UnauthorizedOperation" error during an EC2 instance launch?

High CPU usage shortly after Windows starts (Windows instances only)

Note

This troubleshooting tip is for Windows instances only.

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 Run commands on your Windows instance at launch.

Troubleshoot Amazon EC2 instance stop issues

If your Amazon EBS-backed instance appears stuck in the `stopping` state, the issue might be with the underlying host computer.

To resolve the issue, follow these steps:

1. **Force stop the instance**

   Use the Amazon EC2 console or the AWS CLI to force stop the instance. For the steps, see Force stop an instance.

   The instance will first attempt a graceful shutdown, which includes flushing file system caches and metadata (although you can optionally bypass the graceful shutdown). If the graceful shutdown fails to complete within the timeout period, the instance shuts down forcibly without flushing the file system caches and metadata.

2. **After force stop**

   Perform file system check and repair procedures.

   > **Important**
   >
   > Performing these procedures is crucial because a forced stop prevents flushing of file system caches and metadata.

3. **If force stop fails**

   If, after 10 minutes, the instance has not stopped, do the following:

   a. Post a request for help on AWS re:Post. To help expedite a resolution, include the instance ID, and describe the steps that you've already taken.

   b. Alternatively, if you have a support plan, create a technical support case in the Support Center.

   c. While waiting for assistance, you can create a replacement instance if needed. For the steps, see (Optional) Create a replacement instance.

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.

Contents

Force stop an instance

You can force an instance to stop. If, after 10 minutes, the instance has not stopped, post a request for help on AWS re:Post. 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.

> **Note**
>
> Using the console, you can force an instance to stop while the instance is in the `stopping` state only. Using the AWS CLI, you can force an instance to stop while the instance is in the `pending`, `running`, or `stopping` state.

Console

To force stop an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance state**, **Force stop instance**.

   Note that **Force stop instance** is only available in the console if your instance is in the `stopping` state. If your instance is in another state (except `shutting-down` and `terminated`) you can use the AWS CLI to force stop your instance.
4. (Optional) To bypass the graceful OS shutdown during the force stop, select the **Skip OS shutdown** checkbox.
5. Choose **Force stop**.

AWS CLI

To force stop an instance

Use the stop-instances command with the `--force` option.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

To bypass the graceful OS shutdown during force stop, include the `--skip-os-shutdown` option.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

To force stop an instance

Use the Stop-EC2Instance cmdlet and set `-Enforce` to `true`.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

To bypass the graceful OS shutdown during force stop, include `-SkipOsShutdown $true`.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

(Optional) Create a replacement instance

While you are waiting for assistance from AWS re:Post or the Support Center, you can create a replacement instance if needed. Create an AMI from the stuck instance, and launch a new instance using the new AMI.

> **Important**
>
> You can create a replacement instance if the stuck instance produces system status checks only, as instance status checks will result in the AMI copying over an exact replica of the broken operating system. After you've confirmed the status message, create the AMI and launch a new instance using the new AMI.

Console

To create a replacement instance

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. Clear **Reboot instance**.
   c. Choose **Create image**.

   For more information, see the section called “Create an AMI from an instance”.
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 (delete) instance**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

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 (delete) instance**. 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

1. Create an AMI from the stuck instance using the create-image command with the `--no-reboot` option.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-replacement-ami" \
    --description ""AMI for replacement instance" \
    --no-reboot
```

2. Launch a new instance from the AMI that you just created, using the run-instances command.
3. Verify that the new instance is working.
4. (Optional) Terminate the stuck instance using the terminate-instances command.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To create a replacement instance

1. Create an AMI from the stuck instance using the New-EC2Image cmdlet and set `-NoReboot` to `true`.

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-replacement-ami" `
    -Description "AMI for replacement instance" `
    -NoReboot $true
```

2. Launch a new instance from the AMI that you just created, using the New-EC2Instance cmdlet.
3. Verify that the new instance is working.
4. (Optional) Terminate the stuck instance using the Remove-EC2Instance cmdlet.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

Troubleshoot Amazon EC2 instance termination issues

Shutting down or deleting your instance is known as instance termination. The following information can help you troubleshoot issues when you terminate your instance.

You are not billed for any instance usage while an instance is not in the `running` state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to `shutting-down`.

Instance terminates immediately

Several issues can cause your instance to terminate immediately on start-up. See Instance terminates immediately for more information.

Delayed instance termination

If your instance remains in the `shutting-down` state longer than a few minutes, it might be because:

- The instance is running shutdown scripts.
- There's a problem with the underlying host computer.

After several hours in the `shutting-down` state, Amazon EC2 treats the instance as stuck and forcibly terminates it.

To resolve a stuck instance yourself:

1. **Force terminate the instance**

   Use the Amazon EC2 console or the AWS CLI to force terminate the instance. For the steps, see Force terminate an instance.

   The instance will first attempt a graceful shutdown, which includes flushing file system caches and metadata (although you can optionally bypass the graceful shutdown). If the graceful shutdown fails to complete within the timeout period, the instance shuts down forcibly without flushing the file system caches and metadata.
2. **If force terminate fails**

   If, after several hours, the instance has not terminated and it appears stuck terminating, do the following:

   a. Post a request for help on AWS re:Post. To help expedite a resolution, include the instance ID, and describe the steps that you've already taken.
   b. Alternatively, if you have a support plan, create a technical support case in the Support Center.

Force terminate an instance

If it appears that your instance is stuck terminating, you can force your instance to terminate. If, after several hours, the instance has not terminated, post a request for help to AWS re:Post. 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.

Console

To force terminate an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance state**, **Force terminate instance**.

   Note that **Force terminate instance** is only available in the console if your instance is in the `stopping` state. If your instance is in another state (except `shutting-down` and `terminated`) you can use the AWS CLI to force terminate your instance.

4. (Optional) To bypass the graceful OS shutdown during the force terminate, select the **Skip OS shutdown** checkbox.
5. Choose **Force terminate**.

AWS CLI

To force terminate an instance

Use the terminate-instances command with the `--force` option.

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

To bypass the graceful OS shutdown during force terminate, include the `--skip-os-shutdown` option.

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

To force terminate an instance

Use the Remove-EC2Instance cmdlet and set `-Enforce` to `true`.

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

To bypass the graceful OS shutdown during force terminate, include `-SkipOsShutdown $true`.

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

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.

Error: The instance may not be terminated. Modify its 'disableApiTermination' instance attribute

If you try to terminate an instance and get the `The instance i-1234567890abcdef0 may not be terminated. Modify its 'disableApiTermination' instance attribute` error message, it indicates that the instance has been enabled for termination protection. Termination protection prevents the instance from being accidentally terminated.

You must disable termination protection before you can terminate the instance.

For more information, see Change instance termination protection.

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, find the Auto Scaling group or the fleet that is launching the instances and either set its capacity to 0 or delete it.

Troubleshoot an unreachable Amazon EC2 instance

The following information can help you troubleshoot unreachable Amazon EC2 instances. You can capture screenshots or access console output to help diagnose problems and determine if you should reboot your instance. For unreachable Windows instances, troubleshoot by reviewing screenshots returned by the service.

Contents

- [Instance console output]
- [Capture a screenshot of an unreachable instance]
- [Common screenshots to troubleshoot unreachable Windows instances]
- [Instance recovery when a host computer fails]
- [Instance appeared offline and unexpectedly rebooted]

Instance reboot

The ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

Just as you can reset a computer by pressing the reset button, you can reset EC2 instances using the Amazon EC2 console, CLI, or API. For more information, see Reboot your Amazon EC2 instance.

Instance console output

Console output is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started.

- **Linux instances** – The instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. The console output returns buffered information that was posted shortly after an instance transition state (start, stop, reboot, and terminate). The posted output is not continuously updated; only when it is likely to be of the most value.
- **Windows instances –** The instance console output includes the last three system event log errors.

Only the instance owner can access the console output.

You can retrieve the latest serial console output during the instance lifecycle. This option is only supported on Nitro-based instances.

Console

To get console output

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 **Actions**, **Monitor and troubleshoot**, **Get system log**.

AWS CLI

To get console output

Use the get-console-output command.

```
aws ec2 get-console-output --instance-id i-1234567890abcdef0
```

PowerShell

To get console output

Use the Get-EC2ConsoleOutput cmdlet.

```
Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0
```

Capture a screenshot of an unreachable instance

If you are unable to connect to your instance, you can capture a screenshot of your instance and view it as an image. The image can provide visibility as to the status of the instance, and allows for quicker troubleshooting.

You can generate screenshots while the instance is running or after it has crashed. The image is generated in JPG format and is no larger than 100 kb. There is no data transfer cost for the screenshot.

Limitations

This feature is not supported for the following:

- Bare metal instances (instances of type `*.metal`)
- Instance is using an NVIDIA GRID driver
- Instances powered by Arm-based Graviton processors

- Windows instances on AWS Outposts
- Windows instances on AWS Local Zones

Region support

This feature is not available in the following Regions:

- Asia Pacific (Thailand)
- Mexico (Central)
- GovCloud Regions

Console

To get a screenshot of an 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 capture.
4. Choose **Actions**, **Monitor and troubleshoot**, **Get instance screenshot**.
5. Choose **Download**, or right-click the image to download and save it.

AWS CLI

To capture a screenshot of an instance

Use the get-console-screenshot command. The output is base64-encoded.

```
aws ec2 get-console-screenshot --instance-id i-1234567890abcdef0
```

PowerShell

To capture a screenshot of an instance

Use the Get-EC2ConsoleScreenshot cmdlet. The output is base64-encoded.

```
Get-EC2ConsoleScreenshot -InstanceId i-1234567890abcdef0
```

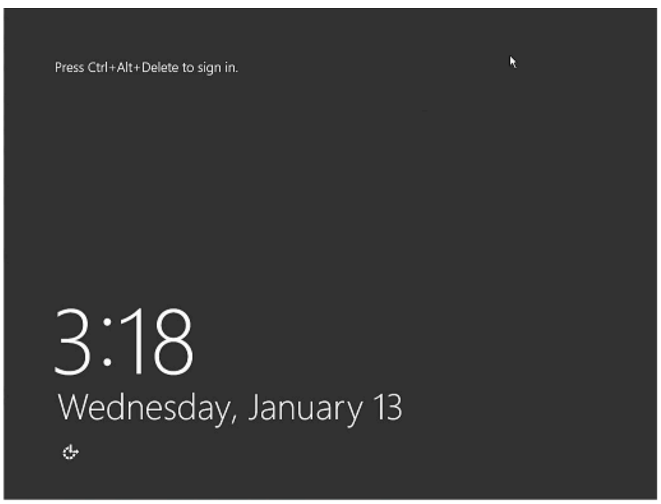
Common screenshots to troubleshoot unreachable Windows instances

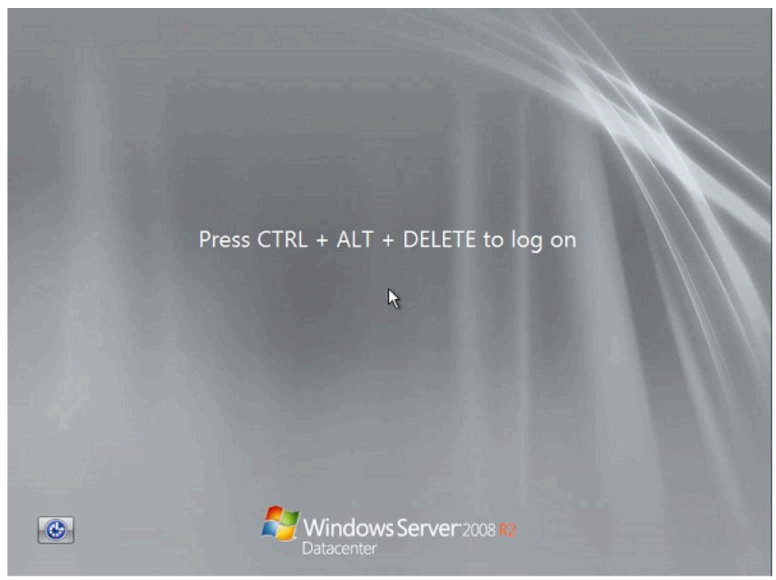
You can use the following information to help you troubleshoot an unreachable Windows instance based on screenshots returned by the service.

- Log on screen (Ctrl+Alt+Delete)
- Recovery console screen
- Windows boot manager screen
- Sysprep screen
- Getting ready screen
- Windows Update screen
- Chkdsk

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

Configuration	Verify
Security group configuration	Verify that port 3389 is open for your security group. Verify you are connecting to the right public IP address. If the instance was not associated with an Elastic IP, the public IP changes after the instance stops/starts. For more information, see Remote Desktop can't connect to the remote computer.
VPC configuration (Network ACLs)	Verify that the access control list (ACL) for your Amazon VPC is not blocking access. For information, see Network ACLs in the *Amazon VPC User Guide*.
VPN configuration	If you are connecting to your VPC using a virtual private network (VPN), verify VPN tunnel connectivity. For more information, see Troubleshooting AWS Client VPN: Tunnel connectivity issues to a VPC.

Windows network configuration

Configuration	Verify
Windows Firewall	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.
Advanced TCP/IP configuration (Use of static IP)	The instance may be unresponsive because you configured a static IP address. For a VPC, create a network interface and attach it to the instance.

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.

> **Tip**
>
> You can use the `AWSSupport-TroubleshootRDP` runbook to check and modify various settings that might affect Remote Desktop Protocol (RDP) connections. For more information, see [`AWSSupport-TroubleshootRDP`] in the *AWS Systems Manager Automation runbook reference*.

Remote Desktop Services configuration

Configuration	Verify
RDS is running	Verify that RDS is running on the instance. Connect to the instance using the Microsoft Management Console (MMC) Services snap-in (`services.msc`). In the list of services, verify that **Remote Desktop Services** is **Running**. If it isn't, start it and then set the startup type to **Automatic**. If you can't connect to the instance by using the Services snap-in, detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume, and modify the Start registry key. When you are finished, reattach the root volume to the original instance.
RDS is enabled	Even if the service is started, it might be disabled. Detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume,

Configuration	Verify
	and enable the service by modifying the **Terminal Server** registry key as described in Enable Remote Desktop on an EC2 instance with remote registry. When you are finished, reattach the root volume to the original instance.

High CPU usage

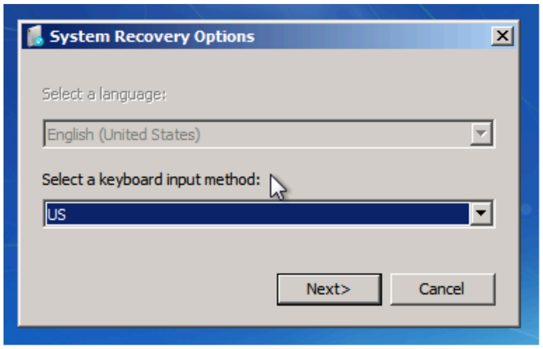
Check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch. If **CPUUtilization (Maximum)** is a high number, wait for the CPU to go down and try connecting again. High CPU usage can be caused by:

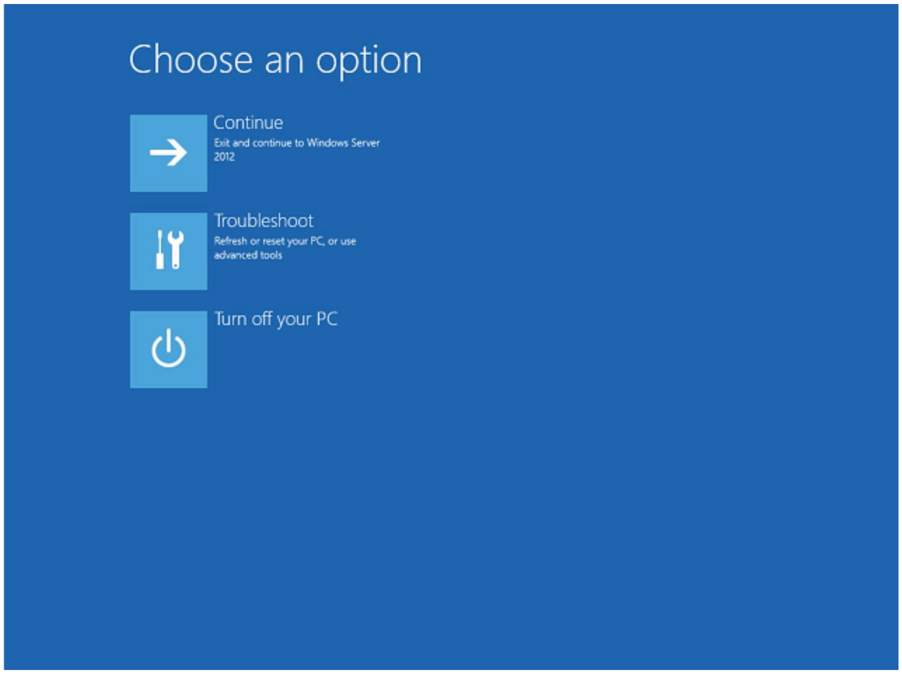
- Windows Update
- Security Software Scan
- Custom Startup Script
- Task Scheduler

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 (Windows instances only).

Recovery console screen

Console Screenshot Service returned the following.





The operating system might 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 public Windows AMIs provided by AWS 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.

   > **Warning**
   >
   > If your temporary instance and the original instance were launched using the same AMI, 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. If you must create a temporary instance using the same AMI, to avoid a disk signature collision, complete the steps in [Disk signature collision](#).

Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses an AMI for Windows Server 2016, launch the temporary instance using an AMI for Windows Server 2019.

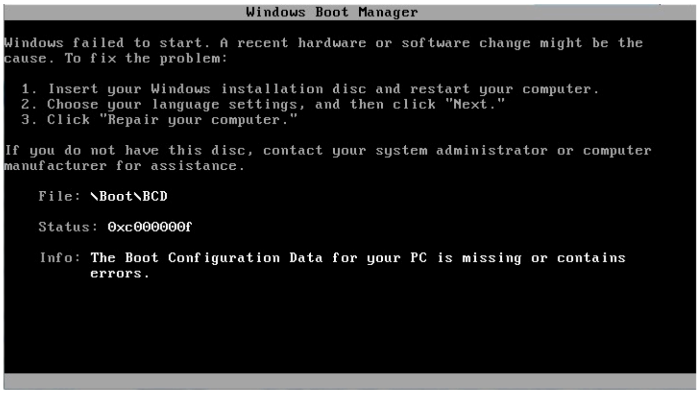
3. Log in to the instance and run the following command from a command prompt to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

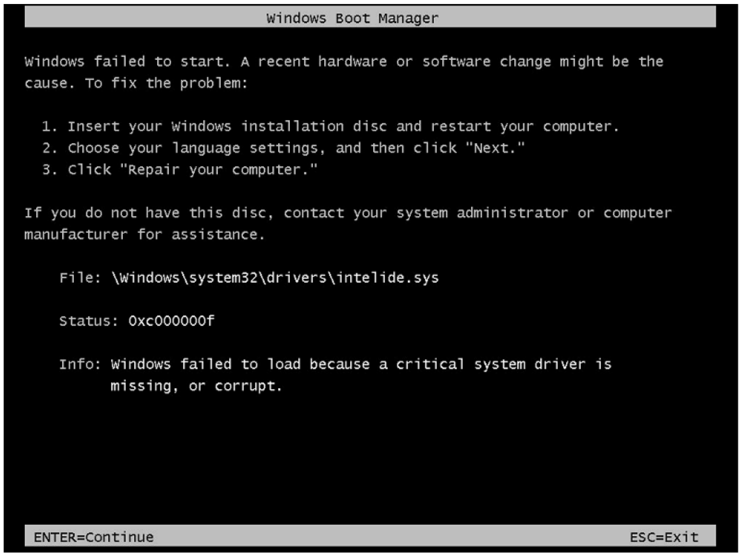
```
bcdedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy
 ignoreallfailures
```

4. Reattach the volume to the unreachable instance and start the instance again.

Windows boot manager screen

Console Screenshot Service returned the following.

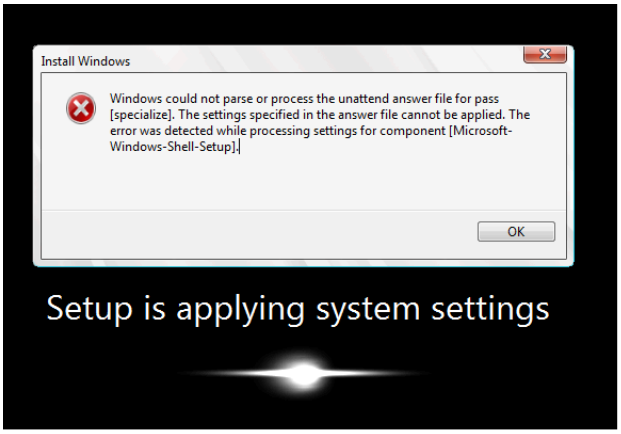


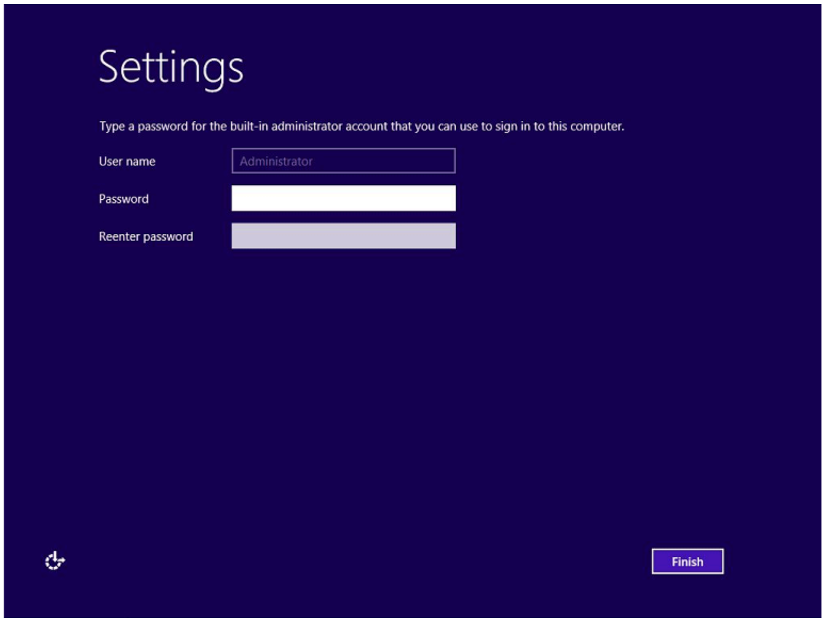


The operating system experienced a fatal corruption in the system file and/or the registry. When the instance is stuck in this state, you should recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach any root volumes from the 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.

Sysprep screen

Console Screenshot Service returned the following.

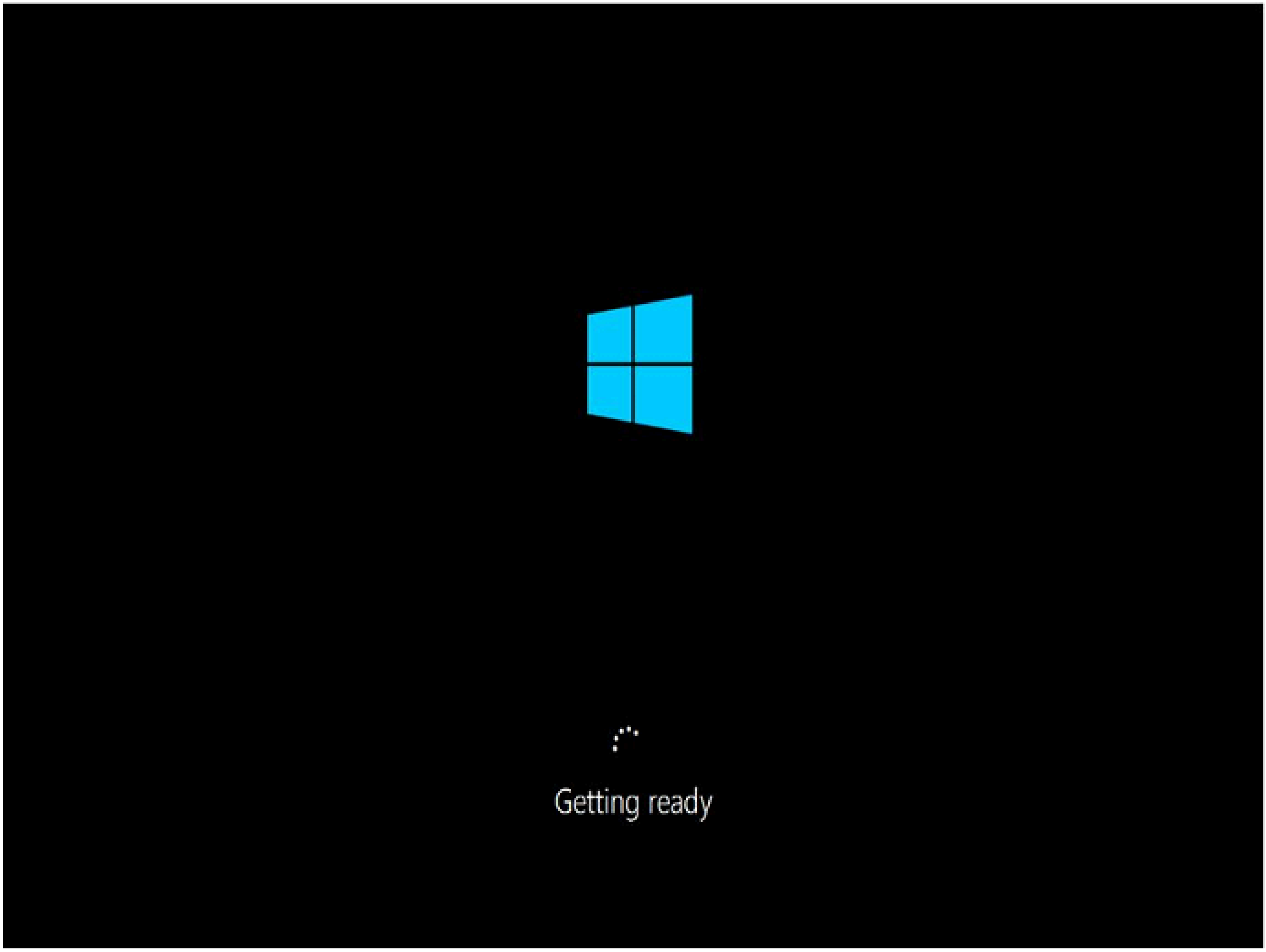




You may see this screen if you did not use the EC2Config Service to call Sysprep or if the operating system failed while running Sysprep. You can reset the password using EC2Rescue. Otherwise, see Create an Amazon EC2 AMI using Windows Sysprep.

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

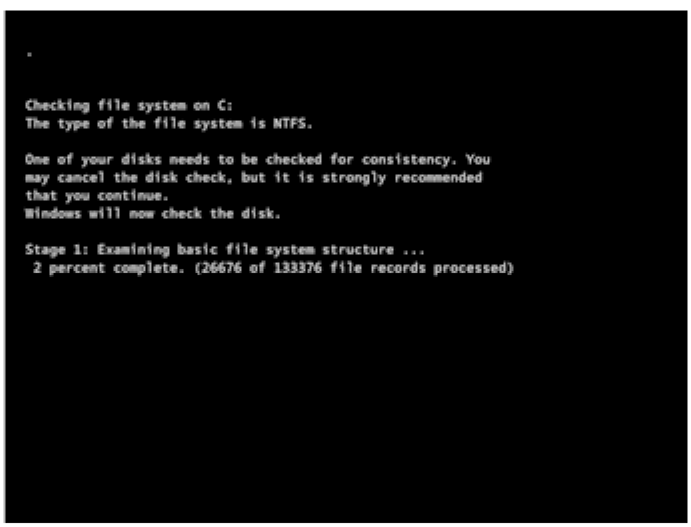
Console Screenshot Service returned the following.

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.

Instance recovery when a host computer fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You are notified of such an event ahead of time by email.

To recover an Amazon EBS-backed instance running on a host computer that failed

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.
4. Restore any important data.

For more information, see Stop and start Amazon EC2 instances.

To recover an instance store-backed instance running on a host computer that failed

1. Create an AMI from the instance.
2. Upload the image to Amazon S3.
3. Back up important data to Amazon EBS or Amazon S3.
4. Terminate the instance.
5. Launch a new instance from the AMI.
6. Restore any important data to the new instance.

Instance appeared offline and unexpectedly rebooted

If your instance appears to have been offline and then unexpectedly rebooted, it might have undergone automatic instance recovery. This happens when AWS detects that the instance is unavailable due to an underlying hardware or software issue, and either simplified automatic recovery or CloudWatch action based recovery is enabled on the instance.

During the recovery process, AWS attempts to restore the instance's availability by migrating it to different hardware. To verify whether automatic instance recovery occurred for your instance, see Verify if automatic instance recovery occurred.

> **Note**
>
> If your workload or application is unresponsive, check whether it's running on the instance. If it's not, start it manually. To prevent this issue in the future, implement a recovery plan to ensure your workload or application functions properly after instance recovery.

Troubleshoot issues connecting to your Amazon EC2 Linux instance

The following information and common errors can help you troubleshoot connecting to your Linux instance.

Connection issues

Common causes for connection issues

We recommend that you start to troubleshoot instance connection problems by verifying that you have accurately performed the following tasks.

Verify the username for your instance

You can connect to your instance using the username for your user account or the default username for the AMI that you used to launch your instance.

- **Get the username for your user account.**

  For more information about how to create a user account, see Manage system users on your Amazon EC2 Linux instance.

- **Get the default username for the AMI that you used to launch your instance.**

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

Verify that your security group rules allow traffic

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. The default security group for the VPC does not allow incoming SSH traffic by default. The security group created by the launch instance wizard enables SSH traffic by default. For steps to add a rule for inbound SSH traffic to your Linux instance, see Rules to connect to instances from your computer. For steps to verify, see Error connecting to your instance: Connection timed out.

Verify that your instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready to accept connection requests. Check your instance to make sure it is running and has passed its status checks.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Verify the following:
    a. In the **Instance state** column, verify that your instance is in the `running` state.
    b. In the **Status check** column, verify that your instance has passed all status checks.

Verify that you've satisfied all prerequisites to connect

Ensure that you have all the information that you need to connect. For more information, see Connect to your Linux instance using SSH.

Connect from Linux or macOS X

If your local computer operating system is Linux or macOS X, check the following for specific prerequisites for connecting to a Linux instance:

- SSH client
- EC2 Instance Connect
- AWS Systems Manager Session Manager

Connect from Windows

If your local computer operating system is Windows, check the following for specific prerequisites for connecting to a Linux instance:

- OpenSSH
- PuTTY
- AWS Systems Manager Session Manager
- Windows Subsystem for Linux

Check if the instance is a managed instance

User-initiated connections to managed instances are not allowed. To determine if the instance is managed, find the **Managed** field for the instance. If the value is **true**, it's a managed instance. For more information, see Amazon EC2 managed instances.

Error connecting to your instance: Connection timed out

If you try to connect to your instance and get the error message `Network error: Connection timed out` or `Error connecting to [instance], reason: -> Connection timed out: connect`, try the following:

Check your security group rules.

You need a security group rule that allows inbound traffic from your local computer's public IPv4 address on the proper port.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Security** tab at the bottom of the console page, under **Inbound rules**, check the list of rules that are in effect for the selected instance. Verify that there is a rule that allows traffic from your local computer to port 22 (SSH).

   If your security group does not have a rule that allows inbound traffic from your local computer, add a rule to your security group. For more information, see Rules to connect to instances from your computer.
4. For the rule that allows inbound traffic, check the **Source** field. If the value is a single IP address, and if the IP address is not static, a new IP address will be assigned each time you restart your computer. This will result in the rule not including your computer's IP address traffic. The IP address might not be static if your computer is on a corporate network, or you're connecting through an internet service provider (ISP), or your computer IP address is dynamic and changes each time you restart your computer. To ensure that your security group rule

allows inbound traffic from your local computer, instead of specifying a single IP address for **Source**, rather specify the range of IP addresses used by your client computers.

For more information about security group rules, see Security group rules in the *Amazon VPC User Guide*.

Check the route table for the subnet.

You need a route that sends all traffic destined outside the VPC to the internet gateway for the VPC.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Networking** tab, make note of the values for **VPC ID** and **Subnet ID**.
4. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
5. In the navigation pane, choose **Internet Gateways**. Verify that there is an internet gateway attached to your VPC. Otherwise, choose **Create internet gateway**, enter a name for the internet gateway, and choose **Create internet gateway**. Then, for the internet gateway you created, choose **Actions**, **Attach to VPC**, select your VPC, and then choose **Attach internet gateway** to attach it to your VPC.
6. In the navigation pane, choose **Subnets**, and then select your subnet.
7. On the **Route table** tab, verify that there is a route with `0.0.0.0/0` as the destination and the internet gateway for your VPC as the target. If you're connecting to your instance using its IPv6 address, verify that there is a route for all IPv6 traffic (`::/0`) that points to the internet gateway. Otherwise, do the following:

   a. Choose the ID of the route table (rtb-*xxxxxxxx*) to navigate to the route table.

   b. On the **Routes** tab, choose **Edit routes**. Choose **Add route**, use `0.0.0.0/0` as the destination and the internet gateway as the target. For IPv6, choose **Add route**, use `::/0` as the destination and the internet gateway as the target.

   c. Choose **Save routes**.

Check the network access control list (ACL) for the subnet.

The network ACLs must allow inbound SSH traffic from your local IP address on port 22. It must also allow outbound traffic to the ephemeral ports (1024-65535).

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Subnets**.
3. Select your subnet.
4. On the **Network ACL** tab, for **Inbound rules**, verify that the rules allow inbound traffic from your computer on the required port. Otherwise, delete or modify the rule that is blocking the traffic.
5. For **Outbound rules**, verify that the rules allow outbound traffic to your computer on the ephemeral ports. Otherwise, delete or modify the rule that is blocking the traffic.

If your computer is on a corporate network

Ask your network administrator whether the internal firewall allows inbound and outbound traffic from your computer on port 22.

If you have a firewall on your computer, verify that it allows inbound and outbound traffic from your computer on port 22.

Check that your instance has a public IPv4 address.

If not, you can associate an Elastic IP address with your instance. For more information, see Elastic IP addresses.

Check the CPU load on your instance; the server may be overloaded.

AWS automatically provides data such as Amazon CloudWatch metrics and instance status, which you can use to see how much CPU load is on your instance and, if necessary, adjust how your loads are handled. For more information, see Monitor your instances using CloudWatch.

- If your load is variable, you can automatically scale your instances up or down using Auto Scaling and Elastic Load Balancing.
- If your load is steadily growing, you can move to a larger instance type. For more information, see Amazon EC2 instance type changes.

To connect to your instance using an IPv6 address, check the following:

- Your subnet must be associated with a route table that has a route for IPv6 traffic (`::/0`) to an internet gateway.

- Your security group rules must allow inbound traffic from your local IPv6 address on port 22.
- Your network ACL rules must allow inbound and outbound IPv6 traffic.
- If you launched your instance from an older AMI, it might not be configured for DHCPv6 (IPv6 addresses are not automatically recognized on the network interface). For more information, see Configure IPv6 on your instances in the *Amazon VPC User Guide*.
- Your local computer must have an IPv6 address, and must be configured to use IPv6.

Error: unable to load key ... Expecting: ANY PRIVATE KEY

If you try to connect to your instance and get the error message, `unable to load key ... Expecting: ANY PRIVATE KEY`, the file in which the private key is stored is incorrectly configured. If the private key file ends in `.pem`, it might still be incorrectly configured. A possible cause for an incorrectly configured private key file is a missing certificate.

If the private key file is incorrectly configured, follow these steps to resolve the error

1. Create a new key pair. For more information, see Create a key pair using Amazon EC2.

   Note

   Alternatively, you can create a new key pair using a third-party tool. For more information, see Create a key pair using a third-party tool and import the public key to Amazon EC2.

2. Add the new key pair to your instance. For more information, see I've lost my private key. How can I connect to my instance?.
3. Connect to your instance using the new key pair.

Error: User key not recognized by server

If you use SSH to connect to your instance

- Use `ssh -vvv` to get triple verbose debugging information while connecting:

```
ssh -vvv -i path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

The following sample output demonstrates what you might see if you were trying to connect to your instance with a key that was not recognized by the server:

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-
interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

If you use PuTTY to connect to your instance

- Verify that your private key (.pem) file has been converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see Connect to your Linux instance using PuTTY.

> **Note**
>
> In PuTTYgen, load your private key file and select **Save Private Key** rather than **Generate**.

- Verify that you are connecting with the appropriate username for your AMI. Enter the username in the **Host name** box in the **PuTTY Configuration** window.

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

- Verify that you have an inbound security group rule to allow inbound traffic to the appropriate port. For more information, see [Rules to connect to instances from your computer].

Error: Permission denied or connection closed by [instance] port 22

If you connect to your instance using SSH and get any of the following errors, `Host key not found in [directory]`, `Permission denied (publickey)`, `Authentication failed,` `permission denied`, or `Connection closed by [instance] port 22`, verify that you are connecting with the appropriate username for your AMI *and* that you have specified the proper private key (`.pem`) file for your instance.

The appropriate usernames are as follows:

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

For example, to use an SSH client to connect to an Amazon Linux instance, use the following command:

```
ssh -i /path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

Confirm that you are using the private key file that corresponds to the key pair that you selected when you launched the instance.

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

2. In the navigation pane, choose **Instances** and then select your instance.
3. On the **Details** tab, under **Instance details**, verify the value of **Key pair name**.
4. If you did not specify a key pair when you launched the instance, you can terminate the instance and launch a new instance, ensuring that you specify a key pair. If this is an instance that you have been using but you no longer have the `.pem` file for your key pair, you can replace the key pair with a new one. For more information, see I've lost my private key. How can I connect to my instance?.

If you generated your own key pair, ensure that your key generator is set up to create RSA keys. DSA keys are not accepted.

If you get a `Permission denied (publickey)` error and none of the above applies (for example, you were able to connect previously), the permissions on the home directory of your instance may have been changed. Permissions for `/home/instance-user-name/.ssh/authorized_keys` must be limited to the owner only.

To verify the permissions on your instance

1. Stop your instance and detach the root volume. For more information, see Stop and start Amazon EC2 instances.
2. Launch a temporary instance in the same Availability Zone as your current instance (use a similar or the same AMI as you used for your current instance), and attach the root volume to the temporary instance.
3. Connect to the temporary instance, create a mount point, and mount the volume that you attached.
4. From the temporary instance, check the permissions of the `/home/instance-user-name/` directory of the attached volume. If necessary, adjust the permissions as follows:

```
[ec2-user ~]$ chmod 600 mount_point/home/instance-user-name/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name
```

5. Unmount the volume, detach it from the temporary instance, and re-attach it to the original instance. Ensure that you specify the correct device name for the root volume; for example, /dev/xvda.
6. Start your instance. If you no longer require the temporary instance, you can terminate it.

Error: Unprotected private key file

Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem
Permission denied (publickey).
```

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key `.ssh/my_private_key.pem` with file permissions of `0777`, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key.

If you are connecting from macOS or Linux, run the following command to fix this error, substituting the path for your private key file.

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

If you are connecting to a Linux instance from Windows, perform the following steps on your local computer.

1. Navigate to your .pem file.
2. Right-click on the .pem file and select **Properties**.
3. Choose the **Security** tab.

4. Select **Advanced**.
5. Verify that you are the owner of the file. If not, change the owner to your username.
6. Select **Disable inheritance** and **Remove all inherited permissions from this object**.
7. Select **Add**, **Select a principal**, enter your username, and select **OK**.
8. From the **Permission Entry** window, grant **Read** permissions and select **OK**.
9. Click **Apply** to ensure all settings are saved.
10. Select **OK** to close the **Advanced Security Settings** window.
11. Select **OK** to close the **Properties** window.
12. You should be able to connect to your Linux instance from Windows using SSH.

From a Windows command prompt, run the following commands.

1. From the command prompt, navigate to the file path location of your .pem file.
2. Run the following command to reset and remove explicit permissions:

```
icacls.exe $path /reset
```

3. Run the following command to grant Read permissions to the current user:

```
icacls.exe $path /GRANT:R "$($env:USERNAME):(R)"
```

4. Run the following command to disable inheritance and remove inherited permissions.

```
icacls.exe $path /inheritance:r
```

5. You should be able to connect to your Linux instance from Windows using SSH.

Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"

If you use a third-party tool, such as **ssh-keygen**, to create an RSA key pair, it generates the private key in the OpenSSH key format. When you connect to your instance, if you use the private key in the OpenSSH format to decrypt the password, you'll get the error `Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"`.

To resolve the error, the private key must be in the PEM format. Use the following command to create the private key in the PEM format:

```
ssh-keygen -m PEM
```

Error: Host key verification failed

This error occurs if there is a mismatch between the host key stored on the instance in the `known_hosts` file and on the client. For example, a mismatch can occur if you connect to an instance using one public IP address, and then try to connect to it again using a different public IP address. This can happen after you add or remove an Elastic IP address, as doing so changes the public IP address of an instance.

To resolve this error, start by confirming that there was an expected change to the host key or the network configuration of the instance. Before you connect to the instance, you might also want to verify the host fingerprint. After you connect to the instance, you can remove the old host key from the `known_hosts` file. For instructions, refer to the documentation for the Linux distribution in use on your instance.

Error: Server refused our key *or* No supported authentication methods available

If you use PuTTY to connect to your instance and get either of the following errors, Error: Server refused our key or Error: No supported authentication methods available, verify that you are connecting with the appropriate username for your AMI. Type the username in **User name** in the **PuTTY Configuration** window.

The appropriate usernames are as follows:

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`

AMI used to launch instance	Default username
RHEL	ec2-user or root
SUSE	ec2-user or root
Ubuntu	ubuntu
Oracle	ec2-user
Bitnami	bitnami
Rocky Linux	rocky
Other	Check with the AMI provider

You should also verify that:

- You are using the latest version of PuTTY. For more information, see the PuTTY web page.
- Your private key (.pem) file has been correctly converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see Connect to your Linux instance using PuTTY.

Cannot ping instance

The `ping` command is a type of ICMP traffic — if you are unable to ping your instance, ensure that your inbound security group rules allow ICMP traffic for the `Echo Request` message from all sources, or from the computer or instance from which you are issuing the command.

If you are unable to issue a `ping` command from your instance, ensure that your outbound security group rules allow ICMP traffic for the `Echo Request` message to all destinations, or to the host that you are attempting to ping.

`Ping` commands can also be blocked by a firewall or time out due to network latency or hardware issues. You should consult your local network or system administrator for help with further troubleshooting.

Error: Server unexpectedly closed network connection

If you are connecting to your instance with PuTTY and you receive the error "Server unexpectedly closed network connection," verify that you have enabled keepalives on the Connection page of the PuTTY Configuration to avoid being disconnected. Some servers disconnect clients when they do not receive any data within a specified period of time. Set the Seconds between keepalives to 59 seconds.

If you still experience issues after enabling keepalives, try to disable Nagle's algorithm on the Connection page of the PuTTY Configuration.

Error: Host key validation failed for EC2 Instance Connect

If you rotate your instance host keys, the new host keys are not automatically uploaded to the AWS trusted host keys database. This causes host key validation to fail when you try to connect to your instance using the EC2 Instance Connect browser-based client, and you're unable to connect to your instance.

To resolve the error, you must run the `eic_harvest_hostkeys` script on your instance, which uploads your new host key to EC2 Instance Connect. The script is located at `/opt/aws/bin/` on Amazon Linux 2 instances, and at `/usr/share/ec2-instance-connect/` on Ubuntu instances.

Amazon Linux 2

To resolve the host key validation failed error on an Amazon Linux 2 instance

1. Connect to your instance using SSH.

   You can connect by using the EC2 Instance Connect CLI or by using the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For Amazon Linux 2, the default username is `ec2-user`.

   For example, if your instance was launched using Amazon Linux 2, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
   ```

For more information about connecting to your instance, see [Connect to your Linux instance using an SSH client](#).

2. Navigate to the following folder.

```
[ec2-user ~]$ cd /opt/aws/bin/
```

3. Run the following command on your instance.

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

Note that a successful call results in no output.

You can now use the EC2 Instance Connect browser-based client to connect to your instance.

Ubuntu

To resolve the host key validation failed error on an Ubuntu instance

1. Connect to your instance using SSH.

You can connect by using the EC2 Instance Connect CLI or by using the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For Ubuntu, the default username is `ubuntu`.

For example, if your instance was launched using Ubuntu, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see [Connect to your Linux instance using an SSH client](#).

2. Navigate to the following folder.

```
[ec2-user ~]$ cd /usr/share/ec2-instance-connect/
```

3. Run the following command on your instance.

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

Note that a successful call results in no output.

You can now use the EC2 Instance Connect browser-based client to connect to your instance.

Can't connect to Ubuntu instance using EC2 Instance Connect

If you use EC2 Instance Connect to connect to your Ubuntu instance and you get an error when attempting to connect, you can use the following information to try to fix the issue.

Possible cause

The `ec2-instance-connect` package on the instance is not the latest version.

Solution

Update the `ec2-instance-connect` package on the instance to the latest version, as follows:

1. Connect to your instance using a method other than EC2 Instance Connect.
2. Run the following command on your instance to update the `ec2-instance-connect` package to the latest version.

```
apt update && apt upgrade
```

I've lost my private key. How can I connect to my instance?

If you lose the private key for an EBS-backed instance, you can regain access to your instance. You must stop the instance, detach its root volume and attach it to another instance as a data volume, modify the `authorized_keys` file with a new public key, move the volume back to the original instance, and restart the instance. For more information about launching, connecting to, and stopping instances, see Amazon EC2 instance state changes.

This procedure is only supported for instances with EBS root volumes. If the root device is an instance store volume, you cannot use this procedure to regain access to your instance; you must have the private key to connect to the instance. To determine the root device type of your instance, open the Amazon EC2 console, choose **Instances**, select the instance, choose the **Storage** tab, and in the **Root device details** section, check the value of **Root device type**.

The value is either `EBS` or `INSTANCE-STORE`.

In addition to the following steps, there are other ways to connect to your Linux instance if you lose your private key. For more information, see [How can I connect to my Amazon EC2 instance if I lost my SSH key pair after its initial launch?](#)

Steps for connecting to an EBS-backed instance with a different key pair

- Step 1: Create a new key pair
- Step 2: Get information about the original instance and its root volume
- Step 3: Stop the original instance
- Step 4: Launch a temporary instance
- Step 5: Detach the root volume from the original instance and attach it to the temporary instance
- Step 6: Add the new public key to authorized_keys on the original volume mounted to the temporary instance
- Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance
- Step 8: Connect to the original instance using the new key pair
- Step 9: Clean up

Step 1: Create a new key pair

Create a new key pair using either the Amazon EC2 console or a third-party tool. If you want to name your new key pair exactly the same as the lost private key, you must first delete the existing key pair. For information about creating a new key pair, see Create a key pair using Amazon EC2 or Create a key pair using a third-party tool and import the public key to Amazon EC2.

Step 2: Get information about the original instance and its root volume

Make note of the following information because you'll need it to complete this procedure.

To get information about your original instance

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

2. Choose **Instances** in the navigation pane, and then select the instance that you'd like to connect to. (We'll refer to this as the *original* instance.)
3. On the **Details** tab, make note of the instance ID and AMI ID.
4. On the **Networking** tab, make note of the Availability Zone.
5. On the **Storage** tab, under **Root device name**, make note of the device name for the root volume (for example, `/dev/xvda`). Then, under **Block devices**, find this device name and make note of the volume ID (for example, vol-0a1234b5678c910de).

Step 3: Stop the original instance

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.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Step 4: Launch a temporary instance

To launch a temporary instance

1. In the navigation pane, choose **Instances**, and then choose **Launch instances**.
2. In the **Name and tags** section, for **Name**, enter **Temporary**.
3. In the **Application and OS Images** section, select the same AMI that you used to launch the original instance. If this AMI is unavailable, you can create an AMI that you can use from the stopped instance. For more information, see Create an Amazon EBS-backed AMI.
4. In the **Instance type** section, keep the default instance type.
5. In the **Key pair** section, for **Key pair name**, select the existing key pair to use or create a new one.
6. In the **Network settings** section, choose **Edit**, and then for **Subnet**, select a subnet in the same Availability Zone as the original instance.
7. In the **Summary** panel, choose **Launch**.

Step 5: Detach the root volume from the original instance and attach it to the temporary instance

1. In the navigation pane, choose **Volumes** and select the root device volume for the original instance (you made note of its volume ID in a previous step). Choose **Actions**, **Detach volume**, and then choose **Detach**. Wait for the state of the volume to become `available`. (You might need to choose the Refresh icon.)
2. With the volume still selected, choose **Actions**, and then choose **Attach volume**. Select the instance ID of the temporary instance, make note of the device name specified under **Device name** (for example, `/dev/sdf`), and then choose **Attach volume**.

> **Note**
>
> If you launched your original instance from an AWS Marketplace AMI and your volume contains AWS Marketplace codes, you must first stop the temporary instance before you can attach the volume.

Step 6: Add the new public key to `authorized_keys` on the original volume mounted to the temporary instance

1. Connect to the temporary instance.
2. From the temporary instance, mount the volume that you attached to the instance so that you can access its file system. For example, if the device name is `/dev/sdf`, use the following commands to mount the volume as `/mnt/tempvol`.

> **Note**
>
> The device name might appear differently on your instance. For example, devices mounted as `/dev/sdf` might show up as `/dev/xvdf` on the instance. Some versions of Red Hat (or its variants, such as CentOS) might even increment the trailing letter by 4 characters, where `/dev/sdf` becomes `/dev/xvdk`.

   a. Use the **lsblk** command to determine if the volume is partitioned.

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

```
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0    8G  0 disk
##xvda1 202:1    0    8G  0 part /
xvdf     202:80   0  101G  0 disk
##xvdf1 202:81   0  101G  0 part
xvdg     202:96   0   30G  0 disk
```

In the preceding example, /dev/xvda and /dev/xvdf are partitioned volumes, and /dev/xvdg is not. If your volume is partitioned, you mount the partition (/dev/xvdf1) instead of the raw device (/dev/xvdf) in the next steps.

b. Create a temporary directory to mount the volume.

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

c. Mount the volume (or partition) at the temporary mount point, using the volume name or device name that you identified earlier. The required command depends on your operating system's file system. Note that the device name might appear differently on your instance. See the note in Step 6 for more information.

- Amazon Linux, Ubuntu, and Debian

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
```

- Amazon Linux 2, CentOS, SUSE Linux 12, and RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

Note

If you get an error stating that the file system is corrupt, run the following command to use the **fsck** utility to check the file system and repair any issues:

```
[ec2-user ~]$ sudo fsck /dev/xvdf1
```

3. From the temporary instance, use the following command to update authorized_keys on the mounted volume with the new public key from the authorized_keys for the temporary instance.

> **Important**
>
> The following examples use the Amazon Linux username `ec2-user`. You might need to substitute a different username, such as `ubuntu` for Ubuntu instances.

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

If this copy succeeded, you can go to the next step.

(Optional) Otherwise, if you don't have permission to edit files in `/mnt/tempvol`, you must update the file using **sudo** and then check the permissions on the file to verify that you are able to log into the original instance. Use the following command to check the permissions on the file.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
total 4
-rw------- 1 222 500 398 Sep 13 22:54 authorized_keys
```

In this example output, *222* is the user ID and *500* is the group ID. Next, use **sudo** to re-run the copy command that failed.

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

Run the following command again to determine whether the permissions changed.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

If the user ID and group ID have changed, use the following command to restore them.

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance

1. From the temporary instance, unmount the volume that you attached so that you can reattach it to the original instance. For example, use the following command to unmount the volume at `/mnt/tempvol`.

   ```
   [ec2-user ~]$ sudo umount /mnt/tempvol
   ```

2. Detach the volume from the temporary instance (you unmounted it in the previous step): From the Amazon EC2 console, choose **Volumes** in the navigation pane, select the root device volume for the original instance (you made note of the volume ID in a previous step), choose **Actions**, **Detach volume**, and then choose **Detach**. Wait for the state of the volume to become `available`. (You might need to choose the Refresh icon.)

3. Reattach the volume to the original instance: With the volume still selected, choose **Actions**, **Attach volume**. Select the instance ID of the original instance, specify the device name that you noted earlier in Step 2 for the original root device attachment (`/dev/sda1` or `/dev/xvda`), and then choose **Attach volume**.

   > **Important**
   >
   > If you don't specify the same device name as the original attachment, you cannot start the original instance. Amazon EC2 expects the root device volume at `sda1` or `/dev/xvda`.

Step 8: Connect to the original instance using the new key pair

Select the original instance, choose **Instance state**, **Start instance**. After the instance enters the `running` state, you can connect to it using the private key file for your new key pair.

> **Note**
>
> If the name of your new key pair and corresponding private key file is different from the name of the original key pair, ensure that you specify the name of the new private key file when you connect to your instance.

Step 9: Clean up

(Optional) You can terminate the temporary instance if you have no further use for it. Select the temporary instance, and choose **Instance state**, **Terminate (delete) instance**.

Troubleshoot Amazon EC2 Linux instances with failed status checks

The following information can help you troubleshoot issues if your Linux instance fails a status check. First determine whether your applications are exhibiting any problems. If you verify that the instance is not running your applications as expected, review the status check information and the system logs.

For examples of problems that can cause status checks to fail, see Status checks for Amazon EC2 instances.

Contents

- Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)
- XENBUS: Device with no driver...
- ... days without being checked, check forced (File system check required)
- fsck died with exit status... (Missing device)
- GRUB prompt (grubdom>)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)
- XENBUS: Timeout connecting to devices (Xenbus timeout)

Review status check information

To investigate impaired instances using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Select the **Status and alarms** tab to see the individual results for all **System status checks**, **Instance status checks**, and **Attached EBS status checks**.

If a status check has failed, you can try one of the following options:

- Create an alarm to recover the instance in response to the failed status check. For more information, see Create alarms that stop, terminate, reboot, or recover an instance.
- (Instance status checks) If you changed the instance type to a Nitro-based instance, status checks fail if you migrated from an instance that does not have the required ENA and NVMe drivers. For more information, see Compatibility for changing the instance type.
- For an EBS-backed instance, stop and restart the instance. For more information, see Stop and start Amazon EC2 instances.
- For an instance-store backed instance, terminate the instance and launch a replacement instance. For more information, see Terminate Amazon EC2 instances.
- Wait for Amazon EC2 to resolve the issue.

- Contact Support or post your issue to AWS re:Post.
- If your instance is in an Auto Scaling group:
  - (System status checks and instance status checks) By default, Amazon EC2 Auto Scaling automatically launches a replacement instance. For more information, see Health checks for instances in an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.
  - (Attached EBS status checks) You must configure Amazon EC2 Auto Scaling to automatically launch a replacement instance. For more information, see Monitor and replace Auto Scaling instances with impaired Amazon EBS volumes in the *Amazon EC2 Auto Scaling User Guide*.
- Retrieve the system log and look for errors. For more information, see Retrieve the system logs.

Retrieve the system logs

If an instance status check fails, you can reboot the instance and retrieve the system logs. The logs may reveal an error that can help you troubleshoot the issue. Rebooting clears unnecessary information from the logs.

To reboot an instance and retrieve the system log

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and select your instance.
3. Choose **Instance state**, **Reboot instance**. It might take a few minutes for your instance to reboot.
4. Verify that the problem still exists; in some cases, rebooting may resolve the problem.
5. When the instance is in the `running` state, choose **Actions**, **Monitor and troubleshoot**, **Get system log**.
6. Review the log that appears on the screen, and use the list of known system log error statements below to troubleshoot your issue.
7. If your issue is not resolved, you can post your issue to AWS re:Post.

Troubleshoot system log errors for Linux instances

For Linux instances that have failed an instance status check, such as the instance reachability check, verify that you followed the steps above to retrieve the system log. The following list contains some common system log errors and suggested actions you can take to resolve the issue for each error.

Memory Errors

- Out of memory: kill process
- ERROR: mmu_update failed (Memory management update failed)

Device Errors

- I/O error (block device failure)
- I/O ERROR: neither local nor remote disk (Broken distributed block device)

Kernel Errors

- request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)
- "FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)
- "FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)
- ERROR Invalid kernel (EC2 incompatible kernel)

File System Errors

- fsck: No such file or directory while trying to open... (File system not found)
- General error mounting filesystems (failed mount)
- VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)
- Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)
- XENBUS: Device with no driver...
- ... days without being checked, check forced (File system check required)
- fsck died with exit status... (Missing device)

Operating System Errors

- GRUB prompt (grubdom>)

- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)
- XENBUS: Timeout connecting to devices (Xenbus timeout)

Out of memory: kill process

An out-of-memory error is indicated by a system log entry similar to the one shown below.

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

Potential cause

Exhausted memory

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to a non-impaired status. The problem will probably occur again unless you change the instance type.
Instance store-backed	Do one of the following: • Terminate the instance and launch a new instance, specifying a different instance

For this instance type	Do this
	type. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.

ERROR: mmu_update failed (Memory management update failed)

Memory management update failures are indicated by a system log entry similar to the following:

```
...
Press `ESC' to enter the menu... 0   [H[J  Booting 'Amazon Linux 2011.09
 (2.6.35.14-95.38.amzn1.i686)'


root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=

en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22
```

Potential cause

Issue with Amazon Linux

Suggested action

Post your issue to AWS re:Post or contact Support.

I/O error (block device failure)

An input/output error is indicated by a system log entry similar to the following example:

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance.

For this instance type	Do this
	2. Detach the volume. 3. Attempt to recover the volume. **Note** It's good practice to snapshot your Amazon EBS volumes often. This dramatically decreases the risk of data loss as a result of failure. 4. Re-attach the volume to the instance. 5. Start the instance.
Instance store-backed	Terminate the instance and launch a new instance. **Note** Data cannot be recovered. Recover from backups. **Note** It's a good practice to use either Amazon S3 or Amazon EBS for backups. Instance store volumes are directly tied to single host and single disk failures.

I/O ERROR: neither local nor remote disk (Broken distributed block device)

An input/output error on the device is indicated by a system log entry similar to the following example:

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...

Aborting journal on device drbd1-8.

block drbd1: IO ERROR: neither local nor remote disk

Buffer I/O error on device drbd1, logical block 557056

lost page write due to I/O error on drbd1

JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested action

Terminate the instance and launch a new instance.

For an Amazon EBS-backed instance you can recover data from a recent snapshot by creating an image from it. Any data added after the snapshot cannot be recovered.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) can cause an interminable loop condition at startup.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

 Xen: 0000000000000000 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use a newer kernel, either GRUB-based or static, using one of the following options: Option 1: Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters. Option 2: 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel. 3. Start the instance.

For this instance type	Do this
Instance store-backed	Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters.

"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

Incompatible kernel and userland

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Modify the configuration to use a newer kernel. 3. Start the instance.
Instance store-backed	Use the following procedure: 1. Create an AMI that uses a newer kernel. 2. Terminate the instance.

For this instance type	Do this
	3. Start a new instance from the AMI you created.

"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)

This condition is indicated by a system log similar to the one shown below.

```
[    0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or
 directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No
 such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device.  Common problems:
 - Boot args (cat /proc/cmdline)
   - Check rootdelay= (did the system wait long enough?)
   - Check root= (did the system wait for the right device?)
 - Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
```

```
ALERT! /dev/sda1 does not exist. Dropping to a shell!


BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

Potential causes

One or more of the following conditions can cause this problem:

- Missing ramdisk
- Missing correct modules from ramdisk
- Amazon EBS root volume not correctly attached as `/dev/sda1`

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Select corrected ramdisk for the Amazon EBS volume. 2. Stop the instance. 3. Detach the volume and repair it. 4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance with the correct ramdisk. 2. Create a new AMI with the correct ramdisk.

ERROR Invalid kernel (EC2 incompatible kernel)

This condition is indicated by a system log similar to the one shown below.

```
...
root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

Potential causes

One or both of the following conditions can cause this problem:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure:

For this instance type	Do this
	1. Stop the instance. 2. Replace with working kernel. 3. Install a fallback kernel. 4. Modify the AMI by correcting the kernel.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance with the correct kernel. 2. Create an AMI with the correct kernel. 3. (Optional) Seek technical assistance for data recovery using Support.

fsck: No such file or directory while trying to open... (File system not found)

This condition is indicated by a system log similar to the one shown below.

```
  Welcome to Fedora
  Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [  OK  ]

Starting udev: [  OK  ]

Setting hostname localhost:  [  OK  ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[  OK  ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh
```

```
/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]


*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

Potential causes

- A bug exists in ramdisk filesystem definitions /etc/fstab
- Misconfigured filesystem definitions in /etc/fstab
- Missing/failed drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance, detach the root volume, repair/modify /etc/fstab the volume, attach the volume to the instance, and start the instance. 2. Fix ramdisk to include modified /etc/fstab (if applicable). 3. Modify the AMI to use a newer ramdisk.

For this instance type	Do this
	The sixth field in the fstab defines availabil ity requirements of the mount – a nonzero value implies that an fsck will be done on that volume and *must* succeed. Using this field can be problematic in Amazon EC2 because a failure typically results in an interactive console prompt that is not currently available in Amazon EC2. Use care with this feature and read the Linux man page for fstab.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance. 2. Detach any errant Amazon EBS volumes and the reboot instance. 3. (Optional) Seek technical assistance for data recovery using Support.

General error mounting filesystems (failed mount)

This condition is indicated by a system log similar to the one shown below.

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting.  Commit interval 5 seconds
```

```
EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	• Detached or failed Amazon EBS volume. • Corrupted filesystem. • Mismatched ramdisk and AMI combination (such as Debian ramdisk with a SUSE AMI).
Instance store-backed	• A failed drive. • A corrupted file system. • A mismatched ramdisk and combination (for example, a Debian ramdisk with a SUSE AMI).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the root volume. 3. Attach the root volume to a known working instance. 4. Run filesystem check (fsck -a /dev/...). 5. Fix any errors. 6. Detach the volume from the known working instance. 7. Attach the volume to the stopped instance. 8. Start the instance. 9. Recheck the instance status.
Instance store-backed	Try one of the following: • Start a new instance. • (Optional) Seek technical assistance for data recovery using Support.

VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
 20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line:  root=/dev/sda1 ro 4
...
Registering block device major 8
...
```

```
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	• Device not attached correctly. • Root device not attached at correct device point. • Filesystem not in expected format. • Use of legacy kernel (such as 2.6.16-XenU). • A recent kernel update on your instance (faulty update, or an update bug)
Instance store-backed	Hardware device failure.

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Stop and then restart the instance. • Modify root volume to attach at the correct device point, possible /dev/sda1 instead of /dev/sda. • Stop and modify to use modern kernel. • Refer to the documentation for your Linux distribution to check for known update bugs. Change or reinstall the kernel.
Instance store-backed	Terminate the instance and launch a new instance using a modern kernel.

Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)

This condition is indicated by a system log similar to the one shown below.

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda or sda instead of sda1)
- Incorrect choice of instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Fix the device mapping problem.

For this instance type	Do this
	4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: 1. Create a new AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance from the AMI you created.

XENBUS: Device with no driver...

This condition is indicated by a system log similar to the one shown below.

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda)
- Incorrect choice of instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Fix the device mapping problem. 4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: 1. Create an AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance using the AMI you created.

... days without being checked, check forced (File system check required)

This condition is indicated by a system log similar to the one shown below.

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

Potential causes

Filesystem check time passed; a filesystem check is being forced.

Suggested actions

- Wait until the filesystem check completes. A filesystem check can take a long time depending on the size of the root filesystem.
- Modify your filesystems to remove the filesystem check (fsck) enforcement using tune2fs or tools appropriate for your filesystem.

fsck died with exit status... (Missing device)

This condition is indicated by a system log similar to the one shown below.

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

Potential causes

- Ramdisk looking for missing drive
- Filesystem consistency check forced
- Drive failed or detached

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Try one or more of the following to resolve the issue: • Stop the instance, attach the volume to an existing running instance. • Manually run consistency checks. • Fix ramdisk to include relevant utilities.

For this instance type	Do this
	• Modify filesystem tuning parameters to remove consistency requirements (not recommended).
Instance store-backed	Try one or more of the following to resolve the issue: • Rebundle ramdisk with correct tooling. • Modify file system tuning parameters to remove consistency requirements (not recommended). • Terminate the instance and launch a new instance. • (Optional) Seek technical assistance for data recovery using Support.

GRUB prompt (grubdom>)

This condition is indicated by a system log similar to the one shown below.

```
    GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

       [ Minimal BASH-like line editing is supported.   For

         the   first   word,  TAB  lists  possible  command

         completions.  Anywhere else TAB lists the possible

         completions of a device/filename. ]

grubdom>
```

Potential causes

Instance type	Potential causes
Amazon EBS-backed	• Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).
Instance store-backed	• Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Option 1: Modify the AMI and relaunch the instance: 1. Modify the source AMI to create a GRUB configuration file at the standard location (/ boot/grub/menu.lst). 2. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.

For this instance type	Do this
	3. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 4. Terminate the instance and launch a new one using the AMI that you created. Option 2: Fix the existing instance: 1. Stop the instance. 2. Detach the root filesystem. 3. Attach the root filesystem to a known working instance. 4. Mount filesystem. 5. Create a GRUB configuration file. 6. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 7. Detach filesystem. 8. Attach to the original instance. 9. Modify kernel attribute to use the appropria te GRUB image (1st disk or 1st partition on 1st disk). 10Start the instance.

For this instance type	Do this
Instance store-backed	Option 1: Modify the AMI and relaunch the instance: 1. Create the new AMI with a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 3. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 4. Terminate the instance and launch a new instance using the AMI you created. Option 2: Terminate the instance and launch a new instance, specifying the correct kernel. **Note** To recover data from the existing instance, contact Support.

Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)

This condition is indicated by a system log similar to the one shown below.

```
...
Bringing up loopback interface:  [  OK  ]

Bringing up interface eth0:  Device eth0 has different MAC address than expected,
 ignoring.
[FAILED]
```

```
Starting auditd: [  OK  ]
```

Potential causes

There is a hardcoded interface MAC in the AMI configuration

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Modify the AMI to remove the hardcoding and relaunch the instance. • Modify the instance to remove the hardcoded MAC address. OR Use the following procedure: 1. Stop the instance. 2. Detach the root volume. 3. Attach the volume to another instance and modify the volume to remove the hardcoded MAC address. 4. Attach the volume to the original instance. 5. Start the instance.
Instance store-backed	Do one of the following: • Modify the instance to remove the hardcoded MAC address. • Terminate the instance and launch a new instance.

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)

This condition is indicated by a system log similar to the one shown below.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

SELinux has been enabled in error:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the failed instance. 2. Detach the failed instance's root volume. 3. Attach the root volume to another running Linux instance (later referred to as a recovery instance). 4. Connect to the recovery instance and mount the failed instance's root volume. 5. Disable SELinux on the mounted root volume. This process varies across Linux distributions; for more information, consult your OS-specific documentation.

For this instance type	Do this
	Note On some systems, you disable SELinux by setting `SELINUX=disabled` in the `/mount_point/etc/sysconfig/selinux` file, where *mount_point* is the location that you mounted the volume on your recovery instance. 6. Unmount and detach the root volume from the recovery instance and reattach it to the original instance. 7. Start the instance.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance. 2. (Optional) Seek technical assistance for data recovery using Support.

XENBUS: Timeout connecting to devices (Xenbus timeout)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found.  Try passing init= option to kernel.
```

Potential causes

- The block device is not connected to the instance
- This instance is using an old instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Modify the AMI and instance to use a modern kernel and relaunch the instance. • Reboot the instance.
Instance store-backed	Do one of the following: • Terminate the instance. • Modify the AMI to use a modern kernel, and launch a new instance using this AMI.

Troubleshoot an Amazon EC2 Linux instance booting from wrong volume

In some situations, a volume other than the volume attached to `/dev/xvda` or `/dev/sda` becomes the root volume of a Linux instance. This can happen when you have attached the root volume of another instance, or a volume created from the snapshot of a root volume, to an instance with an existing root volume.

This is due to how the initial ramdisk in Linux works. It chooses the volume defined as / in the `/etc/fstab`, and in some distributions, this is determined by the label attached to the volume partition. Specifically, you find that your `/etc/fstab` looks something like the following:

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
```

```
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

If you check the label of both volumes, you see that they both contain the / label:

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

In this example, you could end up having `/dev/xvdf1` become the root device that your instance boots to after the initial ramdisk runs, instead of the `/dev/xvda1` volume from which you had intended to boot. To solve this, use the same **e2label** command to change the label of the attached volume that you do not want to boot from.

In some cases, specifying a UUID in `/etc/fstab` can resolve this. However, if both volumes come from the same snapshot, or the secondary is created from a snapshot of the primary volume, they share a UUID.

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

To change the label of an attached ext4 volume

1. Use the **e2label** command to change the label of the volume to something other than /.

   ```
   [ec2-user ~]$ sudo e2label /dev/xvdf1 old/
   ```

2. Verify that the volume has the new label.

   ```
   [ec2-user ~]$ sudo e2label /dev/xvdf1
   old/
   ```

To change the label of an attached xfs volume

- Use the **xfs_admin** command to change the label of the volume to something other than /.

```
[ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
writing all SBs
new label = "old/"
```

After changing the volume label as shown, you should be able to reboot the instance and have the proper volume selected by the initial ramdisk when the instance boots.

> **Important**
>
> If you intend to detach the volume with the new label and return it to another instance to use as the root volume, you must perform the above procedure again and change the volume label back to its original value. Otherwise, the other instance does not boot because the ramdisk is unable to find the volume with the label /.

Troubleshoot issues connecting to your Amazon EC2 Windows instance

The following information and common errors can help you troubleshoot issues when connecting to your Windows instance.

Connection issues

- Remote Desktop can't connect to the remote computer
- Error using the macOS RDP client
- RDP displays a black screen instead of the desktop
- Unable to remotely log on to an instance with a user that is not an administrator
- Troubleshooting Remote Desktop issues using AWS Systems Manager
- Enable Remote Desktop on an EC2 instance with remote registry
- I've lost my private key. How can I connect to my Windows instance?

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 DNS attributes for 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.
- To connect to your instance using an IPv6 address, check that your local computer has an IPv6 address and is configured to use IPv6. 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 on port 3389.
- 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 Amazon EC2 instances and the section called "Linux instance failed status checks".
- 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. Try one of the following options:
  - Use AWSSupport-TroubleshootRDP to disable the Windows Firewall profiles using SSM Agent. This option requires that the instance is configured for AWS Systems Manager.
  - Use AWSSupport-ExecuteEC2Rescue.
  - Stop the instance, detach the root volume, and attach the volume to a temporary instance as a data volume. Connect to the temporary instance and bring the volume online. Load the registry hive from the data volume. Under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicyStandardProfile`, set `EnableFirewall` to 0. Unload the registry hive and then bring the volume offline. Detach the volume from the temporary instance and attach it to the original instance as the root volume. Start the original instance.
- Verify that Network Level Authentication is disabled on instances that are not part of an Active Directory domain (use AWSSupport-TroubleshootRDP to disable NLA).

- Verify that the Remote Desktop Service (TermService) Startup Type is Automatic and the service is started (use AWSSupport-TroubleshootRDP 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 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 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 the Windows administrator password for an Amazon EC2 Windows instance.
- If you attempt to connect using a user 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 Logon 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 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.
- 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 Amazon EC2 instance type changes.

Unable to remotely log on to an instance with a user that is not an administrator

If you are not able to remotely log on to a Windows instance with a user 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 Managed nodes 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. Choose **Documents** from the left navigation pane.
4. On the **Owned by Amazon** tab, enter `AWSSupport-TroubleshootRDP` in the search field. When the `AWSSupport-TroubleshootRDP` document appears, select it.
5. Choose **Execute automation**.
6. For **Execution Mode**, choose **Simple execution**.
7. For **Input parameters**, **InstanceId**, enable **Show interactive instance picker**.
8. Choose your Amazon EC2 instance.
9. Review the examples, then choose **Execute**.
10. 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=i-1234567890abcdef0, Action=Custom" --region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region#documentVersion=$LATEST
```

Example Example: Disable the Windows Firewall

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=i-1234567890abcdef0, Action=Custom, Firewall=Disable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&Firewall=Disable
```

Example Example: Disable Network Level Authentication

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, NLASettingAction=Disable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion
```

Example Example: Set RDP Service Startup Type to Automatic and start the RDP service

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPServiceStartupType=Auto,
 RDPServiceAction=Start" --region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/
AWSSupport-TroubleshootRDP?region=region_code#documentVersion=
$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example Example: Restore the default RDP Port (3389)

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPPortAction=Modify" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RDPPortAction=Modify
```

Example Example: Allow remote connections

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RemoteConnections=Enable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RemoteConnections=Enable
```

AWSSupport-ExecuteEC2Rescue

The AWSSupport-ExecuteEC2Rescue automation document uses EC2Rescue for Windows Server to automatically troubleshoot and restore EC2 instance connectivity and RDP issues. For more information, see Run the EC2Rescue tool on unreachable instances.

The AWSSupport-ExecuteEC2Rescue automation document requires a stop and restart of the instance. Systems Manager Automation stops the instance and creates an Amazon Machine Image (AMI). Data stored in instance store volumes is lost. The public IP address changes if you are not using an Elastic IP address. For more information, see Run the EC2Rescue tool on unreachable instances in the *AWS Systems Manager User Guide*.

To troubleshoot using the AWSSupport-ExecuteEC2Rescue document

1. Open the Systems Manager console.
2. Verify that you are in the same Region as the impaired Amazon EC2 instance.
3. In the navigation panel, choose **Documents**.

4. Search for and select the `AWSSupport-ExecuteEC2Rescue` document, and then choose **Execute automation**.
5. In **Execution Mode**, choose **Simple execution**.
6. In the **Input parameters** section, for **UnreachableInstanceId**, enter the Amazon EC2 instance ID of the unreachable instance.
7. (Optional) For **LogDestination**, enter the Amazon Simple Storage Service (Amazon S3) bucket name if you want to collect operating system logs for troubleshooting your Amazon EC2 instance. Logs are automatically uploaded to the specified bucket.
8. Choose **Execute**.
9. To monitor the execution progress, in **Execution status**, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose the **Step ID**.

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. Detach the root volume of the unreachable instance and attach it to a reachable instance in the same Availability Zone as a storage volume. If you don't have a reachable instance in the same Availability Zone, launch one. Note the device name of the root volume on the unreachable instance.
3. On the reachable instance, open Disk Management. You can do so by running the following command in the Command Prompt window.

```
diskmgmt.msc
```

4. Right click the newly attached volume that came from the unreachable instance, and then choose **Online**.
5. Open the Windows Registry Editor. You can do so by running the following command in the Command Prompt window.

```
regedit
```

6. In Registry Editor, 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 Registry Editor, navigate to `HKEY_LOCAL_MACHINE\your key name\ControlSet001\Control\Terminal Server`, and then, in the details pane, double-click **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. In Registry Editor, choose **HKEY_LOCAL_MACHINE***your-key-name*, then select **File**, **Unload Hive**.
14. Close Registry Editor and Disk Management.
15. From the EC2 console, detach the volume from the reachable instance and then reattach it to the unreachable instance. When attaching the volume to the unreachable instance, enter the device name that you saved earlier in the **device** field.
16. Restart the unreachable instance.

I've lost my private key. How can I connect to my Windows instance?

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 the Windows administrator password for an Amazon EC2 Windows instance. For steps to reset the password using an Systems Manager document, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

Troubleshoot Amazon EC2 Windows instance start issues

The following are troubleshooting tips to help you solve password and activation issues with Amazon EC2 Windows instances.

Issues

- "Password is not available"
- "Password not available yet"
- "Cannot retrieve Windows password"
- "Waiting for the metadata service"
- "Unable to activate Windows"
- "Windows is not genuine (0x80070005)"
- "No Terminal Server License Servers available to provide a license"
- "Some settings are managed by your organization"

"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.

You can generate a password for the Administrator account for instances launched using a custom Windows AMI. To generate the password, you will need to configure some settings in the operating system before the AMI is created. For more information, see Create an Amazon EBS-backed AMI.

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 the Windows administrator password for an Amazon EC2 Windows instance.

"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 the launch agent for your instance is not configured to generate a password. Verify by checking whether the console output is empty. For more information, see Unable to get console output.

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 Use instance metadata to manage your EC2 instance.

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 and attaching it to the instance.
- **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 2016, launch the temporary instance using the AWS Windows AMI for Windows Server 2019.

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 the section called “Install EC2Config”.

   b. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.

c. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).

d. Select the key that you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe -q` as the data.

e. Select the key again, and from the **File** menu, choose **Unload Hive**.

7. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

> **Warning**
>
> The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see [Configure the Registry](#).

a. Open a command prompt, type **regedit.exe**, and press Enter.

b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.

c. Type **Windows Boot Manager** and then choose **Find Next**.

d. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.

e. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).

f. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
```

```
...
```

g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. Using the **Disk Management** utility, bring the drive offline.

> **Note**
>
> The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

9. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
10. Restore the root volume of the affected instance by attaching the volume as `/dev/sda1`.
11. Start the affected instance.

If you are connected to the instance, open an Internet browser from the instance and enter the following URL for the metadata server:

```
http://169.254.169.254/latest/meta-data/
```

If you can't contact the metadata server, try the following to resolve the issue:

- Download and install the latest version of the EC2Config service. For more information about installing this service, see the section called "Install EC2Config".
- Check whether the Windows instance is running Red Hat PV drivers. If so, update to Citrix PV drivers. For more information, see the section called "Upgrade PV drivers".
- Verify that the firewall, IPSec, and proxy settings do not block outgoing traffic to the metadata service (`169.254.169.254`) or the AWS KMS servers (the addresses are specified in `TargetKMSServer` elements in `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`).
- Verify that you have a route to the metadata service (`169.254.169.254`) using the following command.

  ```
  route print
  ```

- Check for network issues that might affect the Availability Zone for your instance. Go to http://status.aws.amazon.com/.

"Unable to activate Windows"

Windows instances use Windows 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 the section called "Install EC2Config".

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:

```
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:

```
PS C:\> Add-Routes
```

3. Call the Set-ActivationSettings function:

```
PS C:\> Set-Activationsettings
```

4. Then, run the following script to activate Windows:

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

For both EC2Config and EC2Launch, if you are still receiving an activation error, verify the following information.

- Verify that you have routes to the 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.

```
route print
```

- Verify that the AWS KMS client key is set. Run the following command and check the output.

  ```
  C:\Windows\System32\slmgr.vbs /dlv
  ```

  If the output contains Error: product key not found, the 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 activation and product keys](), and then run the following command to set the AWS KMS client key.

  ```
  C:\Windows\System32\slmgr.vbs /ipk client_key
  ```

- Verify that the system has the correct time and time zone. If you are using a time zone other than UTC, add the following registry key and set it to 1 to ensure that the time is correct: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal`.
- If Windows Firewall is enabled, temporarily disable it using the following command.

  ```
  netsh advfirewall set allprofiles state off
  ```

"Windows is not genuine (0x80070005)"

Windows instances use Windows 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"]().

"No Terminal Server License Servers available to provide a license"

By default, Windows Server is licensed for two simultaneous users through Remote Desktop. If you need to provide more than two users with simultaneous access to your Windows instance through Remote Desktop, you can purchase a Remote Desktop Services client access license (CAL) and install the Remote Desktop Session Host and Remote Desktop Licensing Server roles.

Check for the following issues:

- You've exceeded the maximum number of concurrent RDP sessions.

- You've installed the Windows Remote Desktop Services role.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance as a user. You can try the following:
  - Connect to the instance from the command line using an `/admin` parameter, for example:

    ```
    mstsc /v:instance /admin
    ```

    For more information, see the following Microsoft article: [Access Remote Desktop Via Command Line](#).
  - Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

"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."

Troubleshoot issues with Amazon EC2 Windows instances

The following are troubleshooting tips to help you solve the issues with Amazon EC2 Windows instances.

Issues

Unable to connect AWS Systems Manager Sessions Manager to a Windows Server 2025 instance

You may encounter an issue connecting AWS Systems Manager Sessions Manager to a Windows Server 2025 instance. To address this issue, log onto the instance, then navigate to `Settings > Apps > Optional Features`, and add `WMIC`. Restart the SSM Agent service or reboot the instance, and Sessions Manager should connect.

You can also use the following PowerShell command to perform the same action:

```
Start-Process -FilePath "$env:SystemRoot\system32\Dism.exe" -ArgumentList @('/
Online', '/Add-Capability', '/CapabilityName:WMIC~~~~') -Wait; Restart-Service -Name
 AmazonSSMAgent
```

EBS volumes don't initialize on Windows Server 2016 and 2019

Instances created from Amazon Machine Images (AMIs) for Windows Server 2016 and 2019 use the EC2Launch v1 agent for a variety of startup tasks, including initializing EBS volumes. By default, EC2Launch v1 doesn't initialize secondary volumes. However, you can configure EC2Launch v1 to initialize these disks automatically, as follows.

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 as follows:

```
{
"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 v1 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
```

The EC2Launch v1 agent can run instance initialization scripts such as `initializeDisks.ps1` in parallel with the `InitializeInstance.ps1` script. If the `InitializeInstance.ps1` script reboots the instance, it might interrupt other scheduled tasks that run at instance startup. To avoid any potential conflicts, we recommend that you add logic to your `initializeDisks.ps1` script to ensure that instance initialization has finished first.

> **Note**
>
> If the EC2Launch script does not initialize the volumes, ensure that the volumes are online. If the volumes are offline, run the following command to bring all disks online.

```
PS C:\> Get-Disk | Where-Object IsOffline -Eq $True | Set-Disk -IsOffline
 $False
```

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.

Driver Versions	DSRM Supported?	Next Steps
Citrix PV 5.9	No	Restore the instance from a backup. You cannot enable DSRM.
AWS PV 7.2.0	No	Though DSRM is not supported for this driver, you can still detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability Zone as a secondary volume. You can then enable DSRM (as described in this section).
AWS PV 7.2.2 and later	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
Enhanced Networking	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).

For information about how to enable enhanced networking, see the section called “Elastic Network Adapter (ENA)”. For information about upgrading AWS PV drivers, see Upgrade PV drivers on Windows instances.

Configure an instance to boot into DSRM

EC2 Windows instances do not have network connectivity before the operating system is running. For this reason, you cannot press the F8 button on your keyboard to select a boot option. You must use one of the following procedures to boot an EC2 Windows Server instance into DSRM.

If you suspect that Active Directory has been corrupted and the instance is still running, you can configure the instance to boot into DSRM using either the System Configuration dialog box or the command prompt.

To boot an online instance into DSRM using the System Configuration dialog box

1. In the **Run** dialog box, type `msconfig` and press Enter.
2. Choose the **Boot** tab.
3. Under **Boot options** choose **Safe boot**.
4. Choose **Active Directory repair** and then choose **OK**. The system prompts you to reboot the server.

To boot an online instance into DSRM using the command line

From a Command Prompt window, run the following command:

```
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 2016, then choose a Windows Server 2019 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:

```
bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
```

9. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
10. In the EC2 console, detach the affected volume from the temporary instance and reattach it to your original instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
11. Start the instance.
12. After the instance passes the health checks in the EC2 console, connect to the instance using Remote Desktop and verify that it boots into DSRM mode.
13. (Optional) Delete or stop the temporary instance you created in this procedure.

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 the section called “Upgrade PV drivers”.
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`. You can also display event log messages in the console, but this feature might not be enabled by default depending on your version of Windows. For more information, see the section called “Windows launch agents”.

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 the section called “Install EC2Config”.

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.

Disk signature collision

You can check for and resolve disk signature collisions using EC2Rescue for Windows Server. Or, you can manually resolve disk signature issues by performing the following steps.

> **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.

1. Open a command prompt, type **regedit.exe**, and press Enter.
2. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
3. Type **Windows Boot Manager** and then choose **Find Next**.
4. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.
5. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).
6. Locate the four-byte disk signature at offset 0x38 in the data. This is the Boot Configuration Database signature (BCD). 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
...
```

7. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

8. Run the `select disk` DiskPart command and specify the disk number for the volume with the disk signature collision.

> **Tip**
>
> To check the disk number for the volume with the disk signature collision, use the **Disk Management** utility. Open a command prompt, type `compmgmt.msc` and press **Enter**. In the left-hand navigation panel, double-click **Disk Management**. In the **Disk Management** utility, check the disk number for the offline volume with the disk signature collision.

```
DISKPART> select disk 1
Disk 1 is now the selected disk.
```

9. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk
Disk ID: 0C764FA8
```

10. If the disk signature shown in the previous step doesn't match the disk signature that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

Reset the Windows administrator password for an Amazon EC2 Windows instance

If you are no longer able to connect to your Amazon EC2 Windows 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 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 agent, 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 Use EC2Rescue for Windows Server with Systems Manager Run Command.

Contents

Reset Windows admin password for EC2 instance using EC2Launch v2

If you have lost your Windows administrator password and are using a supported Windows AMI that includes the EC2Launch v2 agent, 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 agent, see Reset Windows admin password for EC2 instance using EC2Launch.

If you are using a Windows Server AMI earlier than Windows Server 2016 that does not include the EC2Launch v2 agent, see Reset Windows admin password for EC2 instance using EC2Config.

> **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 Use 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 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 agent is running
- Step 2: Detach the root volume from the instance
- Step 3: Attach the volume to a temporary instance
- Step 4: Delete the .run-once file
- Step 5: Restart the original instance

Step 1: Verify that the EC2Launch v2 agent is running

Before you attempt to reset the administrator password, verify that the EC2Launch v2 agent is installed and running. You use the EC2Launch v2 agent to reset the administrator password later in this section.

To verify that the EC2Launch v2 agent 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 agent is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see Capture a screenshot of an unreachable instance.

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 **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. With the instance selected, choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**.

d. In the navigation pane, choose **AMIs**. Wait for the image status to change to **available**. Then, select the image and choose **Launch instance from AMI**.

e. Complete the fields to launch an instance, making sure to select the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch instance**.

f. When prompted, choose the key pair that you created for the new instance, and then choose **Launch instance**.

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.

5. Detach the root volume from the original instance as follows:

   a. Select the original instance and choose the **Storage** tab. Note the name of the root device under **Root device name**. Find the volume with this device name under **Block devices**, and note the volume ID.

   b. In the navigation pane, choose **Volumes**.

   c. In the list of volumes, select the volume that you noted as the root device, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

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

You must now delete the `.run-once` file from the offline volume attached to the instance. This directs EC2Launch v2 to run all tasks with a frequency of `once`, which includes setting the administrator password. The file path in the secondary volume that you attached will be similar to `D:\ProgramData\Amazon\EC2Launch\state\.run-once`.

To delete the .run-once file

1. Open the **Disk Management** utility, and bring the drive online using these instructions: Make an Amazon EBS volume available for use.
2. Locate the `.run-once` file in the disk you brought online.
3. Delete the .run-once file.

> **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 using RDP.

    > **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 Amazon EC2 instance state changes.

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 Windows admin password for EC2 instance 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 2016 or later AMI that does not include the EC2Launch v2 agent, you can use EC2Launch v2 to generate a new password.

If you are using a Windows Server AMI earlier than Windows Server 2016, see Reset Windows admin password for EC2 instance using EC2Config.

> **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 Use 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 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

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 **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. With the instance selected, choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**.

    d. In the navigation pane, choose **AMIs**. Wait for the image status to change to **available**. Then, select the image and choose **Launch instance from AMI**.

    e. Complete the fields to launch an instance, making sure to select the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch instance**.

    f. When prompted, choose the key pair that you created for the new instance, and then choose **Launch instance**.

    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.

5. Detach the root volume from the original instance as follows:

    a. Select the original instance and choose the **Storage** tab. Note the name of the root device under **Root device name**. Find the volume with this device name under **Block devices**, and note the volume ID.

    b. In the navigation pane, choose **Volumes**.

    c. In the list of volumes, select the volume that you noted as the root device, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

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.

   d. On the **Review Instance Launch** page, choose **Launch**.
   e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:
   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.
   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.
   c. For **Device**, type `xvdf` (if it isn't already there), and choose **Attach**.

Step 3: Reset the administrator password

Next, connect to the temporary instance and use EC2Launch to reset the administrator password.

To reset the administrator password

1. Connect to the temporary instance and use the EC2Rescue for Windows Server tool on the instance to reset the administrator password as follows:
   a. Download the EC2Rescue for Windows Server zip file, extract the contents, and run **EC2Rescue.exe**.
   b. On the **License Agreement** screen, read the license agreement, and, if you accept the terms, choose **I Agree**.
   c. On the **Welcome to EC2Rescue for Windows Server** screen, choose **Next**.
   d. On the **Select mode** screen, choose **Offline instance**.
   e. On the **Select a disk** screen, select the **xvdf** device and choose **Next**.
   f. Confirm the disk selection and choose **Yes**.
   g. After the volume has loaded, choose **OK**.

h. On the **Select Offline Instance Option** screen, choose **Diagnose and Rescue**.

i. On the **Summary** screen, review the information and choose **Next**.

j. On the **Detected possible issues** screen, select **Reset Administrator Password** and choose **Next**.

k. On the **Confirm** screen, choose **Rescue**, **OK**.

l. On the **Done** screen, choose **Finish**.

m. Close the EC2Rescue for Windows Server tool, disconnect from the temporary instance, and then return to the Amazon EC2 console.

2. Detach the secondary (`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 using RDP.
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 Windows admin password for EC2 instance 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 agent to generate a new password.

If you are using a Windows Server 2016 or later AMI, see Reset Windows admin password for EC2 instance using EC2Launch or, you can use the EC2Rescue tool, which uses the EC2Launch service to generate a new password.

> **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 Use 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 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
- Step 2: Detach the root volume from the instance

- [Step 3: Attach the volume to a temporary instance]
- [Step 4: Modify the configuration file]
- [Step 5: Restart the original instance]

Step 1: Verify that the EC2Config service is running

Before you attempt to reset the administrator password, verify that the EC2Config service is installed and running. You use the EC2Config service to reset the administrator password later in this section.

To verify that the EC2Config service is running

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then 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 Capture a screenshot of an unreachable instance.

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 **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. With the instance selected, choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**.

    d. In the navigation pane, choose **AMIs**. Wait for the image status to change to **available**. Then, select the image and choose **Launch instance from AMI**.

    e. Complete the fields to launch an instance, making sure to select the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch instance**.

    f. When prompted, choose the key pair that you created for the new instance, and then choose **Launch instance**.

    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.

5. Detach the root volume from the original instance as follows:

    a. Select the original instance and choose the **Storage** tab. Note the name of the root device under **Root device name**. Find the volume with this device name under **Block devices**, and note the volume ID.

    b. In the navigation pane, choose **Volumes**.

    c. In the list of volumes, select the volume that you noted as the root device, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

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

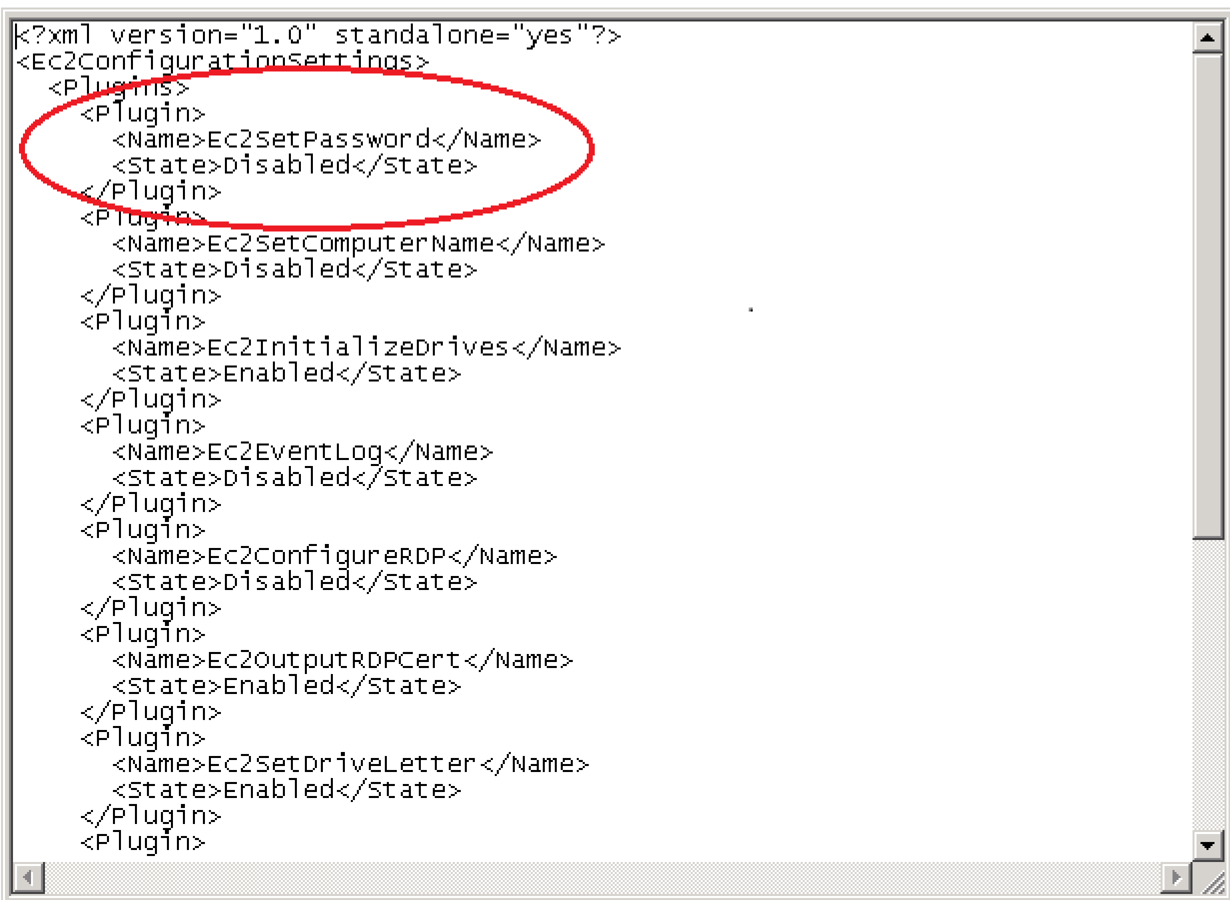
1. From the temporary instance, modify the configuration file on the secondary volume as follows:

   a. Launch and connect to the temporary instance.

   b. Use the following instructions to bring the drive online: [Make 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.



```
<?xml version="1.0" standalone="yes"?>
<Ec2ConfigurationSettings>
  <Plugins>
    <Plugin>
      <Name>Ec2SetPassword</Name>
      <State>Disabled</State>
    </Plugin>
    <Plugin>
      <Name>Ec2SetComputerName</Name>
      <State>Disabled</State>
    </Plugin>
    <Plugin>
      <Name>Ec2InitializeDrives</Name>
      <State>Enabled</State>
    </Plugin>
    <Plugin>
      <Name>Ec2EventLog</Name>
      <State>Disabled</State>
    </Plugin>
    <Plugin>
      <Name>Ec2ConfigureRDP</Name>
      <State>Disabled</State>
    </Plugin>
    <Plugin>
      <Name>Ec2OutputRDPCert</Name>
      <State>Enabled</State>
    </Plugin>
    <Plugin>
      <Name>Ec2SetDriveLetter</Name>
      <State>Enabled</State>
    </Plugin>
    <Plugin>
```

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 using RDP.

   > **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 Amazon EC2 instance state changes.

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 Sysprep issues with Amazon EC2 Windows instances

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, EC2Launch v1, or EC2Launch v2 with Sysprep.

- `%WINDIR%\Panther\Unattendgc` (EC2Config, EC2Launch v1, and EC2Launch v2)
- `%WINDIR%\System32\Sysprep\Panther` (EC2Config, EC2Launch v1, and EC2Launch v2)
- `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 v1 only)
- `%ProgramData%\Amazon\EC2Launch\log\agent.log` (EC2Launch v2 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]
```

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

1. Edit the relevant file, as follows:
   - Windows Server 2012 R2 and earlier – Edit the EC2Config answer file (`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`).
   - Windows Server 2016 and 2019 – Edit the unattend.xml answer file (`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml`).
   - Windows Server 2022 – Edit the unattend.xml answer file (`C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml`).
2. Change `<CopyProfile>true</CopyProfile>` to `<CopyProfile>false</CopyProfile>`.
3. Run Sysprep again. Note that this configuration change will delete the built-in administrator user profile after Sysprep completes.

Troubleshoot impaired Amazon EC2 Linux instance using EC2Rescue

EC2Rescue for Linux is an easy-to-use, open-source tool that can be run on an Amazon EC2 Linux instance to diagnose, troubleshoot, and remediate common issues using its library of over 100 *modules*. Modules are YAML files that contain either a BASH or a Python script and the necessary metadata.

Some generalized use cases for EC2Rescue for Linux instances include:

- Gathering syslog and package manager logs
- Collecting resource utilization data
- Diagnosing and remediating known problematic kernel parameters and common OpenSSH issues

Note

The `AWSSupport-TroubleshootSSH` AWS Systems Manager Automation runbook installs EC2Rescue for Linux and then uses the tool to check or attempt to fix common issues that prevent an SSH connection to a Linux instance. For more information, see AWSSupport-TroubleshootSSH.

If you are using a Windows instance, see the section called “EC2Rescue for Windows instances”.

Topics

Install EC2Rescue on an Amazon EC2 Linux instance

The EC2Rescue for Linux tool can be installed on an Amazon EC2 Linux instance that meets the following prerequisites.

Prerequisites

- Supported operating systems:
  - Amazon Linux 2
  - Amazon Linux 2016.09+
  - SUSE Linux Enterprise Server 12+
  - RHEL 7+
  - Ubuntu 16.04+
- Software requirements:
  - Python 2.7.9+ or 3.2+

Install EC2Rescue

The `AWSSupport-TroubleshootSSH` runbook installs EC2Rescue for Linux and then uses the tool to check or attempt to fix common issues that prevent a remote connection to a Linux machine via SSH. For more information, and to run this automation, see Support-TroubleshootSSH.

If your system has the required Python version, you can install the standard build. Otherwise, you can install the bundled build, which includes a minimal copy of Python.

To install the standard build

1. From a working Linux instance, download the EC2Rescue for Linux tool:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. (*Optional*) Verify the signature of the EC2Rescue for Linux installation file. For more information, see (Optional) Verify the signature of EC2Rescue for Linux.
3. Download the sha256 hash file:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

4. Verify the integrity of the tarball:

```
sha256sum -c ec2rl.tgz.sha256
```

5. Unpack the tarball:

```
tar -xzvf ec2rl.tgz
```

6. Verify the installation by listing out the help file:

```
cd ec2rl-<version_number>
./ec2rl help
```

To install the bundled build

For a link to the download and a list of limitations, see EC2Rescue for Linux on github.

(Optional) Verify the signature of EC2Rescue for Linux

The following is the recommended process of verifying the validity of the EC2Rescue for Linux package for Linux-based operating systems.

When you download an application from the internet, we recommend that you authenticate the identity of the software publisher and check that the application has not been altered or corrupted after it was published. This protects you from installing a version of the application that contains a virus or other malicious code.

If, after running the steps in this topic, you determine that the software for EC2Rescue for Linux is altered or corrupted, do not run the installation file. Instead, contact Amazon Web Services.

EC2Rescue for Linux files for Linux-based operating systems are signed using GnuPG, an open-source implementation of the Pretty Good Privacy (OpenPGP) standard for secure digital signatures. GnuPG (also known as GPG) provides authentication and integrity checking through a digital signature. AWS publishes a public key and signatures that you can use to verify the downloaded EC2Rescue for Linux package. For more information about PGP and GnuPG (GPG), see https://www.gnupg.org/.

The first step is to establish trust with the software publisher. Download the public key of the software publisher, check that the owner of the public key is who they claim to be, and then add the public key to your keyring. Your keyring is a collection of known public keys. After you establish the authenticity of the public key, you can use it to verify the signature of the application.

Tasks

- Authenticate and import the public key
- Verify the signature of the package

Authenticate and import the public key

The next step in the process is to authenticate the EC2Rescue for Linux public key and add it as a trusted key in your GPG keyring.

To authenticate and import the EC2Rescue for Linux public key

1. At a command prompt, use the following command to obtain a copy of our public GPG build key:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.key
```

2. At a command prompt in the directory where you saved `ec2rl.key`, use the following command to import the EC2Rescue for Linux public key into your keyring:

   ```
   gpg2 --import ec2rl.key
   ```

   The command returns results similar to the following:

   ```
   gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
   gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
    imported
   gpg: Total number processed: 1
   gpg:               imported: 1  (RSA: 1)
   ```

   Tip

   If you see an error stating that the command cannot be found, install the GnuPG utility with `apt-get install gnupg2` (Debian-based Linux) or `yum install gnupg2` (Red Hat- based Linux).

Verify the signature of the package

After you've installed the GPG tools, authenticated and imported the EC2Rescue for Linux public key, and verified that the EC2Rescue for Linux public key is trusted, you are ready to verify the signature of the EC2Rescue for Linux installation script.

To verify the EC2Rescue for Linux installation script signature

1. At a command prompt, run the following command to download the signature file for the installation script:

   ```
   curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sig
   ```

2. Verify the signature by running the following command at a command prompt in the directory where you saved `ec2rl.tgz.sig` and the EC2Rescue for Linux installation file. Both files must be present.

```
gpg2 --verify ./ec2rl.tgz.sig
```

The output should look something like the following:

```
gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C  C36A F780 4843 2FAE 2A1C
     Subkey fingerprint: 966B 0D27 85E9 AEEC 1146  7A9D 8851 1153 6991 ED45
```

If the output contains the phrase `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"`, it means that the signature has successfully been verified, and you can proceed to run the EC2Rescue for Linux installation script.

If the output includes the phrase `BAD signature`, check whether you performed the procedure correctly. If you continue to get this response, contact Amazon Web Services and do not run the installation file that you downloaded previously.

The following are details about the warnings that you might see:

- **WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.** This refers to your personal level of trust in your belief that you possess an authentic public key for EC2Rescue for Linux. In an ideal world, you would visit an Amazon Web Services office and receive the key in person. However, more often you download it from a website. In this case, the website is an Amazon Web Services website.
- **gpg2: no ultimately trusted keys found.** This means that the specific key is not "ultimately trusted" by you (or by other people whom you trust).

For more information, see https://www.gnupg.org/.

Run EC2Rescue commands on an Amazon EC2 Linux instance

EC2Rescue is a command line tool. After you have installed EC2Rescue on your Linux instance, you can get general help on how to use the tool by running `./ec2rl help`. You can view the available modules by running `./ec2rl list`, and you can get help on a specific module by running `./ec2rl help module_name`.

The following are common tasks you can perform to get started using this tool.

Tasks

- Run EC2Rescue modules
- Upload the EC2Rescue module results
- Create backups of an Amazon EC2 Linux instance

Run EC2Rescue modules

To run all EC2Rescue modules

Use the **./ec2rl run** command without specifying any additional parameters. Some modules require root access. If you are not a root user, use **sudo** when you run the command.

```
./ec2rl run
```

To run a specific EC2Rescue module

Use the **./ec2rl run** command and for `--only-modules`, specify the name of the module to run. Some modules require *arguments* to use them.

```
./ec2rl run --only-modules=module_name --arguments
```

For example, to run the **dig** module to query the `amazon.com` domain, use the following command.

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

To view the results of an EC2Rescue module

Run the module then view the log file in `cat /var/tmp/ec2rl/logfile_location`. For example, the log file for the **dig** module can be found in the following location:

```
cat /var/tmp/ec2rl/timestamp/mod_out/run/dig.log
```

Upload the EC2Rescue module results

If Support has requested the results for a EC2Rescue module, you can upload the log file using the EC2Rescue tool. You can upload the results either to a location provided by Support or to an Amazon S3 bucket that you own.

To upload results to a location provided by Support

Use the **./ec2rl upload** command. For `--upload-directory`, specify the location of the log file. For `--support-url`, specify the URL provided by Support.

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --support-url="url_provided_by_aws_support"
```

To upload results to an Amazon S3 bucket

Use the **./ec2rl upload** command. For `--upload-directory`, specify the location of the log file. For `--presigned-url`, specify a presigned URL for the S3 bucket. For more information about generating pre-signed URLs for Amazon S3, see Uploading Objects Using Pre-Signed URLs.

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --presigned-url="presigned_s3_url"
```

Create backups of an Amazon EC2 Linux instance

You can use EC2Rescue to backup your Linux instance by creating an AMI or by creating snapshots of it's attached volumes.

To create an AMI

Use the `./ec2rl run` command and for --`backup`, specify `ami`.

```
./ec2rl run --backup=ami
```

To create multi-volume snapshots of all attached volumes

Use the `./ec2rl run` command and for --`backup`, specify `allvolumes`.

```
./ec2rl run --backup=allvolumes
```

To create a snapshot of a specific attached volume

Use the `./ec2rl run` command and for `--backup`, specify the ID of the volume to back up.

```
./ec2rl run --backup=vol-01234567890abcdef
```

Develop EC2Rescue modules for Amazon EC2 Linux instances

Modules are written in YAML, a data serialization standard. A module's YAML file consists of a single document, representing the module and its attributes.

Add module attributes

The following table lists the available module attributes.

Attribute	Description
name	The name of the module. The name should be less than or equal to 18 characters in length.
version	The version number of the module.
title	A short, descriptive title for the module. This value should be less than or equal to 50 characters in length.

helptext	The extended description of the module. Each line should be less than or equal to 75 characters in length. If the module consumes arguments, required or optional, include them in the helptext value.
	
	For example:
	
	`helptext: !!str |`
	`  Collect output from ps for system analysis`
	`  Consumes --times= for number of times to repeat`
	`  Consumes --period= for time period between repetition`

Attribute	Description
placement	The stage in which the module should be run. Supported values: • prediagnostic • run • postdiagnostic
language	The language that the module code is written in. Supported values: • bash • python **Note** Python code must be compatible with both Python 2.7.9+ and Python 3.2+.
remediation	Indicates whether the module supports remediation. Supported values are `True` or `False`. The module defaults to `False` if this is absent, making it an optional attribute for those modules that do not support remediation.
content	The entirety of the script code.
constraint	The name of the object containing the constraint values.

Attribute	Description
domain	A descriptor of how the module is grouped or classified. The set of included modules uses the following domains: • application • net • os • performance
class	A descriptor of the type of task performed by the module. The set of included modules uses the following classes: • collect (collects output from programs) • diagnose (pass/fail based on a set of criteria) • gather (copies files and writes to specific file)
distro	The list of Linux distributions that this module supports. The set of included modules uses the following distributions: • **alami** (Amazon Linux) • **rhel** • **ubuntu** • **suse**
required	The required arguments that the module is consuming from the CLI options.
optional	The optional arguments that the module can use.

Attribute	Description
software	The software executables used in the module. This attribute is intended to specify software that is not installed by default. The EC2Rescue for Linux logic ensures that these programs are present and executable before running the module.
package	The source software package for an executabl e. This attribute is intended to provide extended details on the package with the software, including a URL for downloading or getting further information.
sudo	Indicates whether root access is required to run the module. You do not need to implement sudo checks in the module script. If the value is true, then the EC2Rescue for Linux logic only runs the module when the executing user has root access.
perfimpact	Indicates whether the module can have significant performance impact upon the environment in which it is run. If the value is true and the `--perfimpact=true` argument is not present, then the module is skipped.
parallelexclusive	Specifies a program that requires mutual exclusivity. For example, all modules specifyin g "bpf" run in a serial manner.

Add environment variables

The following table lists the available environment variables.

Environment Variable	Description
EC2RL_CALLPATH	The path to ec2rl.py. This path can be used to locate the lib directory and use vendored Python modules.
EC2RL_WORKDIR	The main tmp directory for the diagnostic tool. Default value: /var/tmp/ec2rl .
EC2RL_RUNDIR	The directory where all output is stored. Default value: /var/tmp/ec2rl/<date&timestamp> .
EC2RL_GATHEREDDIR	The root directory for placing gathered module data. Default value:/var/tmp/ec2rl/<date&timestamp>/mod_out/gathered/ .
EC2RL_NET_DRIVER	The driver in use for the first, alphabetically ordered, non-virtual network interface on the instance. Examples: • xen_netfront • ixgbevf • ena
EC2RL_SUDO	True if EC2Rescue for Linux is running as root; otherwise, false.
EC2RL_VIRT_TYPE	The virtualization type as provided by the instance metadata. Examples:

Environment Variable	Description
	• default-hvm • default-paravirtual
EC2RL_INTERFACES	An enumerated list of interfaces on the system. The value is a string containing names, such as eth0, eth1, etc. This is generated via the functions.bash and is only available for modules that have sourced it.

Use YAML syntax

The following should be noted when constructing your module YAML files:

- The triple hyphen (---) denotes the explicit start of a document.
- The !ec2rlcore.module.Module tag tells the YAML parser which constructor to call when creating the object from the data stream. You can find the constructor inside the module.py file.
- The !!str tag tells the YAML parser to not attempt to determine the type of data, and instead interpret the content as a string literal.
- The pipe character (|) tells the YAML parser that the value is a literal-style scalar. In this case, the parser includes all whitespace. This is important for modules because indentation and newline characters are kept.
- The YAML standard indent is two spaces, which can be seen in the following examples. Ensure that you maintain standard indentation (for example, four spaces for Python) for your script and then indent the entire content two spaces inside the module file.

Example modules

Example one (mod.d/ps.yaml):

```
--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
```

```
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
  Collect output from ps for system analysis
  Requires --times= for number of times to repeat
  Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
      printf "%0.s=" {1..80}
      echo -e "\nERROR: "$BASH_COMMAND" exited with an error on line ${BASH_LINENO[0]}"
      exit 0
  }
  trap error_trap ERR

  # read-in shared function
  source functions.bash
  echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every
 $period seconds."
  for i in $(seq 1 $times); do
      ps auxww
      sleep $period
  done
constraint:
  requires_ec2: !!str False
  domain: !!str performance
  class: !!str collect
  distro: !!str alami ubuntu rhel suse
  required: !!str period times
  optional: !!str
  software: !!str
  sudo: !!str False
  perfimpact: !!str False
  parallelexclusive: !!str
```

Troubleshoot impaired Amazon EC2 Windows instance using EC2Rescue

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. The following are some common issues that EC2Rescue can address:

- Instance connectivity issues due to firewall, Remote Desktop Protocol (RDP), or network interface configuration
- Operating system boot issues due to a stop error, boot loop, or corrupted registry
- Issues that might need advanced log analysis and troubleshooting

EC2Rescue for Windows Server has two different modules:

- A **data collector module** that collects data from all different sources
- 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 2012 and later. When the tool starts, it checks whether it is running on an Amazon EC2 instance.

> **Note**
>
> The `AWSSupport-ExecuteEC2Rescue` AWS Systems Manager Automation 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.

If you are using a Linux instance, see the section called “EC2Rescue for Linux instances”.

Topics

- [Troubleshoot impaired Windows instance with the EC2Rescue GUI]
- [Troubleshoot impaired Windows instance with the EC2Rescue CLI]
- [Troubleshoot impaired Windows instance with EC2Rescue and Systems Manager]

Troubleshoot impaired Windows instance with the EC2Rescue GUI

EC2Rescue for Windows Server can perform the following analysis on **offline instances**:

Option	Description

Diagnose and Rescue	EC2Rescue for Windows Server can detect and address issues with the following service settings:
	
	• 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.
	
	• 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.
	
	• Remote Desktop
	  ◦ **Service Start** - Detects whether the Remote Desktop service is enabled.

Option	Description
	• **Remote Desktop Connections** - Detects whether this is enabled. • **TCP Port** - Detects which port the Remote Desktop service is listening on. • EC2Config (Windows Server 2012 R2 and earlier) • **Installation** - Detects which EC2Config version is installed. • **Service Start** - Detects whether the EC2Config service is enabled. • **Ec2SetPassword** - Generates a new administrator password. • **Ec2HandleUserData** - Allows you to run a user data script on the next boot of the instance. • EC2Launch (Windows Server 2016 and later) • **Installation** - Detects which EC2Launch version is installed. • **Ec2SetPassword** - Generates a new administrator password. • Network Interface • **DHCP Service Startup** - Detects whether the DHCP service is enabled. • **Ethernet detail** - Displays informati on about the network driver version, if detected. • **DHCP on Ethernet** - Detects whether DHCP is enabled.

Option	Description
	• Disk signature status ◦ **Signature on disk** and **Signature on Boot Configuration Database (BCD)** - Detects whether the disk signature and the BCD signature are the same. If the values are different, EC2Rescue attempts to overwrite the disk signature with the signature on BCD.
Restore	Perform one of the following actions: • **Last Known Good Configuration** - Attempts to boot the instance into the last known bootable state. • **Restore registry from backup** - Restores the registry from `\Windows\System32\config\RegBack` .
Capture Logs	Allows you to capture logs on the instance for analysis.

EC2Rescue for Windows Server can collect the following data from **active and offline instances**:

Item	Description
Event Log	Collects application, system, and EC2Config event logs.
Registry	Collects `SYSTEM` and `SOFTWARE` hives.
Windows Update Log	Collects log files generated by Windows Update.

Item	Description
	Note In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.
Sysprep Log	Collects log files generated by the Windows System Preparation tool.
Driver Setup Log	Collects Windows SetupAPI logs (`setupapi.dev.log` and `setupapi.setup.log`).
Boot Configuration	Collects `HKEY_LOCAL_MACHINE\BCD00000000` hive.
Memory Dump	Collects any memory dump files that exist on the instance.
EC2Config File	Collects log files generated by the EC2Config service.
EC2Launch File	Collects log files generated by the EC2Launch scripts.
SSM Agent File	Collects log files generated by SSM Agent and Patch Manager logs.
EC2 ElasticGPUs File	Collects event logs related to elastic GPUs.
ECS	Collects logs related to Amazon ECS.
CloudEndure	Collects log files related to CloudEndure Agent.
AWS Replication Agent for MGN or DRS Log Files	Collects log files related to AWS Application Migration Service or AWS Elastic Disaster Recovery.

EC2Rescue for Windows Server can collect the following additional data from **active instances**:

Item	Description
System Information	Collects MSInfo32.
Group Policy Result	Collects a Group Policy report.

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](https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip) tool and extract the files.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):

   ```
   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.

   > **Note**
   >
   > If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   >  [Net.SecurityProtocolType]::Tls12
   > ```

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):

   ```
   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.

   > **Note**
   >
   > If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:

```
[Net.ServicePointManager]::SecurityProtocol =
 [Net.SecurityProtocolType]::Tls12
```

3. Open the EC2Rescue for Windows Server application and accept the license agreement.
4. Choose **Next**, **Current instance**, **Capture logs**.
5. Select the data items to collect and choose **Collect...**. Read the warning and choose **Yes** to continue.
6. Choose a file name and location for the ZIP file and choose **Save**.
7. After EC2Rescue for Windows Server completes, choose **Open Containing Folder** to view the ZIP file.
8. Choose **Finish**.

Troubleshoot impaired Windows instance with the EC2Rescue CLI

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
- Rescue action

- Restore action

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**.

Log group	Available logs	Description
`all`		Collects all available logs.
`eventlog`	• `'Application'` • `'System'` • `'EC2ConfigService'`	Collects application, system, and EC2Config event logs.
`memory-dump`	• `'Memory Dump File'` • `'Mini Dump Files'`	Collects any memory dump files that exist on the instance.
`ec2config`	• `'Log Files'` • `'Configuration Files'`	Collects log files generated by the EC2Config service.
`ec2launch`	• `'Logs'` • `'Config'`	Collects log files generated by the EC2Launch scripts.
`ssm-agent`	• `'Log Files'` • `'Patch Baseline Logs'` • `'InstanceData'`	Collects log files generated by SSM Agent and Patch Manager logs.
`sysprep`	`'Log Files'`	Collects log files generated by the Windows System Preparation tool.

Log group	Available logs	Description
`driver-setup`	• `'SetupAPI Log Files'` • `'DPInst Log File'` • `'AWS PV Setup Log File'`	Collects Windows SetupAPI logs (`setupapi.dev.log` and `setupapi.setup.log`).
`registry`	• `'SYSTEM'` • `'SOFTWARE'` • `'BCD'`	Collects `SYSTEM` and `SOFTWARE` hives.
`egpu`	• `'Event Log'` • `'System Files'`	Collects event logs related to elastic GPUs.
`boot-config`	`'BCDEDIT Output'`	Collects `HKEY_LOCAL_MACHINE\BCD00000000` hive.
`windows-update`	`'Log Files'`	Collects log files generated by Windows Update. **Note** In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.
`cloudendure`	• `'Migrate Script Logs'` • `'Driver Logs'` • `'CloudEndure File List'`	Collects log files related to CloudEndure Agent.

EC2Rescue for Windows Server can collect the following additional data from **active instances**.

Log group	Available logs	Description
`system-info`	`'MSInfo32 Output'`	Collects `MSInfo32`.
`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:

Service group	Available actions	Description
`all`		
`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.

Service group	Available actions	Description
		• **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 administr ator 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.

Service group	Available actions	Description
`network`	`'DHCP Service Startup'`	Network Interface • **DHCP Service Startup** - Detects whether the DHCP service is enabled.

The following are the available options:

- **/level:<level>** - Optional attribute for the check level that the action should trigger. Allowed values are: `information`, `warning`, `error`, `all`. By default, it is set to `error`.
- **/check-only** - Optional attribute that generates a report but makes no modifications to the offline volume.

> **Note**
>
> If EC2Rescue for Windows Server detects a possible disk signature collision, it corrects the signature after the offline process completes by default, even when you use the `/check-only` option. You must use the `/no-fix-signature` option to prevent the correction.

- **/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:

Service Group	Available Actions	Description
Restore Last Known Good Configuration	`lkgc`	**Last Known Good Configura tion** - Attempts to boot the instance into the last known bootable state.
Restore Windows registry from latest backup	`regback`	**Restore registry from backup** - Restores the registry from `\Windows\System32\config\RegBack` .

The following are the available options:

- **/no-offline**—Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature**—Optional attribute that does not fix a possible disk signature collision after completing the action.

Restore examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Restore last known good configuration on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

Restore the last Windows registry backup on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

Troubleshoot impaired Windows instance with EC2Rescue and Systems Manager

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.

Requirements

To run the **ResetAccess** action, your Amazon EC2 instance must have the a policy attached that grants permissions to write the encrypted password to Parameter Store. After attaching the policy, wait a few minutes before attempting to reset the password of an instance after you have attached this policy to the related IAM role.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    }
  ]
}
```

If you are using a custom KMS key, not the default KMS key, use this policy instead.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<kmskeyid>"
      ]
    }
  ]
}
```

View the JSON for the document

The following procedure describes how to view the JSON for this document.

To view the JSON for the Systems Manager Run Command document

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, expand **Change Management Tools** and choose **Documents**.
3. In the search bar, enter `AWSSupport-RunEC2RescueForWindowsTool`, and then select the `AWSSupport-RunEC2RescueForWindowsTool` document.
4. Choose the **Content** tab.

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 using the AWS CLI, see send-command.

Examples

- Attempt to fix all identified issues on an offline root volume
- Collect logs from the current Amazon EC2 Windows instance
- Reset the local Administrator password

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" --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" --parameters "Command=CollectLogs, Parameters='amzn-s3-demo-bucket'" --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" --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" --parameters "Command=ResetAccess, Parameters=a133dc3c-a2g4-4fc6-a873-6c0720104bf0" --output text
```

Note

In this example, the KMS key is `a133dc3c-a2g4-4fc6-a873-6c0720104bf0`.

EC2 Serial Console for 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.

The serial console can be accessed by using the EC2 console or the AWS CLI.

The serial console is available at no additional cost.

Topics

- Prerequisites for the EC2 Serial Console
- Configure access to the EC2 Serial Console
- Connect to the EC2 Serial Console
- Disconnect from the EC2 Serial Console
- Troubleshoot your Amazon EC2 instance using the EC2 Serial Console

Prerequisites for the EC2 Serial Console

To connect to the EC2 Serial Console and use your chosen tool for troubleshooting, the following prerequisites must be in place:

- AWS Regions
- Wavelength Zones and AWS Outposts
- Local Zones
- Instance types
- Grant access
- Support for browser-based client
- Instance state
- Amazon EC2 Systems Manager
- Configure your chosen troubleshooting tool

AWS Regions

Supported in all AWS Regions except Asia Pacific (Taipei).

Wavelength Zones and AWS Outposts

Not supported.

Local Zones

Supported in all Local Zones.

Instance types

Supported instance types:

- **Linux**
  - All virtualized instances built on the Nitro System.
  - All bare metal instances except:
    - General purpose: `a1.metal`, `mac1.metal`, `mac2.metal`
    - Accelerated computing: `g5g.metal`

  - Memory optimized: `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`
- **Windows**

  All virtualized instances built on the Nitro System. Not supported on bare metal instances.

Grant access

You must complete the configuration tasks to grant access to the EC2 Serial Console. For more information, see Configure access to the EC2 Serial Console.

Support for browser-based client

To connect to the serial console using the browser-based client, 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.

Instance state

Must be `running`.

You can't connect to the serial console if the instance is in the `pending`, `stopping`, `stopped`, `shutting-down`, or `terminated` state.

For more information about the instance states, see Amazon EC2 instance state changes.

Amazon EC2 Systems Manager

If the instance uses Amazon EC2 Systems Manager, then SSM Agent version 3.0.854.0 or later must be installed on the instance. For information about SSM Agent, see Working with SSM Agent in the *AWS Systems Manager User Guide*.

Configure your chosen troubleshooting tool

To troubleshoot your instance via the serial console, you can use GRUB or SysRq on Linux instances, and Special Admin Console (SAC) on Windows instances. Before you can use these tools, you must first perform configuration steps on every instance on which you'll use them.

Use the instructions for your instance's operating system to configure your chosen troubleshooting tool.

(Linux instances) Configure GRUB

To configure GRUB, choose one of the following procedures based on the AMI that was used to launch the instance.

Amazon Linux 2

To configure GRUB on an Amazon Linux 2 instance

1. Connect to your Linux instance using SSH
2. Add or change the following options in `/etc/default/grub`:

   - Set `GRUB_TIMEOUT=1`.
   - Add `GRUB_TERMINAL="console serial"`.
   - Add `GRUB_SERIAL_COMMAND="serial --speed=115200"`.

   The following is an example of `/etc/default/grub`. You might need to change the configuration based on your system setup.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.shell=0"
GRUB_TIMEOUT=1
GRUB_DISABLE_RECOVERY="true"
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

3. Apply the updated configuration by running the following command.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Ubuntu

To configure GRUB on an Ubuntu instance

1. Connect to your instance.
2. Add or change the following options in `/etc/default/grub.d/50-cloudimg-settings.cfg`:

   - Set `GRUB_TIMEOUT=1`.

- Add GRUB_TIMEOUT_STYLE=menu.
- Add GRUB_TERMINAL="console serial".
- Remove GRUB_HIDDEN_TIMEOUT.
- Add GRUB_SERIAL_COMMAND="serial --speed=115200".

The following is an example of /etc/default/grub.d/50-cloudimg-settings.cfg. You might need to change the configuration based on your system setup.

```
# Cloud Image specific Grub settings for Generic Cloud Images
# CLOUD_IMG: This file was created/modified by the Cloud Image build process

# Set the recordfail timeout
GRUB_RECORDFAIL_TIMEOUT=0

# Do not wait on grub prompt
GRUB_TIMEOUT=1
GRUB_TIMEOUT_STYLE=menu

# Set the default commandline
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
 nvme_core.io_timeout=4294967295"

# Set the grub console type
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed 115200"
```

3. Apply the updated configuration by running the following command.

```
[ec2-user ~]$ sudo update-grub
```

RHEL

To configure GRUB on a RHEL instance

1. Connect to your instance.
2. Add or change the following options in /etc/default/grub:

   - Remove GRUB_TERMINAL_OUTPUT.

- Add GRUB_TERMINAL="console serial".
- Add GRUB_SERIAL_COMMAND="serial --speed=115200".

The following is an example of `/etc/default/grub`. You might need to change the configuration based on your system setup.

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 rd.blacklist=nouveau nvme_core.io_timeout=4294967295 crashkernel=auto"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

3. Apply the updated configuration by running the following command.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

For RHEL 9.2 and earlier, use the following command.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

CentOS

For instances that are launched using a CentOS AMI, GRUB is configured for the serial console by default.

The following is an example of `/etc/default/grub`. Your configuration might be different based on your system setup.

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

```
GRUB_CMDLINE_LINUX="console=tty0 crashkernel=auto console=ttyS0,115200"
GRUB_DISABLE_RECOVERY="true"
```

(Linux instances) Configure SysRq

To configure SysRq, you enable the SysRq commands for the current boot cycle. To make the configuration persistent, you can also enable the SysRq commands for subsequent boots.

To enable all SysRq commands for the current boot cycle

1. Connect to your instance.
2. Run the following command.

   ```
   [ec2-user ~]$ sudo sysctl -w kernel.sysrq=1
   ```

   Note

   This setting will clear on the next reboot.

To enable all SysRq commands for subsequent boots

1. Create the file `/etc/sysctl.d/99-sysrq.conf` and open it in your favorite editor.

   ```
   [ec2-user ~]$ sudo vi /etc/sysctl.d/99-sysrq.conf
   ```

2. Add the following line.

   ```
   kernel.sysrq=1
   ```

3. Reboot the instance to apply the changes.

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

4. At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
5. At the `Password` prompt, enter the password, and then press **Enter**.

(Windows instances) Enable SAC and the boot menu

> **Note**
>
> If you enable SAC on an instance, the EC2 services that rely on password retrieval will not work from the Amazon EC2 console. Windows on Amazon EC2 launch agents (EC2Config, EC2Launch v1, and EC2Launch v2) rely on the serial console to execute various tasks. These tasks do not perform successfully when you enable SAC on an instance. For more information about Windows on Amazon EC2 launch agents, see the section called "Configure Windows instances". If you enable SAC, you can disable it later. For more information, see Disable 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 to your instance and perform the following steps from an elevated PowerShell command line.
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
```

Command prompt

To enable SAC and the boot menu on a Windows instance

1. Connect 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
```

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 users. For Linux instances you must also configure a password-based user on every instance so that your users can use the serial console for troubleshooting.

Before commencing, be sure to check the prerequisites.

Contents

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.

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 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.

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.

OS level (Linux instances only)

You can set a user password at the guest OS level. This provides access to the serial console for some use cases. However, to monitor the logs, you don't need a password-based user. For more information, see Set an OS user password on a Linux instance.

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.

Note

This setting is configured at the account level, either directly in the account or by using a declarative policy. It must be configured in each AWS Region where you want to grant access to the serial console. Using a declarative policy allows you to apply the setting across multiple Regions simultaneously, as well as across multiple accounts simultaneously. When a declarative policy is in use, you can't modify the setting directly within an account. This topic describes how to configure the setting directly within an account. For information about using declarative policies, see Declarative policies in the *AWS Organizations User Guide*.

Contents

Grant permission to users to manage account access

To allow your 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

Console

To view account access to the serial 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. On the **EC2 Serial Console** tab, the value of **EC2 Serial Console access** is either **Allowed** or **Prevented**.

AWS CLI

To view account access to the serial console

Use the get-serial-console-access-status command.

```
aws ec2 get-serial-console-access-status
```

The following is example output. A value of `true` indicates that the account is allowed access to the serial console.

```
{
    "SerialConsoleAccessEnabled": true,
    "ManagedBy": "account"
}
```

The `ManagedBy` field indicates the entity that configured the setting. The possible values are `account` (configured directly) or `declarative-policy`. For more information, see Declarative policies in the *AWS Organizations User Guide*.

PowerShell

To view account access to the serial console

Use the Get-EC2SerialConsoleAccessStatus cmdlet.

```
Get-EC2SerialConsoleAccessStatus -Select *
```

The following is example output. A value of `True` indicates that the account is allowed access to the serial console.

```
ManagedBy SerialConsoleAccessEnabled
--------- --------------------------
account   True
```

The `ManagedBy` field indicates the entity that configured the setting. The possible values are `account` (configured directly) or `declarative-policy`. For more information, see Declarative policies in the *AWS Organizations User Guide*.

Grant account access to the serial console

Console

To grant account access to the serial 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 allow access to the EC2 serial console of all instances in the account, select the **Allow** checkbox.
6. Choose **Update**.

AWS CLI

To grant account access to the serial console

Use the enable-serial-console-access command.

```
aws ec2 enable-serial-console-access
```

The following is example output.

```
{
    "SerialConsoleAccessEnabled": true
}
```

PowerShell

To grant account access to the serial console

Use the Enable-EC2SerialConsoleAccess cmdlet.

```
Enable-EC2SerialConsoleAccess
```

The following is example output.

```
True
```

Deny account access to the serial console

Console

To deny account access to the serial 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** checkbox.
6. Choose **Update**.

AWS CLI

To deny account access to the serial console

Use the disable-serial-console-access command.

```
aws ec2 disable-serial-console-access
```

The following is example output.

```
{
    "SerialConsoleAccessEnabled": false
}
```

PowerShell

To deny account access to the serial console

Use the Disable-EC2SerialConsoleAccess cmdlet.

```
Disable-EC2SerialConsoleAccess
```

The following is example output.

```
False
```

Configure IAM policies for EC2 Serial Console access

By default, your users do not have access to the serial console. Your organization must configure IAM policies to grant your 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 a 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 users with this permission can connect to the serial console of all EC2 instances.

Example IAM policies

- Explicitly allow access to the serial console
- Explicitly deny access to the serial console

- Use resource tags to control access to the serial console

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.

Note that the `DescribeInstances`, `DescribeInstanceTypes`, and `GetSerialConsoleAccessStatus` actions do not support resource-level permissions, and therefore all resources, indicated by the * (asterisk), must be specified for these actions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
             "Resource": "*"
        },
        {
            "Sid": "AllowinstanceBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

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",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenySerialConsoleAccess",
            "Effect": "Deny",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

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 a 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*.

Note that the `DescribeInstances`, `DescribeInstanceTypes`, and `GetSerialConsoleAccessStatus` actions do not support resource-level permissions, and therefore all resources, indicated by the `*` (asterisk), must be specified for these actions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowTagBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SerialConsole":
 "${aws:PrincipalTag/SerialConsole}"
                }
            }
        }
    ]
}
```

Set an OS user password on a Linux instance

> **Note**
>
> This section applies to Linux instances only.

You can connect to the serial console without a password. However, to *use* the serial console for troubleshooting a Linux instance, the instance must have a password-based OS user.

You can set the password for any OS user, including the root user. Note that the root user can modify all files, while each OS user might have limited permissions.

You must set a user password for every instance for which you will use the serial console. This is a one-time requirement for each instance.

> **Note**
>
> The following instructions are applicable only if you launched your instance using a Linux AMI provided by AWS because, by default, AMIs provided by AWS are not configured with a password-based user. If you launched your instance using an AMI that already has the root user password configured, you can skip these instructions.

To set an OS user password on a Linux instance

1. Connect to your instance. You can use any method for connecting to your instance, except the EC2 Serial Console connection method.
2. To set the password for a user, use the **passwd** command. In the following example, the user is `root`.

   ```
   [ec2-user ~]$ sudo passwd root
   ```

   The following is example output.

   ```
   Changing password for user root.
   New password:
   ```
3. At the `New password` prompt, enter the new password.

4. At the prompt, re-enter the password.

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 Amazon EC2 instance using the EC2 Serial Console.

Considerations

- Only 1 active serial console connection is supported per instance.
- The serial console connection typically lasts for 1 hour unless you disconnect from it. However, during system maintenance, Amazon EC2 will disconnect the serial console session.

  The duration of the connection is not determined by the duration of your IAM credentials. If your IAM credentials expire, the connection continues to persist until the maximum duration of the serial console connection is reached. When using the EC2 Serial Console console experience, if your IAM credentials expire, terminate the connection by closing the browser page.
- 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 ports: `ttyS0` (Linux instances) and `COM1` (Windows instances)
- When you connect to the serial console, you might observe a slight drop in your instance’s throughput.

Topics

- Connect using the browser-based client
- Connect using your own key and SSH client
- EC2 Serial Console endpoints and fingerprints

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.

Before connecting, make sure you have completed the prerequisites.

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 **Actions**, **Monitor and troubleshoot**, **EC2 Serial Console**, **Connect**.

   Alternatively, select the instance and choose **Connect**, **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.
   - (Linux instances only) **Use SysRq to connect to the serial console.** SysRq does not require that you connect via the browser-based client. For more information, see (Linux instances) Use SysRq to troubleshoot your instance.
   - (Linux instances only) **Restart getty.** If you have SSH access to your instance, then connect to your instance using SSH, and restart getty using the following command.

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```
   - **Reboot your instance.** You can reboot your instance by using SysRq (Linux instances), the EC2 console, or the AWS CLI. For more information, see (Linux instances) Use SysRq to troubleshoot your instance (Linux instances) or Reboot your Amazon EC2 instance.
5. (Linux instances only) At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
6. (Linux instances only) At the `Password` prompt, enter the password, and then press **Enter**.

You are now logged onto the instance and can use the serial console for troubleshooting.

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.

Before connecting, make sure you have completed the prerequisites.

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.

```
aws ec2-instance-connect send-serial-console-ssh-public-key \
    --instance-id i-001234a4bf70dec41EXAMPLE \
    --serial-port 0 \
    --ssh-public-key file://my_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 username format is `instance-id.port0`, which comprises the instance ID and port 0. In the following example, the username is `i-001234a4bf70dec41EXAMPLE.port0`.

   The endpoint of the serial console service is different for each Region. See the EC2 Serial Console endpoints and fingerprints table for each Region's endpoint. In the following example, the serial console service is in the us-east-1 Region.

```
ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-
connect.us-east-1.aws
```

The following example uses `timeout 3600` to set your SSH session to terminate after 1 hour. Processes started during the session may continue running on your instance after the session terminates.

```
timeout 3600 ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-connect.us-east-1.aws
```

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 endpoints and fingerprints.

```
SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
```

> **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.
   - (Linux instances only) **Use SysRq to connect to the serial console.** SysRq does not require that you connect via SSH. For more information, see (Linux instances) Use SysRq to troubleshoot your instance.
   - (Linux instances only) **Restart getty.** If you have SSH access to your instance, then connect to your instance using SSH, and restart getty using the following command.

```
[ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
```

- **Reboot your instance.** You can reboot your instance by using SysRq (Linux instances only), the EC2 console, or the AWS CLI. For more information, see (Linux instances) Use SysRq to troubleshoot your instance (Linux instances only) or Reboot your Amazon EC2 instance.

5. (Linux instances only) At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
6. (Linux instances only) At the `Password` prompt, enter the password, and then press **Enter**.

You are now logged onto the instance and can use the serial console for troubleshooting.

EC2 Serial Console endpoints and fingerprints

The following are the service endpoints and fingerprints for EC2 Serial Console. To connect programmatically to an instance's serial console, you use an EC2 Serial Console endpoint. The EC2 Serial Console endpoints and fingerprints are unique for each AWS Region.

Region Name	Region	Endpoint	Fingerprint
US East (Ohio)	us-east-2	serial-console.ec2-instance-connect.us-east-2.aws	SHA256:EhwPkTzRtTY7TRSzz26XbB0/HvV9jRM7mCZN0xw/d/0
US East (N. Virginia)	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
US West (N. California)	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OHldlcMET8u7QLSX3jmRTRAPFHVtqbyoLZBMUCqiH3Y
US West (Oregon)	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EMCIe23TqKaBI6yGHain

Region Name	Region	Endpoint	Fingerprint
			qZcMwqNkD hhAVHa1O2JxVUc
Africa (Cape Town)	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RM WWZ2fVePe JUqzjO5jL2KIgXsczo Hlz21Ed00biiWI
Asia Pacific (Hong Kong)	ap-east-1	ec2-serial-console.ap-east-1.api.aws	SHA256:T0 Q1lpiXxChoZHplnAkj bP7tkm2xXViC9bJFsj Ynifk
Asia Pacific (Hyderabad)	ap-south-2	ec2-serial-console.ap-south-2.api.aws	SHA256:WJ gPBSwV4/shN +OPITValoewAuYj1 5DVW845JEhDKRs
Asia Pacific (Jakarta)	ap-southeast-3	ec2-serial-console.ap-southeast-3.api.aws	SHA256:5ZwgrCh+lfn s32XITqL/4O0zIfbx4 bZgsYFqy3o8mIk
Asia Pacific (Malaysia)	ap-southeast-5	ec2-serial-console.ap-southeast-5.api.aws	SHA256:cQ XTHQMRcqR dIjmAGoAM BSExeoRob YyRwT67yTjnEiA
Asia Pacific (Melbourne)	ap-southeast-4	ec2-serial-console.ap-southeast-4.api.aws	SHA256:Av aq27hFgLvjn5gTSShZ 0oV7h90p0 GG46wfOeT6ZJvM
Asia Pacific (Mumbai)	ap-south-1	serial-console.ec2-instance-connect.ap-south-1.aws	SHA256:oB LXcYmklqHHEbliARxE gH8IsO51rezTPiSM35 BsU40

Region Name	Region	Endpoint	Fingerprint
Asia Pacific (Osaka)	ap-northeast-3	ec2-serial-console.ap-northeast-3.api.aws	SHA256:Am0/jiBKBnBuFnHr9aXsgEV3G8Tu/vVHFXE/3UcyjsQ
Asia Pacific (Seoul)	ap-northeast-2	serial-console.ec2-instance-connect.ap-northeast-2.aws	SHA256:FoqWXNX+DZ++GuNTztg9PK49WYMqBX+FrcZM2dSrqrI
Asia Pacific (Singapore)	ap-southeast-1	serial-console.ec2-instance-connect.ap-southeast-1.aws	SHA256:PLFNn7WnCQDHx3qmwLu1Gy/O8TUX7LQgZuaC6L45CoY
Asia Pacific (Sydney)	ap-southeast-2	serial-console.ec2-instance-connect.ap-southeast-2.aws	SHA256:yFvMwUK9lEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4
Asia Pacific (Thailand)	ap-southeast-7	ec2-serial-console.ap-southeast-7.api.aws	SHA256:KCAZiRYrR1Q2lqsg7vTwixWmvc2wmjVT31XRgSdEfDY
Asia Pacific (Tokyo)	ap-northeast-1	serial-console.ec2-instance-connect.ap-northeast-1.aws	SHA256:RQfsDCZTOfQawewTRDV1t9Em/HMrFQe+CRlIOT5um4k

Region Name	Region	Endpoint	Fingerprint
Canada (Central)	ca-central-1	serial-console.ec2-instance-connect.ca-central-1.aws	SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gczYMMO7s4
Canada West (Calgary)	ca-west-1	ec2-serial-console.ca-west-1.api.aws	SHA256:s3rc8lI2xhbhr3iedjJNxGAFLPGOLjx7IxxXrGckk6Q
China (Beijing)	cn-north-1	ec2-serial-console.cn-north-1.api.amazonwebservices.com.cn	SHA256:2gHVFy4H7uU3+WaFUxD28v/ggMeqjvSlgngpgLgGT+Y
China (Ningxia)	cn-northwest-1	ec2-serial-console.cn-northwest-1.api.amazonwebservices.com.cn	SHA256:TdgrNZkiQOdVfYEBUhO4SzUA09VWI5rYOZGTogpwmiM
Europe (Frankfurt)	eu-central-1	serial-console.ec2-instance-connect.eu-central-1.aws	SHA256:aCMFS/yIcOdOlkXvOl8AmZ1Toe+bBnrJJ3Fy0k0De2c
Europe (Ireland)	eu-west-1	serial-console.ec2-instance-connect.eu-west-1.aws	SHA256:h2AaGAWO4Hathhtm6ezs3Bj7udgUxi2qTrHjZAwCW6E
Europe (London)	eu-west-2	serial-console.ec2-instance-connect.eu-west-2.aws	SHA256:a69rd5CE/AEG4Amm53I6lkD1ZPvS/BCV3tTPW2RnJg8

Region Name	Region	Endpoint	Fingerprint
Europe (Milan)	eu-south-1	ec2-serial-console.eu-south-1.api.aws	SHA256:lC0kOVJnpgFyBVrxn0A7n99ecLbXSX95cuuS7X7QK30
Europe (Paris)	eu-west-3	serial-console.ec2-instance-connect.eu-west-3.aws	SHA256:q8ldnAf9pymeNe8BnFVngY3RPAr/kxswJUzfrlxeEWs
Europe (Spain)	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:GoCW2DFRlu669QNxqFxEcsR6fZUz/4F4n7T45ZcwoEc
Europe (Stockholm)	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tkGFFUVUDvocDiGSS3Cu8Gdl6w2uI32EPNpKFKLwX84
Europe (Zurich)	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8Ppx2mBMf6WdCw0NUlzKfwM4/IfRz4OaXFutQXWp6mk
Israel (Tel Aviv)	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR6q8v6kNNPi8+QSFQ4dj5dimNmZPTgwgsM1SNvtYyU

Region Name	Region	Endpoint	Fingerprint
Mexico (Central)	mx-central-1	ec2-serial-console.mx-central-1.api.aws	SHA256:BCuVl13iQNk+CcVnt18Ef4p2ZHUrBBAOxlFetB32GS0
Middle East (Bahrain)	me-south-1	ec2-serial-console.me-south-1.api.aws	SHA256:nPjLLKHu2QnLdUq2kVArsoK5xvPJOMRJKCBzCDqC3k8
Middle East (UAE)	me-central-1	ec2-serial-console.me-central-1.api.aws	SHA256:zpb5duKiBZ+l0dFwPeyykB4MPBYhI/XzXNeFSDKBvLE
South America (São Paulo)	sa-east-1	serial-console.ec2-instance-connect.sa-east-1.aws	SHA256:rd2+/32Ognjew1yVIemENaQzC+Botbih62OqAPDq1dI
AWS GovCloud (US-East)	us-gov-east-1	serial-console.ec2-instance-connect.us-gov-east-1.amazonaws.com	SHA256:tIwe19GWsoyLClrtvu38YEEh+DHIkqnDcZnmtebvF28
AWS GovCloud (US-West)	us-gov-west-1	serial-console.ec2-instance-connect.us-gov-west-1.amazonaws.com	SHA256:kfOFRWLaOZfB+utbd3bRf8OlPf8nGO2YZLqXZiIw5DQ

Disconnect from the EC2 Serial Console

If you no longer need to be connected to your instance's EC2 Serial Console, you can disconnect from it. When you disconnect from the serial console, any shell session running on the instance will continue to run. If you want to end the shell session, you'll need to end it before disconnecting from the serial console.

Considerations

- The serial console connection typically lasts for 1 hour unless you disconnect from it. However, during system maintenance, Amazon EC2 will disconnect 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.

The way to disconnect from the serial console depends on the client.

Browser-based client

To disconnect from the serial console, close the serial console in-browser terminal window.

Standard OpenSSH client

To disconnect from the serial console, use the following command to close the SSH connection. This command must be run immediately following a new line.

```
~.
```

The command that you use for closing an SSH connection might be different depending on the SSH client that you're using.

Troubleshoot your Amazon EC2 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.

Use the instructions for your instance's operating system and for the tool you've configured on your instance.

Tools

- (Linux instances) Use GRUB to troubleshoot your instance
- (Linux instances) Use SysRq to troubleshoot your instance
- (Windows instances) Use SAC to troubleshoot your instance

Prerequisites

Before you begin, make sure you have completed the prerequisites, including configuring your chosen troubleshooting tool.

(Linux instances) Use GRUB to troubleshoot your instance

GNU GRUB (short for GNU GRand Unified Bootloader, commonly referred to as GRUB) is the default boot loader for most Linux operating systems. From the GRUB menu, you can select which kernel to boot into, or modify menu entries to change how the kernel will boot. This can be useful when troubleshooting a failing instance.

The GRUB menu is displayed during the boot process. The menu is not accessible via normal SSH, but you can access it via the EC2 Serial Console.

You can boot into single user mode or emergency mode. Single user mode will boot the kernel at a lower runlevel. For example, it might mount the filesystem but not activate the network, giving you the opportunity to perform the maintenance necessary to fix the instance. Emergency mode is similar to single user mode except that the kernel runs at the lowest runlevel possible.

To boot into single user mode

1. Connect to the instance's serial console.
2. Reboot the instance using the following command.

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

3. During reboot, when the GRUB menu appears, press any key to stop the boot process.
4. In the GRUB menu, use the arrow keys to select the kernel to boot into, and press `e` on your keyboard.
5. Use the arrow keys to locate your cursor on the line containing the kernel. The line begins with either `linux` or `linux16` depending on the AMI that was used to launch the instance. For Ubuntu, two lines begin with `linux`, which must both be modified in the next step.
6. At the end of the line, add the word `single`.

   The following is an example for Amazon Linux 2.

```
linux /boot/vmlinuz-4.14.193-149.317.amzn2.aarch64 root=UUID=d33f9c9a-\
dadd-4499-938d-ebbf42c3e499 ro  console=tty0 console=ttyS0,115200n8 net.ifname\
```

```
s=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.she\
ll=0 single
```

7. Press **Ctrl+X** to boot into single user mode.
8. At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
9. At the `Password` prompt, enter the password, and then press **Enter**.

To boot into emergency mode

Follow the the same steps as single user mode, but at step 6, add the word `emergency` instead of `single`.

(Linux instances) Use SysRq to troubleshoot your instance

The System Request (SysRq) key, which is sometimes referred to as "magic SysRq", can be used to directly send the kernel a command, outside of a shell, and the kernel will respond, regardless of what the kernel is doing. For example, if the instance has stopped responding, you can use the SysRq key to tell the kernel to crash or reboot. For more information, see Magic SysRq key in Wikipedia.

You can use SysRq commands in the EC2 Serial Console browser-based client or in an SSH client. The command to send a break request is different for each client.

To use SysRq, choose one of the following procedures based on the client that you are using.

Browser-based client

To use SysRq in the serial console browser-based client

1. Connect to the instance's serial console.
2. To send a break request, press `CTRL+0` (zero). If your keyboard supports it, you can also send a break request using the Pause or Break key.

   ```
   [ec2-user ~]$ CTRL+0
   ```

3. To issue a SysRq command, press the key on your keyboard that corresponds to the required command. For example, to display a list of SysRq commands, press h.

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

The h command outputs something similar to the following.

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

SSH client

To use SysRq in an SSH client

1. Connect to the instance's serial console.
2. To send a break request, press ~B (tilde, followed by uppercase B).

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

3. To issue a SysRq command, press the key on your keyboard that corresponds to the required command. For example, to display a list of SysRq commands, press h.

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

The h command outputs something similar to the following.

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

> **Note**
>
> The command that you use for sending a break request might be different depending on the SSH client that you're using.

(Windows instances) Use SAC to troubleshoot your 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.

> **Note**
>
> If you enable SAC on an instance, the EC2 services that rely on password retrieval will not work from the Amazon EC2 console. Windows on Amazon EC2 launch agents (EC2Config, EC2Launch v1, and EC2Launch v2) rely on the serial console to execute various tasks. These tasks do not perform successfully when you enable SAC on an instance. For more information about Windows on Amazon EC2 launch agents, see the section called "Configure Windows instances". If you enable SAC, you can disable it later. For more information, see Disable SAC and the boot menu.

Tasks

- Use SAC
- Use the boot menu
- Disable SAC and the boot menu

Use SAC

To use SAC

1. Connect to the serial console.

   If SAC is enabled on the instance, the serial console displays the SAC> prompt.

```
Computer is booting, SAC started and initialized.

Use the "ch -?" command for information about using channels.
Use the "?" command for general help.


SAC>?
EVENT: The CMD command is now available.
SAC_
```

2. To display the SAC commands, enter ?, and then press **Enter**.

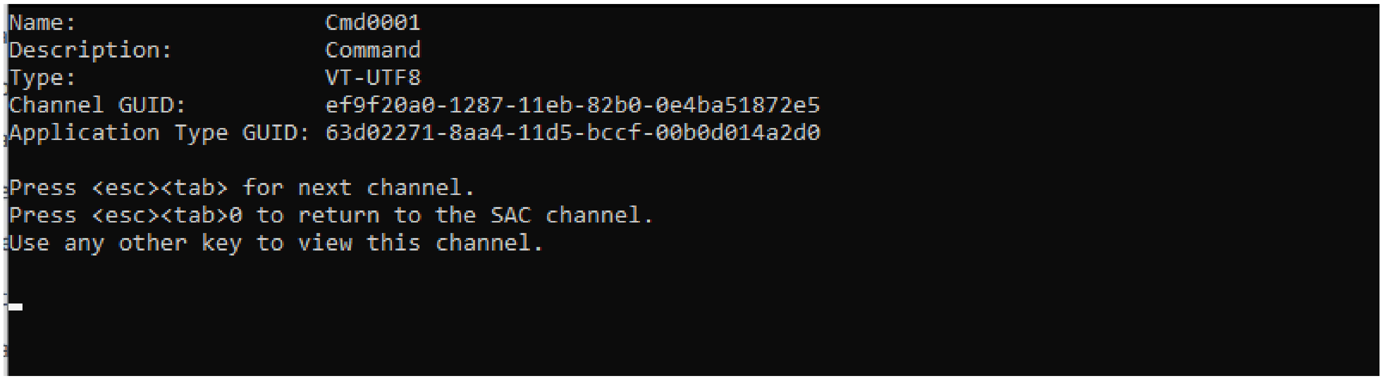
   Expected output

```
SAC>?
ch                    Channel management commands.  Use ch -? for more help.
cmd                   Create a Command Prompt channel.
d                     Dump the current kernel log.
f                     Toggle detailed or abbreviated tlist info.
? or help             Display this list.
i                     List all IP network numbers and their IP addresses.
i <#> <ip> <subnet> <gateway> Set IPv4 addr., subnet and gateway.
id                    Display the computer identification information.
k <pid>               Kill the given process.
l <pid>               Lower the priority of a process to the lowest possible.
lock                  Lock access to Command Prompt channels.
m <pid> <MB-allow>    Limit the memory usage of a process to <MB-allow>.
p                     Toggle paging the display.
r <pid>               Raise the priority of a process by one.
s                     Display the current time and date (24 hour clock used).
s mm/dd/yyyy hh:mm    Set the current time and date (24 hour clock used).
t                     Tlist.
restart               Restart the system immediately.
shutdown              Shutdown the system immediately.
crashdump             Crash the system. You must have crash dump enabled.
```

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**.

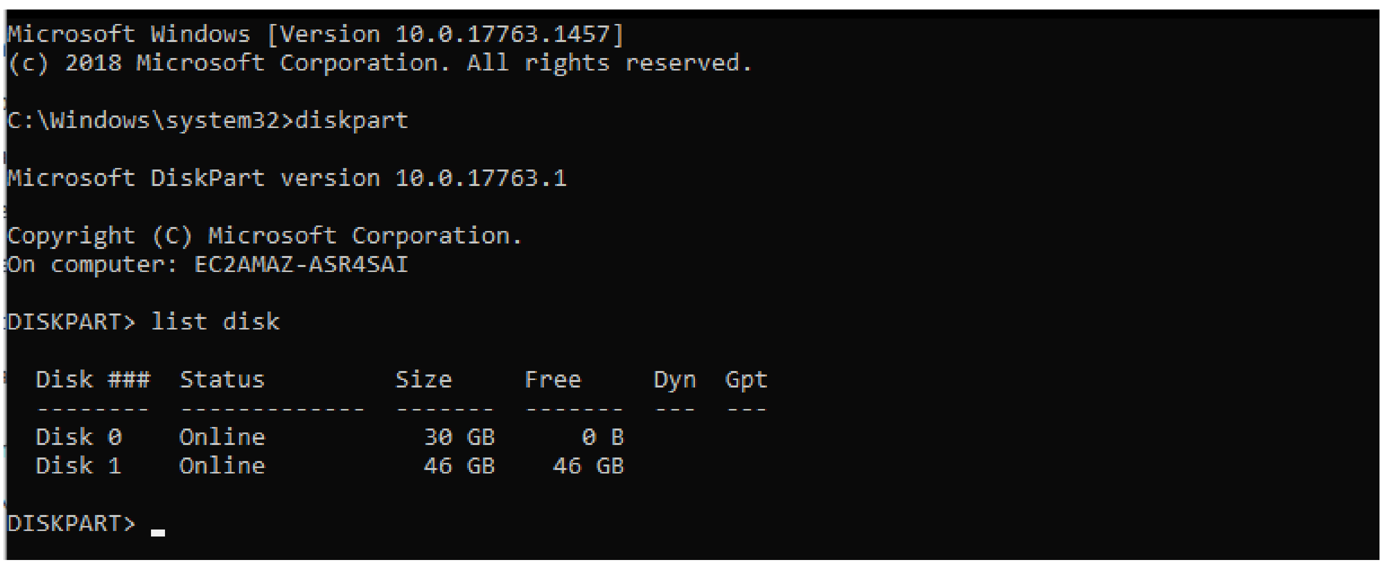
   Expected output



5. 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.

```
PS C:\Windows\system32> $ProgressPreference="SilentlyContinue"
PS C:\Windows\system32> $computerInfo = Get-ComputerInfo
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Name
Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Description
Intel64 Family 6 Model 85 Stepping 4
PS C:\Windows\system32> _
```

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.

```
                          Windows Boot Manager

Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)

    Windows Server [EMS Enabled]                                    >




To specify an advanced option for this choice, press F8.




Tools:

    Windows Memory Diagnostic



 ENTER=Choose                      TAB=Menu                           ESC=Cancel
```

Boot menu commands

ENTER

Starts the selected entry of the operating system.

TAB

Switches to the Tools menu.

ESC

Cancels and restarts the instance.

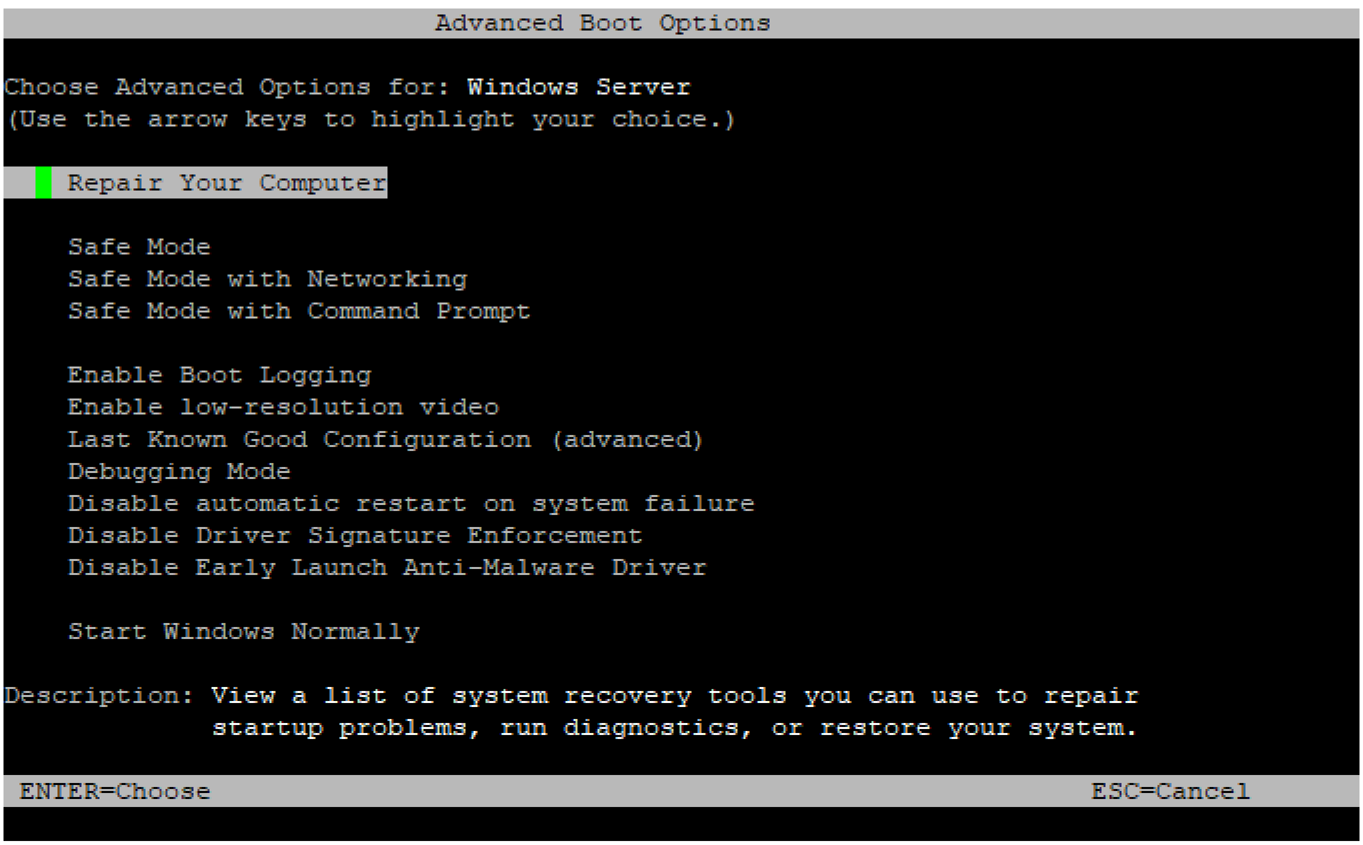
ESC followed by 8

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.



Disable SAC and the boot menu

If you enable SAC and the boot menu, you can disable these features later.

Use one of the following methods to disable SAC and the boot menu on an instance.

PowerShell

To disable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from an elevated PowerShell command line.
2. First disable the boot menu by changing the value to no.

```
bcdedit /set '{bootmgr}' displaybootmenu no
```

3. Then disable SAC by changing the value to `off`.

```
bcdedit /ems '{current}' off
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Command prompt

To disable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from the command prompt.
2. First disable the boot menu by changing the value to no.

```
bcdedit /set {bootmgr} displaybootmenu no
```

3. Then disable SAC by changing the value to `off`.

```
bcdedit /ems {current} off
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Send a diagnostic interrupt to debug an unreachable Amazon EC2 instance

> **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 instance to manually trigger a *kernel panic* for a Linux instance, or a *stop error* (commonly referred to as a *blue screen error*) for a Windows instance.

Linux instances

Linux operating systems typically crash and reboot when a kernel panic occurs. The specific behavior of the operating system depends on its configuration. A kernel panic can also be used to cause the instance's operating system kernel to perform tasks, such as generating a crash dump file. You can then use the information in the crash dump file to conduct root cause analysis and debug the instance. The crash dump data is generated locally by the operating system on the instance itself.

Windows instances

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

Diagnostic interrupt is supported on all Nitro-based instance types, except those powered by AWS Graviton processors. For more information, see instances built on the AWS Nitro System and AWS Graviton.

Prerequisites

Before using a diagnostic interrupt, you must configure your instance's operating system. This ensures that it performs the actions that you need when a kernel panic (Linux instances) or stop error (Windows instances) occurs.

Linux instances

To configure Amazon Linux 2 or Amazon Linux 2023 to generate a crash dump when a kernel panic occurs

1. Connect to your instance.
2. Install **kexec** and **kdump**.

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

3. Configure the kernel to reserve an appropriate amount of memory for the secondary kernel. The amount of memory to reserve depends on the total available memory of your instance. Open the `/etc/default/grub` file using your preferred text editor, locate the line that starts with `GRUB_CMDLINE_LINUX_DEFAULT`, and then add the `crashkernel` parameter in the following format: `crashkernel=`*`memory_to_reserve`*. For example, to reserve 256MB, modify the `grub` file as follows:

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. Save the changes and close the `grub` file.
5. Rebuild the GRUB2 configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Open the `/etc/sysctl.conf` file using your preferred text editor and add the following.

```
kernel.unknown_nmi_panic=1
```

7. Reboot and reconnect to your instance.

8. Verify that the kernel has been booted with the correct `crashkernel` parameter.

```
$ grep crashkernel /proc/cmdline
```

The following example output indicates successful configuration.

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0
```

9. Verify that the **kdump** service is running.

```
[ec2-user ~]$ systemctl status kdump.service
```

The following example output shows the result if the **kdump** service is running.

```
kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
 enabled)
   Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
  Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
 Main PID: 2503 (code=exited, status=0/SUCCESS)
```

Note

By default, the crash dump file is saved to `/var/crash/`. To change the location, modify the `/etc/kdump.conf` file using your preferred text editor.

To configure SUSE Linux Enterprise, Ubuntu, or Red Hat Enterprise Linux

On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI by adjusting the configuration file for your operating system. For information about how to configure the kernel to crash, see the documentation for your operating system:

- SUSE Linux Enterprise
- Ubuntu
- Red Hat Enterprise Linux (RHEL)

Windows instances

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.

AWS CLI

To send a diagnostic interrupt to your instance

Use the send-diagnostic-interrupt command.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

PowerShell

To send a diagnostic interrupt to your instance

Use the Send-EC2DiagnosticInterrupt cmdlet.

```
Send-EC2DiagnosticInterrupt -InstanceId i-1234567890abcdef0
```

Document history for the Amazon EC2 User Guide

The following table describes important additions to the Amazon EC2 User Guide starting in 2019. We also update the guide frequently to address the feedback that you send us.

Change	Description	Date
Amazon EC2 Mac Dedicated Host updates	Amazon EC2 Mac Dedicated Hosts support Dedicated Host auto recovery and reboot-based host maintenance.	August 28, 2025
AMI copies for Local Zones	You can now copy Amazon EBS-backed AMIs to and from Local Zones.	August 28, 2025
AmazonEC2ImageReferencesAccessPolicy – new managed policy	Amazon EC2 added the AmazonEC2ImageReferencesAccessPolicy managed policy to provide permission to scan all resources types supported by the EC2 DescribeImageReferences API.	August 26, 2025
AMI usage reports	If you share your Amazon Machine Images (AMIs) with other AWS accounts—whether with specific AWS accounts, organizations, organizational units (OUs), or publicly—you can see how these AMIs are used by creating AMI usage reports. The reports provide visibility into which AWS accounts are using your AMIs in EC2	August 26, 2025

	instances or launch templates , and how many instances or launch templates are referencing each AMI.	
AMI references	You can identify your AWS resources that reference specified Amazon Machine Images (AMIs), regardless of whether the AMIs are public or private, or who owns them. This visibility helps you ensure your resources use the latest compliant AMIs.	August 26, 2025
Modify EC2 Instance Connect Endpoint	You can now modify an existing EC2 Instance Connect Endpoint using the AWS CLI or an SDK.	August 14, 2025
Capacity Reservation sharing enhancements	You can now share Capacity Reservations that you create in cluster placement groups.	August 12, 2025

Updated managed policy for EC2 Instance Connect Endpoint	To support the modification of existing EC2 Instance Connect Endpoints, Amazon EC2 updated the `Ec2InstanceConnectEndpoint` policy to add permissions to assign and unassign IPv6 addresses and modify security groups on network interfaces created by EC2 Instance Connect Endpoint. Amazon EC2 also updated this policy to replace the `Null` condition operator with the `StringLike` condition operator.	July 31, 2025
Instance topology - additional Region support	The DescribeInstanceTopology API is now available in Asia Pacific (Melbourne).	July 31, 2025
Force terminate	You can now force terminate your instance.	July 31, 2025
Instance topology - additional Region support	The DescribeInstanceTopology API is now available in AWS GovCloud (US-West).	July 24, 2025
Skip OS shutdown during instance termination	You can skip the graceful operating system (OS) shutdown when terminating an instance.	July 23, 2025
Skip OS shutdown during instance stop	You can skip the graceful operating system (OS) shutdown when stopping an instance.	July 23, 2025

P6-B200 Capacity Blocks in us-east-1	P6-B200 Capacity Blocks are now available in US East (N. Virginia) - `us-east-1` .	July 23, 2025
Updated managed policy for the EC2 Fast Launch service linked role	To help prevent orphaned resources, Amazon EC2 updated this policy to add permission to describe volumes, volume attributes and network interfaces, and to delete volumes and net work interfaces that EC2 Fast Launch created.	July 17, 2025
EC2 Serial Console - additiona l Region support	EC2 Serial Console is now also available in Asia Pacific (Thailand) and Mexico (Central).	July 15, 2025
EC2 Instance Connect - additional Region support	EC2 Instance Connect is now also available in Asia Pacific (Thailand) and Mexico (Central).	July 15, 2025
Hardware packet timestamp ing	You can enable hardware packet timestamping on your EC2 instance to add a 64-bit nanosecond-precision timestamp to every incoming network packet.	June 23, 2025
Delete snapshots during AMI deregistration	When deregistering an AMI, you can automatically delete its associated snapshots at the same time.	May 28, 2025

User-initiated reboot migration	When your instance receives an EC2 scheduled reboot event notification, it is enabled by default for instance migration when you reboot the instance before the event.	May 20, 2025
SIP configuration for Amazon EC2 Mac instances	You can configure System Integrity Protection (SIP) settings for x86 Mac instances and Apple silicon Mac instances.	May 20, 2025
Root volume ownership delegation	You can now automate root volume ownership delegatio n for the Amazon EBS root volume when updating the operating system on Apple silicon Mac instances.	May 20, 2025
Instance topology for P6-B200 instances	You can get the instance topology information for `p6-b200.48xlarge` instances.	May 15, 2025

Updated managed policies for the EC2 Fast Launch service linked role	Updated the EC2 Fast Launch service linked role policy with access to create and maintain Amazon EventBridge event rules, and other actions to enhance the EC2 Fast Launch experience. For more information about permissions granted by the service linked role, see Permissions granted by `EC2FastLaunchServiceRolePolicy` .	May 14, 2025
Updated managed policies for the EC2 Fast Launch full access policy	Updated the EC2 Fast Launch full access policy with permission to Create and Modify Launch Template operations, to create network and security group resources , authorize ingress rules, and delete resources that EC2 Fast Launch created. The policy can additionally describe and create CloudFormation stacks, and delete stacks that EC2 Fast Launch created.	May 14, 2025
Simplified onboarding for EC2 Fast Launch	Updated EC2 Fast Launch to simplify the onboarding process.	May 14, 2025

AWS Config integration with Allowed AMIs	Use the ec2-instance-launched-with-allowed-ami AWS Config rule to check if running or stopped instances were launched with AMIs that meet your Allowed AMIs criteria.	March 11, 2025
Updated AWSEC2CapacityReservationFleetRolePolicy managed policy	Amazon EC2 updated the `AWSEC2CapacityReservationFleetRolePolicy` managed policy to use the `ArnLike` condition operator instead of the `StringLike` condition operator.	March 3, 2025
Time-based AMI copies	You can now request a completion duration for EBS-backed AMI copy operations to ensure that AMI copies are completed within a specific timeframe.	February 25, 2025
Updated AmazonEC2ReadOnlyAccess policy	Amazon EC2 added the `GetSecurityGroupsForVpc` operation to the existing AmazonEC2ReadOnlyAccess policy.	December 27, 2024
Managed instances	You can now view Amazon EC2 managed instances in the EC2 console.	December 1, 2024

Declarative policies	You can now use declarative policies to apply account-level settings across multiple Regions and accounts simultaneously. Declarative policies are supported for configuring EC2 serial console access, Allowed AMIs, IMDS defaults, and block public access settings for VPCs, AMIs, and snapshots. For more information, see Declarative policies in the *AWS Organizations User Guide*, as well as the specific documentation for each supported feature.	December 1, 2024
Allowed AMIs	You can now control the discovery and use of AMIs in Amazon EC2 by specifying criteria that AMIs must meet.	December 1, 2024
Instant Capacity Blocks	You can now reserve Capacity Blocks for ML, that can start as soon as 30 minutes.	November 21, 2024
Future-dated Capacity Reservations	You can now request Capacity Reservations for a future date.	November 21, 2024
Extend Capacity Block	You can now extend the duration of existing Capacity Blocks.	November 21, 2024
6 month Capacity Blocks	You can now reserve Capacity Blocks for ML for up to 6 months (182 days).	November 21, 2024

Saved filter sets	You can now create customized groups of filters and reuse them to efficiently view your EC2 resources.	November 20, 2024
Performance protection	When using attribute-based instance type selection for your EC2 Fleet or Spot Fleet, you can now enable performance protection to ensure that the selected instance types are similar to or exceed a specified performance baseline.	November 20, 2024
Capacity Reservations only	You can now specify that instances will only run in a Capacity Reservation or Capacity Reservation resource group.	November 20, 2024
Identify source AMI	You can now identify the source AMI that was used to create an AMI.	November 13, 2024
Split capacity	You can split off capacity from an existing Capacity Reservation to create a new reservation.	October 30, 2024
Move capacity	You can now move capacity from one Capacity Reservation to another.	October 30, 2024

Beginner tutorials	Two new tutorials for beginners: Launch my very first EC2 instance and Launch a test EC2 instance and connect to it.	October 21, 2024
Windows Server 2025 support	Add support for Windows Server 2025.	October 16, 2024
EC2 Global View	EC2 Global View now enables you to view Capacity Reservations and Capacity Blocks in your account across all Regions.	October 16, 2024
Live migration host maintenance	Amazon EC2 Dedicated Hosts now support live migration host maintenance, which automatically migrates sup ported instances from an impaired Dedicated Host to a replacement Dedicated Host without stopping and res tarting them.	October 15, 2024
Billing assignment for shared Capacity Reservations	You can now assign the billing of any available capacity of a shared Capacity Reservati on to a consumer account that belongs to the same AWS organization.	October 14, 2024
PTP hardware clock - additional Region support	The PTP hardware clock is now also available in US East (Ohio) and Asia Pacific (Malaysia).	September 23, 2024

EC2 Instance Connect supports IPv6	You can now use EC2 Instance Connect to connect to the public IPv6 address of your instance.	September 23, 2024
EC2 Instance Connect prefix lists	You can now select a managed prefix list for IPv4 or IPv6 addresses when creating rules in your security groups to allow SSH traffic from the EC2 Instance Connect service.	September 23, 2024
New capabilities to manage On-Demand Capacity Reservations	You can now split your Capacity Reservation, move capacity between Capacity Reservations, and modify the instance eligibility attribute of your Capacity Reservation.	August 14, 2024
Hibernation support for C6g, C6gn, C6gd, C7g, C7gd, M6g, M6gd, M7g, M7gd, R6g, and R6gd	Hibernate your newly-lau nched instances running on C6g, C6gn, C6gd, C7g, C7gd, M6g, M6gd, M7g, M7gd, R6g, and R6gd instance types.	July 30, 2024
Hibernation support for AMIs that support Graviton instance types	Hibernate your newly-lau nched instances that were launched from an Amazon Linux or Ubuntu AMI that supports Graviton instance types.	July 30, 2024
Additional instance types supported for Credential Guard	You can now enable Credentia l Guard for C7i, C7-flex, M7i, M7i-flex, R7i, R7i-flex, and T3 instances.	June 26, 2024

EC2 M1 Ultra Mac instances	New general purpose instance type that features Apple M1 Ultra processors.	June 17, 2024
EC2 instance type finder – additional parameters	The EC2 instance type finder now provides additional parameters for you to specify more detailed requirements for your workload.	June 5, 2024
U7i-12tb, U7in-16tb, U7in-24tb, and U7in-32tb instances	New high memory instance types that feature 4th generation Intel Xeon Sca lable processors.	May 28, 2024
New managed policy for EC2 Fast Launch	Added the EC2FastLa unchFullAccess policy to perform API actions related to the EC2 Fast Launch feature from an instance.	May 14, 2024
AMI deregistration protection	You can turn on deregistr ation protection on an AMI to prevent accidental or malicious deletion.	April 23, 2024
PTP hardware clock – instance type support	The PTP hardware clock is now available on M7a, M7g, M7i, R7a, and R7i instance types.	April 22, 2024
Added Nitro performance considerations for enhanced networking	This page focuses on network considerations to help with performance tuning for your Nitro based Amazon EC2 instances.	April 4, 2024

New managed policy for VSS based EBS snapshots	Amazon EC2 VSS has a new IAM managed policy available that you can add to your ins tance profile role to ensure that your permissions stay up-to-date and follow best practices.	March 28, 2024
PTP hardware clock – US East (N. Virginia)	The PTP hardware clock is now available in the US East (N. Virginia) Region.	March 26, 2024
Set IMDSv2 as the account default	You can set all new EC2 instance launches in your account to use Instance Metadata Service Version 2 (I MDSv2) by default.	March 25, 2024
Tag new Linux AMIs created from a snapshot	When you create a Linux AMI from a snapshot, you can tag the new AMI.	March 7, 2024
Tag new AMIs and snapshots when copying	When you copy an AMI, you can tag the new AMI and the new snapshots with the same tags, or you can tag them with different tags.	March 7, 2024

Remove AWS Management Pack pages	The AWS Management Pack was primarily used with Windows Server 2012 and prior. Those legacy OS platform versions are no longer supported. To manage and troubleshoot your fleet of servers running on AWS and on-premises, see AWS Systems Manager Fleet Manager.	February 12, 2024
EC2 Instance Connect preinstalled on macOS AMIs	EC2 Instance Connect now comes preinstalled on macOS Sonoma 14.2.1 or later, macOS Ventura 13.6.3 or later, and macOS Monterey 12.7.2 or later AMIs.	January 26, 2024
EC2 Instance Connect support for CentOS, macOS, and RHEL	You can now install EC2 Instance Connect on supported CentOS, macOS, and RHEL AMIs.	December 6, 2023
Hibernation support for C7a, C7i, R7a, R7i, and R7iz	Hibernate your newly-lau nched instances running on C7a, C7i, R7a, R7i, and R7iz instance types.	December 1, 2023

[Amazon Q EC2 instance type selector]()	Amazon Q EC2 instance type selector considers your use case, workload type, and CPU manufacturer preference, as well as how you prioritize price and performance. It then uses this data to provide guidance and suggestions for Amazon EC2 instance types that are best suited for your new workloads.	November 28, 2023
[EC2 Free Tier]()	You can track your EC2 Free Tier usage from the EC2 Dashboard.	November 26, 2023
[Console-to-Code]()	Console-to-Code can help you get started with your automation code. Console-to-Code records your console actions, and then uses generative AI to suggest code in your preferred infrastructure-as code format. You can use the code as a starting point, customizing it to make it production-ready for your specific use case.	November 26, 2023

Configurable idle connection tracking timeouts	Security group connections that remain idle can lead to connection tracking exhaustio n and cause connections not to be tracked and packets to be dropped. You can now set the timeout in seconds for security group connection tracking on an Elastic network interface.	November 17, 2023
PTP hardware clock	Supported instances now have a Precision Time Protocol (PTP) hardware clock. The PTP hardware clock supports either NTP or a direct PTP connection.	November 16, 2023
Change instance type of instance enabled for hibernati on	You can now change the instance type of an instance enabled for hibernation when it is in the `stopped` state.	November 16, 2023
Instance topology	You can use the DescribeI nstanceTopology API to detect the location of your instances, and then use this information to optimize HPC and ML jobs by running them on instances that are physicall y closer to each other.	November 13, 2023

EC2 Fast Launch shared AMI support	You can now enable EC2 Fast Launch on an AMI that's shared with you. When you enable EC2 Fast Launch on a shared AMI, the pre-provi sioned snapshots for faster launching are created in your account.	November 6, 2023
Capacity Blocks for ML	You can now reserve GPU instances on a future date to support your short duration machine learning (ML) workloads.	October 31, 2023
Spot Instance hibernation	You can now hibernate your Spot Instances using the same hibernation experience and instance families that are currently available for On-Demand Instances.	October 24, 2023
Default settings for block public access for AMIs	Block public access for AMIs is now enabled by default for all new accounts and for existing accounts with no public AMIs.	October 20, 2023
Amazon EC2 Global View	Amazon EC2 Global View supports additional resource types and customizable display options.	October 18, 2023
Hibernation support for Ubuntu 22.04.2 LTS (Jammy Jellyfish)	Hibernate your newly-lau nched instances that were launched from the Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI.	October 16, 2023

Disable an AMI	You can disable an AMI to prevent it from being used for instance launches.	October 12, 2023
Attached EBS status checks	You can use the attached EBS status checks to monitor if the Amazon EBS volumes attached to an instance are reachable.	October 11, 2023
Hibernation support for Red Hat Enterprise Linux 9	Hibernate your newly-lau nched instances that were launched from the Red Hat Enterprise Linux 9 AMI.	October 2, 2023
Hibernation support for Microsoft Windows Server 2022	Hibernate your newly-lau nched instances that were launched from the Microsoft Windows Server 2022 AMI.	October 2, 2023
Hibernation support for AL2023	Hibernate your newly-lau nched instances that were launched from the AL2023 AMI.	October 2, 2023
Initiate interruption of Spot Instances in a Spot Fleet	You can select a Spot Fleet in the Amazon EC2 console and initiate an interruption of Spot Instances in the fleet so that you can test how the applications on your Spot Instances handle being interrupted.	September 21, 2023
Block public access to AMIs	You can enable *block public access for AMIs* at the account level to block any attempts to make your AMIs public.	September 12, 2023

Hibernation support for M7i and M7i-flex	Hibernate your newly-lau nched instances running on M7i and M7i-flex instance types.	August 22, 2023
EC2-Classic has been deprecated	With EC2-Classic, EC2 instances ran in a single, flat network shared with other customers. Amazon VPC replaces EC2-Classic. With Amazon VPC, your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.	August 8, 2023
Dedicated Hosts	You can allocate Dedicated Hosts on specific hardware assets on an Outpost.	June 20, 2023
EC2 Instance Connect Endpoint	You can now connect to an instance via SSH or RDP without requiring the instance to have a public IPv4 address.	June 13, 2023
IMDS Package Analyzer	You can now use the IMDS Packet Analyzer to identify sources of IMDSv1 calls on your EC2 instances.	June 1, 2023
EC2 Serial Console bare metal instances	The EC2 serial console now supports connectivity to the serial port of selected bare metal instances.	April 11, 2023

Launch template quotas	You can now view your quotas for launch templates and launch template versions in the Service Quotas console and by using the Service Quotas CLI.	April 3, 2023
Capacity Reservation utilizati on notifications	AWS Health now sends notifications when capacity utilization for Capacity Reservations in your account drops below 20 percent.	April 3, 2023
Capacity Reservation groups	You can now add Capacity Reservations that are shared with you to Capacity Reservati on groups that you own.	March 30, 2023
Modify instance metadata options	You can now use the Amazon EC2 console to modify instance metadata options.	March 20, 2023
In-place macOS operating system updates	You can now perform in-place Apple macOS operating system updates on M1 Mac instances.	March 14, 2023
UEFI preferred	You can now create a single AMI that supports both Unified Extensible Firmware Interface (UEFI) and Legacy BIOS boot modes.	March 3, 2023
Modify an AMI for IMDSv2	Modify your existing AMI so that instances launched from the AMI require IMDSv2 by default.	February 28, 2023

Windows Virtualization-based security - Credential Guard	You can enable Credential Guard, a Virtualization-based security (VBS) feature, on supported Amazon EC2 instances.	January 31, 2023
New managed policy for EC2 Instance Connect Endpoint	Added the `Ec2InstanceConnectEndpoint` policy, which is attached to the **AWSServiceRoleForEC2InstanceConnect** service-linked role, to allow Amazon EC2 to perform actions on your behalf when you create an EC2 Instance Connect Endpoint.	January 24, 2023
AMI alias in launch templates	You can specify an AWS Systems Manager parameter instead of the AMI ID in your launch templates to avoid having to update the templates every time the AMI ID changes.	January 19, 2023
Hibernation support for C6i, I3en, and M6i	Hibernate your newly-launched instances running on C6i, I3en, and M6i instance types.	December 19, 2022

Torn write prevention	Improve the performance of your I/O-intensive relational database workloads and reduce latency without negatively impacting data resiliency with torn write prevention, a block storage feature.	November 29, 2022
ENA Express	Increase throughput and minimize tail latency of network traffic between EC2 instances with ENA Express.	November 28, 2022
Copy AMI tags	When you copy an AMI, you can copy your user-defined AMI tags at the same time.	November 18, 2022
AMI size for store and restore	The size of an AMI (before compression) that can be stored and restored to and from an Amazon S3 bucket can now be up to 5,000 GB.	November 16, 2022
priceCapacityOptimized allocation strategy for Spot Instances	A Spot Fleet that uses the `priceCapacityOptimized` allocation strategy looks at both price and capacity to select the Spot Instances pools that are the least likely to be interrupted and have the lowest possible price.	November 10, 2022

price-capacity-optimized allocation strategy for Spot Instances	An EC2 Fleet that uses the `price-capacity-optimized` allocation strategy looks at both price and capacity to select the Spot Instances pools that are the least likely to be interrupted and have the lowest possible price.	November 10, 2022
Cancel having an AMI shared with your account	If an AMI has been shared with your AWS account and you no longer want it shared with your account, you can remove your account from the AMI's launch permissions.	November 4, 2022
Transfer Elastic IP addresses	You can now transfer Elastic IP addresses from one AWS account to another.	October 31, 2022
Replace root volume	You can replace the root Amazon EBS volume for a running instance using an AMI.	October 27, 2022
Automatically connect instance to database	Use the automatic connection feature to quickly connect one or more EC2 instances to an RDS database to allow traffic between them.	October 10, 2022
AMI quotas	Quotas now apply to creating and sharing AMIs.	October 10, 2022

Configure AMI for IMDSv2	Configure your AMI so that instances launched from the AMI require IMDSv2 by default.	October 3, 2022
Initiate Spot Instance interruption	You can select a Spot Instance in the Amazon EC2 console and initiate an interruption so that you can test how the applications on your Spot Instances handle being interrupted.	September 26, 2022
Verified AMI provider	In the Amazon EC2 console, public AMIs that are owned by Amazon or a verified Amazon partner are marked **Verified provider**.	July 22, 2022
Placement groups on AWS Outposts	Added a host spread strategy for placement groups on an Outpost.	June 30, 2022
Dedicated Hosts on AWS Outposts	You can allocate Dedicated Hosts on AWS Outposts.	May 31, 2022
Instance stop protection	To prevent your instance from being accidentally stopped, you can enable stop protectio n for the instance.	May 24, 2022
UEFI Secure Boot	UEFI Secure Boot builds on the long-standing secure boot process of Amazon EC2 and provides additional defense-i n-depth that helps customers secure software from threats that persist across reboots.	May 10, 2022

NitroTPM	Nitro Trusted Platform Module (NitroTPM) is a virtual device that is provided by the AWS Nitro System and conforms to the TPM 2.0 specification.	May 10, 2022
AMI state change events	Amazon EC2 now generates an event when an AMI changes state. You can use Amazon EventBridge to detect and react to these events.	May 9, 2022
Describe public keys	You can query the public key and creation date of an Amazon EC2 key pair.	April 28, 2022
Create key pairs	You can specify the key format (PEM or PPK) when creating a new key pair.	April 28, 2022
Mount Amazon FSx file systems at launch	You can mount a new or existing Amazon FSx for NetApp ONTAP or Amazon FSx for OpenZFS file system at launch using the new launch instance wizard.	April 12, 2022
New launch instance wizard	A new and improved launch experience in the Amazon EC2 console, providing a quicker and easier way to launch an EC2 instance.	April 5, 2022

Automatically deprecate public AMIs	By default, the deprecation date of all public AMIs is set to two years from the AMI creation date.	March 31, 2022
Instance metadata category: autoscaling/target-lifecycle-state	When using Auto Scaling groups, you can access an instance's target lifecycle state from the instance metadata.	March 24, 2022
AMI last launched time	The `lastLaunchedTime` indicates when your AMI was last used to launch an instance.	February 28, 2022
ED25519 keys	ED25519 keys are now supported for EC2 Instance Connect and EC2 Serial Console.	January 20, 2022
Additional RHEL platforms for Capacity Reservations	Additional Red Hat Enterprise Linux platforms for On-Demand Capacity Reservations.	January 11, 2022
Configure Windows AMIs for faster launching	Configure Windows AMIs to launch instances up to 65% faster, using pre-provisioned snapshots.	January 10, 2022
Instance tags in instance metadata	You can access an instance's tags from the instance metadata.	January 6, 2022
Capacity Reservations in cluster placement groups	You can create Capacity Reservations in cluster placement groups.	January 6, 2022

Spot Fleet launch-before-terminate	Spot Fleet can terminate the Spot Instances that receive a rebalance notification after new replacement Spot Instances are launched.	November 4, 2021
EC2 Fleet launch-before-terminate	EC2 Fleet can terminate the Spot Instances that receive a rebalance notification after new replacement Spot Instances are launched.	November 4, 2021
Compare timestamps	You can determine the true time of an event by comparing the timestamp of your Amazon EC2 Linux instance with ClockBound.	November 2, 2021
Share AMIs with organizations and OUs	You can now share AMIs with the following AWS resources : organizations and organizational units (OUs).	October 29, 2021
Spot placement score	Get a recommendation for an AWS Region or Availability Zone based on your Spot capacity requirements.	October 27, 2021
Attribute-based instance type selection for Spot Fleet	Specify the attributes that an instance must have, and Amazon EC2 will identify all the instance types with those attributes.	October 27, 2021

Attribute-based instance type selection for EC2 Fleet	Specify the attributes that an instance must have, and Amazon EC2 will identify all the instance types with those attributes.	October 27, 2021
On-Demand Capacity Reservation Fleet	You can use a Capacity Reservation Fleet to launch a group, or fleet, of Capacity Reservations.	October 5, 2021
Hibernation support for Ubuntu 20.04 LTS - Focal	Hibernate your newly-lau nched instances that were launched from the Ubuntu 20.04 LTS - Focal AMI.	October 4, 2021
EC2 Fleet and targeted On-Demand Capacity Reservati ons	EC2 Fleet can launch On-Demand Instances into `targeted` Capacity Reservati ons.	September 22, 2021
T3 instances on Dedicated Hosts	Support for T3 instances on Amazon EC2 Dedicated Host.	September 14, 2021
Hibernation support for RHEL, Fedora, and CentOS	Hibernate your newly-lau nched instances that were launched from RHEL, Fedora, and CentOS AMIs.	September 9, 2021
Amazon EC2 Global View	Amazon EC2 Global View enables you to view VPCs, subnets, instances, security groups, and volumes across multiple AWS Regions in a single console.	September 1, 2021

Hibernation support for C5d, M5d, and R5d	Hibernate your newly-launched instances running on C5d, M5d, and R5d instance types.	August 19, 2021
Amazon EC2 key pairs	Amazon EC2 now supports ED25519 keys on Linux and Mac instances.	August 17, 2021
Prefixes for network interfaces	You can assign a private IPv4 or IPv6 CIDR range, either automatically or manually, to your network interfaces.	July 22, 2021
Event windows	You can define custom, weekly-recurring event windows for scheduled events that reboot, stop, or terminate your Amazon EC2 instances.	July 15, 2021
Resource IDs and tagging support for security group rules	You can refer to security group rules by resource ID. You can also add tags to your security group rules.	July 7, 2021
Deprecate an AMI	You can now specify when an AMI is deprecated.	June 11, 2021
Windows per-second billing	Amazon EC2 charges for Windows- and SQL Server-based usage by the second, with a one-minute minimum charge.	June 10, 2021
Capacity Reservations on AWS Outposts	You can now use Capacity Reservations on AWS Outposts.	May 24, 2021

Capacity Reservation sharing	You can now share Capacity Reservations created in Local Zones and Wavelength Zones.	May 24, 2021
Root volume replacement	You can now use root volume replacement tasks to replace the root EBS volume for running instances.	April 22, 2021
Store and restore an AMI using S3	Store EBS-backed AMIs in S3 and restore them from S3 to enable cross-partition copying of AMIs.	April 6, 2021
EC2 Serial Console	Troubleshoot boot and network connectivity issues by establishing a connectio n to the serial port of an instance.	March 30, 2021
Boot modes	Amazon EC2 now supports UEFI boot on selected AMD- and Intel-based EC2 instances .	March 22, 2021
Create a reverse DNS record	You can now set up reverse DNS lookup for your Elastic IP addresses.	February 3, 2021
Tag AMIs and snapshots on AMI creation	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.	December 4, 2020

Use Amazon EventBridge to monitor Spot Fleet events	Create EventBridge rules that trigger programmatic actions in response to Spot Fleet state changes and errors.	November 20, 2020
Use Amazon EventBridge to monitor EC2 Fleet events	Create EventBridge rules that trigger programmatic actions in response to EC2 Fleet state changes and errors.	November 20, 2020
Delete instant fleets	Delete an EC2 Fleet of type `instant` and terminate all the instances in the fleet in a single API call.	November 18, 2020
Hibernation support for T3 and T3a	Hibernate your newly-lau nched instances running on T3 and T3a instance types.	November 17, 2020
Amazon EFS Quick Create	You can create and mount an Amazon EFS file system to an instance at launch using Amazon EFS Quick Create.	November 9, 2020
Instance metadata category: events/recommendations/rebalance	The approximate time, in UTC, when the EC2 instance rebalance recommendation notification is emitted for the instance.	November 4, 2020
EC2 instance rebalance recommendation	A signal that notifies you when a Spot Instance is at elevated risk of interruption.	November 4, 2020
Capacity Reservations in Wavelength Zones	Capacity Reservations can now be created and used in Wavelength Zones.	November 4, 2020

Capacity Rebalancing	Configure Spot Fleet or EC2 Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance rec ommendation.	November 4, 2020
Hibernation support for I3, M5ad, and R5ad	Hibernate your newly-lau nched instances running on I3, M5ad, and R5ad instance types.	October 21, 2020
Spot Instance vCPU limits	Spot Instance limits are now managed in terms of the number of vCPUs that your running Spot Instances are either using or will use pending the fulfillment of open requests.	October 1, 2020
Capacity Reservations in Local Zones	Capacity Reservations can now be created and used in Local Zones.	September 30, 2020
Hibernation support for M5a and R5a	Hibernate your newly-lau nched instances running on M5a and R5a instance types.	August 28, 2020
Instance metadata provides instance location and placement information	New instance metadata fields under the `placement` category: Region, placement group name, partition number, host ID, and Availabil ity Zone ID.	August 24, 2020

Capacity Reservation groups	You can use AWS Resource Groups to create logical collections of Capacity Reservations, and then target instance launches into those groups.	July 29, 2020
EC2Launch v2	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.	June 30, 2020
Bring your own IPv6 addresses	You can bring part or all of your IPv6 address range from your on-premises network to your AWS account.	May 21, 2020
Launch instances using a Systems Manager parameter	You can specify a AWS Systems Manager parameter instead of an AMI when you launch an instance.	May 5, 2020
Customize scheduled event notifications	You can customize scheduled event notifications to include tags in the email notification.	May 4, 2020
Windows Server on Dedicated Hosts	You can use Windows Server AMIs provided by Amazon to run the latest versions of Windows Server on Dedicated Hosts.	April 7, 2020

Stop and start a Spot Instance	Stop your Spot Instances backed by Amazon EBS and start them at will, instead of relying on the stop interrupt ion behavior.	January 13, 2020
Resource tagging	You can tag egress-only internet gateways, local gateways, local gateway route tables, local gateway virtual interfaces, local gateway virtual interface groups, local gateway route table VPC associations, and local gateway route table virtual interface group associations.	January 10, 2020
Connect to your instance using Session Manager	You can start a Session Manager session with an instance from the Amazon EC2 console.	December 18, 2019
Dedicated Hosts and host resource groups	Dedicated Hosts can now be used with host resource groups.	December 2, 2019
Dedicated Host sharing	You can now share your Dedicated Hosts across AWS accounts.	December 2, 2019
Default credit specification at the account level	You can set the default credit specification per burstable performance instance family at the account level per AWS Region.	November 25, 2019
Instance type discovery	You can find an instance type that meets your needs.	November 22, 2019

Dedicated Hosts	You can now configure a Dedicated Host to support multiple instance types in an instance family.	November 21, 2019
Instance Metadata Service Version 2	You can use Instance Metadata Service Version 2, which is a session-oriented method for requesting instance metadata.	November 19, 2019
Elastic Fabric Adapter	Elastic Fabric Adapters can now be used with Intel MPI 2019 Update 6.	November 15, 2019
Hibernation support for On-Demand Windows instances	You can hibernate On-Demand Windows instances.	October 14, 2019
Queued purchases of Reserved Instances	You can queue the purchase of a Reserved Instance up to three years in advance.	October 4, 2019
Diagnostic interrupt	You can send a diagnostic interrupt to an unreachable or unresponsive instance to trigger a kernel panic.	August 14, 2019
Capacity optimized allocation strategy	Using EC2 Fleet or Spot Fleet, you can launch Spot Instances from Spot pools with optimal capacity for the number of instances that are launching.	August 12, 2019
On-Demand Capacity Reservation sharing	You can now share your Capacity Reservations across AWS accounts.	July 29, 2019

Elastic Fabric Adapter	EFA now supports Open MPI 3.1.4 and Intel MPI 2019 Update 4.	July 26, 2019
EC2 Instance Connect	EC2 Instance Connect is a simple and secure way to connect to your instances usi ng Secure Shell (SSH).	June 27, 2019
Host recovery	Automatically restart your instances on a new host in the event of an unexpected har dware failure on a Dedicated Host.	June 5, 2019
VSS application-consistent snapshots	Take application-consistent snapshots of all Amazon EBS volumes attached to your Windows instances using AWS Systems Manager Run Command.	May 13, 2019
Windows to Linux Replatfor ming Assistant for Microsoft SQL Server Databases	Move existing Microsoft SQL Server workloads from a Windows to a Linux operating system. Updated link points to the Microsoft SQL Server on Amazon EC2 User Guide.	May 8, 2019
Windows Automated Upgrade	Perform automated upgrades of EC2 Windows instances using AWS Systems Manager.	May 6, 2019
Elastic Fabric Adapter	You can attach an Elastic Fabric Adapter to your instances to accelerate High Performance Computing (HPC) applications.	April 29, 2019

For information about the instance type releases for Amazon EC2, see Document history in the *Amazon EC2 Instance Types Guide*.

History for 2018 and earlier

The following table describes important additions to the Amazon EC2 User Guide in 2018 and earlier years.

Feature	API version	Description	Release date
Partition placement groups	2016-11-15	Partition placement groups spread instances across logical partitions, ensuring that instances in one partition do not share underlying hardware with instances in other partitions. For more information, see Partition placement groups.	20 December 2018
Hibernate EC2 Linux instances	2016-11-15	You can hibernate a Linux instance if it's enabled for hibernation and it meets the hibernation prerequisites. For more information, see Hibernate your Amazon EC2 instance.	28 November 2018
Amazon Elastic Inference Accelerators	2016-11-15	You can attach an Amazon EI accelerator to your instances to add GPU-powered acceleration to reduce the cost of running deep learning inference.	28 November 2018
Spot console recommends a fleet of instances	2016-11-15	The Spot console recommends a fleet of instances based on Spot best practice (instance diversification) to meet the minimum hardware specifications (vCPUs, memory, and storage) for your application need. For more information, see Create a Spot Fleet.	20 November 2018
New EC2 Fleet request type: instant	2016-11-15	EC2 Fleet now supports a new request type, `instant`, that you can use to synchronously provision capacity across instance	14 November 2018

Feature	API version	Description	Release date
		types and purchase models. The `instant` request returns the launched instances in the API response, and takes no further action, enabling you to control if and when instances are launched. For more information, see EC2 Fleet and Spot Fleet request types.	
Spot savings information	2016-11-15	You can view the savings made from using Spot Instances for a single Spot Fleet or for all Spot Instances. For more information, see Savings from purchasing Spot Instances.	5 November 2018
Console support for optimizing CPU options	2016-11-15	When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs using the Amazon EC2 console. For more information, see CPU options for Amazon EC2 instances.	31 October 2018
Console support for creating a launch template from an instance	2016-11-15	You can create a launch template using an instance as the basis for a new launch template using the Amazon EC2 console. For more information, see Create an Amazon EC2 launch template.	30 October 2018
On-Demand Capacity Reservations	2016-11-15	You can reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This allows you to create and manage capacity reservations independently from the billing discounts offered by Reserved Instances (RI). For more information, see Reserve compute capacity with EC2 On-Demand Capacity Reservations.	25 October 2018

Feature	API version	Description	Release date
Bring Your Own IP Addresses (BYOIP)	2016-11-15	You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources. For more information, see Bring your own IP addresses (BYOIP) to Amazon EC2.	23 October 2018
Dedicated Host tag on create and console support	2016-11-15	You can tag your Dedicated Hosts on creation, and you can manage your Dedicated Host tags using the Amazon EC2 console. For more information, see Allocate an Amazon EC2 Dedicated Host for use in your account.	08 October 2018
Console support for scheduled scaling for Spot Fleet	2016-11-15	Increase or decrease the current capacity of the fleet based on the date and time. For more information, see Scheduled scaling: Scale Spot Fleet on a schedule.	20 September 2018
Allocation strategies for EC2 Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity.	26 July 2018

Feature	API version	Description	Release date
Allocation strategies for Spot Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity.	26 July 2018
Automate snapshot lifecycle	2016-11-15	You can use Amazon Data Lifecycle Manager to automate creation and deletion of snapshots for your EBS volumes. For more information, see Amazon Data Lifecycle Manager.	12 July 2018
Launch template CPU options	2016-11-15	When you create a launch template using the command line tools, you can optimize the CPU options to suit specific workloads or business needs. For more information, see Create an Amazon EC2 launch template.	11 July 2018
Tag Dedicated Hosts	2016-11-15	You can tag your Dedicated Hosts.	3 July 2018
Get latest console output	2016-11-15	You can retrieve the latest console output for some instance types when you use the get-console-output AWS CLI command.	9 May 2018
Optimize CPU options	2016-11-15	When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs. For more information, see CPU options for Amazon EC2 instances.	8 May 2018

Feature	API version	Description	Release date
EC2 Fleet	2016-11-15	You can use EC2 Fleet to launch a group of instances across different EC2 instance types and Availability Zones, and across On-Demand Instance, Reserved Instance, and Spot Instance purchasing models. For more information, see EC2 Fleet and Spot Fleet.	2 May 2018
On-Demand Instances in Spot Fleets	2016-11-15	You can include a request for On-Demand capacity in your Spot Fleet request to ensure that you always have instance capacity. For more information, see EC2 Fleet and Spot Fleet.	2 May 2018
Tag EBS snapshots on creation	2016-11-15	You can apply tags to snapshots during creation.	2 April 2018
Change placement groups	2016-11-15	You can move an instance in or out of a placement group, or change its placement group. For more information, see Change the placement for an EC2 instance.	1 March 2018
Longer resource IDs	2016-11-15	You can enable the longer ID format for more resource types.	9 February 2018
Network performance improvements	2016-11-15	Instances outside of a cluster placement group can now benefit from increased bandwidth when sending or receiving network traffic between other instances or Amazon S3.	24 January 2018
Tag Elastic IP addresses	2016-11-15	You can tag your Elastic IP addresses.	21 December 2017

Feature	API version	Description	Release date
Amazon Time Sync Service	2016-11-15	You can use the Amazon Time Sync Service to keep accurate time on your instance. For more information, see Precision clock and time synchronization on your EC2 instance.	29 November 2017
T2 Unlimited	2016-11-15	T2 Unlimited instances can burst above the baseline for as long as required. For more information, see Burstable performance instances.	29 November 2017
Launch templates	2016-11-15	A launch template can contain all or some of the parameters to launch an instance, so that you don't have to specify them every time you launch an instance. For more information, see Store instance launch parameters in Amazon EC2 launch templates.	29 November 2017
Spread placement	2016-11-15	Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. For more information, see Spread placement groups.	29 November 2017
Spot Instance hibernation	2016-11-15	The Spot service can hibernate Spot Instances in the event of an interruption.	28 November 2017
Spot Fleet target tracking	2016-11-15	You can set up target tracking scaling policies for your Spot Fleet. For more information, see Target tracking scaling: Scale Spot Fleet by targeting a value for a specific metric.	17 November 2017
Spot Fleet integrates with Elastic Load Balancing	2016-11-15	You can attach one or more load balancers to a Spot Fleet.	10 November 2017

Feature	API version	Description	Release date
Merge and split Convertible Reserved Instances	2016-11-15	You can exchange (merge) two or more Convertible Reserved Instances for a new Convertible Reserved Instance. You can also use the modification process to split a Convertible Reserved Instance into smaller reservations. For more information, see Exchange Convertible Reserved Instances.	6 November 2017
Modify VPC tenancy	2016-11-15	You can change the instance tenancy attribute of a VPC from `dedicated` to `default`. For more information, see Change the instance tenancy of a VPC.	16 October 2017
Per second billing	2016-11-15	Amazon EC2 charges for Linux-based usage by the second, with a one-minute minimum charge.	2 October 2017
Stop on interruption	2016-11-15	You can specify whether Amazon EC2 should stop or terminate Spot Instances when they are interrupted. For more information, see Behavior of Spot Instance interruptions.	18 September 2017
Tag NAT gateways	2016-11-15	You can tag your NAT gateway. For more information, see Tag your resources.	7 September 2017
Security group rule descriptions	2016-11-15	You can add descriptions to your security group rules.	31 August 2017
Elastic Graphics	2016-11-15	Attach Elastic Graphics accelerators to your instances to accelerate the graphics performance of your applications.	29 August 2017

Feature	API version	Description	Release date
Recover Elastic IP addresses	2016-11-15	If you release an Elastic IP address for use in a VPC, you might be able to recover it.	11 August 2017
Tag Spot Fleet instances	2016-11-15	You can configure your Spot Fleet to automatically tag the instances that it launches.	24 July 2017
Tag resources during creation	2016-11-15	You can apply tags to instances and volumes during creation. For more information, see Tag your resources. 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 Amazon EC2 resources during creation.	28 March 2017
Perform modifications on attached EBS volumes	2016-11-15	With most EBS volumes attached to most EC2 instances, you can modify volume size, type, and IOPS without detaching the volume or stopping the instance.	13 February 2017
Attach an IAM role	2016-11-15	You can attach, detach, or replace an IAM role for an existing instance. For more information, see IAM roles for Amazon EC2.	9 February 2017
Dedicated Spot Instances	2016-11-15	You can run Spot Instances on single-tenant hardware in a virtual private cloud (VPC). For more information, see Launch on single-tenant hardware.	19 January 2017
IPv6 support	2016-11-15	You can associate an IPv6 CIDR with your VPC and subnets, and assign IPv6 addresses to instances in your VPC. For more information, see Amazon EC2 instance IP addressing.	1 December 2016

Feature	API version	Description	Release date
Automatic scaling for Spot Fleet		You can now set up scaling policies for your Spot Fleet. For more information, see Understand automatic scaling for Spot Fleet.	1 September 2016
Elastic Network Adapter (ENA)	2016-04-01	You can now use ENA for enhanced networking. For more information, see Enhanced networking on Amazon EC2 instances.	28 June 2016
Enhanced support for viewing and modifying longer IDs	2016-04-01	You can now view and modify longer ID settings for other IAM users, IAM roles, or the root user.	23 June 2016
Copy encrypted Amazon EBS snapshots between AWS accounts	2016-04-01	You can now copy encrypted EBS snapshots between AWS accounts.	21 June 2016
Capture a screenshot of an instance console	2015-10-01	You can now obtain additional information when debugging instances that are unreachable. For more information, see Capture a screenshot of an unreachable instance.	24 May 2016
Two new EBS volume types	2015-10-01	You can now create Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes.	19 April 2016
Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2		Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2. For more information, see Instance metrics.	23 March 2016
CloudWatch metrics for Spot Fleet		You can now get CloudWatch metrics for your Spot Fleet. For more information, see Monitor your EC2 Fleet or Spot Fleet using CloudWatch.	21 March 2016

Feature	API version	Description	Release date
Scheduled Instances	2015-10-01	Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration.	13 January 2016
Longer resource IDs	2015-10-01	We're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. During the opt-in period, you can enable the longer ID format for supported resource types.	13 January 2016
ClassicLink DNS support	2015-10-01	You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses.	11 January 2016
Dedicated hosts	2015-10-01	An Amazon EC2 Dedicated host is a physical server with instance capacity dedicated for your use. For more information, see Amazon EC2 Dedicated Hosts.	23 November 2015
Spot Instance duration	2015-10-01	You can now specify a duration for your Spot Instances. Spot blocks is not supported (January 2023).	6 October 2015
Spot Fleet modify request	2015-10-01	You can now modify the target capacity of your Spot Fleet request. For more information, see Modify a Spot Fleet request.	29 September 2015

Feature	API version	Description	Release date
Spot Fleet diversified allocation strategy	2015-04-15	You can now allocate Spot Instances in multiple Spot pools using a single Spot Fleet request. For more information, see Use allocation strategies to determine how EC2 Fleet or Spot Fleet fulfills Spot and On-Demand capacity.	15 September 2015
Spot Fleet instance weighting	2015-04-15	You can now define the capacity units that each instance type contributes to your application's performance, and adjust the amount you are willing to pay for Spot Instances for each Spot pool accordingly. For more information, see Use instance weighting to manage cost and performance of your EC2 Fleet or Spot Fleet.	31 August 2015
New reboot alarm action and new IAM role for use with alarm actions		Added the reboot alarm action and new IAM role for use with alarm actions. For more information, see Create alarms that stop, terminate, reboot, or recover an instance.	23 July 2015
Spot Fleets	2015-04-15	You can manage a collection, or fleet, of Spot Instances instead of managing separate Spot Instance requests. For more information, see EC2 Fleet and Spot Fleet.	18 May 2015
Migrate Elastic IP addresses to EC2-Classic	2015-04-15	You can migrate an Elastic IP address that you've allocated for use in EC2-Classic to be used in a VPC..	15 May 2015
Importing VMs with multiple disks as AMIs	2015-03-01	The VM Import process now supports importing VMs with multiple disks as AMIs. For more information, see Importing a VM as an Image Using VM Import/Export in the *VM Import/Export User Guide* .	23 April 2015

Feature	API version	Description	Release date
Systems Manager		Systems Manager enables you to configure and manage your EC2 instances.	17 February 2015
Systems Manager for Microsoft SCVMM 1.5		You can now use Systems Manager for Microsoft SCVMM to launch an instance and to import a VM from SCVMM to Amazon EC2.	21 January 2015
Automatic recovery for EC2 instances		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 Automatic instance recovery.	12 January 2015
ClassicLink	2014-10-01	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.	7 January 2015

Feature	API version	Description	Release date
Spot Instance termination notices		The best way to protect against Spot Instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot Instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot Instance. For more information, see Spot Instance interruption notices.	5 January 2015
Systems Manager for Microsoft SCVMM		Systems Manager for Microsoft SCVMM provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM.	29 October 2014
`DescribeVolumes` pagination support	2014-09-01	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*.	23 October 2014
Added support for Amazon CloudWatch Logs		You can use Amazon CloudWatch Logs to monitor, store, and access your system, application, and custom log files from your instances or other sources. You can then retrieve the associated log data from CloudWatch Logs using the Amazon CloudWatch console, the CloudWatch Logs commands in the AWS CLI, or the CloudWatch Logs SDK.	10 July 2014

Feature	API version	Description	Release date
New **EC2 Service Limits** page		Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.	19 June 2014
Amazon EBS General Purpose SSD Volumes	2014-05-01	General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose SSD volumes can range in size from 1 GiB to 1 TiB.	16 June 2014
AWS Management Pack		AWS Management Pack now supports for System Center Operations Manager 2012 R2.	22 May 2014
Amazon EBS encryption	2014-05-01	Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using 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.	21 May 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2.	28 January 2014

Feature	API version	Description	Release date
Importing Linux virtual machines	2013-10-15	The VM Import process now supports the importation of Linux instances. For more information, see the VM Import/Export User Guide.	16 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Identity and access management for Amazon EC2.	20 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launch an Amazon EC2 instance from an AWS Marketplace AMI.	11 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launch an EC2 instance using the launch instance wizard in the console.	10 October 2013
Modifying instance types of Reserved Instances	2013-10-01	You can now modify the instance type of Linux Reserved Instances within the same family (for example, M1, M2, M3, C1). For more information, see Modify Reserved Instances.	09 October 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a Region. For more information, see Modify Reserved Instances.	11 September 2013

Feature	API version	Description	Release date
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assign a public IPv4 address at launch.	20 August 2013
Granting resource-level permissions	2013-06-15	Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see Identity-based policies for Amazon EC2.	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies.	11 June 2013
AWS Management Pack		The AWS Management Pack links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager.	8 May 2013
New **Tags** page		There is a new **Tags** page in the Amazon EC2 console. For more information, see Tag your Amazon EC2 resources.	04 April 2013
Copy an AMI from one Region to another	2013-02-01	You can copy an AMI from one Region to another, enabling you to launch consistent instances in more than one AWS Region quickly and easily. For more information, see Copy an Amazon EC2 AMI.	11 March 2013

Feature	API version	Description	Release date
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either EC2-Classic or a VPC, or only into a VPC, on a region-by-region basis. If you can launch instances only into a VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.	11 March 2013
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs).	17 December 2012
Updated EBS metrics and status checks for Provisioned IOPS SSD volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS SSD volumes. Also added new status checks for Provisioned IOPS SSD volumes.	20 November 2012
Spot Instance request status	2012-10-01	Spot Instance request status makes it easy to determine the state of your Spot requests.	14 October 2012
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012

Feature	API version	Description	Release date
Provisioned IOPS SSD for Amazon EBS	2012-07-20	Provisioned IOPS SSD volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times.	31 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services.	11 June 2012

Feature	API version	Description	Release date
Spot Instance features that make it easier to get started and handle the potential of interruption.		You can now manage your Spot Instances as follows: • 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 [Request Spot Instances](#) in the *Amazon EC2 Auto Scaling User Guide*. • Get notifications when instances are launched or terminated. • Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources.	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for exporting Windows Server instances that you originally imported into EC2. Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	25 May 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012

Feature	API version	Description	Release date
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic network interfaces.	21 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitor the status of your Amazon EC2 instances.	16 November 2011
Spot Instances in Amazon VPC	2011-07-15	Added information about the support for Spot Instances in Amazon VPC. With this update, users can launch Spot Instances a virtual private cloud (VPC). By launching Spot Instances in a VPC, users of Spot Instances can enjoy the benefits of Amazon VPC.	11 October 2011

Feature	API version	Description	Release date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of `ImportInstance` and `ImportVolume`, which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of `ResumeImport`, users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide.	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes.	27 June 2011

Feature	API version	Description	Release date
Spot Instances Availability Zone pricing changes	2011-05-15	Added information about the Spot Instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot Instance requests and Spot price history. These additions make it easier to determine the price required to launch a Spot Instance into a particular Availability Zone.	26 May 2011
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Identity and access management for Amazon EC2.	26 April 2011
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Amazon EC2 Dedicated Instances.	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances.	27 March 2011

Feature	API version	Description	Release date
Metadata informati on	2011-01-0 1	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Use instance metadata to manage your EC2 instance and Instance metadata categories.	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrate s with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2.	3 March 2011
Force volume detachment		You can now use the AWS Managemen t Console to force the detachment of an Amazon EBS volume from an instance.	23 February 2011
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Change instance termination protection.	23 February 2011
VM Import	2010-11-1 5	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more informati on, see the VM Import/Export User Guide.	15 December 2010
Basic monitoring for instances	2010-08-3 1	Added information about basic monitoring for EC2 instances.	12 December 2010

Feature	API version	Description	Release date
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Find your Amazon EC2 resources and Tag your Amazon EC2 resources.	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances.	19 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Identity and access management for Amazon EC2.	2 September 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes.	14 June 2010
Reserved Instances with Windows		Amazon EC2 now supports Reserved Instances with Windows.	22 February 2010