



User Guide for Linux Instances

Amazon Elastic Compute Cloud



Amazon Elastic Compute Cloud: User Guide for Linux Instances

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

What is Amazon EC2?	1
Features	2
Get started	3
Related services	4
Access EC2	6
Pricing	7
Estimates, billing, and cost optimization	8
Set up	9
Sign up for an AWS account	9
Create a user with administrative access	10
Create a key pair	11
Create a security group	12
Get started tutorial	19
Overview	19
Prerequisites	20
Step 1: Launch an instance	20
Step 2: Connect to your instance	22
Step 3: Track your Free Tier usage	22
Step 4: Clean up your instance	25
Next steps	26
Best practices	27
Tutorials	30
Install LAMP	30
Amazon Linux 2	31
Amazon Linux	43
Configure SSL/TLS	56
Amazon Linux 2	56
Amazon Linux	74
Host a WordPress blog	90
Amazon Linux 2	91
Amazon Machine Images	103
Use an AMI	104
Create your own AMI	104
Buy, share, and sell AMIs	105

Deregister your AMI	105
AL2023 and Amazon Linux 2	105
AMI types	106
Launch permissions	106
Storage for the root device	107
Virtualization types	111
Boot modes	113
Launch an instance	115
AMI boot mode parameter	120
Instance type boot mode	122
Instance boot mode	124
Operating system boot mode	126
Set AMI boot mode	127
UEFI variables	132
UEFI Secure Boot	133
Find a Linux AMI	148
Find a Linux AMI using the Amazon EC2 console	149
Find an AMI using the AWS CLI	150
Find the latest Amazon Linux AMI using Systems Manager	151
Use a Systems Manager parameter to find an AMI	152
Shared AMIs	156
Verified provider	156
Find shared AMIs	157
Make an AMI public	161
Share an AMI with organizations or OUs	168
Share an AMI with specific AWS accounts	177
Cancel having an AMI shared with your account	181
Use bookmarks	183
Guidelines for shared Linux AMIs	183
Paid AMIs	190
Sell your AMI	191
Find a paid AMI	191
Purchase a paid AMI	192
Get the product code for your instance	193
Use paid support	194
Bills for paid and supported AMIs	194

Manage your AWS Marketplace subscriptions	194
AMI lifecycle	195
Create an AMI	196
Modify an AMI	249
Copy an AMI	250
Store and restore an AMI	261
Deprecate an AMI	271
Disable an AMI	278
Archive AMI snapshots	284
Deregister (delete) an AMI	285
Automate the EBS-backed AMI lifecycle	295
Use encryption with EBS-backed AMIs	295
Instance-launching scenarios	295
Image-copying scenarios	299
Monitor AMI events	301
AMI events	302
Create Amazon EventBridge rules	305
Understand AMI billing	308
AMI billing fields	309
Find AMI billing information	311
Verify AMI charges on your bill	314
Amazon Linux	314
Amazon Linux availability	315
Connect to an Amazon Linux instance	315
Identify Amazon Linux images	316
Amazon Linux 2 AMI boot mode	317
AWS command line tools	317
Package repository	318
Extras library (Amazon Linux 2)	322
Amazon Linux 2 supported kernels	323
Access source packages for reference	325
cloud-init	325
Subscribe to Amazon Linux notifications	327
Run Amazon Linux 2 on premises	330
Kernel Live Patching	336
User provided kernels	345

HVM AMIs (GRUB)	345
Paravirtual AMIs (PV-GRUB)	345
Configure the MATE desktop connection	352
Prerequisite	353
Configure the RDP connection	354
AMI quotas	356
Request a quota increase for AMIs	357
Instances	359
Instances and AMIs	360
Instances	360
AMIs	363
Instance types	363
Available instance types	364
Hardware specifications	365
AMI virtualization types	367
Find an instance type	367
Get recommendations	369
Change the instance type	377
Burstable performance instances	387
Mac instances	438
Considerations	439
Instance readiness	441
EC2 macOS AMIs	441
EC2 macOS Init	441
Amazon EC2 System Monitor for macOS	442
Related resources	442
Launch a Mac instance	442
Connect to your Mac instance	445
Update the operating system and software on Mac instances	448
Increase the size of an EBS volume on your Mac instance	456
Stop and terminate your Mac instance	457
Find supported macOS versions for Dedicated Host	458
Subscribe to macOS AMI notifications	459
EC2 macOS AMIs release notes	460
EBS optimization	462
Supported instance types	463

Get maximum performance	535
View instances types that support EBS optimization	536
Enable EBS optimization at launch	537
Enable EBS optimization for an existing instance	538
Instance purchasing options	539
Determine the instance lifecycle	540
On-Demand Instances	541
Reserved Instances	544
Spot Instances	614
Dedicated Hosts	713
Dedicated Instances	774
Capacity Reservations	783
Instance lifecycle	867
Instance launch	869
Instance stop and start	870
Instance hibernate	870
Instance reboot	871
Instance termination	871
Differences between reboot, stop, hibernate, and terminate	872
Launch	874
Stop and start	953
Hibernate	961
Reboot	992
Terminate	993
Retire	1003
Recover	1008
Connect	1014
Connect to your Linux instance	1014
Connect using EC2 Instance Connect Endpoint	1081
Connect your instance to a resource	1104
Configure instances	1146
Common configuration scenarios	1146
Manage software	1147
Manage users	1157
Processor state control	1162
I/O scheduler	1172

Set the time	1174
GPU instances	1190
Optimize CPU options	1241
CPU features	1363
Change the hostname	1369
Set up dynamic DNS	1373
Run commands at launch	1376
Instance metadata and user data	1386
Identify instances	1517
Inspect the instance identity document	1517
Inspect the system UUID	1517
Inspect the system virtual machine generation identifier	1518
Fleets	1525
EC2 Fleet	1526
EC2 Fleet limitations	1528
Burstable performance instances	1528
EC2 Fleet request types	1529
EC2 Fleet configuration strategies	1556
Work with EC2 Fleets	1595
Spot Fleet	1622
Spot Fleet request types	1622
Spot Fleet configuration strategies	1623
Work with Spot Fleets	1662
CloudWatch metrics for Spot Fleet	1695
Automatic scaling for Spot Fleet	1699
Monitor fleet events	1708
EC2 Fleet event types	1709
Spot Fleet event types	1715
Create EventBridge rules	1722
Tutorials	1732
Tutorial: Use EC2 Fleet with instance weighting	1733
Tutorial: Use EC2 Fleet with On-Demand as the primary capacity	1736
Tutorial: Launch On-Demand Instances using targeted Capacity Reservations	1738
Tutorial: Launch instances into Capacity Blocks	1745
Tutorial: Use Spot Fleet with instance weighting	1747
Example configurations	1750

EC2 Fleet example configurations	1750
Spot Fleet example configurations	1771
Fleet quotas	1790
Request a quota increase for target capacity	1791
Monitor	1792
Automated and manual monitoring	1793
Automated monitoring tools	1794
Manual monitoring tools	1795
Best practices for monitoring	1796
Monitor the status of your instances	1796
Instance status checks	1797
State change events	1805
Scheduled events	1807
Monitor your instances using CloudWatch	1839
Instance alarms	1840
Enable detailed monitoring	1841
List available metrics	1844
Install and configure the CloudWatch agent	1868
Get statistics for metrics	1872
Graph metrics	1882
Create an alarm	1883
Create alarms that stop, terminate, reboot, or recover an instance	1884
Automate using EventBridge	1897
Amazon EC2 event types	1898
Log API calls with AWS CloudTrail	1899
Amazon EC2 and Amazon EBS information in CloudTrail	1899
Understand Amazon EC2 and Amazon EBS log file entries	861
Audit users that connect via EC2 Instance Connect	1901
Networking	1904
Regions and Zones	1905
Regions	1905
Availability Zones	1912
Local Zones	1916
Wavelength Zones	1919
AWS Outposts	1922
Instance IP addressing	1924

Private IPv4 addresses	1925
Public IPv4 addresses	1926
Elastic IP addresses (IPv4)	1927
IPv6 addresses	1927
Work with the IPv4 addresses for your instances	1929
Work with the IPv6 addresses for your instances	1932
Multiple IP addresses	1934
EC2 instance hostnames	1947
Link-local addresses	1947
Instance hostname types	1948
Types of EC2 hostnames	1948
Where you see Resource name and IP name	1950
How to decide whether to choose Resource name or IP name	1952
Modify Hostname type and DNS Hostname configurations	1952
Bring your own IP addresses	1954
BYOIP definitions	1955
Requirements and quotas	1955
Onboarding prerequisites	1956
Onboard your BYOIP	1965
Work with your address range	1970
Validate your BYOIP	1971
Regional availability	1975
Local Zone availability	1975
Learn more	1976
Elastic IP addresses	1976
Elastic IP address pricing	1977
Elastic IP address basics	1977
Work with Elastic IP addresses	1978
Elastic IP address quota	1993
Network interfaces	1994
Network interface basics	1995
Network cards	1997
IP addresses per network interface per instance type	1999
Work with network interfaces	2000
Best practices for configuring network interfaces	2011
Scenarios for network interfaces	2014

Requester-managed network interfaces	2017
Assign prefixes	2019
Network bandwidth	2035
Available instance bandwidth	2036
Monitor instance bandwidth	2038
Enhanced networking	2038
Enhanced networking support	2039
Enable enhanced networking on your instance	2039
Elastic Network Adapter (ENA)	2040
ENA Express	2054
Intel 82599 VF	2075
Operating system optimizations	2084
Network performance metrics	2085
Troubleshoot ENA	2094
Improve network latency on Linux instances	2108
Nitro performance considerations	2111
Elastic Fabric Adapter	2118
EFA basics	2120
Supported interfaces and libraries	2121
Supported instance types	2121
Supported operating systems	2122
EFA limitations	2123
EFA pricing	2124
Get started with P5 instances and EFA	2124
Get started with EFA and MPI	2128
Get started with EFA and NCCL	2144
Work with EFA	2183
Monitor an EFA	2186
Verify the EFA installer using a checksum	2187
Instance topology	2199
How it works	2200
Prerequisites	2203
Examples	2205
Placement groups	2216
Placement strategies	2217
Rules and limitations	2221

Work with placement groups	2223
Share a placement group	2236
Placement groups on AWS Outposts	2242
Network MTU	2243
Jumbo frames (9001 MTU)	2244
Path MTU Discovery	2245
Check the path MTU between two hosts	2246
Check and set the MTU on your Linux instance	2247
Troubleshoot	2248
Virtual private clouds	2249
Your default VPCs	2249
Create additional VPCs	2250
Access the internet from your instances	2251
SSH access to your instances	2252
Shared subnets	2252
IPv6-only subnets	2252
Security	2253
Infrastructure security	2254
Network isolation	2254
Isolation on physical hosts	2255
Controlling network traffic	2255
Resilience	2256
Data protection	2257
Amazon EBS data security	2258
Encryption at rest	2258
Encryption in transit	2259
Identity and access management	2261
Network access to your instance	2262
Amazon EC2 permission attributes	2262
IAM and Amazon EC2	2262
IAM policies	2264
AWS managed policies	2335
IAM roles	2337
Key pairs	2353
Create a key pair	2354
Tag a key pair	2361

Describe your key pairs	2363
Delete your key pair	2369
Add or remove a public key on your instance	2371
Verify the fingerprint	2372
Security groups	2375
Security group rules	2377
Connection tracking	2379
Default and custom security groups	2384
Work with security groups	2386
Security group rules for different use cases	2396
AWS PrivateLink	2403
Create an interface VPC endpoint	2404
Create an endpoint policy	2404
Update management	2405
Compliance validation	2406
NitroTPM	2407
Considerations	2408
Prerequisites	2408
Create a Linux AMI for NitroTPM support	2409
Verify whether an AMI is enabled for NitroTPM	2410
Enable or stop using NitroTPM on an instance	2411
Storage	2413
Amazon EBS	2414
Instance store	2415
Instance store volume and data lifetime	2416
Instance store volumes	2418
Add instance store volumes	2420
SSD instance store volumes	2425
Instance store swap volumes	2428
Optimize disk performance	2432
File storage	2433
Amazon S3	2433
Amazon EFS	2436
Amazon FSx	2439
Amazon File Cache	2445
Instance volume limits	2445

Volume limits for instances built on the Nitro System	2446
Volume limits for Xen-based instances	2448
Root device volume	2448
Root volume type	2449
Choose an AMI by root volume type	2452
Determine the root device type of your instance	2453
Change the root volume to persist	2453
Change the initial size of the root volume	2457
Replace a root volume	2458
Device names	2469
Available device names	2469
Device name considerations	2470
Block device mappings	2471
Block device mapping concepts	2472
AMI block device mapping	2476
Instance block device mapping	2479
Torn write prevention	2485
Pricing	2486
Supported block sizes and block boundary alignments	2486
Requirements	2486
Check torn write prevention support and configuration	2487
Configure your software stack for torn write prevention	2489
Resources and tags	2491
Recycle Bin	2491
How does it work?	2492
Supported resources	2493
Considerations	2493
Quotas	2497
Related services	2497
Pricing	2497
Required IAM permissions	2498
Work with retention rules	2503
Work with resources in the Recycle Bin	2517
Monitor Recycle Bin	2527
Resource locations	2546
Resource IDs	2547

List and filter your resources	2548
Console steps	2548
CLI and API steps	2554
Global View (cross-Region)	2557
Global View	2557
Tag your resources	2560
Tag basics	2561
Tag your resources	2562
Tag restrictions	2566
Tags and access management	2567
Tag your resources for billing	2568
Work with tags using the console	2568
Work with tags using the command line	2574
Work with instance tags in instance metadata	2578
Add tags to a resource using CloudFormation	2582
Service quotas	2583
View your current quotas	2583
Request an increase	2584
Restriction on email sent using port 25	2585
Usage reports	2585
Track your Free Tier usage	2586
Troubleshoot	2589
Troubleshoot launch issues	2589
Invalid device name	2590
Instance limit exceeded	2590
Insufficient instance capacity	2591
The requested configuration is currently not supported. Please check the documentation for supported configurations.	2592
Instance terminates immediately	2592
Insufficient permissions	2594
Connect to your instance	2595
Common causes for connection issues	2595
Error connecting to your instance: Connection timed out	2598
Error: unable to load key ... Expecting: ANY PRIVATE KEY	2601
Error: User key not recognized by server	2601
Error: Permission denied or connection closed by [instance] port 22	2604

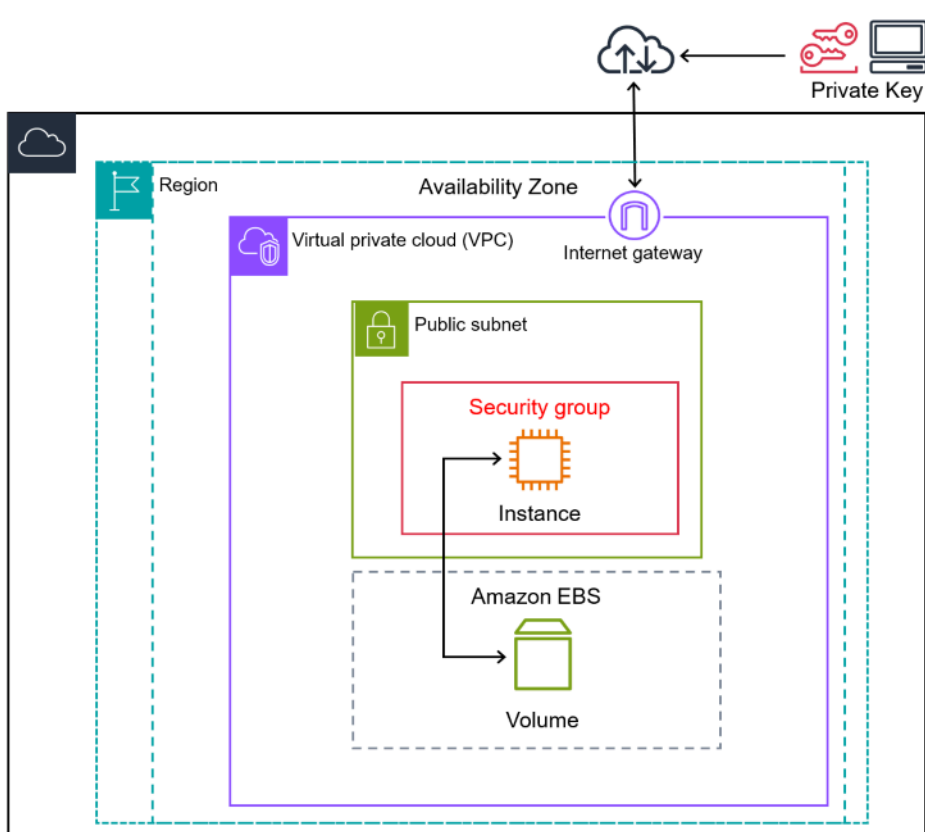
Error: Unprotected private key file	2606
Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"	2607
Error: Server refused our key or No supported authentication methods available	2608
Cannot ping instance	2609
Error: Server unexpectedly closed network connection	2609
Error: Host key validation failed for EC2 Instance Connect	2610
Can't connect to Ubuntu instance using EC2 Instance Connect	2612
I've lost my private key. How can I connect to my Linux instance?	2612
Stop your instance	2619
Force stop the instance	2619
Create a replacement instance	2620
Terminate your instance	2622
Instance terminates immediately	2623
Delayed instance termination	2623
Terminated instance still displayed	2623
Error: The instance may not be terminated. Modify its 'disableApiTermination' instance attribute	2623
Instances automatically launched or terminated	2624
Failed status checks	2624
Review status check information	2625
Retrieve the system logs	2626
Troubleshoot system log errors for Linux-based instances	2627
Out of memory: kill process	2628
ERROR: mmu_update failed (Memory management update failed)	2629
I/O error (block device failure)	2630
I/O ERROR: neither local nor remote disk (Broken distributed block device)	2632
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)	2632
"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)	2634
"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)	2635
ERROR Invalid kernel (EC2 incompatible kernel)	2637
fsck: No such file or directory while trying to open... (File system not found)	2638
General error mounting filesystems (failed mount)	2640
VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)	2642

Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)	2644
XENBUS: Device with no driver...	2645
... days without being checked, check forced (File system check required)	2646
fsck died with exit status... (Missing device)	2647
GRUB prompt (grubdom>)	2648
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)	2651
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)	2653
XENBUS: Timeout connecting to devices (Xenbus timeout)	2654
Troubleshoot an unreachable instance	2655
Instance reboot	2656
Instance console output	2656
Capture a screenshot of an unreachable instance	2657
Instance recovery when a host computer fails	2659
Boot from the wrong volume	2660
EC2Rescue for Linux	2662
Install EC2Rescue for Linux	2662
(Optional) Verify the signature of EC2Rescue for Linux	2664
Work with EC2Rescue for Linux	2667
Develop EC2Rescue modules	2669
EC2 Serial Console	2676
Prerequisites	2677
Configure access to the EC2 Serial Console	2683
Connect to the EC2 Serial Console	2692
Disconnect from the EC2 Serial Console	2701
Troubleshoot your instance using the EC2 Serial Console	2702
Send a diagnostic interrupt	2706
Supported instance types	2706
Prerequisites	2706
Send a diagnostic interrupt	2710
Related information	2711
Document history	2713
History for previous years	2752

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.

The following diagram shows a basic architecture of an Amazon EC2 instance deployed within an Amazon Virtual Private Cloud (VPC). In this example, the EC2 instance is within an Availability Zone in the Region. The EC2 instance is secured with a security group, which is a virtual firewall that controls incoming and outgoing traffic. A private key is stored on the local computer and a public key is stored on the instance. Both keys are specified as a key pair to prove the identity of the user. In this scenario, the instance is backed by an Amazon EBS volume. The VPC communicates with the internet using an internet gateway. For more information about Amazon VPC, see the [Amazon VPC User Guide](#).



Tip

This user guide provides information specific to running Linux-based instances on Amazon EC2. See the [EC2 User Guide for Windows Instances](#) for information to help you run Windows-based instances on EC2.

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](#).

If you are looking for technical guidance about Amazon EC2, try [AWS re:Post](#).

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

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.

Key pairs

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

Instance store volumes

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

Amazon EBS volumes

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

Regions and Zones

Multiple physical locations for your resources, such as instances and Amazon EBS volumes.

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.

Elastic IP addresses

Static IPv4 addresses for dynamic cloud computing.

Tags

Metadata that you can create and assign to your Amazon EC2 resources.

Virtual private clouds (VPCs)

Virtual networks you can create that are logically isolated from the rest of the AWS Cloud. You can optionally connect these virtual networks to your own network.

For details about all of the features of Amazon EC2, see [Amazon EC2 features](#).

For options to run your website on AWS, see [Web Hosting](#).

Get started with Amazon EC2

The following topics can help you get started with Amazon EC2. After you set up to use EC2, you can walk through [Tutorial: Get started with Amazon EC2 Linux instances](#) to launch, connect to, and clean up an instance. The remaining topics point to more information about the high-level features of EC2.

Set up and use an EC2 instance

- [Set up to use Amazon EC2](#)
- [Tutorial: Get started with Amazon EC2 Linux instances](#)
- [Connect to your Linux instance](#)
- [Transfer files](#)

Learn the basics of Amazon EC2

- [Instances and AMIs](#)
- [Regions and Zones](#)
- [Amazon EC2 instance types](#)

Read about networking and security

- [Key pairs](#)
- [Security groups](#)
- [Elastic IP addresses](#)
- [Virtual private clouds](#)

Review your storage options

- [Amazon EBS](#)
- [Instance store](#)

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 Tools for PowerShell

A set of PowerShell modules that are built on the functionality exposed by the AWS 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 Windows PowerShell User Guide](#). You can find the cmdlets for Amazon EC2, in the [AWS Tools for PowerShell Cmdlet 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*.

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*.

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](#).

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 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](#).

Set up to use Amazon EC2

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

1. [Sign up for an AWS account](#)
2. [Create a user with administrative access](#)
3. [Create a key pair](#)
4. [Create a security group](#)

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

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

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

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

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform [tasks that require root user access](#).

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/> and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the [AWS Management Console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

For help signing in by using root user, see [Signing in as the root user](#) in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

For instructions, see [Enable a virtual MFA device for your AWS account root user \(console\)](#) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

For instructions, see [Enabling AWS IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

For a tutorial about using the IAM Identity Center directory as your identity source, see [Configure user access with the default IAM Identity Center directory](#) in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see [Signing in to the AWS access portal](#) in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

For instructions, see [Create a permission set](#) in the *AWS IAM Identity Center User Guide*.

2. Assign users to a group, and then assign single sign-on access to the group.

For instructions, see [Add groups](#) in the *AWS IAM Identity Center User Guide*.

Create a key pair

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

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

To create your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Choose **Create key pair**.
4. For **Name**, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
5. For **Key pair type**, choose either **RSA** or **ED25519**. Note that **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. Choose **Create key pair**.

8. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is determined by the file format you chose. Save the private key file in a safe place.

⚠ Important

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

9. 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](#).

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

Create a security group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using SSH. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

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

Prerequisites

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

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

Console

To create a security group with least privilege

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the top navigation bar, select an AWS Region for the security group. Security groups are specific to a Region, so you should select the same Region in which you created your key pair.
3. In the left navigation pane, choose **Security Groups**.
4. Choose **Create security group**.
5. For **Basic details**, do the following:
 - a. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your user name, followed by `_SG_`, plus the Region name. For example, `me_SG_uswest2`.
 - b. In the **VPC** list, select your default VPC for the Region.
6. For **Inbound rules**, create rules that allow specific traffic to reach your instance. For example, use the following rules for a web server that accepts HTTP and HTTPS traffic. For more examples, see [Security group rules for different use cases](#).
 - a. Choose **Add rule**. For **Type**, choose **HTTP**. For **Source**, choose **Anywhere-IPv4** to allow inbound HTTP traffic from any IPv4 address, or **Anywhere-IPv6** to allow inbound HTTP traffic from any IPv6 address.
 - b. Choose **Add rule**. For **Type**, choose **HTTPS**. For **Source**, choose **Anywhere-IPv4** to allow inbound HTTPS traffic from any IPv4 address, or **Anywhere-IPv6** to allow inbound HTTPS traffic from any IPv6 address.
 - c. Choose **Add rule**. For **Type**, choose **SSH**. For **Source**, do one of the following:
 - Choose **My IP** to automatically add the public IPv4 address of your local computer.
 - Choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix `/32`, for example, `203.0.113.25/32`. If your company or your router allocates addresses from a range, specify the entire range, such as `203.0.113.0/24`.

⚠ Warning

For security reasons, do not choose **Anywhere-IPv4** or **Anywhere-IPv6** for **Source** with a rule for SSH. This would allow access to your instance from all IP addresses on the internet. This is acceptable for a short time in a test environment, but it is unsafe for production environments.

7. For **Outbound rules**, keep the default rule, which allows all outbound traffic.
8. Choose **Create security group**.

AWS CLI

When you use the AWS CLI to create a security group, an outbound rule that allows all outbound traffic is automatically added to the security group. An inbound rule isn't automatically added; you'll need to add it.

In this procedure, you'll combine the [create-security-group](#) and [authorize-security-group-ingress](#) AWS CLI commands to create the security group and add the inbound rule that allows the specified inbound traffic. An alternative to the following procedure is to run the commands separately, first creating a security group, and then adding an inbound rule to the security group.

To create a security group and add an inbound rule to the security group

Use the [create-security-group](#) and [authorize-security-group-ingress](#) AWS CLI commands as follows:

```
aws ec2 authorize-security-group-ingress \
  --region us-west-2 \
  --group-id $(aws ec2 create-security-group \
    --group-name myname_SG_uswest2 \
    --description "Security group description" \
    --vpc-id vpc-12345678 \
    --output text \
    --region us-west-2) \
  --ip-permissions \
  IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges='[{"CidrIp=0.0.0.0/0,Description="HTTP from anywhere"}]' \
```

```
IpProtocol=tcp,FromPort=443,ToPort=443,IpRanges=' [{CidrIp=0.0.0.0/0,Description="HTTPS
from anywhere"}]' \

IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges=' [{CidrIp=172.31.0.0/16,Description="SSH
from private network"}]' \

IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges=' [{CidrIp=203.0.113.25/32,Description="SSH
from public IP"}]'
```

For:

- `--region` – Specify the Region in which to create the inbound rules.
- `--group-id` – Specify the `create-security-group` command and the following parameters to create the security group:
 - `--group-name` – Specify a name for the new security group. Use a name that is easy for you to remember, such as your user name, followed by `_SG_`, plus the Region name. For example, `myname_SG_uswest2`.
 - `--description` – Specify a description that will help you know what traffic the security group allows.
 - `--vpc-id` – Specify your default VPC for the Region.
 - `--output` – Specify text as the output format for the command.
 - `--region` – Specify the Region in which to create the security group. It should be the same Region that you specified for the inbound rules.
- `--ip-permissions` – Specify the inbound rules to add to the security group. The rules in this example are for a web server that accepts HTTP and HTTPS traffic from anywhere, and that accepts SSH traffic from a private network (if your company or your router allocates addresses from a range) and a specified public IP address (such as the public IPv4 address of your computer or network in CIDR notation).

Warning

For security reasons, do not specify `0.0.0.0/0` for `CidrIp` with a rule for SSH. This would allow access to your instance from all IP addresses on the internet. This is acceptable for a short time in a test environment, but it is unsafe for production environments.

PowerShell

When you use the AWS Tools for Windows PowerShell to create a security group, an outbound rule that allows all outbound traffic is automatically added to the security group. An inbound rule isn't automatically added; you'll need to add it.

In this procedure, you'll combine the [New-EC2SecurityGroup](#) and [Grant-EC2SecurityGroupIngress](#) AWS Tools for Windows PowerShell commands to create the security group and add the inbound rule that allows the specified inbound traffic. An alternative to the following procedure is to run the commands separately, first creating a security group, and then adding an inbound rule to the security group.

To create a security group

Use the [New-EC2SecurityGroup](#) and [Grant-EC2SecurityGroupIngress](#) AWS Tools for Windows PowerShell commands as follows.

```
Import-Module AWS.Tools.EC2
New-EC2SecurityGroup -GroupName myname_SG_uswest2 -Description 'Security group description' -VpcId vpc-12345678 -Region us-west-2 | `
    Grant-EC2SecurityGroupIngress `
    -GroupName $_ `
    -Region us-west-2 `
    -IpPermission @(
        (New-Object -TypeName Amazon.EC2.Model.IpPermission -Property @{
            IpProtocol = 'tcp';
            FromPort   = 80;
            ToPort     = 80;
            Ipv4Ranges = @( @{CidrIp = '0.0.0.0/0'; Description = 'HTTP from anywhere' } )
        } ),
        (New-Object -TypeName Amazon.EC2.Model.IpPermission -Property @{
            IpProtocol = 'tcp';
            FromPort   = 443;
            ToPort     = 443;
            Ipv4Ranges = @( @{CidrIp = '0.0.0.0/0'; Description = 'HTTPS from anywhere' } )
        } ),
        (New-Object -TypeName Amazon.EC2.Model.IpPermission -Property @{
            IpProtocol = 'tcp';
            FromPort   = 3389;
            ToPort     = 3389;
```

```
        Ipv4Ranges = @(
            @{CidrIp = '172.31.0.0/16'; Description = 'RDP from private
network'},
            @{CidrIp = '203.0.113.25/32'; Description = 'RDP from public
IP'}
        )
    })
)
```

For the security group:

- -GroupName – Specify a name for the new security group. Use a name that is easy for you to remember, such as your user name, followed by `_SG_`, plus the Region name. For example, `myname_SG_uswest2`.
- -Description – Specify a description that will help you know what traffic the security group allows.
- -VpcId – Specify your default VPC for the Region.
- -Region – Specify the Region in which to create the security group.

For the inbound rules:

- -GroupName – Specify `$_` to reference the security group you're creating.
- -Region – Specify the Region in which to create the inbound rules. It should be the same Region that you specified for the security group.
- -IpPermission – Specify the inbound rules to add to the security group. The rules in this example are for a web server that accepts HTTP and HTTPS traffic from anywhere, and that accepts RDP traffic from a private network (if your company or your router allocates addresses from a range) and a specified public IP address (such as the public IPv4 address of your computer or network in CIDR notation).

Warning

For security reasons, do not specify `0.0.0.0/0` for `CidrIp` with a rule for RDP. This would allow access to your instance from all IP addresses on the internet. This is acceptable for a short time in a test environment, but it is unsafe for production environments.

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

Tutorial: Get started with Amazon EC2 Linux instances

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

When you sign up for AWS, you can get started with Amazon EC2 using the [AWS Free Tier](#). If you created your AWS account less than 12 months ago, and have not already exceeded the Free Tier benefits for Amazon EC2, it 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 until you terminate the instance (which is the final task of this tutorial), even if it remains idle.

Related tutorials

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

Contents

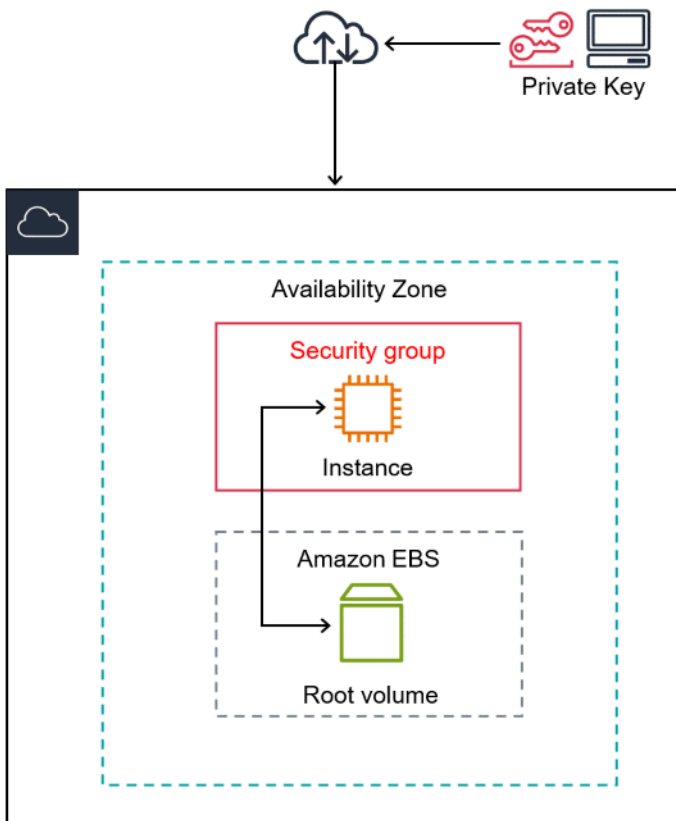
- [Overview](#)
- [Prerequisites](#)
- [Step 1: Launch an instance](#)
- [Step 2: Connect to your instance](#)
- [Step 3: Track your Free Tier usage](#)
- [Step 4: Clean up your instance](#)
- [Next steps](#)

Overview

The instance launched in this tutorial is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs,

or let Amazon EC2 select an Availability Zone for you. Availability Zones are multiple, isolated locations within each AWS Region. You can think of an Availability Zone as an isolated data center.

When you launch your instance, you secure it by specifying a key pair (to prove your identity) and a security group (which acts as a virtual firewall to control ingoing and outgoing traffic). When you connect to your instance, you must provide the private key of the key pair that you specified when you launched your instance.



Prerequisites

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

Step 1: Launch an instance

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

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**.
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
4. Under **Application and OS Images (Amazon Machine Image)**, do the following:
 - a. Choose **Quick Start**, and then choose Amazon Linux. This is the operating system (OS) for your instance.
 - b. From **Amazon Machine Image (AMI)**, select an HVM version of Amazon Linux 2. Notice that these AMIs are marked **Free Tier eligible**. An *Amazon Machine Image (AMI)* is a basic configuration that serves as a template for your instance.

Note


AL2023 is the successor to Amazon Linux 2. For more information, see [Launching AL2023 using the Amazon EC2 console](#).

5. Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the `t2.micro` instance type, which is selected by default. The `t2.micro` instance type is eligible for the Free Tier. In Regions where `t2.micro` is unavailable, you can use a `t3.micro` instance under the Free Tier. For more information, see [AWS Free Tier](#).
6. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created when getting set up.

Warning

Do not choose **Proceed without a key pair (Not recommended)**. If you launch your instance without a key pair, then you can't connect to it.

7. Next to **Network settings**, choose **Edit**. For **Security group name**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
 - a. Choose **Select existing security group**.

- b. From **Common security groups**, choose your security group from the list of existing security groups.
8. Keep the default selections for the other configuration settings for your instance.
9. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.
10. 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.
11. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. If the **Public IPv4 DNS** column is hidden, choose the settings icon () in the top-right corner, toggle on **Public IPv4 DNS**, and choose **Confirm**.
12. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the **Status check** column.

Step 2: Connect to your instance

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


Important

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

Step 3: Track your Free Tier usage

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 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.

 **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. In the navigation pane, choose **EC2 Dashboard**.
2. Find the **EC2 Free Tier** box (at top right).

EC2 Free Tier [Info](#)

Offers for all AWS Regions.


3 EC2 free tier offers in use


End of month forecast
⚠️ 2 offers forecasted to exceed free tier limit.


Exceeds free tier
⚠️ 1 offers exceeded and is now pay-as-you-go pricing.


[View Global EC2 resources](#)

Offer usage (monthly)

Windows EC2 Instances
 12%
662 hours remaining

Linux EC2 Instances
 100%
⚠️ Offer limit reached


Storage space on EBS
 85%
4.59 GB remaining

[View all AWS Free Tier offers](#) 

3. 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.

4. 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, go to the next step in this tutorial.
 - 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 [Amazon EC2 Global View](#).
5. 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 [Using the AWS Free Tier](#) in the *AWS Billing User Guide*.

Step 4: 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.

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

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

Next steps

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

- Learn how to remotely manage your EC2 instance using the Run command. For more information, see [AWS Systems Manager Run Command](#) in the *AWS Systems Manager User Guide*.
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see [Tracking your AWS Free Tier usage](#) in the *AWS Billing User Guide*.
- Add an EBS volume. For more information, see [Create an Amazon EBS volume](#) in the *Amazon EBS User Guide*.
- Install the LAMP stack. For more information, see [Install LAMP on Amazon Linux 2](#).
- Learn about instance purchasing options. For more information, see [Instance purchasing options](#).
- Get advice about instance types. For more information, see [Get instance type recommendations for a new workload](#).

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. For more information, see [Security group rules](#).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating AL2023, see [Updating AL2023](#) in the *AL2023 User Guide*. For more information about updating Amazon Linux 2 or the Amazon Linux AMI, see [Manage software on your Linux instance](#) in the *Amazon EC2 User Guide for Linux 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 [Storage for the root device](#).
- 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 [Instance metadata and user data](#) 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](#).
- 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.

Tutorials for Amazon EC2 instances running Linux

The following tutorials show you how to perform common tasks using EC2 instances running Linux. AWS provides Amazon Linux 2023, Amazon Linux 2, and the Amazon Linux AMI. For more information, see [Amazon Linux 2023](#), [Amazon Linux 2](#), and [Amazon Linux AMI](#). For video tutorials, see [AWS Instructional Videos and Labs](#).

Note

For AL2023 tutorials, see [Tutorials](#) in the *Amazon Linux 2023 User Guide*.

Tutorials

- [Install LAMP](#)
- [Configure SSL/TLS](#)
- [Host a WordPress blog](#)

Install LAMP

This section includes tutorials that show you how to install an Apache web server with PHP and MariaDB on an Amazon EC2 instance.

Note

For the AL2023 LAMP tutorial, see [Tutorial: Install a LAMP server on AL2023](#) in the *Amazon Linux 2023 User Guide*.

Install LAMP on

- [Install LAMP on Amazon Linux 2](#)
- [Install LAMP on Amazon Linux](#)

Install LAMP on Amazon Linux 2

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

Important

If you are trying to set up a LAMP web server on a different distribution, such as Ubuntu or Red Hat Enterprise Linux, this tutorial will not work. For Amazon Linux AMI, see [Install LAMP on Amazon Linux](#). For Ubuntu, see the following Ubuntu community documentation: [ApacheMySQLPHP](#). For other distributions, see their specific documentation.

Option: Complete this tutorial using automation

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

Tasks

- [Step 1: Prepare the LAMP server](#)
- [Step 2: Test your LAMP server](#)
- [Step 3: Secure the database server](#)
- [Step 4: \(Optional\) Install phpMyAdmin](#)
- [Troubleshoot](#)
- [Related topics](#)

Step 1: Prepare the LAMP server

Prerequisites

- This tutorial assumes that you have already launched a new instance using Amazon Linux 2, with a public DNS name that is reachable from the internet. For more information, see [Step 1: Launch an instance](#). You must also have configured your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see [Security group rules](#).

- The following procedure installs the latest PHP version available on Amazon Linux 2, currently php8.2. If you plan to use PHP applications other than those described in this tutorial, you should check their compatibility with php8.2.

To prepare the LAMP server

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

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

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

3. Install the mariadb10.5 Amazon Linux Extras repositories to get the latest version of the MariaDB package.

```
[ec2-user ~]$ sudo amazon-linux-extras install mariadb10.5
```

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

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

4. Install the php8.2 Amazon Linux Extras repositories to get the latest version of the PHP package for Amazon Linux 2.

```
[ec2-user ~]$ sudo amazon-linux-extras install php8.2
```

5. Now that your instance is current, you can install the Apache web server, MariaDB, and PHP software packages. Use the `yum install` command to install multiple software packages and all related dependencies at the same time

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

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

```
yum info package_name
```

6. Start the Apache web server.

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

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

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

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

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

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

Port range	Protocol	Source
22	tcp	0.0.0.0/0

⚠ Warning

Using `0.0.0.0/0` allows all IPv4 addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you authorize only a specific IP address or range of addresses to access your instance.

- d. Choose the link for the security group. Using the procedures in [Add rules to a security group](#), add a new inbound security rule with the following values:
 - **Type:** HTTP

- **Protocol:** TCP
 - **Port Range:** 80
 - **Source:** Custom
9. Test your web server. In a web browser, type the public DNS address (or the public IP address) of your instance. If there is no content in `/var/www/html`, you should see the Apache test page. You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, choose **Show/Hide Columns** (the gear-shaped icon) and choose **Public DNS**).

Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see [Add rules to a security group](#).

Important

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

Test Page

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

If you are a member of the general public:

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

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

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

If you are the website administrator:

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

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



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

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

To set file permissions

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

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

2. Log out and then log back in again to pick up the new group, and then verify your membership.

- a. Log out (use the **exit** command or close the terminal window):

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

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

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

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

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

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

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

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

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

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

To secure your web server (Optional)

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

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

Step 2: Test your LAMP server

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

To test your LAMP server

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

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

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

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

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

You should see the PHP information page:

PHP Version 7.2.0	
System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64 #1 SMP Wed Dec 6 00:07:49 UTC 2017 x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

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

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

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

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

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

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

Step 3: Secure the database server

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

To secure the MariaDB server

1. Start the MariaDB server.

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

2. Run **mysql_secure_installation**.

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

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

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

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

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

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

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

Step 4: (Optional) Install phpMyAdmin

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

Important

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

To install phpMyAdmin

1. Install the required dependencies.

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

2. Restart Apache.

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

3. Restart php-fpm.

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

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

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


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

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

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

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

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

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

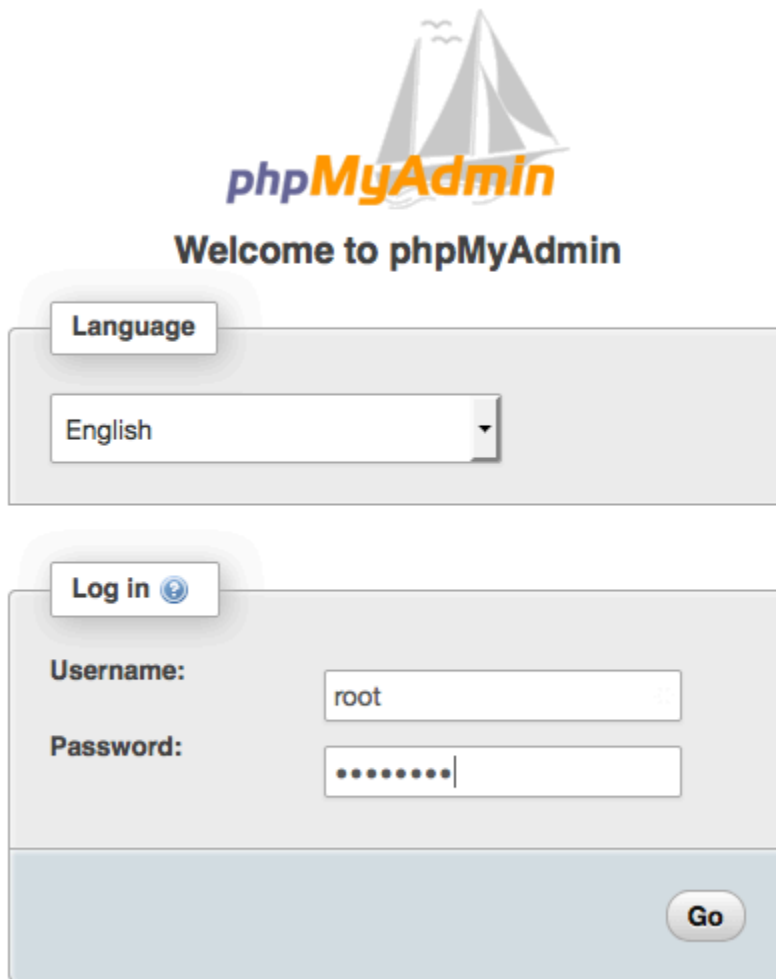
8. (Optional) If the MySQL server is not running, start it now.

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

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

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

You should see the phpMyAdmin login page:



The image shows the phpMyAdmin login interface. At the top, there is a logo for phpMyAdmin with a sailboat icon. Below the logo, it says "Welcome to phpMyAdmin". There are two main sections: a "Language" dropdown menu currently set to "English", and a "Log in" section. The "Log in" section has a "Username:" label and a text input field containing "root". Below that is a "Password:" label and a text input field with masked characters (dots). At the bottom right of the login form is a "Go" button.

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

Your installation must still be configured before you put it into service. We suggest that you begin by manually creating the configuration file, as follows:

- a. To start with a minimal configuration file, use your favorite text editor to create a new file, and then copy the contents of `config.sample.inc.php` into it.
- b. Save the file as `config.inc.php` in the phpMyAdmin directory that contains `index.php`.
- c. Refer to post-file creation instructions in the [Using the Setup script](#) section of the phpMyAdmin installation instructions for any additional setup.

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

Troubleshoot

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

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

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

- **Is the web server running?**

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

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

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

- **Is the firewall correctly configured?**

Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see [Add rules to a security group](#).

I can't connect to my server using HTTPS

Perform the following checks to see if your Apache web server is configured to support HTTPS.

- **Is the web server correctly configured?**

After you install Apache, the server is configured for HTTP traffic. To support HTTPS, enable TLS on the server and install an SSL certificate. For information, see [Configure SSL/TLS on Amazon Linux 2](#).

- **Is the firewall correctly configured?**

Verify that the security group for the instance contains a rule to allow HTTPS traffic on port 443. For more information, see [Add rules to a security group](#).

Related topics

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

- [Transfer files to your Linux instance using WinSCP](#)
- [Transfer files to Linux instances using an SCP client](#)
- [Host a WordPress blog on Amazon Linux 2](#)

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

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

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

Install LAMP on Amazon Linux

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

Important

If you are trying to set up a LAMP web server on a different distribution, such as Ubuntu or Red Hat Enterprise Linux, this tutorial will not work. For Amazon Linux 2, see [Install LAMP on Amazon Linux 2](#). For Ubuntu, see the following Ubuntu community documentation: [ApacheMySQLPHP](#). For other distributions, see their specific documentation.

Option: Complete this tutorial using automation

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

Tasks

- [Step 1: Prepare the LAMP server](#)
- [Step 2: Test your LAMP server](#)
- [Step 3: Secure the database server](#)
- [Step 4: \(Optional\) Install phpMyAdmin](#)
- [Troubleshoot](#)
- [Related topics](#)

Step 1: Prepare the LAMP server

Prerequisites

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

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

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

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

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

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

Important

Some applications may not be compatible with the following recommended software environment. Before installing these packages, check whether your LAMP applications

are compatible with them. If there is a problem, you may need to install an alternative environment. For more information, see [The application software I want to run on my server is incompatible with the installed PHP version or other software](#)

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

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

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

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

4. Start the Apache web server.

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

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

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

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

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

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

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

6. Add a security rule to allow inbound HTTP (port 80) connections to your instance if you have not already done so. By default, a **launch-wizard-N** security group was set up for your instance during initialization. This group contains a single rule to allow SSH connections.

- a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- b. Choose **Instances** and select your instance.
- c. On the **Security** tab, view the inbound rules. You should see the following rule:

Port range	Protocol	Source
22	tcp	0.0.0.0/0

⚠ Warning

Using `0.0.0.0/0` allows all IPv4 addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you authorize only a specific IP address or range of addresses to access your instance.

- d. Choose the link for the security group. Using the procedures in [Add rules to a security group](#), add a new inbound security rule with the following values:
 - **Type:** HTTP
 - **Protocol:** TCP
 - **Port Range:** 80
 - **Source:** Custom
7. Test your web server. In a web browser, type the public DNS address (or the public IP address) of your instance. You can get the public DNS address for your instance using the Amazon EC2 console. If there is no content in `/var/www/html`, you should see the Apache test page. When you add content to the document root, your content appears at the public DNS address of your instance instead of the test page.

Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see [Add rules to a security group](#).

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

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

```
[ec2-user ~]$ ls -l /var/www
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug  7 00:02 error
drwxr-xr-x 2 root root 4096 Jan  6 2012 html
drwxr-xr-x 3 root root 4096 Aug  7 00:02 icons
drwxr-xr-x 2 root root 4096 Aug  7 21:17 noindex
```

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

To set file permissions

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

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

2. Log out and then log back in again to pick up the new group, and then verify your membership.

- a. Log out (use the `exit` command or close the terminal window):

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

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

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

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

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

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

```
[ec2-user ~]$ sudo chmod 2775 /var/www
```



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

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

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

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

(Optional) Secure your web server

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

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

Step 2: Test your LAMP server

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

To test your LAMP web server

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

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

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

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

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

You should see the PHP information page:

PHP Version 7.2.0	
System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64 #1 SMP Wed Dec 6 00:07:49 UTC 2017 x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

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

```
[ec2-user ~]$ sudo yum list installed httpd24 php72 mysql57-server php72-mysqlnd
Loaded plugins: priorities, update-motd, upgrade-helper
Installed Packages
httpd24.x86_64                2.4.25-1.68.amzn1           @amzn-
updates
mysql56-server.x86_64        5.6.35-1.23.amzn1           @amzn-
updates
php70.x86_64                 7.0.14-1.20.amzn1           @amzn-
updates
php70-mysqlnd.x86_64         7.0.14-1.20.amzn1           @amzn-
updates
```

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

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

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

Step 3: Secure the database server

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

To secure the database server

1. Start the MySQL server.

```
[ec2-user ~]$ sudo service mysqld start
Initializing MySQL database:
...

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
...

Starting mysqld: [ OK ]
```

2. Run `mysql_secure_installation`.

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

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

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

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

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

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

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

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

Step 4: (Optional) Install phpMyAdmin

To install phpMyAdmin

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

Important

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

Note

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

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

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

3. Restart Apache.

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

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

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

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

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

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

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

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

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

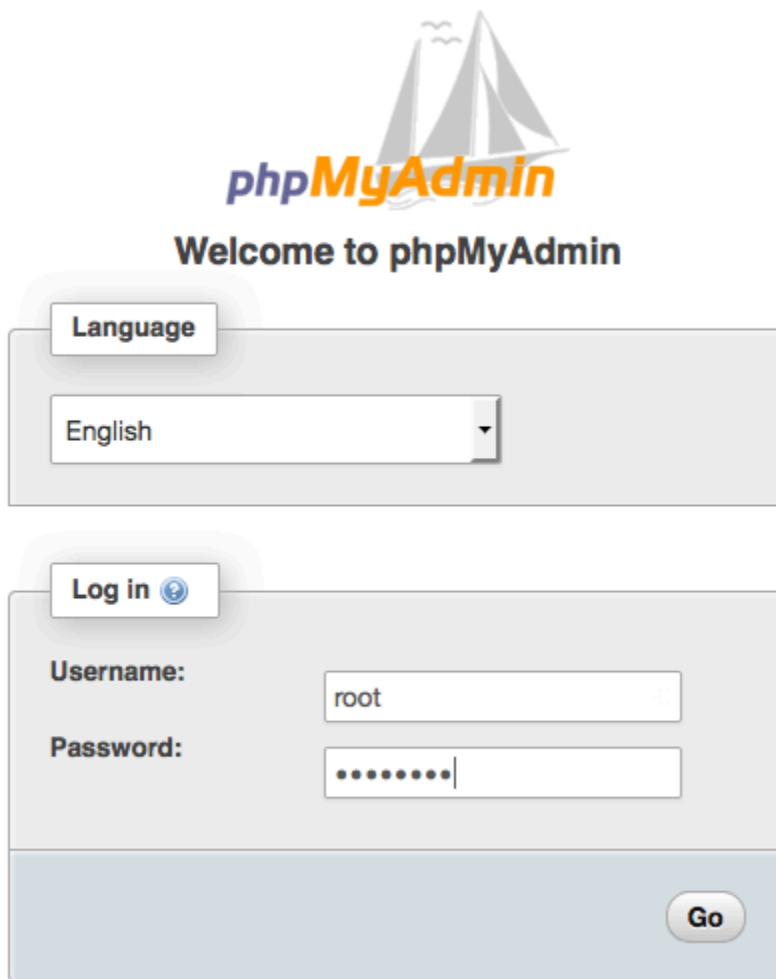
8. (Optional) If the MySQL server is not running, start it now.

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

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

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

You should see the phpMyAdmin login page:



The screenshot shows the phpMyAdmin login interface. At the top, there is a logo of a sailboat with the text 'phpMyAdmin' and 'Welcome to phpMyAdmin'. Below the logo, there is a 'Language' dropdown menu currently set to 'English'. Underneath the language menu is a 'Log in' button with a user icon. The login form consists of two input fields: 'Username:' with the value 'root' and 'Password:' with masked characters. A 'Go' button is located at the bottom right of the form.

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

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

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

Troubleshoot

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

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

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

- **Is the web server running?**

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

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

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

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

- **Is the firewall correctly configured?**

Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see [Add rules to a security group](#).

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

This tutorial recommends installing the most up-to-date versions of Apache HTTP Server, PHP, and MySQL. Before installing an additional LAMP application, check its requirements to confirm that it is compatible with your installed environment. If the latest version of PHP is not supported, it is possible (and entirely safe) to downgrade to an earlier supported configuration. You can also

install more than one version of PHP in parallel, which solves certain compatibility problems with a minimum of effort. For information about configuring a preference among multiple installed PHP versions, see [Amazon Linux AMI 2016.09 Release Notes](#).

How to downgrade

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

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

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

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

Next, install the replacement environment:

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

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

```
[ec2-user ~]$ sudo yum remove -y httpd24 php56 mysql55-server php56-mysqlnd perl-DBD-MySQL56
```

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

Related topics

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

- [Transfer files to your Linux instance using WinSCP](#)
- [Transfer files to Linux instances using an SCP client](#)

- [Host a WordPress blog on Amazon Linux 2](#)

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

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

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

Configure SSL/TLS

This section includes tutorials that show you how to configure SSL/TLS on web server running on an Amazon EC2 instance.

Note

For the AL2023 SSL/TLS tutorial, see [Tutorial: Configure SSL/TLS on AL2023](#) in the *Amazon Linux 2023 User Guide*.

Configure SSL/TLS on

- [Configure SSL/TLS on Amazon Linux 2](#)
- [Configure SSL/TLS on Amazon Linux](#)

Configure SSL/TLS on Amazon Linux 2

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on an EC2 instance with Amazon Linux 2 and

Apache web server. This tutorial assumes that you are not using a load balancer. If you are using Elastic Load Balancing, you can choose to configure SSL offload on the load balancer, using a certificate from [AWS Certificate Manager](#) instead.

For historical reasons, web encryption is often referred to simply as SSL. While web browsers still support SSL, its successor protocol TLS is less vulnerable to attack. Amazon Linux 2 disables server-side support for all versions of SSL by default. [Security standards bodies](#) consider TLS 1.0 to be unsafe. TLS 1.0 and TLS 1.1 were formally [deprecated](#) in March 2021. This tutorial contains guidance based exclusively on enabling TLS 1.2. TLS 1.3 was finalized in 2018 and is available in Amazon Linux 2 as long as the underlying TLS library (OpenSSL in this tutorial) is supported and enabled. [Clients must support TLS 1.2 or later by June 28, 2023](#). For more information about the updated encryption standards, see [RFC 7568](#) and [RFC 8446](#).

This tutorial refers to modern web encryption simply as TLS.

Important

These procedures are intended for use with Amazon Linux 2. We also assume that you are starting with a new Amazon EC2 instance. If you are trying to set up an EC2 instance running a different distribution, or an instance running an old version of Amazon Linux 2, some procedures in this tutorial might not work. For the Amazon Linux AMI, see [Configure SSL/TLS on Amazon Linux](#). For Ubuntu, see the following community documentation: [Open SSL on Ubuntu](#). For Red Hat Enterprise Linux, see the following: [Setting up the Apache HTTP Web Server](#). For other distributions, see their specific documentation.

Note

Alternatively, you can use AWS Certificate Manager (ACM) for AWS Nitro enclaves, which is an enclave application that allows you to use public and private SSL/TLS certificates with your web applications and servers running on Amazon EC2 instances with AWS Nitro Enclaves. Nitro Enclaves is an Amazon EC2 capability that enables creation of isolated compute environments to protect and securely process highly sensitive data, such as SSL/TLS certificates and private keys.

ACM for Nitro Enclaves works with **nginx** running on your Amazon EC2 Linux instance to create private keys, to distribute certificates and private keys, and to manage certificate renewals.

To use ACM for Nitro Enclaves, you must use an enclave-enabled Linux instance.

For more information, see [What is AWS Nitro Enclaves?](#) and [AWS Certificate Manager for Nitro Enclaves](#) in the *AWS Nitro Enclaves User Guide*.

Contents

- [Prerequisites](#)
- [Step 1: Enable TLS on the server](#)
- [Step 2: Obtain a CA-signed certificate](#)
- [Step 3: Test and harden the security configuration](#)
- [Troubleshoot](#)

Prerequisites

Before you begin this tutorial, complete the following steps:

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

For more information, see [Security group rules](#).

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

Step 1: Enable TLS on the server

Option: Complete this tutorial using automation

To complete this tutorial using AWS Systems Manager Automation instead of the following tasks, run the [automation document](#).

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

Note

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

To enable TLS on a server

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

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

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

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

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

Note

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

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

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

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

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

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

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

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

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

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

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

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

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

```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCCKgwgwSkAgEAAoIBAQD2KKx/8Zk94m1q
3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLj00CI8u1PTcGmAah5kEitCEc0wzmNeo
BC10wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ971deakHWeRMiEJFXg6kZZ0vr
GvwnKoMh3D1k44D9dX7IDua2P1Yx5+eroA+1Lqf32ZSaA00bBIMIYTHigwbHMZoT
...
56tE7THvH7v0Ef4/iU0sIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcPODFs
27hDzPDinrquSEvoZlggkDM1h2irTiipJ/GhkvtP0Q1v0fK/VXw8vSgeaBuhwJvS
```

```
LXU9HvYq0U604FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhnNz6eqqdsccCS09VtRAo
4QQvAq0a8UheYeoXLdWcHaLP
-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIEazCCA10gAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb211U3RhdGUxETAPBgNVBACMCFNvbWVudXR5MRkwFwYDVQK
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYWxV
bm10MRkwFwYDVQDDBBpcC0xNzItMzEtMjAtMjMMSQwIgwYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpKKZMz0vDedREjNKAvk4ws6F0
CuIjvubtUysVyQoMVPQ971deakHWeRMiEJFXg6kZZ0vrGvwnKoMh3D1K44D9d1U3
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zC0sc1knYhHrCVD2vnB1ZJKSZvak
3ZazhBxtQSukFM0nWPP2a0DMMFGYUH0d0BQE8sBJxg==
-----END CERTIFICATE-----
```

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

Note

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

5. Open the `/etc/httpd/conf.d/ssl.conf` file using your favorite text editor (such as **vim** or **nano**) as root user and comment out the following line, because the self-signed dummy certificate also contains the key. If you do not comment out this line before you complete the next step, the Apache service fails to start.

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

6. Restart Apache.

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

Note

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

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

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

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

Note

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

Step 2: Obtain a CA-signed certificate

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

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

A self-signed TLS X.509 host certificate is cryptologically identical to a CA-signed certificate. The difference is social, not mathematical. A CA promises, at a minimum, to validate a domain's ownership before issuing a certificate to an applicant. Each web browser contains a list of CAs trusted by the browser vendor to do this. An X.509 certificate consists primarily of a public key that corresponds to your private server key, and a signature by the CA that is cryptographically

ted to the public key. When a browser connects to a web server over HTTPS, the server presents a certificate for the browser to check against its list of trusted CAs. If the signer is on the list, or accessible through a *chain of trust* consisting of other trusted signers, the browser negotiates a fast encrypted data channel with the server and loads the page.

Certificates generally cost money because of the labor involved in validating the requests, so it pays to shop around. A few CAs offer basic-level certificates free of charge. The most notable of these CAs is the [Let's Encrypt](#) project, which also supports the automation of the certificate creation and renewal process. For more information about using a Let's Encrypt certificate, see [Get Certbot](#).

If you plan to offer commercial-grade services, [AWS Certificate Manager](#) is a good option.

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

Important

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

To obtain a CA-signed certificate

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

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


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

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

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

Important

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

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

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

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

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

Note

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

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

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

The preceding commands yield the following result.

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

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

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

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

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

Name	Description	Example
Country Name	The two-letter ISO abbreviation for your country.	US (=United States)
State or Province Name	The name of the state or province where your organization is located. This name cannot be abbreviated.	Washington
Locality Name	The location of your organization, such as a city.	Seattle
Organization Name	The full legal name of your organization. Do not abbreviate your organization name.	Example Corporation

Name	Description	Example
Organizational Unit Name	Additional organizational information, if any.	Example Dept
Common Name	This value must exactly match the web address that you expect users to enter into a browser. Usually, this means a domain name with a prefixed hostname or alias in the form www.example.com . In testing with a self-signed certificate and no DNS resolution, the common name may consist of the hostname alone. CAs also offer more expensive certificates that accept wild-card names such as *.example.com .	www.example.com
Email Address	The server administrator's email address.	someone@example.com

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

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

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

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

Note

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

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

The file should also end with the following line.

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

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

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

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

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

Note

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

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

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

These commands should yield the following result.

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

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

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

These commands should yield the following result.

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

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

Note

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

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

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

These commands should yield the following result.

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


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

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

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

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

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

 **Note**

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

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

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

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

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

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

Step 3: Test and harden the security configuration

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

Important

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

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

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

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

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

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

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

To correct and future-proof the TLS configuration

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

```
#SSLProtocol all -SSLv3
```

2. Add the following directive:

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

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

Note

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

To modify the list of allowed ciphers

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

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

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

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

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

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

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

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

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

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

Note

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

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

```
#SSLHonorCipherOrder on
```

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

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

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

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

Each update to OpenSSL introduces new ciphers and removes support for old ones. Keep your EC2 Amazon Linux 2 instance up-to-date, watch for security announcements from [OpenSSL](#), and be alert to reports of new security exploits in the technical press.

Troubleshoot

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

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

You can remove the encryption and password requirement from the key. Assuming that you have a private encrypted RSA key called `custom.key` in the default directory, and that the password on it is **abcde12345**, run the following commands on your EC2 instance to generate an unencrypted version of the key.

```
[ec2-user ~]$ cd /etc/pki/tls/private/
[ec2-user private]$ sudo cp custom.key custom.key.bak
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out
custom.key.nocrypt
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ sudo systemctl restart httpd
```

Apache should now start without prompting you for a password.

- **I get errors when I run `sudo yum install -y mod_ssl`.**

When you are installing the required packages for SSL, you may see errors similar to the following.

```
Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64
Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
```

This typically means that your EC2 instance is not running Amazon Linux 2. This tutorial only supports instances freshly created from an official Amazon Linux 2 AMI.

Configure SSL/TLS on Amazon Linux

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on an EC2 instance with the Amazon Linux AMI and Apache web server. This tutorial assumes that you are not using a load balancer. If you are

using Elastic Load Balancing, you can choose to configure SSL offload on the load balancer, using a certificate from [AWS Certificate Manager](#) instead.

For historical reasons, web encryption is often referred to simply as SSL. While web browsers still support SSL, its successor protocol TLS is less vulnerable to attack. The Amazon Linux AMI disables server-side support for all versions of SSL by default. [Security standards bodies](#) consider TLS 1.0 to be unsafe. TLS 1.0 and TLS 1.1 were formally [deprecated](#) in March 2021. This tutorial contains guidance based exclusively on enabling TLS 1.2. TLS 1.3 was finalized in 2018 and is available in Amazon Linux 2 as long as the underlying TLS library (OpenSSL in this tutorial) is supported and enabled. [Clients must support TLS 1.2 or later by June 28, 2023](#). For more information about the updated encryption standards, see [RFC 7568](#) and [RFC 8446](#).

This tutorial refers to modern web encryption simply as TLS.

Important

These procedures are intended for use with the Amazon Linux AMI. If you are trying to set up a LAMP web server on an instance with a different distribution, some procedures in this tutorial might not work for you. For Amazon Linux 2, see [Configure SSL/TLS on Amazon Linux 2](#). For Ubuntu, see the following community documentation: [Open SSL on Ubuntu](#). For Red Hat Enterprise Linux, see the following: [Setting up the Apache HTTP Web Server](#). For other distributions, see their specific documentation.

Note

Alternatively, you can use AWS Certificate Manager (ACM) for AWS Nitro enclaves, which is an enclave application that allows you to use public and private SSL/TLS certificates with your web applications and servers running on Amazon EC2 instances with AWS Nitro Enclaves. Nitro Enclaves is an Amazon EC2 capability that enables creation of isolated compute environments to protect and securely process highly sensitive data, such as SSL/TLS certificates and private keys.

ACM for Nitro Enclaves works with **nginx** running on your Amazon EC2 Linux instance to create private keys, to distribute certificates and private keys, and to manage certificate renewals.

To use ACM for Nitro Enclaves, you must use an enclave-enabled Linux instance.

For more information, see [What is AWS Nitro Enclaves?](#) and [AWS Certificate Manager for Nitro Enclaves](#) in the *AWS Nitro Enclaves User Guide*.

Contents

- [Prerequisites](#)
- [Step 1: Enable TLS on the server](#)
- [Step 2: Obtain a CA-signed certificate](#)
- [Step 3: Test and harden the security configuration](#)
- [Troubleshoot](#)

Prerequisites

Before you begin this tutorial, complete the following steps:

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

For more information, see [Security group rules](#).

- Install Apache web server. For step-by-step instructions, see [Tutorial: Installing a LAMP Web Server on Amazon Linux](#). Only the httpd24 package and its dependencies are needed; you can ignore the instructions involving PHP and MySQL.
- To identify and authenticate web sites, the TLS public key infrastructure (PKI) relies on the Domain Name System (DNS). To use your EC2 instance to host a public web site, you need to register a domain name for your web server or transfer an existing domain name to your Amazon EC2 host. Numerous third-party domain registration and DNS hosting services are available for this, or you can use [Amazon Route 53](#).

Step 1: Enable TLS on the server

Option: Complete this tutorial using automation

To complete this tutorial using AWS Systems Manager instead of the following tasks, run the [automation document](#).

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

Note

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

To enable TLS on a server

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

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

If necessary, start Apache.

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

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

Note

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

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

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

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

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

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

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

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

An automatically generated, 2048-bit RSA private key for your Amazon EC2 host. During installation, OpenSSL used this key to generate a self-signed host certificate, and you can also use this key to generate a certificate signing request (CSR) to submit to a certificate authority (CA).

```
/etc/pki/tls/certs/localhost.crt
```

An automatically generated, self-signed X.509 certificate for your server host. This certificate is useful for testing that Apache is properly set up to use TLS.

The `.key` and `.crt` files are both in PEM format, which consists of Base64-encoded ASCII characters framed by "BEGIN" and "END" lines, as in this abbreviated example of a certificate:

```
-----BEGIN CERTIFICATE-----
MIIeazCCA10gAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb211U3RhdGUxETAPBgNVBACMFNvbWVwVWVWYXR5YDVQKQ
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYWxV
bm10MRkwFwYDVQQDDDBBpcC0xNzItMzEtMjMjAtMjM2MSQwIgwYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpKKZMz0vDedREjNKAvk4ws6F0
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zC0sc1knYhHrCVD2vnBlZJKSZvak
3ZazhBxtQSukFM0nWPP2a0DMMFGYUH0d0BQE8sBJxg==
-----END CERTIFICATE-----
```

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

Note

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

4. Restart Apache.

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

5. Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by typing the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix **https://**. Because you are connecting to a site with a self-signed, untrusted host certificate, your browser may display a series of security warnings.

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

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

Step 2: Obtain a CA-signed certificate

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

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

A self-signed TLS X.509 host certificate is cryptologically identical to a CA-signed certificate. The difference is social, not mathematical; a CA promises to validate, at a minimum, a domain's ownership before issuing a certificate to an applicant. Each web browser contains a list of CAs trusted by the browser vendor to do this. An X.509 certificate consists primarily of a public key that corresponds to your private server key, and a signature by the CA that is cryptographically tied to the public key. When a browser connects to a web server over HTTPS, the server presents

a certificate for the browser to check against its list of trusted CAs. If the signer is on the list, or accessible through a chain of trust consisting of other trusted signers, the browser negotiates a fast encrypted data channel with the server and loads the page.

Certificates generally cost money because of the labor involved in validating the requests, so it pays to shop around. A few CAs offer basic-level certificates free of charge. The most notable of these CAs is the [Let's Encrypt](#) project, which also supports the automation of the certificate creation and renewal process. For more information about using a Let's Encrypt certificate, see [Get Certbot](#).

If you plan to offer commercial-grade services, [AWS Certificate Manager](#) is a good option.

Underlying the host certificate is the key. As of 2017, [government](#) and [industry](#) groups recommend using a minimum key (modulus) size of 2048 bits for RSA keys intended to protect documents through 2030. The default modulus size generated by OpenSSL in Amazon Linux is 2048 bits, which means that the existing auto-generated key is suitable for use in a CA-signed certificate. An alternative procedure is described below for those who desire a customized key, for instance, one with a larger modulus or using a different encryption algorithm.

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

To obtain a CA-signed certificate

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

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

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

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

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

Important

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

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

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

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

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

Note

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

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

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

The commands above should yield the following result:

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

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

3. Create a CSR using your preferred key; the example below uses **custom.key**:

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

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

Name	Description	Example
Country Name	The two-letter ISO abbreviation for your country.	US (=United States)
State or Province Name	The name of the state or province where your organization is located. This name cannot be abbreviated.	Washington
Locality Name	The location of your organization, such as a city.	Seattle
Organization Name	The full legal name of your organization. Do not abbreviate your organization name.	Example Corporation
Organizational Unit Name	Additional organizational information, if any.	Example Dept
Common Name	This value must exactly match the web address that you expect users to type into	www.example.com

Name	Description	Example
	a browser. Usually, this means a domain name with a prefixed host name or alias in the form www.example.com . In testing with a self-signed certificate and no DNS resolution, the common name may consist of the host name alone. CAs also offer more expensive certificates that accept wild-card names such as *.example.com .	
Email Address	The server administrator's email address.	someone@example.com

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

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

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

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

Note

Your CA may send you files in multiple formats intended for various purposes. For this tutorial, you should only use a certificate file in PEM format, which is usually (but not always) marked with a `.pem` or `.crt` extension. If you are uncertain which file to

use, open the files with a text editor and find the one containing one or more blocks beginning with the following:

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

The file should also end with the following:

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

You can also test a file at the command line as follows:

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

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

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

Note

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

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

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

The commands above should yield the following result:

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

The permissions for the intermediate certificate file are less stringent (owner=root, group=root, owner can write, group can read, world can read). The commands would be:

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

The commands above should yield the following result:

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

6. If you used a custom key to create your CSR and the resulting host certificate, remove or rename the old key from the `/etc/pki/tls/private/` directory, and then install the new key there.

Note

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

From inside the `/etc/pki/tls/private` directory, check that the file ownership, group, and permission settings match the highly restrictive Amazon Linux defaults (owner=root, group=root, read/write for owner only). The commands would be as follows:

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

The commands above should yield the following result:

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

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

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

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

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

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

 **Note**

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

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

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

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

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

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

Step 3: Test and harden the security configuration

After your TLS is operational and exposed to the public, you should test how secure it really is. This is easy to do using online services such as [Qualys SSL Labs](#), which performs a free and thorough

analysis of your security setup. Based on the results, you may decide to harden the default security configuration by controlling which protocols you accept, which ciphers you prefer, and which you exclude. For more information, see [how Qualys formulates its scores](#).

Important

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

On the [Qualys SSL Labs](#) site, type the fully qualified domain name of your server, in the form `www.example.com`. After about two minutes, you receive a grade (from A to F) for your site and a detailed breakdown of the findings. Though the overview shows that the configuration is mostly sound, the detailed report flags several potential problems. For example:

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

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

To correct the TLS configuration

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

```
#SSLProtocol all -SSLv3
#SSLProxyProtocol all -SSLv3
```

2. Add the following directives:

```
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
SSLProxyProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```


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

Note

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

To modify the list of allowed ciphers

1. Open the configuration file `/etc/httpd/conf.d/ssl.conf` and find the section with commented-out examples for configuring **SSLCipherSuite** and **SSLProxyCipherSuite**.

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

Leave these as they are, and below them add the following directives:

Note

Though shown here on several lines for readability, each of these two directives must be on a single line without spaces between the cipher names.

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!
aNULL:!eNULL:!EXPORT:!DES:
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-
SHA

SSLProxyCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:
```

```
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:  
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!  
aNULL:!eNULL:!EXPORT:!DES:  
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
```

These ciphers are a subset of the much longer list of supported ciphers in OpenSSL. They were selected and ordered according to the following criteria:

- Support for forward secrecy
- Strength
- Speed
- Specific ciphers before cipher families
- Allowed ciphers before denied ciphers

The high-ranking ciphers have *ECDHE* in their names, for *Elliptic Curve Diffie-Hellman Ephemeral*; the *ephemeral* indicates forward secrecy. Also, RC4 is now among the forbidden ciphers near the end.

We recommend that you use an explicit list of ciphers instead relying on defaults or terse directives whose content isn't visible. The cipher list shown here is just one of many possible lists; for instance, you might want to optimize a list for speed rather than forward secrecy.

If you anticipate a need to support older clients, you can allow the DES-CBC3-SHA cipher suite.

Each update to OpenSSL introduces new ciphers and deprecates old ones. Keep your EC2 Amazon Linux instance up to date, watch for security announcements from [OpenSSL](#), and be alert to reports of new security exploits in the technical press.

2. Uncomment the following line by removing the "#":

```
#SSLHonorCipherOrder on
```

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

- Restart Apache. If you test the domain again on [Qualys SSL Labs](#), you should see that the RC4 vulnerability is gone.

Troubleshoot

- My Apache webserver won't start unless I enter a password**

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

You can remove the encryption and password requirement from the key. Assuming that you have a private encrypted RSA key called `custom.key` in the default directory, and that the password on it is **abcde12345**, run the following commands on your EC2 instance to generate an unencrypted version of the key.

```
[ec2-user ~]$ cd /etc/pki/tls/private/
[ec2-user private]$ sudo cp custom.key custom.key.bak
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out
custom.key.nocrypt
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key
[ec2-user private]$ sudo chown root.root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ sudo service httpd restart
```

Apache should now start without prompting you for a password.

Host a WordPress blog

This section includes tutorials that show you how to install, configure, and secure a WordPress blog on an Amazon EC2 instance.

Note

For the AL2023 WordPress tutorial, see [Tutorial: Host a WordPress blog on AL2023](#) in the *Amazon Linux 2023 User Guide*.

Host a WordPress blog on

- [Host a WordPress blog on Amazon Linux 2](#)

Host a WordPress blog on Amazon Linux 2

The following procedures will help you install, configure, and secure a WordPress blog on your Amazon Linux 2 instance. This tutorial is a good introduction to using Amazon EC2 in that you have full control over a web server that hosts your WordPress blog, which is not typical with a traditional hosting service.

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

Important

These procedures are intended for use with Amazon Linux 2. For more information about other distributions, see their specific documentation. Many steps in this tutorial do not work on Ubuntu instances. For help installing WordPress on an Ubuntu instance, see [WordPress](#) in the Ubuntu documentation. You can also use [CodeDeploy](#) to accomplish this task on Amazon Linux, macOS, or Unix systems.

Topics

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

Prerequisites

This tutorial assumes that you have launched an Amazon Linux 2 instance with a functional web server with PHP and database (either MySQL or MariaDB) support by following all of the steps in [Install LAMP on Amazon Linux](#) for [Install LAMP on Amazon Linux 2](#) for Amazon Linux 2. This tutorial also has steps for configuring a security group to allow HTTP and HTTPS traffic, as well as

several steps to ensure that file permissions are set properly for your web server. For information about adding rules to your security group, see [Add rules to a security group](#).

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

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

Install WordPress

Option: Complete this tutorial using automation

To complete this tutorial using AWS Systems Manager Automation instead of the following tasks, run the [automation document](#).

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

To download and unzip the WordPress installation package

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

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

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

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

To create a database user and database for your WordPress installation

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

1. Start the database server.

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

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

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

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


3. Create a user and password for your MySQL database. Your WordPress installation uses these values to communicate with your MySQL database.

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

Enter the following command, substituting a unique user name and password.

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

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

 **Note**

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

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

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

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

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

```
FLUSH PRIVILEGES;
```

7. Exit the mysql client.

```
exit
```

## To create and edit the wp-config.php file

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

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

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

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

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

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

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

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

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

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

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

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

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

#### Note

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

```
define('AUTH_KEY', ' #U$$+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/
Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY', 'Zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?
60P$eJT@;+(ndLg');
define('LOGGED_IN_KEY', 'ju}qwre3V*+8f_z0Wf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY', 'P(g62HeZxEes|LnI^i=H,[Xwk9I&[2s|:~?0N}VJM%?;v2v]v+;
+^9eXUahg@:~Cj');
define('AUTH_SALT', 'C$DpB4Hj[JK:~{qL`sRva:{:7yShy(9A@5wg+`JJVb1fk%-
Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.$#+S{n~1M&%@~gL>U>NV<zpD-@2-
Es7Q10-bp28EKv');
define('LOGGED_IN_SALT', ';j{00P*owZf)kVD+FVLn-~ >.|Y%Ug4#I^*LVd9QeZ^&XmK|
e(76miC+&W&+^0P/');
define('NONCE_SALT', '-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|
_e1tS)8_B/,.6[=UK<J_y9?JWG');
```



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

## To install your WordPress files under the Apache document root

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

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

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

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

### Important

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

## To allow WordPress to use permalinks

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

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

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

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

```
<Directory "/var/www/html">
#
Possible values for the Options directive are "None", "All",
or any combination of:
Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
Note that "MultiViews" must be named *explicitly* --- "Options All"
doesn't give it to you.
#
The Options directive is both complicated and important. Please see
http://httpd.apache.org/docs/2.4/mod/core.html#options
for more information.
#
Options Indexes FollowSymLinks

#
AllowOverride controls what directives may be placed in .htaccess files.
It can be "All", "None", or any combination of the keywords:
Options FileInfo AuthConfig Limit
#
AllowOverride None

#
Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

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

**Note**

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

```
AllowOverride All
```

4. Save the file and exit your text editor.

## To install the PHP graphics drawing library on Amazon Linux 2

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

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

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

To verify the installed version, use the following command:

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

The following is example output:

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

## To fix file permissions for the Apache web server

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

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

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

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

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

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

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

4. Recursively change the file permissions of `/var/www` and its subdirectories.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
```

#### Note

If you intend to also use WordPress as an FTP server, you'll need more permissive Group settings here. Please review the recommended [steps and security settings in WordPress](#) to accomplish this.

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

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

Run the WordPress installation script with Amazon Linux 2

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

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

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

2. Verify that the database server is running.

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

If the database service is not running, start it.

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

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

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

If the httpd service is not running, start it.

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

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

Next steps

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

Use a custom domain name

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

Configure your blog

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

Increase capacity

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

- Expand the storage space on your instance. For more information, see [Amazon EBS Elastic Volumes](#) in the *Amazon EBS User Guide*.
- Move your MySQL database to [Amazon RDS](#) to take advantage of the service's ability to scale easily.

Improve network performance of your internet traffic

If you expect your blog to drive traffic from users located around the world, consider [AWS Global Accelerator](#). Global Accelerator helps you achieve lower latency by improving internet traffic performance between your users' client devices and your WordPress application running on AWS. Global Accelerator uses the [AWS global network](#) to direct traffic to a healthy application endpoint in the AWS Region that is closest to the client.

Learn more about WordPress

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

For information about making your WordPress blog more secure, go to <https://wordpress.org/support/article/hardening-wordpress/>. For information about keeping your WordPress blog up-to-date, go to <https://wordpress.org/support/article/updating-wordpress/>.

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

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

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

To change your WordPress site URL with the wp-cli

1. Connect to your EC2 instance with SSH.

- Note the old site URL and the new site URL for your instance. The old site URL is likely the public DNS name for your EC2 instance when you installed WordPress. The new site URL is the current public DNS name for your EC2 instance. If you are not sure of your old site URL, you can use **curl** to find it with the following command.

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

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

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

- Download the **wp-cli** with the following command.

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

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

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

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

Amazon Machine Images (AMI)

An Amazon Machine Image (AMI) is a supported and maintained image provided by AWS that provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you require multiple instances with the same configuration. You can use different AMIs to launch instances when you require instances with different configurations.

An AMI includes the following:

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

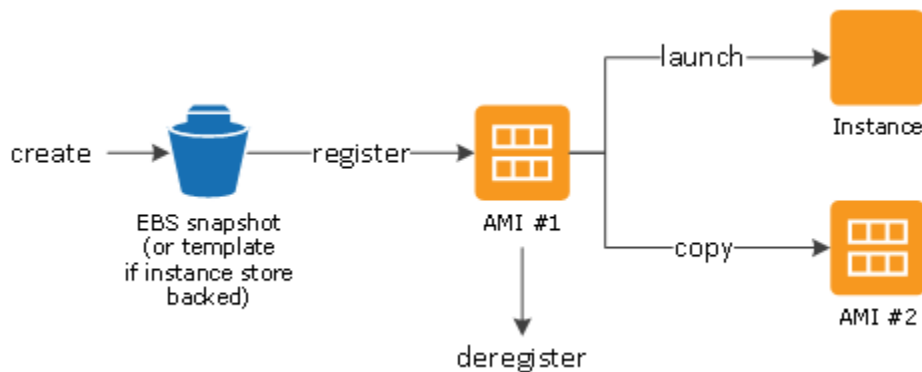
Amazon Machine Image (AMI) topics

- [Use an AMI](#)
- [Create your own AMI](#)
- [Buy, share, and sell AMIs](#)
- [Deregister your AMI](#)
- [AL2023 and Amazon Linux 2](#)
- [AMI types](#)
- [Linux AMI virtualization types](#)
- [Boot modes](#)
- [Find a Linux AMI](#)
- [Shared AMIs](#)
- [Paid AMIs](#)
- [AMI lifecycle](#)
- [Use encryption with EBS-backed AMIs](#)
- [Monitor AMI events using Amazon EventBridge](#)
- [Understand AMI billing information](#)
- [Amazon Linux](#)

- [User provided kernels](#)
- [Configure the Amazon Linux 2 MATE desktop connection](#)
- [AMI quotas](#)

Use an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI within the same AWS Region or to different AWS Regions. When you no longer require an AMI, you can deregister it.



You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community. For more information, see [AMI types](#) and [Find a Linux AMI](#).

After you launch an instance from an AMI, you can connect to it. When you are connected to an instance, you can use it just like you use any other server. For information about launching, connecting, and using your instance, see [Tutorial: Get started with Amazon EC2 Linux instances](#).

Create your own AMI

You can launch an instance from an existing AMI, customize the instance (for example, [install software](#) on the instance), and then save this updated configuration as a custom AMI. Instances launched from this new custom AMI include the customizations that you made when you created the AMI.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon Elastic Block Store (Amazon EBS) volume or an

instance store volume. For more information about the root device volume, see [Amazon EC2 instance root volume](#).

- To create an Amazon EBS-backed AMI, see [Create an Amazon EBS-backed Linux AMI](#).
- To create an instance store-backed AMI, see [Create an instance store-backed Linux AMI](#).

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

Buy, share, and sell AMIs

After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it. Building a safe, secure, usable AMI for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information about how to create and use shared AMIs, see [Shared AMIs](#).

You can purchase AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users. For more information about buying or selling AMIs, see [Paid AMIs](#).

Deregister your AMI

You can deregister an AMI when you have finished with it. After you deregister an AMI, it can't be used to launch new instances. Existing instances launched from the AMI are not affected. For more information, see [Deregister \(delete\) an AMI](#).

AL2023 and Amazon Linux 2

The latest release of Amazon Linux, AL2023, is optimized for Amazon EC2 and is provided at no additional cost to Amazon EC2 users. The features of AL2023 include a predictable release cadence, frequent updates, and long-term support.

For more information about AL2023 features and launching an AL2023 AMI, see:

- [AL2023 Features](#)
- [Get started with AL2023](#)

Amazon Linux 2 provides a stable, secure, and high performance execution environment for applications running on Amazon EC2. For information about Amazon Linux 2 EOL, see [Amazon Linux 2 FAQs](#).

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

AMI types

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

- Region (see [Regions and Zones](#))
- Operating system
- Architecture (32-bit or 64-bit)
- [Launch permissions](#)
- [Storage for the root device](#)

Launch permissions

The owner of an AMI determines its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts, organizations, 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 [Shared AMIs](#). Developers can charge for their AMIs. For more information, see [Paid AMIs](#).

Storage for the root device

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.
- 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.

For more information, see [Amazon EC2 instance root volume](#).

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

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
Boot time for an instance	Usually less than 1 minute	Usually less than 5 minutes
Size limit for a root device	64 TiB**	10 GiB
Root device volume	EBS volume	Instance store volume
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.
Modifications		

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
	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
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

* 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 [Change the root volume to persist](#).

** 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

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

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.

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

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

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

Stopped state

You can stop an instance that has an EBS volume for its root device, but you can't stop an instance that has an instance store volume for its root device.

Stopping causes the instance to stop running (its status goes from `running` to `stopping` to `stopped`). A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because instances with an instance store volume for the root device can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see [Stop and start Amazon EC2 instances](#).

Default data storage and persistence

Instances that have an instance store volume for the root device automatically have instance store available (the root volume contains the root partition and you can store additional data). You can add persistent storage to your instance by attaching one or more EBS volumes. Any data on an instance store volume is deleted when the instance fails or terminates. For more information, see [Instance store volume and data lifetime](#).

Instances that have Amazon EBS for the root device automatically have an EBS volume attached. The volume appears in your list of volumes like any other. With most instance types, instances that have an EBS volume for the root device don't have instance store volumes by default. You can add instance store volumes or additional EBS volumes using a block device mapping. For more information, see [Block device mappings](#).

Boot times

Instances launched from an Amazon EBS-backed AMI launch faster than instances launched from an instance store-backed AMI. When you launch an instance from an instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available. With an Amazon EBS-backed AMI, only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available. However, the performance of an instance that uses an EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume. When you stop and restart the instance, it launches quickly, because the state is stored in an EBS volume.

AMI creation

To create Linux AMIs backed by instance store, you must create an AMI from your instance on the instance itself using the Amazon EC2 AMI tools.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an AMI from a running instance. For more information, see [Create an Amazon EBS-backed Linux AMI](#).

How you're charged

With AMIs backed by instance store, you're charged for instance usage and storing your AMI in Amazon S3. With AMIs backed by Amazon EBS, you're charged for instance usage, EBS volume storage and usage, and storing your AMI as an EBS snapshot.

With Amazon EC2 instance store-backed AMIs, each time you customize an AMI and create a new one, all of the parts are stored in Amazon S3 for each AMI. So, the storage footprint for each customized AMI is the full size of the AMI. For Amazon EBS-backed AMIs, each time you customize an AMI and create a new one, only the changes are stored. So, the storage footprint for subsequent AMIs that you customize after the first is much smaller, resulting in lower AMI storage charges.

When an instance with an EBS volume for its root device is stopped, you're not charged for instance usage; however, you're still charged for volume storage. As soon as you start your instance, we charge a minimum of one minute for usage. After one minute, we charge only for the seconds used. For example, if you run an instance for 20 seconds and then stop it, we charge for a full one minute. If you run an instance for 3 minutes and 40 seconds, we charge for exactly 3 minutes and 40 seconds of usage. We charge you for each second, with a one-minute minimum, that you keep the instance running, even if the instance remains idle and you don't connect to it.

Linux AMI virtualization types

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

For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch your instances. For more information about current generation instance types, see [Amazon EC2 Instance Types](#). If you are using previous generation instance types and would like to upgrade, see [Upgrade Paths](#) and [Change the instance type](#).

The following table compares HVM and PV AMIs.

	HVM	PV
Description	HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.	PV AMIs boot with a special boot loader called PV-GRUB, which starts the boot cycle and then chain loads the kernel specified in the <code>menu.lst</code> file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization. Historically, PV guests had better performance than HVM guests in many cases, but because of enhancements in HVM virtualization and the availability of PV drivers for HVM AMIs, this is no longer true. For more information about PV-GRUB and its use in Amazon EC2, see User provided kernels .

	HVM	PV
Support for hardware extensions	<p>Yes. Unlike PV guests, HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. For more information on CPU virtualization extensions available in Amazon EC2, see Intel Virtualization Technology on the Intel website. HVM AMIs are required to take advantage of enhanced networking and GPU processing. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform; HVM virtualization provides this access. For more information, see Enhanced networking on Linux.</p>	<p>No, they cannot take advantage of special hardware extensions such as enhanced networking or GPU processing.</p>
Supported instance types	<p>All current generation instance types support HVM AMIs.</p>	<p>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.</p>

	HVM	PV
Supported Regions	All Regions support HVM instances.	Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Sydney), Europe (Frankfurt), Europe (Ireland) , South America (São Paulo), US East (N. Virginia), US West (N. California), and US West (Oregon)
How to find	Verify that the virtualization type of the AMI is set to hvm, using the console or the describe-images command. For more information, see Find a Linux AMI .	Verify that the virtualization type of the AMI is set to paravirtual , using the console or the describe-images command. For more information, see Find a Linux AMI .

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.

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

Running Intel and AMD instances types on UEFI

[Most Intel and AMD instance types](#) can run on both UEFI and Legacy BIOS. To use UEFI, you must select an AMI with the boot mode parameter set either to `uefi` or `uefi-preferred`, and the operating system contained in the AMI must be configured to support UEFI.

Boot mode topics

- [Launch an instance](#)
- [Determine the boot mode parameter of an AMI](#)
- [Determine the supported boot modes of an instance type](#)
- [Determine the boot mode of an instance](#)
- [Determine the boot mode of the operating system](#)
- [Set the boot mode of an AMI](#)
- [UEFI variables](#)
- [UEFI Secure Boot](#)

Launch an instance

You can launch an instance in UEFI or Legacy BIOS boot mode.

Topics

- [Limitations](#)
- [Considerations](#)
- [Requirements for launching an instance on UEFI](#)

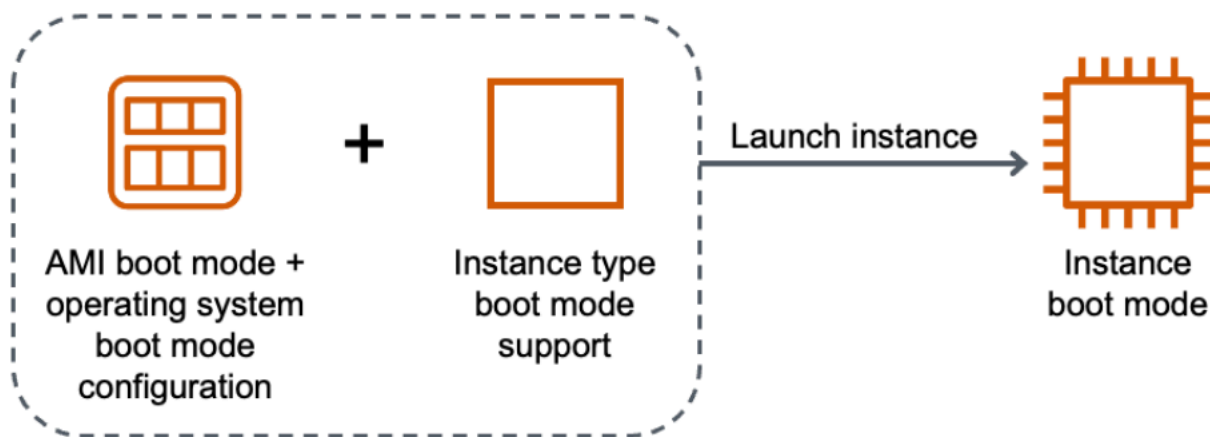
Limitations

UEFI boot is not supported in Local Zones, Wavelength Zones, or with AWS Outposts.

Considerations

Consider the following when launching an instance:

- The boot mode of the instance is determined by the configuration of the AMI, the operating system contained in it, and the instance type, illustrated by the following image:



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

AMI boot mode parameter	Operating system boot mode configuration	Instance type boot mode support	Resulting instance boot mode
No boot mode specified - x86	Legacy BIOS	UEFI and Legacy BIOS	Legacy BIOS

- Default boot modes:
 - Graviton instance types: UEFI
 - Intel and AMD instance types: Legacy BIOS
- Intel and AMD instance types that support UEFI, in addition to Legacy BIOS:
 - All instances built on the AWS Nitro System, except: bare metal instances, DL1, G4ad, P4, u-3tb1, u-6tb1, u-9tb1, u-12tb1, u-18tb1, u-24tb1, and VT1

To see the available instance types for Linux that support UEFI in a specific Region

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 with the `--region` parameter. If you omit the `--region` parameter, your [default Region](#) is used in the request. 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 CLI

```
$ aws ec2 describe-instance-types --filters Name=supported-boot-mode,Values=uefi --
query "InstanceTypes[*].[InstanceType]" --output text | sort

a1.2xlarge
a1.4xlarge
a1.large
a1.medium
a1.metal
a1.xlarge
c5.12xlarge
...
```

PowerShell

```
PS C:\> Get-EC2InstanceType | `
  Where-Object {$_.SupportedBootModes -Contains "uefi"} | `
  Sort-Object InstanceType | `
  Format-Table InstanceType -GroupBy CurrentGeneration

CurrentGeneration: False

InstanceType
-----
a1.2xlarge
a1.4xlarge
a1.large
a1.medium
a1.metal
a1.xlarge

CurrentGeneration: True

InstanceType
-----
c5.12xlarge
c5.18xlarge
c5.24xlarge
c5.2xlarge
c5.4xlarge
c5.9xlarge
...
```

To see the available instance types for Linux that support UEFI Secure Boot and persist non-volatile variables in a specific Region

Currently, bare metal instances do not support UEFI Secure Boot and non-volatile variables. Use the [describe-instance-types](#) command as described in the preceding example, but filter out the bare metal instances by including the `Name=bare-metal,Values=false` filter. For information about UEFI Secure Boot, see [UEFI Secure Boot](#).

AWS CLI

```
$ aws ec2 describe-instance-types --filters Name=supported-boot-mode,Values=uefi
Name=bare-metal,Values=false --query "InstanceTypes[*].[InstanceType]" --output
text | sort
```

```
a1.2xlarge
a1.4xlarge
a1.large
a1.medium
...
```

PowerShell

```
PS C:\> Get-EC2InstanceType | `
    Where-Object { `
        $_.SupportedBootModes -Contains "uefi" -and `
        $_.BareMetal -eq $False
    } | `
    Sort-Object InstanceType | `
    Format-Table InstanceType, SupportedBootModes, BareMetal,
    @{Name="SupportedArchitectures";
    Expression={$_.ProcessorInfo.SupportedArchitectures}}
```

InstanceType	SupportedBootModes	BareMetal	SupportedArchitectures
a1.2xlarge	{uefi}	False	arm64
a1.4xlarge	{uefi}	False	arm64
a1.large	{uefi}	False	arm64
a1.medium	{uefi}	False	arm64
a1.xlarge	{uefi}	False	arm64
c5.12xlarge	{legacy-bios, uefi}	False	x86_64
c5.18xlarge	{legacy-bios, uefi}	False	x86_64

Requirements for launching an instance on UEFI

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

Instance type

When launching an instance, you must select an instance type that supports UEFI. For more information, see [Determine the supported boot modes of an instance type](#).

AMI

When launching an instance, you must select an AMI that is configured for UEFI. The AMI must be configured 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](#).
- **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 AMI](#).

AWS only provides Linux AMIs that are configured to support UEFI for Graviton-based instance types. To use Linux on other UEFI instance types, you must [configure the AMI](#), import the AMI through [VM Import/Export](#), or import the AMI through [CloudEndure](#).

For the supported Windows AMIs, see [Requirements for launching an instance on UEFI](#) in the *Amazon EC2 User Guide for Windows Instances*.

Determine the boot mode parameter of an AMI

The AMI boot mode parameter is optional. An AMI can have one of the following boot mode parameter values: `uefi`, `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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**, and then select the AMI.
3. 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 (console)

When launching an instance using the launch instance wizard, at the step to select an AMI, inspect the **Boot mode** field. For more information, see [Application and OS Images \(Amazon Machine Image\)](#).

AWS CLI

To determine the boot mode parameter of an AMI (AWS CLI)

Use the [describe-images](#) operation to determine the boot mode of an AMI.

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

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

In the output, the `BootMode` field 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.

PowerShell

To determine the boot mode parameter of an AMI (Tools for PowerShell)

Use the [Get-EC2Image](#) Cmdlet to determine the boot mode of an AMI.

```
PS C:\> Get-EC2Image -Region us-east-1 -ImageId ami-0abcdef1234567890 | Format-List  
Name, BootMode, TpmSupport
```

```
Name          : TPM-Windows_Server-2016-English-Full-Base-2023.05.10  
BootMode      : uefi  
TpmSupport    : v2.0
```

In the output, the `BootMode` field 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 instance type

You can use the AWS CLI or the Tools for PowerShell to determine the supported boot modes of an instance type.

To determine the supported boot modes of an instance type

You can use the following methods to determine the supported boot modes of an instance type.

AWS CLI

You can use the [describe-instance-types](#) command to determine the supported boot modes of an instance type. By including the `--query` parameter, you can filter the output. In this example, the output is filtered to return only the supported boot modes.

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

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

Expected output:

```
[
  [
    "legacy-bios",
    "uefi"
  ]
]
```

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

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

Expected output:

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

PowerShell

You can use the [Get-EC2InstanceType](#) (Tools for PowerShell) 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 -Region us-east-1 -InstanceType m5.2xlarge | Format-List
InstanceType, SupportedBootModes
```

Expected output:

```
InstanceType      : m5.2xlarge
SupportedBootModes : {legacy-bios, uefi}
```

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

```
Get-EC2InstanceType -Region us-east-1 -InstanceType t2.xlarge | Format-List
InstanceType, SupportedBootModes
```

Expected output:

```
InstanceType      : t2.xlarge
SupportedBootModes : {legacy-bios}
```

Determine the boot mode of an 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 (console)

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

AWS CLI

To determine the boot mode of an instance (AWS CLI)

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 create the instance.

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

{
  "Reservations": [
    {
      "Groups": [],
      "Instances": [
        {
          "AmiLaunchIndex": 0,
          "ImageId": "ami-0e2063e7f6dc3bee8",
          "InstanceId": "i-1234567890abcdef0",
          "InstanceType": "m5.2xlarge",
          ...
        },
        {
          "BootMode": "uefi",
          "CurrentInstanceBootMode": "uefi"
        }
      ],
      "OwnerId": "1234567890",
      "ReservationId": "r-1234567890abcdef0"
    }
  ]
}
```

PowerShell

To determine the boot mode of an instance (Tools for PowerShell)

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 create the instance.

[Get-EC2Image](#) (AWS Tools for Windows PowerShell)

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances | Format-List BootMode,
CurrentInstanceBootMode, InstanceType, ImageId
```

```
BootMode           : uefi
```

```
CurrentInstanceBootMode : uefi
InstanceType             : c5a.large
ImageId                 : ami-0265446f88eb4021b
```

In the output, the following parameters describe the boot mode:

- **BootMode** – The boot mode of the AMI that was used to create the instance.
- **CurrentInstanceBootMode** – The boot mode that is used to boot the instance at launch or start.

Determine the boot mode of the operating system

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.

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
```

Set the boot mode of an AMI

When you create an AMI using the [register-image](#) command, you can set the boot mode of the AMI to either `uefi`, `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.

Note

If you set the AMI boot mode to `uefi-preferred`, the operating system must support the ability to boot both UEFI and Legacy BIOS.

Currently, you can't use the [register-image](#) command to create an AMI that supports both [NitroTPM](#) and UEFI Preferred.

⚠ 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`.

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

You can't set the boot mode of an AMI using the [create-image](#) command. With [create-image](#), the AMI inherits the boot mode of the EC2 instance used for creating the AMI. For example, if you create an AMI from an EC2 instance running on Legacy BIOS, the AMI boot mode will be configured as `legacy-bios`. If you create an AMI from an EC2 instance that was launched using an AMI with a boot mode set to `uefi-preferred`, the AMI that is created will also have its boot mode set to `uefi-preferred`.

⚠ Warning

Setting the AMI boot mode parameter does not automatically configure the operating system for the specified boot mode. Before proceeding with these steps, 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 will not be usable. The modifications that are required are operating system-specific. For more information, see the manual for your operating system.

To set the boot mode of an AMI (AWS CLI)

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.

Note

If you don't perform this step, the AMI will not be usable.

2. To find the volume ID of the instance, use the [describe-instances](#) command. You'll create a snapshot of this volume in the next step.

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

Expected output

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

3. To create a snapshot of the volume, use the [create-snapshot](#) command. Use the volume ID from the previous step.

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

Expected output

```
{
  "Description": "add text",
  "Encrypted": false,
  "OwnerId": "123",
  "Progress": "",
  "SnapshotId": "snap-01234567890abcdef",
  "StartTime": "",
  "State": "pending",
```

```
"VolumeId": "vol-1234567890abcdef0",
"VolumeSize": 30,
"Tags": []
}
```

4. Note the snapshot ID in the output from the previous step.
5. Wait until the snapshot creation is completed before going to the next step. To query the state of the snapshot, use the [describe-snapshots](#) command.

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

Example output

```
{
  "Snapshots": [
    {
      "Description": "This is my snapshot",
      "Encrypted": false,
      "VolumeId": "vol-049df61146c4d7901",
      "State": "completed",
      "VolumeSize": 8,
      "StartTime": "2019-02-28T21:28:32.000Z",
      "Progress": "100%",
      "OwnerId": "012345678910",
      "SnapshotId": "snap-01234567890abcdef",
      ...
    }
  ]
}
```

6. To create a new AMI, use the [register-image](#) command. Use the snapshot ID that you noted in the earlier step.
 - To set the boot mode to UEFI, add the `--boot-mode` parameter to the command and specify `uefi` as the value.

```
aws ec2 register-image \
  --region us-east-1 \
  --description "add description" \
  --name "add name" \
  --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-01234567890abcdef,DeleteOnTermination=true}" \
  --architecture x86_64 \
  --root-device-name /dev/sda1 \
  --virtualization-type hvm \
```

```
--ena-support \  
--boot-mode uefi
```

- To set the boot mode to `uefi-preferred`, add the `--boot-mode` parameter to the command and specify `uefi-preferred` as the value.

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

Expected output

```
{  
  "ImageId": "ami-new_ami_123"  
}
```

7. To verify that the newly-created AMI has the boot mode that you specified in the previous step, use the [describe-images](#) command.

```
aws ec2 describe-images --region us-east-1 --image-id ami-new_ami_123
```

Expected output

```
{  
  "Images": [  
    {  
      "Architecture": "x86_64",  
      "CreationDate": "2021-01-06T14:31:04.000Z",  
      "ImageId": "ami-new_ami_123",  
      "ImageLocation": "",  
      ...  
      "BootMode": "uefi"  
    }  
  ]  
}
```

```
]
}
```

8. Launch a new instance using the newly-created AMI.

If the AMI boot mode is `uefi` or `legacy-bios`, instances created from this AMI will have the same boot mode as the AMI. If the AMI boot mode is `uefi-preferred`, the instance will boot using UEFI if the instance type supports UEFI; otherwise, the instance will boot using Legacy BIOS. For more information, see [Considerations](#).

9. To verify that the new instance has the expected boot mode, use the [describe-instances](#) command.

UEFI variables

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

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.

Topics

- [How UEFI Secure Boot works](#)
- [Launch an instance with UEFI Secure Boot support](#)
- [Verify whether an instance is enabled for UEFI Secure Boot](#)
- [Create a Linux AMI to support UEFI Secure Boot](#)
- [How the AWS binary blob is created](#)

How UEFI Secure Boot works

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.

Launch an instance with UEFI Secure Boot support

When you [launch an instance](#) with the following prerequisites, the instance will automatically validate UEFI boot binaries against its UEFI Secure Boot database. 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. Typically, UEFI Secure Boot is configured as part of the AMI. If you create a new AMI with different parameters from the base AMI, such as changing the `UefiData` within the AMI, you can disable UEFI Secure Boot.

Prerequisites**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 to support UEFI Secure Boot](#).

Windows AMIs

To launch a Windows instance, the Windows AMI must have UEFI Secure Boot enabled.

The following Windows AMIs are preconfigured to enable UEFI Secure Boot with Microsoft keys:

- TPM-Windows_Server-2022-English-Core-Base
- TPM-Windows_Server-2022-English-Full-Base
- TPM-Windows_Server-2022-English-Full-SQL_2022_Enterprise
- TPM-Windows_Server-2022-English-Full-SQL_2022_Standard
- TPM-Windows_Server-2019-English-Core-Base
- TPM-Windows_Server-2019-English-Full-Base
- TPM-Windows_Server-2019-English-Full-SQL_2019_Enterprise
- TPM-Windows_Server-2019-English-Full-SQL_2019_Standard
- TPM-Windows_Server-2016-English-Core-Base
- TPM-Windows_Server-2016-English-Full-Base

Currently, we do not support importing Windows with UEFI Secure Boot by using the [import-image](#) command.

Instance type

- **Supported:** All virtualized instance types that support UEFI also support UEFI Secure Boot. For the instance types that support UEFI Secure Boot, see [Considerations](#).
- **Not supported:** Bare metal instance types do not support UEFI Secure Boot.

Verify whether an instance 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, see [Install software packages on an Amazon Linux instance](#). For other distributions, see their specific documentation.

To verify whether a Linux instance is enabled for UEFI Secure Boot

Run the following command as root on the instance.

```
mokutil --sb-state
```

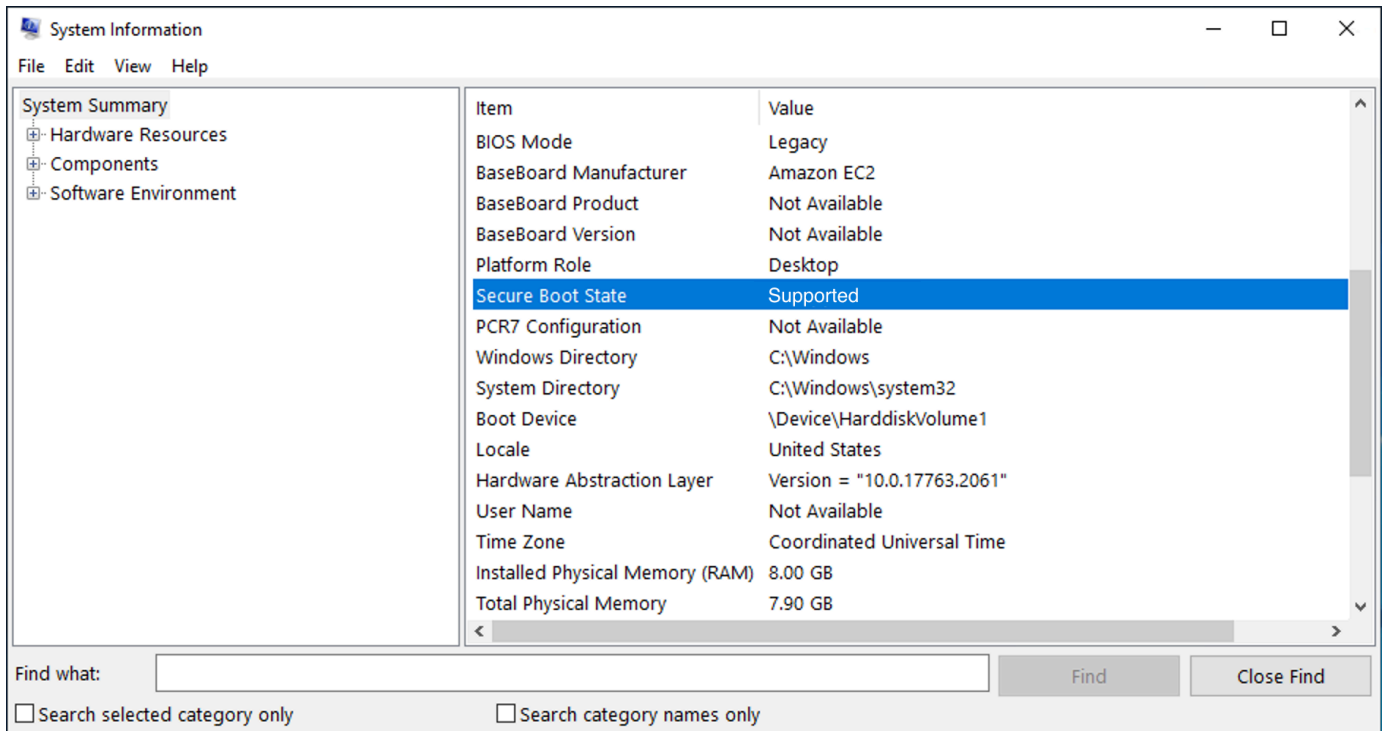
Expected output:

- 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. Open the `msinfo32` tool.
2. Check the **Secure Boot State** field. **Supported** indicates that UEFI Secure Boot is enabled.



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 website*.

Create a Linux AMI to support UEFI Secure Boot

The following procedures describe how to create your own UEFI variable store for secure boot with custom-made private keys. Amazon Linux supports UEFI Secure Boot starting with AL2023 release 2023.1. For more information, see [UEFI Secure Boot](#) in the *AL2023 User Guide*.

⚠ Important

The following procedures for creating an AMI to support UEFI Secure Boot are 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](#) AWS CLI 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 to support UEFI Secure Boot, first create the three key pairs, and then complete either Option A or Option B:

- [Create three key pairs](#)
- [Option A: Add keys to the variable store from within the instance](#)
- [Option B: Create a binary blob containing a pre-filled variable store](#)

Note

These instructions can only be used to create a Linux AMI. If you need a Windows AMI, use one of the supported Windows AMIs. For more information, see [Launch an instance with UEFI Secure Boot support](#) in the *Amazon EC2 User Guide for Windows Instances*.

Create three 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.

Topics

- [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.

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.

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 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 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 [Considerations](#).
- 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-0123456789example
```

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 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 Linux 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.

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.

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/aws-labs/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 -0 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](#) 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
```

How the AWS binary blob is created

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. Download an updated dbx change request from the following link.

```
https://uefi.org/revocationlistfile
```

7. The dbx change request that you downloaded in the previous step is already signed with the Microsoft KEK, so you need to strip or unpack it. You can use the following links.

```
https://gist.github.com/out0xb2/f8e0bae94214889a89ac67fceb37f8c0
```

```
https://support.microsoft.com/en-us/topic/microsoft-guidance-for-applying-secure-  
boot-dbx-update-e3b9e4cb-a330-b3ba-a602-15083965d9ca
```

8. Build a UEFI variable store using the `uefivars.py` script.

```
./uefivars.py -i none -o aws -0 uefiblob-microsoft-keys-empty-pk.bin -P ~/PK.esl -K  
~/MS_Win_KEK.esl --db ~/MS_db.esl --dbx ~/dbx-2021-April.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  
~/dbx-2021-April.bin
```

Expected output

```
Replacing PK  
Replacing KEK  
Replacing db  
Replacing dbx
```

Find a Linux AMI

Before you can launch an instance, you must select an AMI from which to launch the instance. When you select an AMI, consider the following requirements you might have for the instances that you want to launch:

- The Region – AMI IDs are unique to each AWS Region.
- The operating system
- The architecture: 32-bit (i386), 64-bit (x86_64), or 64-bit ARM (arm64)
- The root device type: Amazon EBS or instance store
- The provider (for example, Amazon Web Services)
- Additional software (for example, SQL Server)

If you want to find a Windows AMI, see [Find a Windows AMI](#) in the *Amazon EC2 User Guide for Windows Instances*.

If you want to find an AL2023 AMI, see [Get started with AL2023](#) in the *AL2023 User Guide*.

If you want to find an Ubuntu AMI, see their [EC2 AMI Locator](#).

If you want to find a RedHat AMI, see the RHEL [knowledgebase article](#).

Find a Linux AMI topics

- [Find a Linux AMI using the Amazon EC2 console](#)

- [Find an AMI using the AWS CLI](#)
- [Find the latest Amazon Linux AMI using Systems Manager](#)
- [Use a Systems Manager parameter to find an AMI](#)

Find a Linux AMI using the Amazon EC2 console

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

To find a Linux AMI using the launch 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.
3. From the console dashboard, choose **Launch instance**.
4. (New console) 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 \(Amazon Machine Image\)](#).

(Old console) On the **Quick Start** tab, select from one of the commonly used AMIs in the list. If you don't see the AMI that you want to use, choose the **My AMIs**, **AWS Marketplace**, or **Community AMIs** tab to find additional AMIs. For more information, see [Step 1: Choose an Amazon Machine Image \(AMI\)](#).

To find a Linux 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.
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 Linux 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**. 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 [Storage for the root device](#).
 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 instance using the new launch instance wizard](#). If you're not ready to launch the instance now, make note of the AMI ID for later.

For more information about finding an AL2023 AMI, see [Get started with AL2023](#) in the *AL2023 User Guide*.

Find an AMI using the AWS CLI

You can use AWS CLI commands for Amazon EC2 to list only the Linux AMIs that match your requirements. After locating an AMI that matches your requirements, make note of its ID so that you can use it to launch instances. For more information, see [Launch your instance](#) in the *AWS Command Line Interface User Guide*.

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

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

You can add the following filter to the previous command to display only AMIs backed by Amazon EBS.

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

⚠ Important

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

Find the latest Amazon Linux AMI using Systems Manager

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

To find the latest AL2023 AMI using AWS Systems Manager, see [Get started with AL2023](#).

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

```
/aws/service/ami-amazon-linux-latest
```

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

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

To launch an instance using a public parameter

The following example uses the EC2-provided public parameter to launch an `m5.xlarge` instance using the latest Amazon Linux 2 AMI.

To specify the parameter in the command, use the following syntax: `resolve:ssm:public-parameter`, where `resolve:ssm` is the standard prefix and `public-parameter` is the path and name of the public parameter.

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

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



```
--key-name MyKeyPair
```

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

Use a Systems Manager parameter to find an AMI

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

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

When you create a parameter that points to an AMI ID, make sure that you specify the data type as `aws:ec2:image`. 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*.

Systems Manager parameter topics

- [Use cases](#)
- [Permissions](#)
- [Limitations](#)
- [Launch an instance using a Systems Manager parameter](#)

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 `ssm:DescribeParameters` and `ssm:GetParameters` to your IAM policy. `ssm:DescribeParameters` grants your users permission to view and select Systems Manager parameters. `ssm:GetParameters` grants your users permission to retrieve the values of the Systems Manager parameters. You can also restrict access to specific Systems Manager parameters. For more information, see [Use the EC2 launch 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.

Launch an instance using a Systems Manager parameter

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

New console

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

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose **Launch instance**.
4. 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 instance using the new launch instance wizard](#).

Old console

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

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.

3. From the console dashboard, choose **Launch instance**.
4. Choose **Search by Systems Manager parameter** (at top right).
5. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.
6. Choose **Search**. The AMIs that match the AMI ID appear in the list.
7. Select the AMI from the list, and choose **Select**.

For more information about launching an instance from an AMI using the launch instance wizard, see [Step 1: Choose an Amazon Machine Image \(AMI\)](#).

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

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

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

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

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

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

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

The following example uses version 2 of the parameter.

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

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

To launch an instance using a public parameter provided by AWS

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

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

Shared AMIs

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

You use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center, and perform the appropriate due diligence. We recommend that you get an AMI from a trusted source, 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` 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](#)
- [Make an AMI public](#)
- [Share an AMI with specific organizations or organizational units](#)
- [Share an AMI with specific AWS accounts](#)
- [Cancel having an AMI shared with your AWS account](#)
- [Use bookmarks](#)
- [Guidelines for shared Linux AMIs](#)

If you're looking for information about other topics

- For information about creating an AMI, see [Create an instance store-backed Linux AMI](#) or [Create an Amazon EBS-backed Linux AMI](#).
- For information about building, delivering, and maintaining your applications on the AWS Marketplace, see the [AWS Marketplace Documentation](#).

Find shared AMIs

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

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 AMI](#).

Topics

- [Find a shared AMI \(console\)](#)
- [Find a shared AMI \(AWS CLI\)](#)
- [Use shared AMIs](#)

Find a shared AMI (console)

To find a shared private AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Private images**. All AMIs that have been shared with you are listed. To granulate your search, choose the **Search** bar and use the filter options provided in the menu.

To find a shared public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Public images**. To granulate your search, choose the **Search** field and use the filter options provided in the menu.

To find Amazon's shared public AMIs using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Public images**.
4. Choose the **Search** field and then, from the menu options that appear, choose **Owner alias**, then **=**, and then **amazon** to display only Amazon's public images.

To find a shared public AMI from a verified provider using the console

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. The **Verified provider** label indicates the AMIs that are from Amazon or a verified partner.

Find a shared AMI (AWS CLI)

Use the [describe-images](#) command (AWS CLI) to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

Example: List all public AMIs

The following command lists all public AMIs, including any public AMIs that you own.

```
aws ec2 describe-images --executable-users all
```

Example: List AMIs with explicit launch permissions

The following command lists the AMIs for which you have explicit launch permissions. This list does not include any AMIs that you own.

```
aws ec2 describe-images --executable-users self
```

Example: List AMIs owned by 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` 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
```

Example: List AMIs owned by an account

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

```
aws ec2 describe-images --owners 123456789012
```

Example: Scope AMIs using a filter

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

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


Use shared AMIs

Before you use a shared AMI, take the following steps to confirm that there are no pre-installed credentials that would allow unwanted access to your instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party. Check the documentation for the Linux distribution used by the AMI for information about improving the security of the system.

To ensure that you don't accidentally lose access to your instance, we recommend that you initiate two SSH sessions and keep the second session open until you've removed credentials that you don't recognize and confirmed that you can still log into your instance using SSH.

1. Identify and disable any unauthorized public SSH keys. The only key in the file should be the key you used to launch the AMI. The following command locates `authorized_keys` files:

```
[ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
```

2. Disable password-based authentication for the root user. Open the `sshd_config` file and edit the `PermitRootLogin` line as follows:

```
PermitRootLogin without-password
```

Alternatively, you can disable the ability to log into the instance as the root user:

```
PermitRootLogin No
```

Restart the `sshd` service.

3. Check whether there are any other 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](#).

Make an AMI public

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.

To allow only specific accounts to use your AMI to launch instances, see [Share an AMI with specific AWS accounts](#).

Topics

- [Considerations](#)
- [Share an AMI with all AWS accounts \(share publicly\)](#)
- [Block public access to your AMIs](#)

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 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 [Guidelines for 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}]"
```

Block public access to your AMIs

To prevent the public sharing of your AMIs, you can enable *block public access for AMIs*. This setting is enabled at the account level, but you need to enable it in each AWS Region in which you want to prevent the public sharing of your AMIs.

When block public access is enabled, any attempt to make an AMI public is automatically blocked. However, if you already have public AMIs, they will remain publicly available.

If you want to publicly share AMIs, you'll need to 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.

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](#)
- [Required IAM permissions](#)
- [Enable block public access for AMIs](#)
- [Disable block public access for AMIs](#)
- [View the block public access state 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.

Required IAM permissions

To use block public access for AMIs, you must have the following IAM permissions:

- `EnableImageBlockPublicAccess`
- `DisableImageBlockPublicAccess`
- `GetImageBlockPublicAccessState`

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.

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** check box, and then choose **Update**.

Note

The API can take up to 10 minutes to configure this setting. During this time, the value will be **New public sharing allowed**. When the API has completed the configuration, the value will automatically change to **New public sharing blocked**.

AWS CLI

To enable block public access for AMIs in the specified Region

Use the [enable-image-block-public-access](#) command and specify the Region in which to enable block public access for AMIs. For the `--image-block-public-access-state` parameter, specify `block-new-sharing`.

```
aws ec2 enable-image-block-public-access \  
  --region us-east-1 \  
  --image-block-public-access-state block-new-sharing
```

Expected output

```
{  
  "ImageBlockPublicAccessState": "block-new-sharing"  
}
```

Note

The API can take up to 10 minutes to configure this setting. During this time, if you run the [get-image-block-public-access-state](#) command, the response will be `unblocked`. When the API has completed the configuration, the response will be `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.

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** check box, and then choose **Update**.
7. Enter **confirm** when prompted for confirmation, and then choose **Allow public sharing**.

Note

The API can take up to 10 minutes to configure this setting. During this time, the value will be **New public sharing blocked**. When the API has completed the configuration, the value will automatically change to **New public sharing allowed**.

AWS CLI

To disable block public access for AMIs in the specified Region

Use the [disable-image-block-public-access](#) command and specify the Region in which to disable block public access for AMIs.

```
aws ec2 disable-image-block-public-access --region us-east-1
```

Expected output

```
{
  "ImageBlockPublicAccessState": "unblocked"
}
```

Note

The API can take up to 10 minutes to configure this setting. During this time, if you run the [get-image-block-public-access-state](#) command, the response will be `block-new-sharing`. When the API has completed the configuration, the response will be `unblocked`.

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.

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 in the specified Region

Use the [get-image-block-public-access-state](#) command and specify the Region in which to get the block public access state for AMIs.

```
aws ec2 get-image-block-public-access-state --region us-east-1
```

Expected output – The value is either `block-new-sharing` or `unblocked`.

```
{
  "ImageBlockPublicAccessState": "block-new-sharing"
}
```

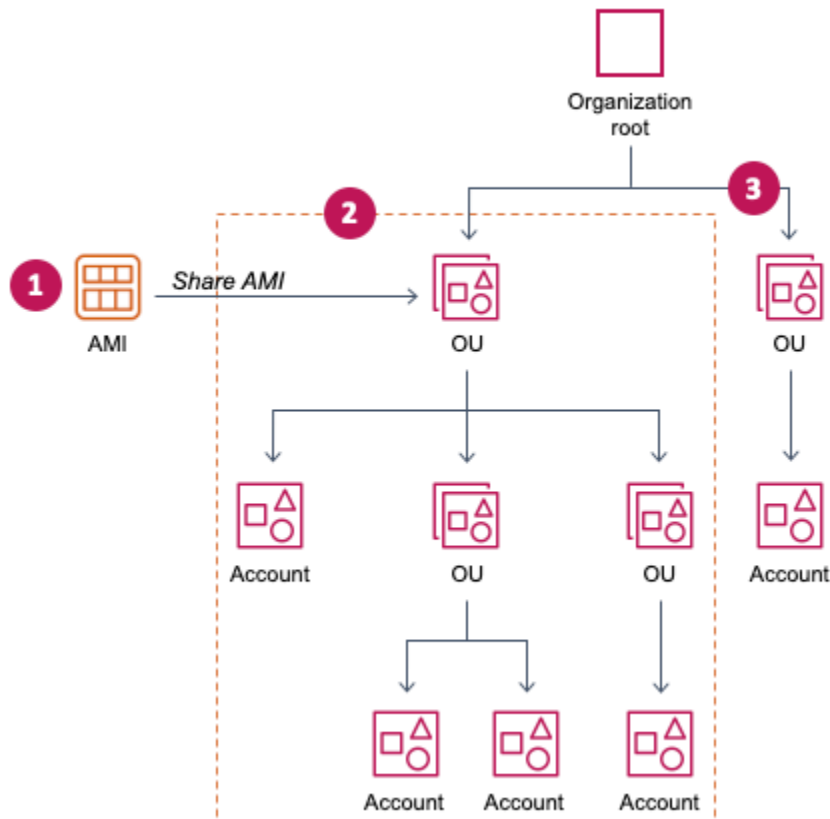
Share an AMI with specific organizations or organizational units

[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.



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::account-id:organization/organization-id`
- OU ARN: `arn:aws:organizations::account-id:ou/organization-id/ou-id`

Where:

- *account-id* is the 12-digit management account number, for example, 123456789012. 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](#).
- *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 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.

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 OUs to use the AWS KMS keys that were used to encrypt the snapshots.

Use the `aws:PrincipalOrgID` and `aws:PrincipalOrgPaths` keys to compare the AWS Organizations path for the principal who is making the request to the path in the policy. That principal can be a user, IAM role, federated user, or AWS account root user. In a policy, this condition key ensures that the requester is an account member within the specified organization root or OUs in AWS Organizations. For more example condition statements, see [aws:PrincipalOrgID](#) and [aws:PrincipalOrgPaths](#) in the *IAM User Guide*.

For information about editing a key policy, see [Allowing users in other accounts to use a KMS key](#) in the *AWS Key Management Service Developer Guide*.

To give an organization or OU 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*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalOrgID": "o-123example"
    }
  }
}
```

To share a KMS key with multiple OUs, 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*"
  ],
  "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/*"
      ]
    }
  }
}
```

Share an AMI

You can use the Amazon EC2 console or the AWS CLI to share an AMI with an organization or OU.


Share an AMI (console)

To share an AMI with an organization or an OU using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions, 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.

 **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 Amazon 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](#).

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](#).

Share an AMI (AWS CLI)

Use the [modify-image-attribute](#) command (AWS CLI) to share an AMI.

To share an AMI with an organization using the AWS CLI

The [modify-image-attribute](#) command grants launch permissions for the specified AMI to the specified organization. Note that you must specify the full ARN, not just the ID.

```
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 using the AWS CLI

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}]]"
```

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 Amazon 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](#).

Stop sharing an AMI

You can use the Amazon EC2 console or the AWS CLI to stop sharing an AMI with an organization or OU.

Stop sharing an AMI (console)

To stop sharing an AMI with an organization or OU using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions, 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**.

Stop sharing an AMI (AWS CLI)

Use the [modify-image-attribute](#) or [reset-image-attribute](#) commands (AWS CLI) to stop sharing an AMI.

To stop sharing an AMI with an organization or OU using the AWS CLI

The [modify-image-attribute](#) command removes launch permissions for the specified AMI from the specified organization. Note that you must specify the ARN.

```
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 using the AWS CLI

The [reset-image-attribute](#) command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

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

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.

View the organizations and OUs with which an AMI is shared

You can use the Amazon EC2 console or the AWS CLI to check with which organizations and OUs you've shared your AMI.

View the organizations and OUs with which an AMI is shared (console)

To check with which organizations and OUs you've shared your AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.

3. Select your AMI 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](#).

View the organizations and OUs with which an AMI is shared (AWS CLI)

You can check which organizations and OUs you've shared your AMI with by using the [describe-image-attribute](#) command (AWS CLI) and the `launchPermission` attribute.

To check with which organizations and OUs you've shared your AMI using the AWS CLI

The [describe-image-attribute](#) command describes the `launchPermission` attribute for the specified AMI, and returns the organizations and OUs with which you've shared the AMI.

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

Example response

```
{  
  "ImageId": "ami-0abcdef1234567890",  
  "LaunchPermissions": [  
    {  
      "OrganizationalUnitArn": "arn:aws:organizations::111122223333:ou/  
o-123example/ou-1234-5example"  
    }  
  ]  
}
```

Get the ARN

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 management account number.

Before you can get the ARNs, you must have the permission to describe organizations and organizational units. The following policy provides the necessary permission.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "organizations:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

To get the ARN of an organization

Use the [describe-organization](#) command and the `--query` parameter set to `'Organization.Arn'` to return only the organization ARN.

```
aws organizations describe-organization --query 'Organization.Arn'
```

Example response

```
"arn:aws:organizations::123456789012:organization/o-123example"
```

To get the ARN of an organizational unit

Use the [describe-organizational-unit](#) command, specify the OU ID, and set the `--query` parameter to `'OrganizationalUnit.Arn'` to return only the organizational unit ARN.

```
aws organizations describe-organizational-unit --organizational-unit-id ou-1234-5example --query 'OrganizationalUnit.Arn'
```

Example response

```
"arn:aws:organizations::123456789012:ou/o-123example/ou-1234-5example"
```

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 [Viewing 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.
- **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 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.

Share an AMI (console)

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions, 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.

Note

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

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](#).

Share an AMI (AWS CLI)

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

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account. In the following example, replace the example AMI ID with a valid AMI ID, and replace *account-id* with the 12-digit AWS account ID.

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

Note

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

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account. In the following example, replace the example AMI ID with a valid AMI ID, and replace *account-id* with the 12-digit AWS account ID.

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

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command. In the following example, replace the example AMI ID with a valid AMI ID.

```
aws ec2 reset-image-attribute \  
  --image-id 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.

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](#)
- [Cancel having an AMI shared with your account](#)
- [Find AMIs that are shared with your account](#)

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 AWS account

Use the [cancel-image-launch-permission](#) command and specify the AMI ID.

```
aws ec2 cancel-image-launch-permission \  
  --image-id ami-0123456789example \  
  --region us-east-1
```

Expected output

```
{  
  "Return": true  
}
```

PowerShell

To cancel having an AMI shared with your AWS account using the AWS Tools for PowerShell

Use the [Stop-EC2ImageLaunchPermission](#) command and specify the AMI ID.

```
Stop-EC2ImageLaunchPermission \  
  -ImageId ami-0123456789example \  
  -Region us-east-1
```

Expected output

```
True
```

Find AMIs that are shared with your account

To find the AMIs that are shared with your AWS account, see [Find shared AMIs](#).

Use bookmarks

If you have created a public AMI, or shared an AMI with another AWS account, you can create a *bookmark* that allows a user to access your AMI and launch an instance in their own account immediately. This is an easy way to share AMI references, so users don't have to spend time finding your AMI in order to use it.

Note that your AMI must be public, or you must have shared it with the user to whom you want to send the bookmark.

To create a bookmark for your AMI

1. Type a URL with the following information, where *region* is the Region in which your AMI resides:

```
https://console.aws.amazon.com/ec2/v2/home?  
region=region#LaunchInstanceWizard:ami=ami_id
```

For example, this URL launches an instance from the ami-0abcdef1234567890 AMI in the US East (N. Virginia) us-east-1 Region:

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

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

Guidelines for shared Linux AMIs

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

⚠ Important

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

Contents

- [Update the AMI tools before using them](#)
- [Disable password-based remote logins for the root user](#)
- [Disable local root access](#)
- [Remove SSH host key pairs](#)
- [Install public key credentials](#)
- [Disabling sshd DNS checks \(optional\)](#)
- [Protect yourself](#)

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](#)

Update the AMI tools before using them

For AMIs backed by instance store, we recommend that your AMIs download and upgrade the Amazon EC2 AMI creation tools before you use them. This ensures that new AMIs based on your shared AMIs have the latest AMI tools.

For [Amazon Linux 2](#), install the `aws-amitools-ec2` package and add the AMI tools to your PATH with the following command. For the [Amazon Linux AMI](#), `aws-amitools-ec2` package is already installed by default.

```
[ec2-user ~]$ sudo yum install -y aws-amitools-ec2 && export PATH=$PATH:/opt/aws/bin  
> /etc/profile.d/aws-amitools-ec2.sh && . /etc/profile.d/aws-amitools-ec2.sh
```

Upgrade the AMI tools with the following command:

```
[ec2-user ~]$ sudo yum upgrade -y aws-amitools-ec2
```

For other distributions, make sure you have the latest AMI tools.

Disable password-based remote logins for 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** may not remove all copies of a file from your storage media. Hidden copies of files may be created by journalling file systems (including Amazon Linux default ext4), snapshots, backups, RAID, and temporary caching. For more information see the **shred** [documentation](#).

⚠ Important

If you forget to remove the existing SSH host key pairs from your public AMI, our routine auditing process notifies you and all customers running instances of your AMI of the potential security risk. After a short grace period, we mark the AMI private.

Install public key credentials

After configuring the AMI to prevent logging in using a password, you must make sure users can log in using another mechanism.

Amazon EC2 allows users to specify a public-private key pair name when launching an instance. When a valid key pair name is provided to the `RunInstances` API call (or through the command line API tools), the public key (the portion of the key pair that Amazon EC2 retains on the server after a call to `CreateKeyPair` or `ImportKeyPair`) is made available to the instance through an HTTP query against the instance metadata.

To log in through SSH, your AMI must retrieve the key value at boot and append it to `/root/.ssh/authorized_keys` (or the equivalent for any other user account on the AMI). Users can launch instances of your AMI with a key pair and log in without requiring a root password.

Many distributions, including Amazon Linux and Ubuntu, use the `cloud-init` package to inject public key credentials for a configured user. If your distribution does not support `cloud-init`, you can add the following code to a system start-up script (such as `/etc/rc.local`) to pull in the public key you specified at launch for the root user.

ℹ 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.

Disabling sshd DNS checks (optional)

Disabling sshd DNS checks slightly weakens your sshd security. However, if DNS resolution fails, SSH logins still work. If you do not disable sshd checks, DNS resolution failures prevent all logins.

To disable sshd DNS checks

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

```
#UseDNS yes
```

2. Change the line to:

```
UseDNS no
```

Note

The location of this configuration file can differ for your distribution or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Protect yourself

We recommend against storing sensitive data or software on any AMI that you share. Users who launch a shared AMI might be able to rebundle it and register it as their own. Follow these guidelines to help you to avoid some easily overlooked security risks:

- We recommend using the `--exclude directory` option on `ec2-bundle-vol` to skip any directories and subdirectories that contain secret information that you would not like to include in your bundle. In particular, exclude all user-owned SSH public/private key pairs and SSH `authorized_keys` files when bundling the image. The Amazon public AMIs store these in `/root/.ssh` for the root 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).

Paid AMIs

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS, including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the [AWS Marketplace](#) website.

Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required. The instance is charged according to the rates set by the owner of the AMI, as well as the standard usage fees for the related web services, for example, the hourly rate for running an m1.small instance type in Amazon EC2. Additional taxes might also apply. The owner of the paid AMI can confirm whether a specific instance was launched using that paid AMI.

Important

Amazon DevPay is no longer accepting new sellers or products. AWS Marketplace is now the single, unified e-commerce platform for selling software and services through AWS. For information about how to deploy and sell software from AWS Marketplace, see [Selling in AWS Marketplace](#). AWS Marketplace supports AMIs backed by Amazon EBS.

Contents

- [Sell your AMI](#)
- [Find a paid AMI](#)

- [Purchase a paid AMI](#)
- [Get the product code for your instance](#)
- [Use paid support](#)
- [Bills for paid and supported AMIs](#)
- [Manage your AWS Marketplace subscriptions](#)

Sell your AMI

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

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

Find a paid AMI

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

Find a paid AMI using the console

To find a paid AMI using the console

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

Find a paid AMI using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open [AWS Marketplace](#).

2. Enter the name of the operating system in the search field, and then choose the search button (magnifying glass).
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either AMI or Software as a Service.

Find a paid AMI using the AWS CLI

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

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

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

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

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

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

Purchase a paid AMI

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

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

Purchase a paid AMI using the console

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

Subscribe to a product using AWS Marketplace

To use the AWS Marketplace, you must have an AWS account. To launch instances from AWS Marketplace products, you must be signed up to use the Amazon EC2 service, and you must be subscribed to the product from which to launch the instance. There are two ways to subscribe to products in the AWS Marketplace:

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

Get the product code for your instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. For more information about retrieving metadata, see [Instance metadata and user data](#).

To retrieve a product code, use the following command:

IMDSv2

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

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/product-codes
```

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

Use paid support

Amazon EC2 also enables developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. During sign-up for the support product, the developer gives you a product code, which you must then associate with your own AMI. This enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the terms for the product specified by the developer.

Important

You can't use a support product with Reserved Instances. You always pay the price that's specified by the seller of the support product.

To associate a product code with your AMI, use one of the following commands, where *ami_id* is the ID of the AMI and *product_code* is the product code:

- [modify-image-attribute](#) (AWS CLI)

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

- [Edit-EC2ImageAttribute](#) (AWS Tools for Windows PowerShell)

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami_id -ProductCode product_code
```

After you set the product code attribute, it cannot be changed or removed.

Bills for paid and supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using any paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill. For more information, see [Paying for products](#) in the *AWS Marketplace Buyer Guide*.

Manage your AWS Marketplace subscriptions

On the AWS Marketplace website, you can check your subscription details, view the vendor's usage instructions, manage your subscriptions, and more.

To check your subscription details

1. Log in to the [AWS Marketplace](#).
2. Choose **Your Marketplace Account**.
3. Choose **Manage your software subscriptions**.
4. All your current subscriptions are listed. Choose **Usage Instructions** to view specific instructions for using the product, for example, a user name for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Instances**.
 - c. Select the instance, and then choose **Instance state, Terminate instance**.
 - d. Choose **Terminate** when prompted for confirmation.
2. Log in to the [AWS Marketplace](#), and choose **Your Marketplace Account**, then **Manage your software subscriptions**.
3. Choose **Cancel subscription**. You are prompted to confirm your cancellation.

Note

After you've canceled your subscription, you are no longer able to launch any instances from that AMI. To use that AMI again, you need to resubscribe to it, either on the AWS Marketplace website, or through the launch wizard in the Amazon EC2 console.

AMI lifecycle

You can create your own AMIs, copy them, back them up, and maintain them until you are ready to deprecate or deregister them.

Contents

- [Create an AMI](#)
- [Modify an AMI](#)

- [Copy an AMI](#)
- [Store and restore an AMI using S3](#)
- [Deprecate an AMI](#)
- [Disable an AMI](#)
- [Archive AMI snapshots](#)
- [Deregister \(delete\) an AMI](#)
- [Automate the EBS-backed AMI lifecycle](#)

Create an AMI

You can create Amazon EBS-backed Linux AMIs and instance store-backed AMIs.

Topics

- [Create an Amazon EBS-backed Linux AMI](#)
- [Create an instance store-backed Linux AMI](#)

For information about how to create a Windows AMI, see [Create a custom Windows AMI](#).

Create an Amazon EBS-backed Linux AMI

To create an Amazon EBS-backed Linux AMI, start from an instance that you've launched from an existing Amazon EBS-backed Linux AMI. This can be an AMI you have obtained from the AWS Marketplace, an AMI you have created using the [AWS Server Migration Service](#) or [VM Import/Export](#), or any other AMI you can access. After you customize the instance to suit your needs, create and register a new AMI, which you can use to launch new instances with these customizations.

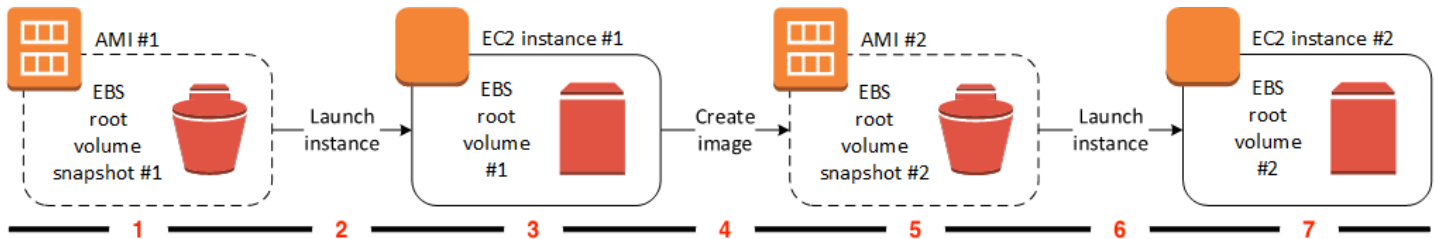
The procedures described below work for Amazon EC2 instances backed by encrypted Amazon Elastic Block Store (Amazon EBS) volumes (including the root volume) as well as for unencrypted volumes.

The AMI creation process is different for instance store-backed AMIs. For information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Storage for the root device](#). For information about creating an instance store-backed Linux AMI, see [Create an instance store-backed Linux AMI](#).

For information about creating an Amazon EBS-backed Windows AMI, see [Create an Amazon EBS-backed Windows AMI](#) in the *Amazon EC2 User Guide for Windows Instances*.

Overview of creating Amazon EBS-backed AMIs

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 you have created using the [AWS Server Migration Service](#) or [VM Import/Export](#), or any other AMI 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](#).

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 a Linux AMI from an instance

You can create an AMI using the AWS Management Console or the command line.

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 **No reboot**, either keep the **Enable** check box cleared (the default), or select it.
 - If the **Enable** check box for **No reboot** is cleared, 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 the **Enable** check box for **No reboot** is selected, when Amazon EC2 creates the new AMI, it does not shut down and reboot the instance.

⚠ Warning

If you choose to enable **No reboot**, 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 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. **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.

- f. 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

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

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

Create a Linux AMI from a snapshot

If you have a snapshot of the root device volume of an instance, you can create an AMI from this snapshot using the AWS Management Console or the command line.

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 on Linux instances](#).
 - e. For **Virtualization type**, choose the virtualization type to be used by instances launched from this AMI. For more information, see [Linux AMI 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 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**.

AWS CLI

To create an AMI from a snapshot using the command line

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

- [register-image](#) (AWS CLI)
- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

Launch an instance from an AMI you created

You can launch an instance from an AMI that you created from an instance or snapshot.

To launch an instance from your AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Set the filter to **Owned by me** and select your AMI.
4. Choose **Launch instance from AMI**.
5. Accept the default values or specify custom values in the launch instance wizard. For more information, see [Launch an instance using the new launch instance wizard](#).

Create an instance store-backed Linux AMI

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

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

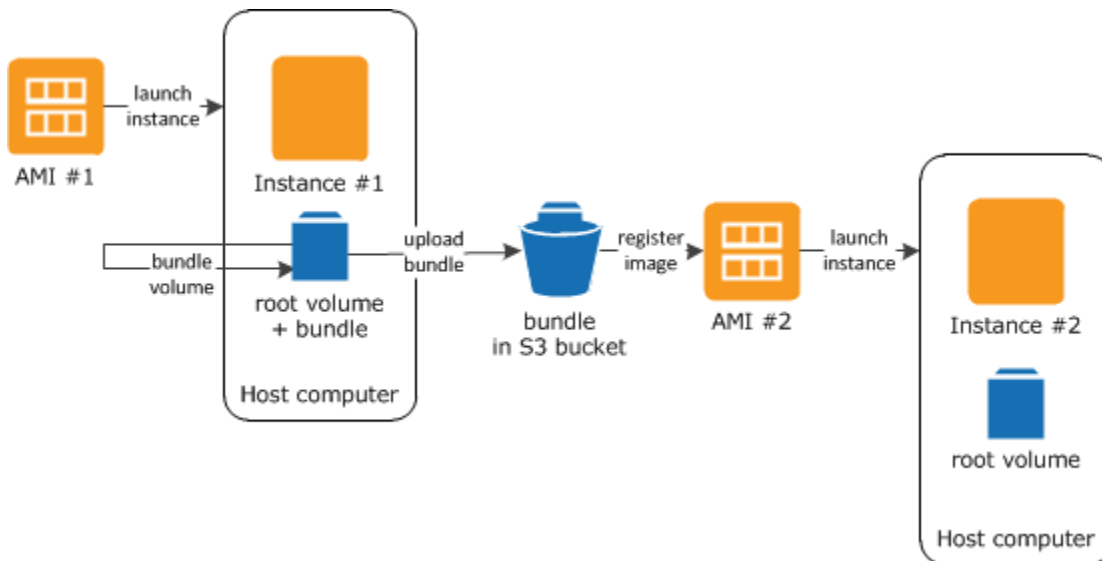
Important

Only the following instance types support an instance store volume as the root device: C1, C3, D2, I2, M1, M2, M3, R3, and X1.

The AMI creation process is different for Amazon EBS-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Storage for the root device](#). If you need to create an Amazon EBS-backed Linux AMI, see [Create an Amazon EBS-backed Linux AMI](#).

Overview of the creation process for instance store-backed AMIs

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



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

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 uploaded objects using S3 Object Ownership](#).

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 \(delete\) an 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](#).

Prerequisites

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

- Install the AMI tools. For more information, see [Set up the AMI tools](#).
- Install the AWS CLI. For more information, see [Getting Set Up with the AWS Command Line Interface](#).
- 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 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 [Best Practices for AWS accounts](#) in the *AWS Account Management Reference 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.

Tasks

- [Set up the AMI tools](#)
- [Create an AMI from an instance store-backed Amazon Linux instance](#)
- [Create an AMI from an instance store-backed Ubuntu instance](#)
- [Convert your instance store-backed AMI to an Amazon EBS-backed AMI](#)

Set up the AMI tools

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

To set up the AMI tools using the RPM

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

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

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

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

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

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

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

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

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

- Header SHA1: a1f662d6f25f69871104e6a62187fa4df508f880
- MD5: 9faff05258064e2f7909b66142de6782

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

4. Install the RPM using the following command:

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

5. Verify your AMI tools installation using the [ec2-ami-tools-version](#) 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 OFHPLP4ZULTHYPMSYEX704BEXAMPLE --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 OFHPLP4ZULTHYPMSYEX704BEXAMPLE
```

Create an AMI from an instance store-backed instance

The following procedures are for creating an instance store-backed AMI from an instance store-backed instance. Before you begin, ensure that you've read the [Prerequisites](#).

Topics

- [Create an AMI from an instance store-backed Amazon Linux instance](#)
- [Create an AMI from an instance store-backed Ubuntu instance](#)

Create an AMI from an instance store-backed Amazon Linux instance

This section describes the creation of an AMI from an Amazon Linux instance. The following procedures may not work for instances running other Linux distributions. For Ubuntu-specific procedures, see [Create an AMI from an instance store-backed Ubuntu instance](#).

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-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem /
path/to/cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem ec2-
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.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-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.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](#).

- 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 my-s3-bucket/bundle_folder/bundle_name -m /tmp/  
image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

⚠ Important

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

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

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

⚠ Important

If you specified a path with the `-d /path/to/bundle/storage` option in [Step 2](#), use that path instead of `/tmp`.

6. To register your AMI, run the [register-image](#) command as follows.

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

⚠ Important

If you previously specified a Region for the [ec2-upload-bundle](#) command, specify that Region again for this command.

Create an AMI from an instance store-backed Ubuntu instance

This section describes the creation of an AMI from an Ubuntu Linux instance with an instance store volume as the root volume. The following procedures may not work for instances running other Linux distributions. For procedures specific to Amazon Linux, see [Create an AMI from an instance store-backed Amazon Linux instance](#).

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

The AMI tools require GRUB Legacy to boot properly. However, Ubuntu is configured to use GRUB 2. You must check to see that your instance uses GRUB Legacy, and if not, you need to install and configure it.

HVM instances also require partitioning tools to be installed for the AMI tools to work properly.

1. GRUB Legacy (version 0.9x or less) must be installed on your instance. Check to see if GRUB Legacy is present and install it if necessary.

- a. Check the version of your GRUB installation.

```
ubuntu:~$ grub-install --version
grub-install (GRUB) 1.99-21ubuntu3.10
```

In this example, the GRUB version is greater than 0.9x, so you must install GRUB Legacy. Proceed to [Step 1.b](#). If GRUB Legacy is already present, you can skip to [Step 2](#).

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

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

2. Install the following partition management packages using the package manager for your distribution.

- gdisk (some distributions may call this package gptfdisk instead)
- kpartx
- parted

Use the following command.

```
ubuntu:~$ sudo apt-get install -y gdisk kpartx parted
```

3. Check the kernel parameters for your instance.

```
ubuntu:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-
aee7-72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```


Note the options following the kernel and root device parameters: `ro`, `console=ttyS0`, and `xen_emul_unplug=unnecessary`. Your options may differ.

4. Check the kernel entries in `/boot/grub/menu.lst`.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=hvc0
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
kernel /boot/memtest86+.bin
```

Note that the `console` parameter is pointing to `hvc0` instead of `ttyS0` and that the `xen_emul_unplug=unnecessary` parameter is missing. Again, your options may differ.

5. Edit the `/boot/grub/menu.lst` file with your favorite text editor (such as **vim** or **nano**) to change the console and add the parameters you identified earlier to the boot entries.

```
title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual
root           (hd0)
kernel        /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
ro console=ttyS0 xen_emul_unplug=unnecessary
initrd        /boot/initrd.img-3.2.0-54-virtual

title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery mode)
root           (hd0)
kernel        /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
single console=ttyS0 xen_emul_unplug=unnecessary
initrd        /boot/initrd.img-3.2.0-54-virtual

title          Ubuntu 12.04.3 LTS, memtest86+
root           (hd0)
kernel        /boot/memtest86+.bin
```

6. Verify that your kernel entries now contain the correct parameters.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=ttyS0
xen_emul_unplug=unnecessary
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
console=ttyS0 xen_emul_unplug=unnecessary
kernel /boot/memtest86+.bin
```

7. [For Ubuntu 14.04 and later only] Starting with Ubuntu 14.04, instance store backed Ubuntu AMIs use a GPT partition table and a separate EFI partition mounted at `/boot/efi`. The `ec2-bundle-vol` command will not bundle this boot partition, so you need to comment out the `/etc/fstab` entry for the EFI partition as shown in the following example.

```

LABEL=cloudimg-rootfs /          ext4 defaults          0 0
#LABEL=UEFI           /boot/efi      vfat  defaults          0 0
/dev/xvdb              /mnt          auto  defaults,nobootwait,comment=cloudconfig 0 2

```

To create an AMI from an instance store-backed Ubuntu instance

This procedure assumes that you have satisfied the prerequisites in [Prerequisites](#).

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:

```
ubuntu:~$ mkdir /tmp/cert
```

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

- b. Copy your X.509 certificate and private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as [scp](#). 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-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem /
path/to/cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem ec2-
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem 100% 717    0.7KB/s   00:00
cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.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 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 needs as root. For most commands, you can use `sudo` to gain elevated permissions, but in this case, you should run `sudo -E su` to keep your environment variables.

```
ubuntu:~$ sudo -E su
```

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

```
root@ubuntu:~#
```

- b. To create the AMI bundle, run the [ec2-bundle-vol](#) command as follows.

```
root@ubuntu:~# ec2-bundle-vol -k /tmp/cert/pk-  
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-  
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64 -e /tmp/  
cert --partition gpt
```

⚠ Important

For Ubuntu 14.04 and later HVM instances, add the `--partition mbr` flag to bundle the boot instructions properly; otherwise, your newly-created AMI will not boot.

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

- c. Exit from the root shell.

```
root@ubuntu:~# exit
```

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

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

```
ubuntu:~$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

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

```
ubuntu:~$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/  
image.manifest.xml
```

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

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

```

    <virtual>ephemeral1</virtual>
    <device>sdc</device>
  </mapping>
  <mapping>
    <virtual>root</virtual>
    <device>/dev/sda1</device>
  </mapping>
</block_device_mapping>

```

- d. Save the `image.manifest.xml` file and exit your text editor.
4. To upload your bundle to Amazon S3, run the [ec2-upload-bundle](#) command as follows.

```

ubuntu:~$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/
image.manifest.xml -a your_access_key_id -s your_secret_access_key

```

Important

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

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

```

ubuntu:~$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image

```

Important

If you specified a path with the `-d /path/to/bundle/storage` option in [Step 2](#), use that same path below, instead of `/tmp`.

6. To register your AMI, run the [register-image](#) AWS CLI command as follows.

```

ubuntu:~$ aws ec2 register-image --image-location my-s3-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --
virtualization-type hvm

```

⚠ Important

If you previously specified a Region for the [ec2-upload-bundle](#) command, specify that Region again for this command.

7. [Ubuntu 14.04 and later] Uncomment the EFI entry in `/etc/fstab`; otherwise, your running instance will not be able to reboot.

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

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

⚠ Important

You can't convert an instance store-backed Windows AMI to an Amazon EBS-backed Windows AMI and you cannot convert an AMI that you do not own.

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

1. Launch an Amazon Linux instance from an Amazon EBS-backed AMI. For more information, see [Launch an instance using the new launch instance wizard](#). 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-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem ec2-
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
```

3. Configure your environment variables to use the AWS CLI. For more information, see [Creating a key pair](#).

- 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.

```
[ec2-user ~]$ 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.

```
[ec2-user ~]$ aws ec2 attach-volume --volume-id volume_id --instance-
id instance_id --device /dev/sdb --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 my-s3-bucket/bundle_folder/bundle_name -m image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

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 volume_id --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 volume_id
```

- b. Check to see that your snapshot is complete.

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

- 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-id
--output text
IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon
available public machine aki-fc8f11cc instance-store paravirtual xen
```

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

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

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --
name your_new_ami_name --block-device-mappings DeviceName=device-
name,Ebs={SnapshotId=snapshot_id} --virtualization-type paravirtual --
architecture x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
```

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

```
aws ec2 delete-volume --volume-id volume_id
```

AMI tools reference

You can use the AMI tools commands to create and manage instance store-backed Linux AMIs. To set up the tools, see [Set up the AMI tools](#).

For information about your access keys, see [Best Practices for AWS accounts](#) in the *AWS Account Management Reference 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 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-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem -c cert-
HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.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 Linux 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 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

`--ramdiskramdisk_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-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem -c
cert-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
Copying / into the image file /mnt/image...
Excluding:
  sys
  dev/shm
  proc
  dev/pts
  proc/sys/fs/binfmt_misc
  dev
  media
  mnt
```

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

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

ec2-delete-bundle

Description

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

Syntax

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

Options

-b, --bucket *bucket*

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

Required: Yes

`-a, --access-key access_key_id`

The AWS access key ID.

Required: Yes

`-s, --secret-key secret_access_key`

The AWS secret access key.

Required: Yes

`-t, --delegation-token token`

The delegation token to pass along to the AWS request. For more information, see the [Using Temporary Security Credentials](#).

Required: Only when you are using temporary security credentials.

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

`--region region`

The Region to use in the request signature.

Default: `us-east-1`

Required: Required if using signature version 4

`--sigv version`

The signature version to use when signing the request.

Valid values: `2 | 4`

Default: `4`

Required: No

`-m, --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 DOC-EXAMPLE-BUCKET1 -a your_access_key_id -  
s your_secret_access_key  
Deleting files:  
DOC-EXAMPLE-BUCKET1/  
image.manifest.xml  
DOC-EXAMPLE-BUCKET1/  
image.part.00  
DOC-EXAMPLE-BUCKET1/  
image.part.01  
DOC-EXAMPLE-BUCKET1/  
image.part.02  
DOC-EXAMPLE-BUCKET1/
```

```
image.part.03
DOC-EXAMPLE-BUCKET1/
image.part.04
DOC-EXAMPLE-BUCKET1/
image.part.05
DOC-EXAMPLE-BUCKET1/image.part.06
Continue? [y/n]
y
Deleted DOC-EXAMPLE-BUCKET1/image.manifest.xml
Deleted DOC-EXAMPLE-BUCKET1/image.part.00
Deleted DOC-EXAMPLE-BUCKET1/image.part.01
Deleted DOC-EXAMPLE-BUCKET1/image.part.02
Deleted DOC-EXAMPLE-BUCKET1/image.part.03
Deleted DOC-EXAMPLE-BUCKET1/image.part.04
Deleted DOC-EXAMPLE-BUCKET1/image.part.05
Deleted DOC-EXAMPLE-BUCKET1/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

Description

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

Syntax

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

Options

-b, --bucket *bucket*

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

Required: Yes

-a, --access-key *access_key_id*

The AWS access key ID.

Required: Yes

`-s, --secret-key secret_access_key`

The AWS secret access key.

Required: Yes

`-k, --privatekey path`

The private key used to decrypt the manifest.

Required: Yes

`--url url`

The Amazon S3 service URL.

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

Required: No

`--region region`

The Region to use in the request signature.

Default: `us-east-1`

Required: Required if using signature version 4

`--sigv version`

The signature version to use when signing the request.

Valid values: 2 | 4

Default: 4

Required: No

`-m, --manifest file`

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

Required: No

`-p, --prefix prefix`

The filename prefix for the bundled AMI files.

Default: image

Required: No

-d, --directory *directory*

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

Default: The current working directory.

Required: No

--retry

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

Required: No

Output

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

Example

This example creates the bundled directory (using the Linux **mkdir** command) and downloads the bundle from the *DOC-EXAMPLE-BUCKET1* Amazon S3 bucket.

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b DOC-EXAMPLE-BUCKET1/bundles/bundle_name
-m image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from DOC-EXAMPLE-BUCKET1 to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.00 ...
Downloaded image.part.00 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.01 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.01 ...
Downloaded image.part.01 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.02 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.02 ...
Downloaded image.part.02 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.03 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.03 ...
```

```

Downloaded image.part.03 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.04 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.04 ...
Downloaded image.part.04 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.05 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.05 ...
Downloaded image.part.05 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.06 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to
mybundle/image.part.06 ...
Downloaded image.part.06 from DOC-EXAMPLE-BUCKET1

```

ec2-migrate-manifest

Description

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

Syntax

```

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

```

Options

-c, --cert *path*

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

Required: Yes

-k, --privatekey *path*

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

Required: Yes

--manifest *path*

The path to the manifest file.

Required: Yes

`-a, --access-key access_key_id`

The AWS access key ID.

Required: Required if using automatic mapping.

`-s, --secret-key secret_access_key`

The AWS secret access key.

Required: Required if using automatic mapping.

`--region region`

The Region to look up in the mapping file.

Required: Required if using automatic mapping.

`--no-mapping`

Disables automatic mapping of kernels and RAM disks.

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

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

`--ec2cert path`

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

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

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](#).

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-HKZYKTAIG2ECMXIYIBH3HXV4ZBZQ55CL0.pem --privatekey pk-  
HKZYKTAIG2ECMXIYIBH3HXV4ZBZQ55CL0.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 Linux 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 uploaded objects using S3 Object Ownership](#).

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 the [Using Temporary Security Credentials](#).

Required: Only when you are using temporary security credentials.

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

`-m, --manifest path`

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

Required: Yes

`--url url`

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

The Amazon S3 endpoint service URL.

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

Required: No

`--region region`

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

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

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

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

Default: `us-east-1`

Required: Required if using signature version 4

`--sigv version`

The signature version to use when signing the request.

Valid values: `2` | `4`

Default: `4`

Required: No

`--acl acl`

The access control list policy of the bundled image.

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

Default: `aws-exec-read`

Required: No

`-d, --directory directory`

The directory containing the bundled AMI parts.

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

Required: No

`--part part`

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

Required: No

`--retry`

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

Required: No

`--skipmanifest`

Does not upload the manifest.

Required: No

`--location location`

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

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

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

Required: No

Output

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

Example

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

```
[ec2-user ~]$ ec2-upload-bundle -b DOC-EXAMPLE-BUCKET1/bundles/bundle_name -m
  image.manifest.xml -a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket DOC-EXAMPLE-BUCKET1 ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
```

```
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

Common options for AMI tools

Most of the AMI tools accept the following optional parameters.

`--help, -h`

Displays the help message.

`--version`

Displays the version and copyright notice.

`--manual`

Displays the manual entry.

`--batch`

Runs in batch mode, suppressing interactive prompts.

`--debug`

Displays information that can be useful when troubleshooting problems.

Modify an AMI

You can modify a limited set of Amazon Machine Image (AMI) attributes, such as the AMI's description and sharing properties. However, AMI content (volume binary data) can't be modified. To modify the AMI content, you must [create a new AMI](#).

Important

You can't modify the content (volume binary data) of an EBS-backed AMI because the snapshots that back them are immutable. You also can't modify the content (volume binary data) of an instance store-backed (S3-backed) AMI because the content is signed, and instance launches will fail if the signatures do not match.

For the AMI attributes that can be modified, see [ModifyImageAttribute](#) in the *Amazon EC2 API Reference*.

The following topics provide instructions for using the Amazon EC2 console and AWS CLI to modify the attributes of an AMI:

- [Make an AMI public](#)
- [Share an AMI with specific organizations or organizational units](#)
- [Share an AMI with specific AWS accounts](#)
- [Use paid support](#)
- [Configure the AMI](#)

Copy an AMI

You can copy an Amazon Machine Image (AMI) within or across AWS Regions. You can copy both Amazon EBS-backed AMIs and instance store-backed AMIs. You can copy EBS-backed AMIs with encrypted snapshots, and also change encryption status during the copy process. You can copy AMIs that are shared with you.

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

- [Considerations](#)
- [Costs](#)
- [IAM permissions](#)
- [Copy an AMI](#)
- [Stop a pending AMI copy operation](#)
- [Cross-Region copying](#)
- [Cross-account copying](#)
- [Encryption and copying](#)

Considerations

- **Permission to copy AMIs** – You can use IAM policies to grant or deny users permission to copy AMIs. Resource-level permissions specified for the CopyImage action apply only to the new AMI. You cannot specify resource-level permissions for the source AMI.
- **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 target AMI and its backing snapshots.

Costs

There are no charges for copying an AMI. However, standard storage and data transfer rates apply. If you copy an EBS-backed AMI, you will incur charges for the storage of any additional EBS snapshots.

IAM permissions

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

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.

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

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. This includes the permission to tag the new backing snapshots of the target AMI. Permission to tag the new backing snapshots can be explicitly denied.

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.

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

```
{
  "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:::ami-source-bucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:GetBucketAcl",
      "s3:PutObjectAcl",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::amis-for-account-in-region-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 using the AWS Management Console, the AWS Command Line Interface or SDKs, or the Amazon EC2 API, all of which support the `CopyImage` action.

Prerequisite

Create or obtain an AMI to copy. Note that you can use the Amazon EC2 console to search a wide variety of AMIs provided by AWS. For more information, see [Create an Amazon EBS-backed Linux AMI](#) and [Finding an AMI](#).

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 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](#).
 - d. **Copy tags:** Select this check box 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. (EBS-backed AMIs only) **Encrypt EBS snapshots of AMI copy:** Select this check box 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** check box is selected and cannot be cleared. For more information, see [Encryption and copying](#).
 - f. (EBS-backed AMIs only) **KMS key:** The KMS key to used to encrypt the target snapshots.
 - g. **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.

- h. 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 using the AWS CLI

You can copy an AMI using the [copy-image](#) command. You must specify both the source and destination Regions. You specify the source Region using the `--source-region` parameter. You can specify the destination Region using either the `--region` parameter or an environment variable. For more information, see [Configuring the AWS Command Line Interface](#).

(EBS-backed AMIs only) When you encrypt a target snapshot during copying, you must specify these additional parameters: `--encrypted` and `--kms-key-id`.

For example commands, see the [Examples](#) under [copy-image](#) in the *AWS CLI Command Reference*.

PowerShell

To copy an AMI using the Tools for Windows PowerShell

You can copy an AMI using the [Copy-EC2Image](#) command. You must specify both the source and destination Regions. You specify the source Region using the `-SourceRegion` parameter. You can specify the destination Region using either the `-Region` parameter or the `Set-AWSDefaultRegion` command. For more information, see [Specifying AWS Regions](#).

(EBS-backed AMIs only) When you encrypt a target snapshot during copying, you must specify these additional parameters: `-Encrypted` and `-KmsKeyId`.

Stop a pending AMI copy operation

You can stop a pending AMI copy using the AWS Management Console or the command line.

Console

To stop an AMI copy operation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the destination Region from the Region selector.
3. In the navigation pane, choose **AMIs**.
4. Select the AMI to stop copying, and then choose **Actions, Deregister AMI**.
5. When asked for confirmation, choose **Deregister AMI**.

Command line

To stop an AMI copy operation using the command line

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

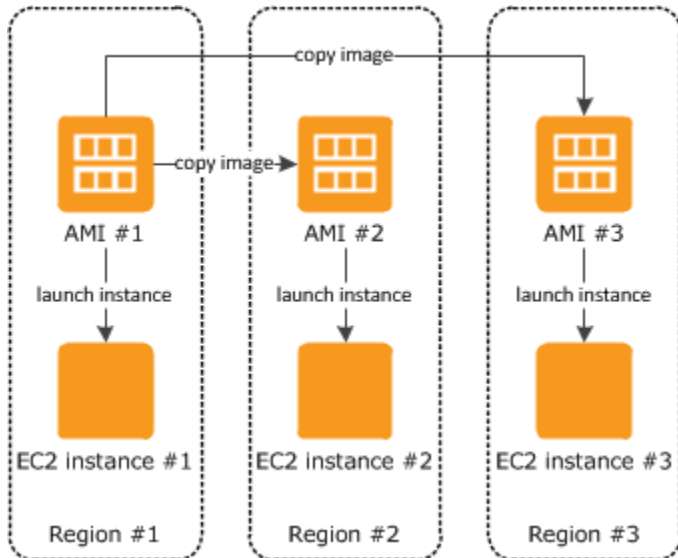
- [deregister-image](#) (AWS CLI)
- [Unregister-EC2Image](#) (AWS Tools for Windows PowerShell)

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 100 concurrent AMI copies.
- You can't copy a paravirtual (PV) AMI to a Region that does not support PV AMIs. For more information, see [Linux AMI virtualization types](#).

Cross-account copying

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

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

Resource permissions

To copy an AMI that was shared with you from another account, the owner of the source AMI must grant you read permissions for the storage that backs the AMI. 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 as well. For more information about granting resource permissions, for EBS snapshots, see [Share an Amazon EBS snapshot](#) in the *Amazon EBS User Guide*, and for S3 buckets, see [Identity and access management in Amazon S3](#) in the *Amazon Simple Storage Service User Guide*.

Note

To copy an AMI with its tags, you must have launch permissions for the source AMI.

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

Scenario	Description	Supported
4	Encrypted-to-unencrypted	No

Note

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

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

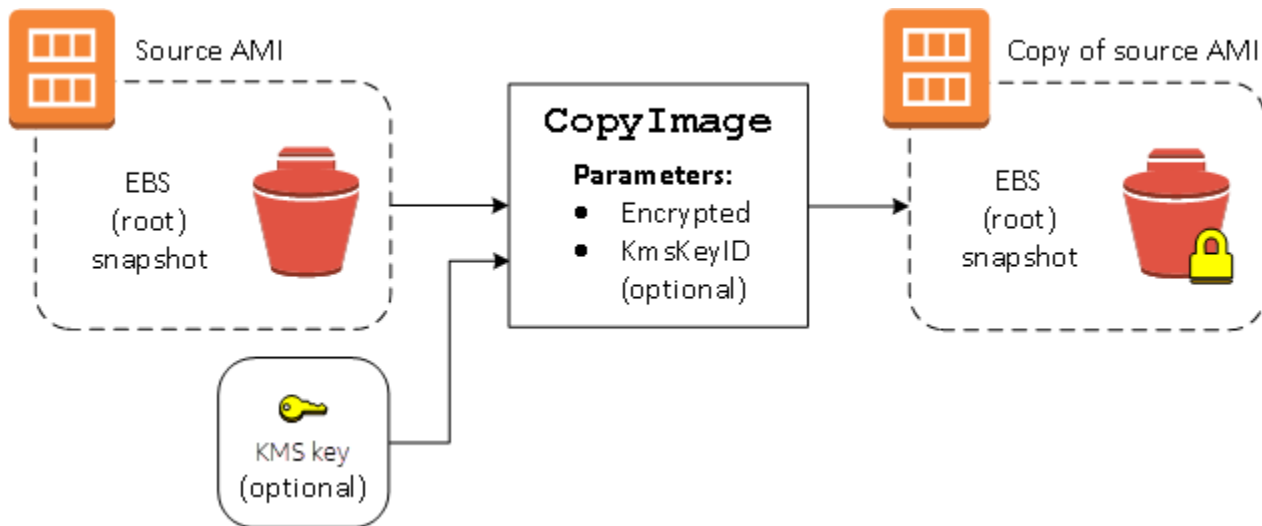
If you specify encryption parameters while copying an AMI, you can encrypt or re-encrypt its backing snapshots. The following example shows a non-default case that supplies encryption parameters to the CopyImage action in order to change the target AMI's encryption state.

Copy an unencrypted source AMI to an encrypted target AMI

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

Note

Enabling encryption by default 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.

Topics

- [Use cases](#)
- [How the AMI store and restore APIs work](#)
- [Limitations](#)
- [Costs](#)
- [Securing your AMIs](#)
- [Permissions for storing and restoring AMIs using S3](#)
- [Work with the AMI store and restore APIs](#)
- [Use file paths in S3](#)

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

- [Copy an AMI from one AWS partition to another AWS partition](#)
- [Make archival copies of AMIs](#)

Copy an AMI from one AWS partition to another AWS partition

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

To copy an AMI from one partition to another, follow these steps:

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

- Monitor the progress of the store task by using `DescribeStoreImageTasks`. The object becomes visible in the S3 bucket when the task is completed. For an example command, see [Describe the progress of an AMI store task](#).
- 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`. For an example command, see [Restore an AMI from an S3 bucket](#).
 - Monitor the progress of the restore task by describing the AMI to check when its state becomes available. You can also monitor the progress percentages of the snapshots that make up the restored AMI by describing the snapshots.

Make archival copies of AMIs

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

The AMI is packed into a single object in S3, and all of the AMI metadata (excluding sharing information) is preserved as part of the stored AMI. The AMI data is compressed as part of the storage process. AMIs that contain data that can easily be compressed will result in smaller objects

in S3. To reduce costs, you can use less expensive S3 storage tiers. For more information, see [Amazon S3 Storage Classes](#) and [Amazon S3 pricing](#)

How the AMI store and restore APIs work

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

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

How the APIs work

- [CreateStoreImageTask](#)
- [DescribeStoreImageTasks](#)
- [CreateRestoreImageTask](#)

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 600 GB of EBS snapshot data per account. Further task creation will be rejected until the tasks in progress are less than the limit. For

example, if an AMI with 100 GB of snapshot data and another AMI with 200 GB of snapshot data are currently being stored, another request will be accepted, because the total in progress is 300 GB, which is less than the limit. But if a single AMI with 800 GB of snapshot data is currently being stored, further tasks are rejected until the task is completed.

DescribeStoreImageTasks

The [DescribeStoreImageTasks](#) 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.

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: 600 GB of storage work (snapshot data) in progress.
- Quota on [restore image](#) requests: 300 GB of restore work (snapshot data) in progress.
- For the duration of the store task, the snapshots must not be deleted and the IAM principal doing the store must have access to the snapshots, otherwise the store process will fail.
- You can't create multiple copies of an AMI in the same S3 bucket.
- An AMI that is stored in an S3 bucket can't be restored with its original AMI ID. You can mitigate this by using [AMI aliasing](#).
- Currently the store and restore APIs are only supported by using the AWS Command Line Interface, AWS SDKs, and Amazon EC2 API. You can't store and restore an AMI using the Amazon EC2 console.

Costs

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

Securing your AMIs

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

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

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

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

Permissions for storing and restoring AMIs using S3

If your IAM principals will store or restore AMIs using 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.

```
{
  "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": "*"
}
]
}

```

Work with the AMI store and restore APIs

Topics

- [Store an AMI in an S3 bucket](#)
- [Describe the progress of an AMI store task](#)
- [Restore an AMI from an S3 bucket](#)

Store an AMI in an S3 bucket

To store an AMI (AWS CLI)

Use the [create-store-image-task](#) command. Specify the ID of the AMI and the name of the S3 bucket in which to store the AMI.

```

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

```

Expected output

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

Describe the progress of an AMI store task

To describe the progress of an AMI store task (AWS CLI)

Use the [describe-store-image-tasks](#) command.

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

Expected output

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

Restore an AMI from an S3 bucket

To restore an AMI (AWS CLI)

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

Note

The restored AMI gets a new AMI ID.

```
aws ec2 create-restore-image-task \
  --object-key ami-1234567890abcdef0.bin \
  --bucket myamibucket \
```

```
--name "New AMI Name"
```

Expected output

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

Use file paths in S3

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.

You can use file paths when using the AWS CLI and SDKs.

Example: Use a file path when storing and restoring an AMI (AWS CLI)

The following example first stores an AMI in S3, with the file path appended to the bucket name. The example then restores the AMI from S3, with the file path prepended to the object key parameter.

1. Store the AMI. For `--bucket`, specify the file path after the bucket name, as follows:

```
aws ec2 create-store-image-task \  
  --image-id ami-1234567890abcdef0 \  
  --bucket myamibucket/path1/path2
```

Expected output

```
{  
  "ObjectKey": "path1/path2/ami-1234567890abcdef0.bin"  
}
```

2. Restore the AMI. For `--object-key`, specify the value from the output in the previous step, which includes the file path.

```
aws ec2 create-restore-image-task \  
  --object-key path1/path2/ami-1234567890abcdef0.bin \  
  --bucket myamibucket \  
  --name "New AMI Name"
```

Deprecate an AMI

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

You can also create Amazon Data Lifecycle Manager EBS-backed AMI policies to automate the deprecation of EBS-backed AMIs. For more information, see [Automate AMI lifecycles](#).

Note

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](#).

Topics

- [Costs](#)
- [Limitations](#)
- [Deprecate an AMI](#)
- [Describe deprecated AMIs](#)
- [Cancel the deprecation of an AMI](#)

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.

Limitations

- To deprecate an AMI, you must be the owner of the AMI.

Deprecate an AMI

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

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** check box, and then enter the deprecation date and time.

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.

6. Choose **Save**.

AWS CLI

To deprecate an AMI on a specific date

Use the [enable-image-deprecation](#) command. Specify the ID of the AMI and the date and time on which to deprecate the AMI. If you specify a value for seconds, Amazon EC2 rounds the seconds to the nearest minute.

The upper limit for `deprecate-at` 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.

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

Expected output

```
{  
  "Return": "true"  
}
```

Check when an AMI was last used

`LastLaunchedTime` is a timestamp that indicates when your AMI was last used to launch an instance. AMIs that have not been used recently to launch an instance might be good candidates for deprecation or [deregistering](#).

Note

- When an AMI is used to launch an instance, there is a 24-hour delay before that usage is reported.
- `lastLaunchedTime` 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 navigator, choose **AMIs**.
3. From the filter bar, choose **Owned by me**.
4. Select the AMI, and then check the **Last launched time** field (if you selected the check box next to the AMI, it's located on the **Details** tab). The field shows the date and time when the AMI was last used to launch an instance.

AWS CLI

To view the last launched time of an AMI

Run the [describe-image-attribute](#) command and specify `--attribute lastLaunchedTime`. You must be the AMI owner to run this command.

```
aws ec2 describe-image-attribute \  
  --image-id ami-1234567890example \  
  --attribute lastLaunchedTime
```

Example output

```
{  
  "LastLaunchedTime": {  
    "Value": "2022-02-10T02:03:18Z"  
  },  
  "ImageId": "ami-1234567890example",  
}
```

Describe deprecated AMIs

You can view the deprecation date and time of an AMI, and filter all the AMIs by deprecation date. You can also use the AWS CLI to describe all the AMIs that have been deprecated, where the deprecation date is in the past.

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 check box 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 using the [describe-images](#) command, the results are different depending on whether you are an AMI user or the AMI owner.

- If you are an AMI user:

By default, when you describe all AMIs using the [describe-images](#) command, deprecated AMIs that are not owned by you, but which are shared with you, do not appear in the results. This is because the default is `--no-include-deprecated`. To include deprecated AMIs in the results, you must specify the `--include-deprecated` parameter.

- If you are the AMI owner:

When you describe all AMIs using the [describe-images](#) command, all the AMIs that you own, including deprecated AMIs, appear in the results. You do not need to specify the `--include-deprecated` parameter. Furthermore, you cannot exclude deprecated AMIs that you own from the results by using `--no-include-deprecated`.

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

Note

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

To include all deprecated AMIs when describing all AMIs

Use the [describe-images](#) command and specify the `--include-deprecated` parameter to include all deprecated AMIs that are not owned by you in the results.

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

To describe the deprecation date of an AMI

Use the [describe-images](#) command and specify the ID of the AMI.

Note that if you specify `--no-include-deprecated` together with the AMI ID, the deprecated AMI will be returned in the results.

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

Expected output

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

```

{
  "Images": [
    {
      "VirtualizationType": "hvm",
      "Description": "Provided by Red Hat, Inc.",
      "PlatformDetails": "Red Hat Enterprise Linux",
      "EnaSupport": true,
      "Hypervisor": "xen",
      "State": "available",
      "SriovNetSupport": "simple",
      "ImageId": "ami-1234567890EXAMPLE",
      "DeprecationTime": "2021-05-10T13:17:12.000Z"
      "UsageOperation": "RunInstances:0010",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sda1",
          "Ebs": {
            "SnapshotId": "snap-111222333444aaabb",
            "DeleteOnTermination": true,
            "VolumeType": "gp2",
            "VolumeSize": 10,
            "Encrypted": false
          }
        }
      ],
      "Architecture": "x86_64",
      "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-
GP2",
      "RootDeviceType": "ebs",
      "OwnerId": "123456789012",
      "RootDeviceName": "/dev/sda1",
      "CreationDate": "2019-05-10T13:17:12.000Z",
      "Public": true,
      "ImageType": "machine",
      "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
    }
  ]
}

```

Cancel the deprecation of an AMI

You can cancel the deprecation of an AMI, which removes the date and time from the **Deprecation time** field (console) or the `DeprecationTime` field from the [describe-images](#) output (AWS CLI). 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** check box, and then choose **Save**.

AWS CLI

To cancel the deprecation of an AMI

Use the [disable-image-deprecation](#) command and specify the ID of the AMI.

```
aws ec2 disable-image-deprecation \  
  --image-id ami-1234567890abcdef0
```

Expected output

```
{  
  "Return": "true"  
}
```

Disable an 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.

⚠ 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.

You can disable both private and public AMIs.

Topics

- [Costs](#)
- [Prerequisites](#)
- [Required IAM permissions](#)
- [Disable an AMI](#)
- [Describe disabled AMIs](#)
- [Re-enable a disabled AMI](#)

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 [Delete resources associated with your Amazon EBS-backed 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 [disable-image](#) command and specify the ID of the AMI.

```
aws ec2 disable-image --image-id ami-1234567890abcdef0
```

Expected output

```
{  
  "Return": "true"  
}
```

Describe disabled AMIs

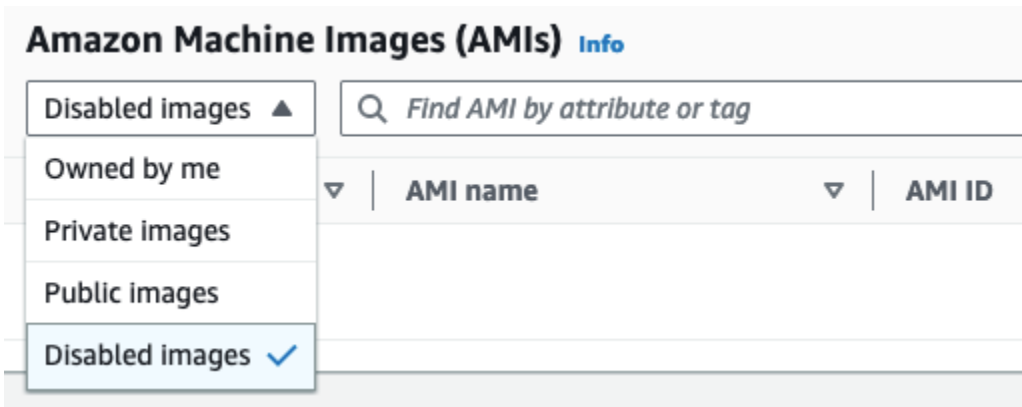
You can view disabled AMIs in the EC2 console and by using the AWS CLI.

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 use the [describe-images](#) command to describe all AMIs, disabled AMIs do not appear in the results. This is because the default is `--no-include-disabled`. To include disabled AMIs in the results, you must specify the `--include-disabled` parameter.

To include all disabled AMIs when describing all AMIs

Use the [describe-images](#) command and specify the `--include-disabled` parameter to retrieve disabled AMIs in addition to all other AMIs. Optionally, specify `--owners self` to retrieve only the AMIs that you own.

```
aws ec2 describe-images \
  --region us-east-1 \
  --owners self
  --include-disabled
```

If you specify the ID of a disabled AMI, but do not specify `--include-disabled`, the disabled AMI is returned in the results.

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

To retrieve only disabled AMIs

Specify `--filters Name=state,Values=disabled`. You must also specify `--include-disabled`, otherwise you'll get an error.

```
aws ec2 describe-images \  
  --include-disabled \  
  --filters Name=state,Values=disabled
```

Example output

The State field displays the state of an AMI. disabled indicates that the AMI is disabled.

```
{  
  "Images": [  
    {  
      "VirtualizationType": "hvm",  
      "Description": "Provided by Red Hat, Inc.",  
      "PlatformDetails": "Red Hat Enterprise Linux",  
      "EnaSupport": true,  
      "Hypervisor": "xen",  
      "State": "disabled",  
      "SriovNetSupport": "simple",  
      "ImageId": "ami-1234567890EXAMPLE",  
      "DeprecationTime": "2023-05-10T13:17:12.000Z"  
      "UsageOperation": "RunInstances:0010",  
      "BlockDeviceMappings": [  
        {  
          "DeviceName": "/dev/sda1",  
          "Ebs": {  
            "SnapshotId": "snap-1112223333444aaabb",  
            "DeleteOnTermination": true,  
            "VolumeType": "gp2",  
            "VolumeSize": 10,  
            "Encrypted": false  
          }  
        }  
      ],  
      "Architecture": "x86_64",  
      "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-  
GP2",  
      "RootDeviceType": "ebs",  
      "OwnerId": "123456789012",  
      "RootDeviceName": "/dev/sda1",  
      "CreationDate": "2019-05-10T13:17:12.000Z",  
      "Public": false,  
      "ImageType": "machine",  
      "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
```

```
    ]  
  }  
}
```

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 [enable-image](#) command and specify the ID of the AMI.

```
aws ec2 enable-image --image-id ami-1234567890abcdef0
```

Expected output

```
{  
  "Return": "true"  
}
```

Archive AMI snapshots

You can archive the snapshots that are associated with your disabled EBS-backed AMIs. This can help you to reduce the storage costs associated with your rarely-used AMIs that need to be

retained for long periods. For more information, see [Archive Amazon EBS snapshots](#) in the *Amazon EBS User Guide*.

To archive snapshots associated with an AMI

1. [Disable the AMI](#).
2. [Archive the snapshots](#).

You can't use an AMI while it is disabled and its associated snapshots are archived.

To restore a disabled AMI with archived snapshots for use

1. [Restore the archived snapshots](#) associated with the AMI.
2. [Enable the AMI](#).

Deregister (delete) an AMI

When you deregister an AMI, Amazon EC2 permanently deletes it. Once deregistered, you can't use the AMI 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.

Deregistering an AMI has no effect on any instances that were launched from the AMI. You can continue to use these instances. 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. For more information, see [Avoid costs from unused resources](#).

Contents

- [Considerations](#)
- [Deregister an AMI](#)
- [Check when an AMI was last used](#)
- [Protect an AMI from deregistration](#)

- [Avoid costs from unused resources](#)

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

Use any of the following methods to deregister an EBS-backed AMI or instance store-backed AMI.

Tip

To avoid incurring unnecessary costs, you should delete any resources that you do not need. For example, for EBS-backed AMIs, if you do not need the snapshots associated with the deregistered AMI, you should delete them. For more information, see [Avoid costs from unused resources](#).

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. When you are prompted for confirmation, 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 [deregister-image](#) command and specify the ID of the AMI to deregister.

```
aws ec2 deregister-image --image-id ami-0123456789example
```

Powershell

To deregister an AMI

Use the [Unregister-EC2Image](#) cmdlet and specify the ID of the AMI to deregister.

```
Unregister-EC2Image -ImageId ami-0123456789example
```

Check when an AMI was last used

LastLaunchedTime is a timestamp that indicates when your AMI was last used to launch an instance. AMIs that have not been used recently to launch an instance might be good candidates for deregistering or [deprecation](#).

Note

- When the AMI is used to launch an instance, there is a 24-hour delay before that usage is reported.
- LastLaunchedTime 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 AMI, and then check the **Last launched time** field (if you selected the check box next to the AMI, it's located on the **Details** tab). The field shows the date and time when the AMI was last used to launch an instance.

AWS CLI

You can use either the [describe-images](#) or [describe-image-attribute](#) command to view the last launched time of an AMI.

To view the last launched time of an AMI by using describe-images

Use the [describe-images](#) command and specify the ID of the AMI.

```
aws ec2 describe-images --image-id ami-0123456789example
```

Example output

Note

The LastLaunchedTime field only appears in the output for AMIs that you own.

```
{
  "Images": [
    {
      ...
      "LastLaunchedTime": {
        "Value": "2024-04-02T02:03:18Z"
      },
      ...
    }
  ]
}
```

To view the last launched time of an AMI

Use the [describe-image-attribute](#) command and specify `--attribute lastLaunchedTime`. You must be owner of the AMI to run this command.

```
aws ec2 describe-image-attribute \
```

```
--image-id ami-0123456789example \  
--attribute lastLaunchedTime
```

Example output

```
{  
  "ImageId": "ami-1234567890example",  
  "LastLaunchedTime": {  
    "Value": "2022-02-10T02:03:18Z"  
  }  
}
```

Protect an 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 any of the following methods to turn on deregistration protection on an AMI. To do this, you must be the owner of the AMI.

Console

To turn on deregistration protection on 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 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 on an AMI

Use the [enable-image-deregistration-protection](#) command and specify the AMI ID. To include the optional 24-hour cooldown period, include `--with-cooldown` set to `true`. To exclude the cooldown period, omit the `--with-cooldown` parameter.

```
aws ec2 enable-image-deregistration-protection \  
  --image-id ami-0123456789example \  
  --with-cooldown true
```

Turn off deregistration protection

Use any of the following methods to turn off deregistration protection on an AMI. To do this, you must be the owner of the AMI.

Note

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 on 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 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 on an AMI

Use the [disable-image-deregistration-protection](#) command and specify the AMI ID.

```
aws ec2 disable-image-deregistration-protection --image-id ami-0123456789example
```

Avoid costs from unused resources

When you deregister an AMI, you don't delete 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. For more information, see [How you're charged](#).

To avoid incurring these types of unnecessary costs, we recommend deleting any resources that you don't need.

To determine whether your AMI is EBS-backed or instance store-backed, see [Determine the root device type of your AMI](#).

Delete resources associated with your Amazon EBS-backed AMI

Use any of the following methods to delete the resources associated with your EBS-backed AMI.

Console

To delete resources associated with your EBS-backed AMI

1. [Deregister the AMI.](#)

Note the AMI ID—this can help you find the snapshots to delete in the next step.

2. [Delete snapshots](#) that you don't need.

The ID of the associated AMI is displayed in the **Description** column on the **Snapshots** screen.

3. [Terminate instances](#) that you don't need.

AWS CLI

To delete resources associated with your EBS-backed AMI

1. Deregister the AMI by using the [deregister-image](#) command.

```
aws ec2 deregister-image --image-id ami-0123456789example
```

2. Delete snapshots that you don't need by using the [delete-snapshot](#) command.

```
aws ec2 delete-snapshot --snapshot-id snap-0123456789example
```

3. Terminate instances that you don't need by using the [terminate-instances](#) command.

```
aws ec2 terminate-instances --instance-ids i-0123456789example
```

PowerShell

To delete resources associated with your EBS-backed AMI

1. Deregister the AMI by using the [Unregister-EC2Image](#) cmdlet.

```
Unregister-EC2Image -ImageId ami-0123456789example
```

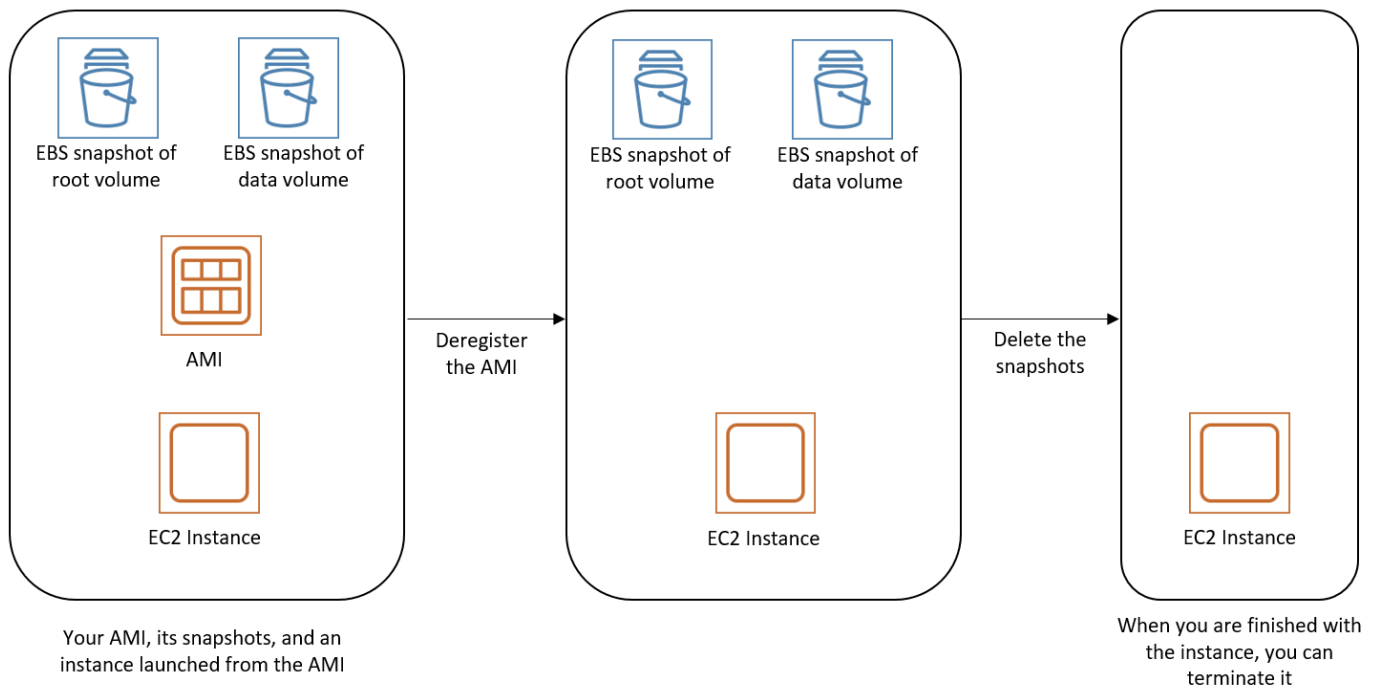
2. Delete snapshots that you don't need by using the [Remove-EC2Snapshot](#) cmdlet.

```
Remove-EC2Snapshot -SnapshotId snap-0123456789example
```

3. Terminate instances that you don't need by using the [Remove-EC2Instance](#) cmdlet.

```
Remove-EC2Instance -InstanceId i-0123456789example
```

The following diagram illustrates the flow for you to delete resources associated with an EBS-backed AMI.



Delete resources associated with your instance store-backed AMI

Use the following method to delete the resources associated with your instance store-backed AMI.

To delete resources associated with your instance store-backed AMI

1. Deregister the AMI by using the [deregister-image](#) command.

```
aws ec2 deregister-image --image-id ami-0123456789example
```

2. Delete the bundle in Amazon S3 by using the [ec2-delete-bundle](#) (AMI tools) command.

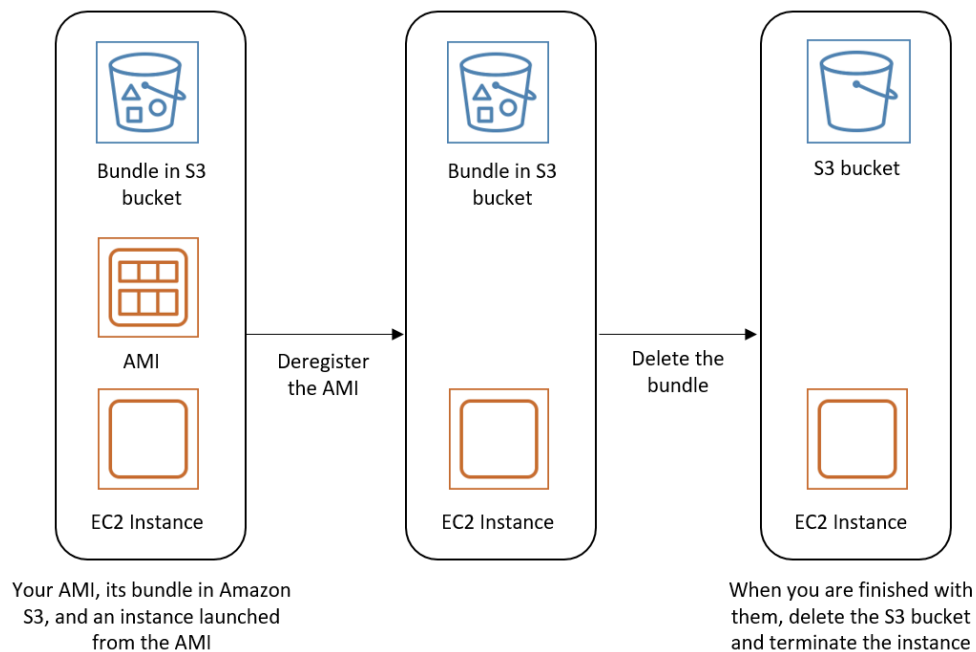
```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s your_secret_access_key -p image
```

3. Terminate instances that you don't need by using the [terminate-instances](#) command.

```
aws ec2 terminate-instances --instance-ids i-0123456789example
```

4. If you are finished with the Amazon S3 bucket that you uploaded the bundle to, you can delete the bucket. To delete an Amazon S3 bucket, open the Amazon S3 console, select the bucket, choose **Actions**, and then choose **Delete**.

The following diagram illustrates the flow for you to delete resources associated with your instance store-backed AMI.



Automate the EBS-backed AMI lifecycle

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

Use encryption with EBS-backed AMIs

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

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

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

AMIs with encrypted snapshots can be shared across AWS accounts. For more information, see [Shared AMIs](#).

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 about block device mapping, see [Block device mapping](#). For examples of controlling block device mapping from the AWS CLI, see [Launch, List, and Terminate EC2 Instances](#).

By default, without explicit encryption parameters, a RunInstances action maintains the existing encryption state of an AMI's source snapshots while restoring EBS volumes from them.

If encryption by default 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.

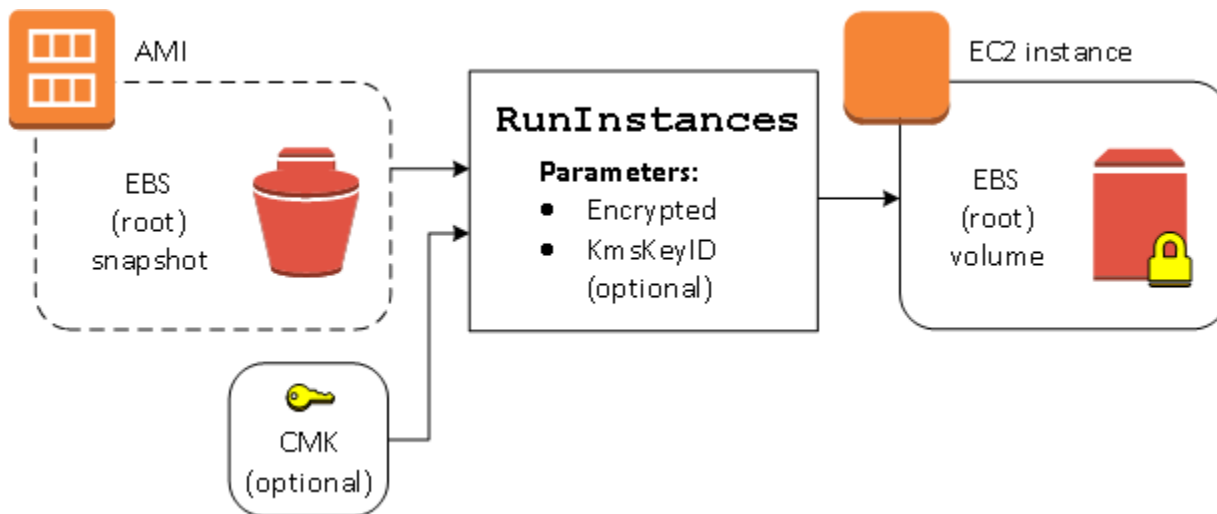
Submitting a `KmsKeyId` without also setting the `Encrypted` parameter results in an error.

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 your instance](#).

Encrypt a volume during launch

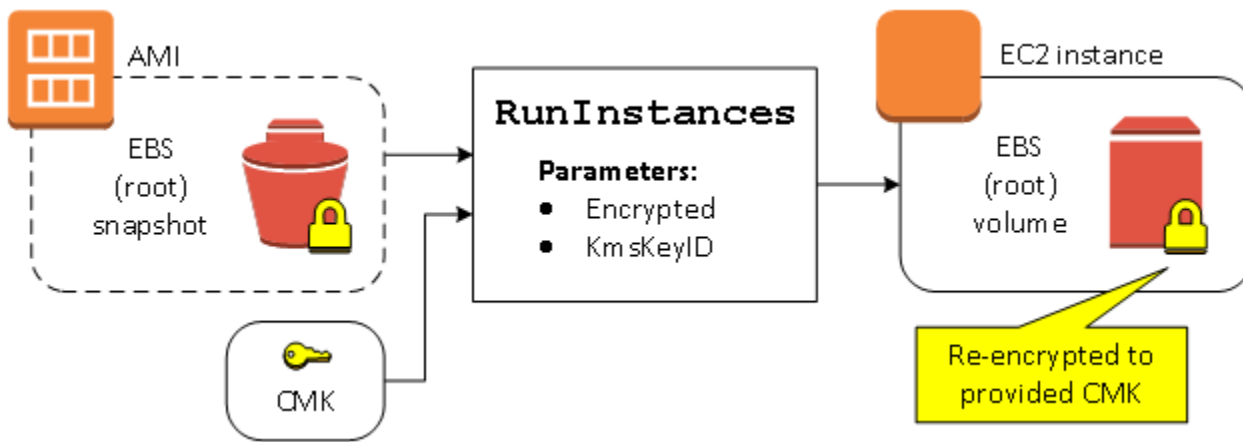
In this example, an AMI backed by an unencrypted snapshot is used to launch an EC2 instance with an encrypted EBS volume.



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

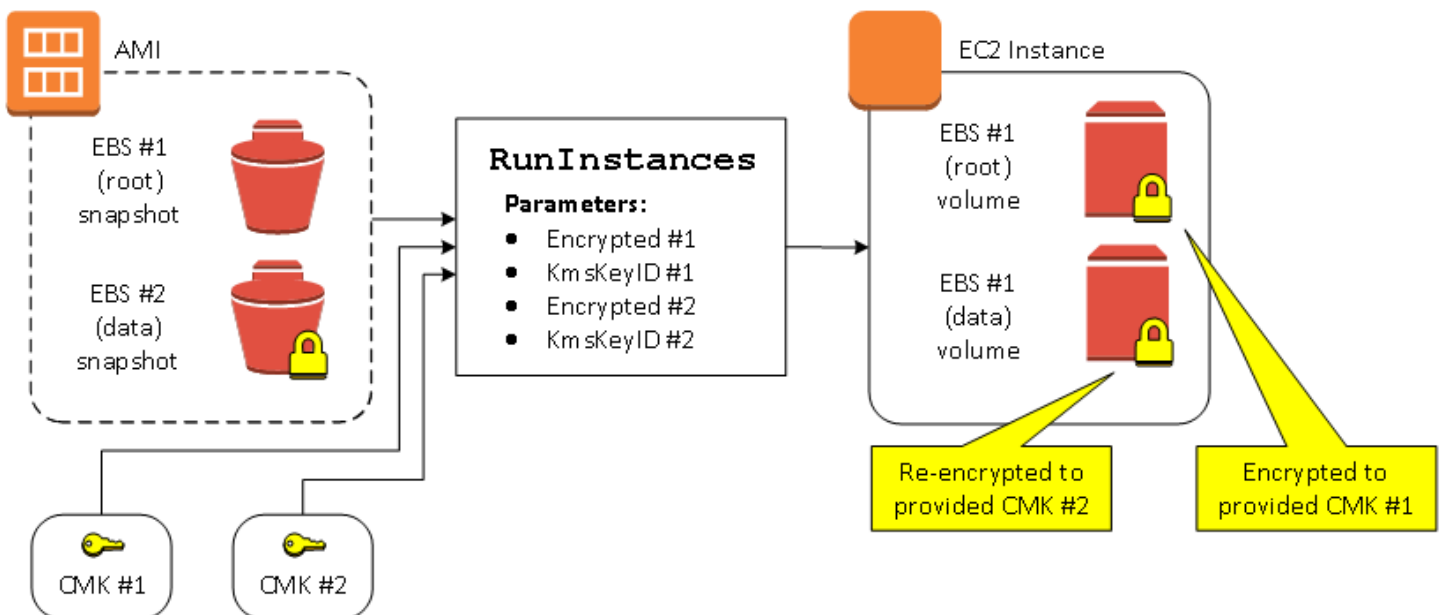
In this example, an AMI backed by an encrypted snapshot is used to launch an EC2 instance with an EBS volume encrypted by a new KMS key.



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.

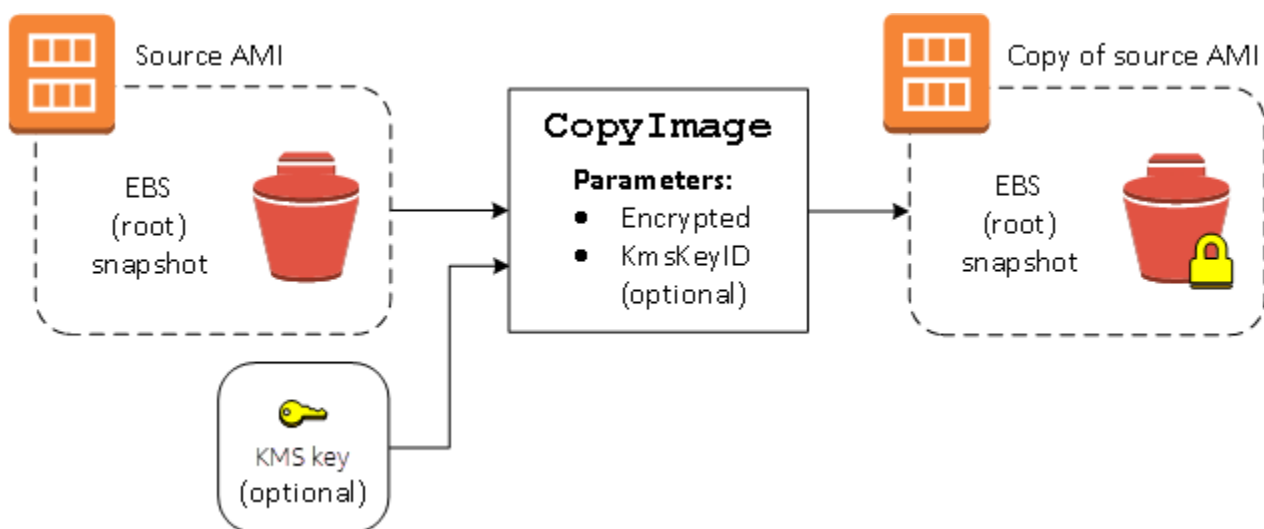
For detailed instructions using the console, see [Copy an AMI](#).

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.

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). 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 event when an AMI enters any of the following states:

- available
- failed
- deregistered
- disabled

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		

AMI operations	available	failed	deregistered	disabled
DeregisterImage			Yes	
DisableImage				Yes
EnableImage	Yes			
RegisterImage	Yes	Yes		

Events are generated on a best effort basis.

Topics

- [AMI events](#)
- [Create Amazon EventBridge rules](#)

AMI events

There are four EC2 AMI State Change events:

- [available](#)
- [failed](#)
- [deregistered](#)
- [disabled](#)

The events are sent to the default EventBridge event bus in JSON format.

The following fields in the event can be used 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-0123456789example", "State": "available", }
```

Provides the following information:

- The AMI ID – If you want to track a specific AMI.
- The state of the AMI (available, failed, deregistered, or disabled).

available

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-0123456789example"],
  "detail": {
    "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
    "ImageId": "ami-0123456789example",
    "State": "available",
    "ErrorMessage": ""
  }
}
```

failed

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-0123456789example"],
"detail": {
  "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
  "ImageId": "ami-0123456789example",
  "State": "failed",
  "ErrorMessage": "Description of failure"
}
}
```

deregistered

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-0123456789example"],
  "detail": {
    "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
    "ImageId": "ami-0123456789example",
    "State": "deregistered",
    "ErrorMessage": ""
  }
}
```

disabled

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-0123456789example"],
  "detail": {
    "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
    "ImageId": "ami-0123456789example",
    "State": "disabled",
    "ErrorMessage": ""
  }
}
```

Create Amazon EventBridge rules

You can create an Amazon EventBridge [rule](#) that specifies an action to take when EventBridge receives an [event](#) that matches the [event pattern](#) in the rule. When an event matches, EventBridge sends the event to the specified [target](#) and triggers the action defined in the rule.

Event patterns have the same structure as the events they match. An event pattern either matches an event or it doesn't.

When creating a rule for an AMI state change event, you can include the following fields in the event pattern:

```
"source": "aws.ec2"
```

Identifies that the event is from Amazon EC2.

```
"detail-type": "EC2 AMI State Change"
```

Identifies the event name.

```
"detail": { "ImageId": "ami-0123456789example", "State": "available", }
```

Provides the following information:

- The AMI ID – If you want to track a specific AMI.
- The state of the AMI (available, failed, deregistered, or disabled).

Example: 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 when any AMI is in the `available` state after the `CreateImage` operation has completed successfully.

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 AMI is created and in the `available` 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 any EC2 AMI State Change event that is generated when an AMI enters the available state:

```
{
  "source": ["aws.ec2"],
  "detail-type": ["EC2 AMI State Change"],
  "detail": {"State": ["available"]}
}
```

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**.
 - D. For **Event type**, choose **EC2 AMI State Change**.
 - E. To customize the template, choose **Edit pattern** and make your changes to match the example event pattern.
 - ii. 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 the following topics in the *Amazon EventBridge User Guide*:

- [Amazon EventBridge events](#)
- [Amazon EventBridge event patterns](#)
- [Amazon EventBridge rules](#)

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*.

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](#)
- [Finding AMI billing and usage details](#)
- [Verify AMI charges on your bill](#)

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

Platform details	Usage operation ²
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
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

Platform details	Usage operation ²
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

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

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

- **Platform details**
- **Usage operation**
- **AMI ID**

Find AMI billing information (console)

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

Look up AMI billing information from the AMIs page

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

Look up AMI billing information from the Instances page

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

Find AMI billing information (AWS CLI)

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

To find the AMI ID

If you know the instance ID, you can get the AMI ID for the instance by using the [describe-instances](#) command.

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

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

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

To find the AMI billing information

If you know the AMI ID, you can use the [describe-images](#) command to get the AMI platform and usage operation details.

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

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

```
{
  "Images": [
    {
      "VirtualizationType": "hvm",
      "Description": "Provided by Red Hat, Inc.",
      "Hypervisor": "xen",
      "EnaSupport": true,
      "SriovNetSupport": "simple",
      "ImageId": "ami-0123456789EXAMPLE",
      "State": "available",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sda1",
          "Ebs": {
            "SnapshotId": "snap-111222333444aaabb",
            "DeleteOnTermination": true,
            "VolumeType": "gp2",
            "VolumeSize": 10,
            "Encrypted": false
          }
        }
      ],
      "Architecture": "x86_64",
      "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-
GP2",
      "RootDeviceType": "ebs",
      "OwnerId": "123456789012",
      "PlatformDetails": "Red Hat Enterprise Linux",
      "UsageOperation": "RunInstances:0010",
      "RootDeviceName": "/dev/sda1",
      "CreationDate": "2019-05-10T13:17:12.000Z",
      "Public": true,
      "ImageType": "machine",
      "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
    }
  ]
}
```

Verify AMI charges on your bill

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

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

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

- **Platform details** = Red Hat Enterprise Linux
- **Usage operation** = RunInstances:0010

If you launched an instance using this AMI, you can find the instance ID in your CUR, and check the corresponding value in the [lineitem/Operation](#) column. In this example, the value should be `RunInstances:0010`.

Amazon Linux

Amazon Linux is provided by Amazon Web Services (AWS). It is designed to provide a stable, secure, and high-performance execution environment for applications running on Amazon EC2. It also includes packages that enable easy integration with AWS, including launch configuration tools and many popular AWS libraries and tools. AWS provides ongoing security and maintenance updates for all instances running Amazon Linux. Many applications developed on CentOS (and similar distributions) run on Amazon Linux.

Contents

- [Amazon Linux availability](#)
- [Connect to an Amazon Linux instance](#)
- [Identify Amazon Linux images](#)
- [Amazon Linux 2 AMI boot mode](#)
- [AWS command line tools](#)
- [Package repository](#)

- [Extras library \(Amazon Linux 2\)](#)
- [Amazon Linux 2 supported kernels](#)
- [Access source packages for reference](#)
- [cloud-init](#)
- [Subscribe to Amazon Linux notifications](#)
- [Run Amazon Linux 2 as a virtual machine on premises](#)
- [Kernel Live Patching on Amazon Linux 2](#)

Amazon Linux availability

AWS provides AL2023, Amazon Linux 2, and the Amazon Linux AMI. If you are migrating from another Linux distribution to Amazon Linux, we recommend that you migrate to AL2023.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

For more information about Amazon Linux, see [AL2023](#), [Amazon Linux 2](#), and [Amazon Linux AMI](#).

For Amazon Linux container images, see [Amazon Linux container image](#) in the *Amazon Elastic Container Registry User Guide*.

Connect to an Amazon Linux instance

Amazon Linux does not allow remote root secure shell (SSH) by default. Also, password authentication is disabled to prevent brute-force password attacks. To enable SSH logins to an Amazon Linux instance, you must provide your key pair to the instance at launch. You must also set the security group used to launch your instance to allow SSH access. By default, the only account that can log in remotely using SSH is `ec2-user`; this account also has **sudo** privileges. If you enable remote root login, be aware that it is less secure than relying on key pairs and a secondary user.

Identify Amazon Linux images

Each image contains a unique `/etc/image-id` file that identifies it. This file contains the following information about the image:

- `image_name`, `image_version`, `image_arch` – Values from the build recipe that Amazon used to construct the image.
- `image_stamp` – A unique, random hex value generated during image creation.
- `image_date` – The UTC time of image creation, in `YYYYMMDDhhmmss` format.
- `recipe_name`, `recipe_id` – The name and ID of the build recipe Amazon used to construct the image.

Amazon Linux contains an `/etc/system-release` file that specifies the current release that is installed. This file is updated using **yum** and is part of the `system-release` RPM Package Manager (RPM).

Amazon Linux also contains a machine-readable version of `/etc/system-release` that follows the Common Platform Enumeration (CPE) specification; see `/etc/system-release-cpe`.

Amazon Linux 2

The following is an example of `/etc/image-id` for the current version of Amazon Linux 2.

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn2-ami-hvm"
image_version="2"
image_arch="x86_64"
image_file="amzn2-ami-hvm-2.0.20180810-x86_64.xfs.gpt"
image_stamp="8008-2abd"
image_date="20180811020321"
recipe_name="amzn2 ami"
recipe_id="c652686a-2415-9819-65fb-4dee-9792-289d-1e2846bd"
```

The following is an example of `/etc/system-release` for the current version of Amazon Linux 2.

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux 2
```

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

```
[ec2-user ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
```

Amazon Linux AMI

The following is an example of `/etc/image-id` for the current Amazon Linux AMI.

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2018.03"
image_arch="x86_64"
image_file="amzn-ami-hvm-2018.03.0.20180811-x86_64.ext4.gpt"
image_stamp="cc81-f2f3"
image_date="20180811012746"
recipe_name="amzn ami"
recipe_id="5b283820-dc60-a7ea-d436-39fa-439f-02ea-5c802dbd"
```

The following is an example of `/etc/system-release` for the current Amazon Linux AMI.

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux AMI release 2018.03
```

Amazon Linux 2 AMI boot mode

Amazon Linux 2 AMIs don't have a boot mode parameter set. Instances launched from Amazon Linux 2 AMIs follow the default boot mode value of the instance type. For more information, see [Boot modes](#).

AWS command line tools

The AWS Command Line Interface (AWS CLI) is an open source tool that provides a consistent interface to interact with AWS services using commands in your command-line shell. For more

information, see [What is the AWS Command Line Interface?](#) in the *AWS Command Line Interface User Guide*.

Amazon Linux 2 and the Amazon Linux AMI have version 1 of the AWS CLI preinstalled. The current release of Amazon Linux, AL2023, has version 2 of the AWS CLI preinstalled. For more information about using the AWS CLI on AL2023, see [Get started with AL2023](#) in the *AL2023 User Guide*.

Package repository

This information applies to Amazon Linux 2 and the Amazon Linux AMI. For information about AL2023, see [Managing packages and operating system updates](#) in the *AL2023 User Guide*.

Amazon Linux 2 and the Amazon Linux AMI are designed to be used with online package repositories hosted in each Amazon EC2 AWS Region. The repositories are available in all Regions and are accessed using **yum** update tools. Hosting repositories in each Region enables us to deploy updates quickly and without any data transfer charges.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

If you don't need to preserve data or customizations for your instances, you can launch new instances using the current Amazon Linux 2 AMI. If you need to preserve data or customizations for your instances, you can maintain those instances through the Amazon Linux package repositories. These repositories contain all the updated packages. You can choose to apply these updates to your running instances. Older versions of the AMI and update packages continue to be available for use, even as new versions are released.

Note

To update and install packages without internet access on an EC2 instance, see [How can I update yum or install packages without internet access on my Amazon EC2 instances running Amazon Linux, Amazon Linux 2, or AL2023?](#)

To install packages, use the following command:

```
[ec2-user ~]$ sudo yum install package
```

If you find that Amazon Linux doesn't contain an application that you need, you can install the application directly on your Amazon Linux instance. Amazon Linux uses RPMs and yum for package management, and that is likely the simplest way to install new applications. You should check to see if an application is available in our central Amazon Linux repository first, because many applications are available there. These applications can be easily added to your Amazon Linux instance.

To upload your applications onto a running Amazon Linux instance, use **scp** or **sftp** and then configure the application by logging in to your instance. Your applications can also be uploaded during the instance launch by using the **PACKAGE_SETUP** action from the built-in cloud-init package. For more information, see [cloud-init](#).

Security updates

Security updates are provided using the package repositories as well as updated AMI security alerts are published in the [Amazon Linux Security Center](#). For more information about AWS security policies or to report a security problem, see [AWS Cloud Security](#).

Amazon Linux and Amazon Linux 2 are configured to download and install critical or important security updates at launch time. Kernel updates are not included in this configuration.

In AL2023, this configuration has changed compared to Amazon Linux AMI and Amazon Linux 2. For more information about security updates for AL2023, see [Security updates and features](#) in the *AL2023 User Guide*.

We recommend that you make the necessary updates for your use case after launch. For example, you may want to apply all updates (not just security updates) at launch, or evaluate each update and apply only the ones applicable to your system. This is controlled using the following cloud-init setting: `repo_upgrade`. The following snippet of cloud-init configuration shows how you can change the settings in the user data text you pass to your instance initialization:

```
#cloud-config
repo_upgrade: security
```

The possible values for `repo_upgrade` are as follows:

critical

Apply outstanding critical security updates.

important

Apply outstanding critical and important security updates.

medium

Apply outstanding critical, important, and medium security updates.

low

Apply all outstanding security updates, including low-severity security updates.

security

Apply outstanding critical or important updates that Amazon marks as security updates.

bugfix

Apply updates that Amazon marks as bug fixes. Bug fixes are a larger set of updates, which include security updates and fixes for various other minor bugs.

all

Apply all applicable available updates, regardless of their classification.

none

Don't apply any updates to the instance on start up.

The default setting for `repo_upgrade` is `security`. That is, if you don't specify a different value in your user data, by default, Amazon Linux performs the security upgrades at launch for any packages installed at that time. Amazon Linux also notifies you of any updates to the installed packages by listing the number of available updates upon login using the `/etc/motd` file. To install these updates, you need to run **`sudo yum upgrade`** on the instance.

Repository configuration

For Amazon Linux and Amazon Linux 2, AMIs are a snapshot of the packages available at the time the AMI was created, with the exception of security updates. Any packages not on the original AMI, but installed at run time, will be the latest version available. To get the latest packages available for Amazon Linux 2, run **`yum update -y`**.

Troubleshooting tip

If you get a `cannot allocate memory` error running **yum update** on nano instance types, such as `t3.nano`, you might need to allocate swap space to enable the update.

For AL2023, the repository configuration has changed compared to the Amazon Linux AMI and Amazon Linux 2. For more information about the AL2023 repository, see [Managing packages and operating system updates](#).

Versions up to AL2023 were configured to deliver a continuous flow of updates to roll from one minor version of Amazon Linux to the next version, also called *rolling releases*. As a best practice, we recommend you update your AMI to the latest available AMI. Don't launch Amazon Linux AMIs that use an older version, such as 2017.09.

In-place upgrades are not supported between major Amazon Linux versions, such as from Amazon Linux to Amazon Linux 2 or from Amazon Linux 2 to AL2023. For more information, see [Amazon Linux availability](#).

Using lock-on-launch in Amazon Linux

You can disable rolling releases by enabling the *lock-on-launch* feature. The lock-on-launch feature locks your instance to receive updates only from the specified release of the AMI. For example, you can launch a 2017.09 AMI and have it receive only the updates that were released prior to the 2018.03 AMI, until you are ready to migrate to the 2018.03 AMI.

Important

If you enable the lock-on-launch feature and choose a version of the repositories that is not the latest, you don't receive further updates. To receive rolling releases, you must use the latest AMI, or consistently update your AMI with the repositories pointed to latest.

To enable the lock-on-launch feature in new instances, launch it with the following user data passed to cloud-init:

```
#cloud-config
repo_releasever: 2017.09
```

To lock existing instances to their current AMI version

1. Edit `/etc/yum.conf`.
2. Comment out `releasever=latest`.
3. To clear the cache, run **yum clean all**.

Extras library (Amazon Linux 2)

With Amazon Linux 2, you can use the Extras Library to install application and software updates on your instances. These software updates are known as *topics*. You can install a specific version of a topic or omit the version information to use the most recent version.

To list the available topics, use the following command:

```
[ec2-user ~]$ amazon-linux-extras list
```

To enable a topic and install the latest version of its package to ensure freshness, use the following command:

```
[ec2-user ~]$ sudo amazon-linux-extras install topic
```

To enable topics and install specific versions of their packages to ensure stability, use the following command:

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

To remove a package installed from a topic, use the following command:

```
[ec2-user ~]$ sudo yum remove $(yum list installed | grep amzn2extra-topic | awk '{ print $1 }')
```

Note

This command does not remove packages that were installed as dependencies of the extra.

To disable a topic and make the packages inaccessible to the yum package manager, use the following command:

```
[ec2-user ~]$ sudo amazon-linux-extras disable topic
```

Important

This command is intended for advanced users. Improper usage of this command could cause package compatibility conflicts.

Amazon Linux 2 supported kernels

Supported kernel versions

Currently, Amazon Linux 2 (AL2) AMIs are available with kernel versions 4.14 and 5.10, with version 5.10 being a default. You also have an option of upgrading the kernel on AL2 to version 5.15 by using the extras repository. Note that an upgrade to 5.15 requires a reboot for the new kernel to take effect. Review new features and limitations of the kernel version 5.15 on AL2 before deciding whether an upgrade is required for your use case. If you require live patching support, we recommend you use AL2 AMI with kernel 5.10.

AL2023 AMIs are available with kernel version 6.1. For more information, see [AL2023 Kernel changes from Amazon Linux 2](#) in the *AL2023 User Guide*.

New features in kernel 5.15

- [Kernel-based Virtual Machine](#) (KVM) now defaults to the new x86 TDP MMU and adds AMD SVM 5-level paging to allow for greater parallelism and scalability compared to the original KVM x86 MMU code.
- [OverlayFS](#) has improved performance and now also handles copying immutable/append/sync/ noatime attributes.
- New optimizations and improvements for EXT4 are added, such as addition of a new `orphan_file` feature to eliminate bottlenecks in cases of large parallel truncates, file deletions and moving the DISCARD work out of the JBD2 commit thread to help with devices having slow DISCARD behavior and not blocking the JBD2 commit KThread.
- New optimizations and improvements for XFS are added, such as batch inode activations in per-CPU background threads that improve directory tree deletion times and enablement of pipelining to help with performance around handling lots of metadata updates.

- [DAMON](#) is better supported as the data access monitoring framework for proactive memory reclamation and performance analysis.

Limitations for kernel 5.15

- LustreFSx is not supported (support will be added later).
- Kernel live patching is not supported.

Instructions for installing kernel 5.15

You can upgrade to kernel 5.15 from both Amazon Linux 2 AMI with kernel 4.14 and AL2 AMI with kernel 5.10 using the following commands:

1. Enable the `kernel-5.15` topic in `amazon-linux-extras` and install kernel 5.15 on the host.

```
sudo amazon-linux-extras install kernel-5.15
```

2. Reboot the host with the installed kernel 5.15.

```
sudo reboot
```

3. Check the system kernel version.

```
uname -r
```

Support Timeframe

All Linux kernels available on Amazon Linux 2 (4.14, 5.10, and 5.15) will be supported until Amazon Linux 2 AMI reaches the end of standard support.

Live patching support

Amazon Linux 2 kernel version	Kernel live patching supported
4.14	Yes
5.10	Yes
5.15	No

Access source packages for reference

You can view the source of packages you have installed on your instance for reference purposes by using tools provided in Amazon Linux. Source packages are available for all of the packages included in Amazon Linux and the online package repository. Simply determine the package name for the source package you want to install and use the **yumdownloader --source** command to view source within your running instance. For example:

```
[ec2-user ~]$ yumdownloader --source bash
```

The source RPM can be unpacked, and, for reference, you can view the source tree using standard RPM tools. After you finish debugging, the package is available for use.

cloud-init

The cloud-init package is an open-source application built by Canonical used to bootstrap Linux images in a cloud computing environment, such as Amazon EC2. Amazon Linux contains a customized version of cloud-init. With the customized version of cloud-init, you can specify the actions that should happen to your instance at boot time. You can pass desired actions to cloud-init through the user data fields when launching an instance. This means you can use common AMIs for many use cases and configure them dynamically at startup. Amazon Linux also uses cloud-init to perform initial configuration of the ec2-user account.

For more information, see the [cloud-init documentation](#).

Amazon Linux uses the cloud-init actions found in `/etc/cloud/cloud.cfg.d` and `/etc/cloud/cloud.cfg`. You can create your own cloud-init action files in `/etc/cloud/cloud.cfg.d`. All files in this directory are read by cloud-init. They are read in lexical order, and later files overwrite values in earlier files.

The cloud-init package performs these (and other) common configuration tasks for instances at boot:

- Set the default locale.
- Set the hostname.
- Parse and handle user data.
- Generate host private SSH keys.

- Add a user's public SSH keys to `.ssh/authorized_keys` for easy login and administration.
- Prepare the repositories for package management.
- Handle package actions defined in user data.
- Execute user scripts found in user data.
- Mount instance store volumes, if applicable.
 - By default, the `ephemeral0` instance store volume is mounted at `/media/ephemeral0` if it is present and contains a valid file system; otherwise, it is not mounted.
 - By default, any swap volumes associated with the instance are mounted (only for `m1.small` and `c1.medium` instance types).
 - You can override the default instance store volume mount with the following cloud-init directive:

```
#cloud-config
mounts:
- [ ephemeral0 ]
```

For more control over mounts, see [Mounts](#) in the cloud-init documentation.

- Instance store volumes that support TRIM are not formatted when an instance launches, so you must partition and format them before you can mount them. For more information, see [Instance store volume TRIM support](#). You can use the `disk_setup` module to partition and format your instance store volumes at boot. For more information, see [Disk Setup](#) in the cloud-init documentation.

Supported user-data formats

The cloud-init package supports user-data handling of a variety of formats:

- Gzip
 - If user-data is gzip compressed, cloud-init decompresses the data and handles it appropriately.
- MIME multipart
 - Using a MIME multipart file, you can specify more than one type of data. For example, you could specify both a user-data script and a cloud-config type. Each part of the multipart file can be handled by cloud-init if it is one of the supported formats.
- Base64 decoding

- If user-data is base64-encoded, cloud-init determines if it can understand the decoded data as one of the supported types. If it understands the decoded data, it decodes the data and handles it appropriately. If not, it returns the base64 data intact.
- User-Data script
 - Begins with `#!` or `Content-Type: text/x-shellscript`.
 - The script is run by `/etc/init.d/cloud-init-user-scripts` during the first boot cycle. This occurs late in the boot process (after the initial configuration actions are performed).
- Include file
 - Begins with `#include` or `Content-Type: text/x-include-url`.
 - This content is an include file. The file contains a list of URLs, one per line. Each of the URLs is read, and their content passed through this same set of rules. The content read from the URL can be gzip compressed, MIME-multi-part, or plaintext.
- Cloud Config Data
 - Begins with `#cloud-config` or `Content-Type: text/cloud-config`.
 - This content is cloud-config data. For a commented example of supported configuration formats, see the examples.
- Upstart job (not supported on Amazon Linux 2)
 - Begins with `#upstart-job` or `Content-Type: text/upstart-job`.
 - This content is stored in a file in `/etc/init`, and upstart consumes the content as per other upstart jobs.
- Cloud Boothook
 - Begins with `#cloud-boothook` or `Content-Type: text/cloud-boothook`.
 - This content is boothook data. It is stored in a file under `/var/lib/cloud` and then runs immediately.
 - This is the earliest *hook* available. There is no mechanism provided for running it only one time. The boothook must take care of this itself. It is provided with the instance ID in the environment variable `INSTANCE_ID`. Use this variable to provide a once-per-instance set of boothook data.

Subscribe to Amazon Linux notifications

To be notified when new Amazon Linux AMIs are released, you can subscribe using Amazon SNS.

For information about subscribing to notifications for AL2023, see [Receiving notifications on new updates](#) in the *AL2023 User Guide*.

 **Note**

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

To subscribe to Amazon Linux notifications

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select the Region in which the SNS notification that you are subscribing to was created.
3. In the navigation pane, choose **Subscriptions, Create subscription**.
4. For the **Create subscription** dialog box, do the following:
 - a. [Amazon Linux 2] For **Topic ARN**, copy and paste the following Amazon Resource Name (ARN): `arn:aws:sns:us-east-1:137112412989:amazon-linux-2-ami-updates`.
 - b. [Amazon Linux] For **Topic ARN**, copy and paste the following Amazon Resource Name (ARN): `arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates`.
 - c. For **Protocol**, choose **Email**.
 - d. For **Endpoint**, enter an email address that you can use to receive the notifications.
 - e. Choose **Create subscription**.
5. You receive a confirmation email with the subject line "AWS Notification - Subscription Confirmation". Open the email and choose **Confirm subscription** to complete your subscription.

Whenever AMIs are released, we send notifications to the subscribers of the corresponding topic. To stop receiving these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon Linux notifications

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must use the Region in which the SNS notification was created.
3. In the navigation pane, choose **Subscriptions**, select the subscription, and choose **Actions, Delete subscriptions**.
4. When prompted for confirmation, choose **Delete**.

Amazon Linux AMI SNS message format

The schema for the SNS message is as follows.

```
{
  "description": "Validates output from AMI Release SNS message",
  "type": "object",
  "properties": {
    "v1": {
      "type": "object",
      "properties": {
        "ReleaseVersion": {
          "description": "Major release (ex. 2018.03)",
          "type": "string"
        },
        "ImageVersion": {
          "description": "Full release (ex. 2018.03.0.20180412)",
          "type": "string"
        },
        "ReleaseNotes": {
          "description": "Human-readable string with extra information",
          "type": "string"
        },
        "Regions": {
          "type": "object",
          "description": "Each key will be a region name (ex. us-east-1)",
          "additionalProperties": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "Name": {
```

```

        "description": "AMI Name (ex. amzn-ami-
hvm-2018.03.0.20180412-x86_64-gp2)",
        "type": "string"
    },
    "ImageId": {
        "description": "AMI Name (ex.ami-467ca739)",
        "type": "string"
    }
},
"required": [
    "Name",
    "ImageId"
]
}
}
},
"required": [
    "ReleaseVersion",
    "ImageVersion",
    "ReleaseNotes",
    "Regions"
]
}
},
"required": [
    "v1"
]
}
}

```

Run Amazon Linux 2 as a virtual machine on premises

Use the Amazon Linux 2 virtual machine (VM) images for on-premises development and testing. We offer a different Amazon Linux 2 VM image for each of the supported virtualization platforms. You can view the list of supported platforms on the [Amazon Linux 2 virtual machine images](#) page.

To use the Amazon Linux 2 virtual machine images with one of the supported virtualization platforms, do the following:

- [Step 1: Prepare the seed.iso boot image](#)
- [Step 2: Download the Amazon Linux 2 VM image](#)
- [Step 3: Boot and connect to your new VM](#)

Step 1: Prepare the seed.iso boot image

The `seed.iso` boot image includes the initial configuration information that is needed to boot your new VM, such as the network configuration, host name, and user data.

Note

The `seed.iso` boot image includes only the configuration information required to boot the VM. It does not include the Amazon Linux 2 operating system files.

To generate the `seed.iso` boot image, you need two configuration files:

- `meta-data` – This file includes the hostname and static network settings for the VM.
- `user-data` – This file configures user accounts, and specifies their passwords, key pairs, and access mechanisms. By default, the Amazon Linux 2 VM image creates an `ec2-user` user account. You use the `user-data` configuration file to set the password for the default user account.

To create the seed.iso boot disc

1. Create a new folder named `seedconfig` and navigate into it.
2. Create the `meta-data` configuration file.
 - a. Create a new file named `meta-data`.
 - b. Open the `meta-data` file using your preferred editor and add the following.

```
local-hostname: vm_hostname
# eth0 is the default network interface enabled in the image. You can configure
static network settings with an entry like the following.
network-interfaces: |
  auto eth0
  iface eth0 inet static
  address 192.168.1.10
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.254
```

Replace *vm_hostname* with a VM host name of your choice, and configure the network settings as required.

- c. Save and close the meta-data configuration file.

For an example meta-data configuration file that specifies a VM hostname (`amazonlinux.onprem`), configures the default network interface (`eth0`), and specifies static IP addresses for the necessary network devices, see the [sample Seed.iso file](#).

3. Create the user-data configuration file.
 - a. Create a new file named `user-data`.
 - b. Open the `user-data` file using your preferred editor and add the following.

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name `ec2-user` is created in the image by default.
- default
chpasswd:
  list: |
    ec2-user:plain_text_password
# In the above line, do not add any spaces after 'ec2-user:'.
```

Replace *plain_text_password* with a password of your choice for the default `ec2-user` user account.

- c. (Optional) By default, cloud-init applies network settings each time the VM boots. Add the following to prevent cloud-init from applying network settings at each boot, and to retain the network settings applied during the first boot.

```
# NOTE: Cloud-init applies network settings on every boot by default. To retain
network settings
# from first boot, add the following 'write_files' section:
write_files:
- path: /etc/cloud/cloud.cfg.d/80_disable_network_after_firstboot.cfg
  content: |
    # Disable network configuration after first boot
    network:
      config: disabled
```

- d. Save and close the `user-data` configuration file.

You can also create additional user accounts and specify their access mechanisms, passwords, and key pairs. For more information about the supported directives, see [Module reference](#). For an example `user-data` file that creates three additional users and specifies a custom password for the default `ec2-user` user account, see the [sample Seed.iso file](#).

4. Create the `seed.iso` boot image using the meta-data and user-data configuration files.

For Linux, use a tool such as **genisoimage**. Navigate into the `seedconfig` folder, and run the following command.

```
$ genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

For macOS, use a tool such as **hdiutil**. Navigate one level up from the `seedconfig` folder, and run the following command.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata  
seedconfig/
```

Step 2: Download the Amazon Linux 2 VM image

We offer a different Amazon Linux 2 VM image for each of the supported virtualization platforms. You can view the list of supported platforms and download the correct VM image for your chosen platform from the [Amazon Linux 2 virtual machine images](#) page.

Step 3: Boot and connect to your new VM

To boot and connect to your new VM, you must have the `seed.iso` boot image (created in [Step 1](#)) and an Amazon Linux 2 VM image (downloaded in [Step 2](#)). The steps vary depending on your chosen VM platform.

VMware vSphere

The VM image for VMware is made available in the OVF format.

To boot the VM using VMware vSphere

1. Create a new datastore for the `seed.iso` file, or add it to an existing datastore.

2. Deploy the OVF template, but do not start the VM yet.
3. In the **Navigator** panel, right-click the new virtual machine and choose **Edit Settings**.
4. On the **Virtual Hardware** tab, for **New device**, choose **CD/DVD Drive**, and then choose **Add**.
5. For **New CD/DVD Drive**, choose **Datastore ISO File**. Select the datastore to which you added the seed .iso file, browse to and select the seed .iso file, and then choose **OK**.
6. For **New CD/DVD Drive**, select **Connect**, and then choose **OK**.

After you have associated the datastore with the VM, you should be able to boot it.

KVM virt-manager

To boot the VM using KVM

1. Open the **Create new VM** wizard.
2. For Step 1, choose **Import existing disk image**.
3. For Step 2, browse to and select the VM image. For **OS type** and **Version**, choose **Linux** and **Red Hat Enterprise Linux 7.0** respectively.
4. For Step 3, specify the amount of RAM and the number of CPUs to use.
5. For Step 4, enter a name for the new VM and select **Customize configuration before install**, and choose **Finish**.
6. In the Configuration window for the VM, choose **Add Hardware**.
7. In the **Add New Virtual Hardware** window, choose **Storage**.
8. In the Storage configuration, choose **Select or create custom storage**. For **Device type**, choose **CDROM device**. Choose **Manage, Browse Local**, and then navigate to and select the seed .iso file. Choose **Finish**.
9. Choose **Begin Installation**.

Oracle VirtualBox

To boot the VM using Oracle VirtualBox

1. Open Oracle VirtualBox and choose **New**.
2. For **Name**, enter a descriptive name for the virtual machine, and for **Type** and **Version**, select **Linux** and **Red Hat (64-bit)** respectively. Choose **Continue**.

3. For **Memory size**, specify the amount of memory to allocate to the virtual machine, and then choose **Continue**.
4. For **Hard disk**, choose **Use an existing virtual hard disk file**, browse to and open the VM image, and then choose **Create**.
5. Before you start the VM, you must load the seed `.iso` file in the virtual machine's virtual optical drive:
 - a. Select the new VM, choose **Settings**, and then choose **Storage**.
 - b. In the **Storage Devices** list, under **Controller: IDE**, choose the *Empty* optical drive.
 - c. In the **Attributes** section for the optical drive, choose the browse button, select **Choose Virtual Optical Disk File**, and then select the seed `.iso` file. Choose **OK** to apply the changes and close the Settings.

After you have added the seed `.iso` file to the virtual optical drive, you should be able to start the VM.

Microsoft Hyper-V

The VM image for Microsoft Hyper-V is compressed into a zip file. You must extract the contents of the zip file.

To boot the VM using Microsoft Hyper-V

1. Open the **New Virtual Machine Wizard**.
2. When prompted to select a generation, select **Generation 1**.
3. When prompted to configure the network adapter, for **Connection** choose **External**.
4. When prompted to connect a virtual hard disk, choose **Use an existing virtual hard disk**, choose **Browse**, and then navigate to and select the VM image. Choose **Finish** to create the VM.
5. Right-click the new VM and choose **Settings**. In the **Settings** window, under **IDE Controller 1**, choose **DVD Drive**.
6. For the DVD drive, choose **Image file** and then browse to and select the seed `.iso` file.
7. Apply the changes and start the VM.

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

Kernel Live Patching on Amazon Linux 2

With Kernel Live Patching for Amazon Linux 2, you can apply security vulnerability and critical bug patches to a running Linux kernel, without reboots or disruptions to running applications. You benefit from improved service and application availability, while keeping your infrastructure secure and up to date.

For information about Kernel Live Patching for AL2023, see [Kernel Live Patching on AL2023](#) in the *AL2023 User Guide*.

AWS releases two types of kernel live patches for Amazon Linux 2:

- **Security updates** – Include updates for Linux common vulnerabilities and exposures (CVE). These updates are typically rated as *important* or *critical* using the Amazon Linux Security Advisory ratings. They generally map to a Common Vulnerability Scoring System (CVSS) score of 7 and higher. In some cases, AWS might provide updates before a CVE is assigned. In these cases, the patches might appear as bug fixes.
- **Bug fixes** – Include fixes for critical bugs and stability issues that are not associated with CVEs.

AWS provides kernel live patches for an Amazon Linux 2 kernel version for up to 3 months after its release. After the 3-month period, you must update to a later kernel version to continue to receive kernel live patches.

Amazon Linux 2 kernel live patches are made available as signed RPM packages in the existing Amazon Linux 2 repositories. The patches can be installed on individual instances using existing **yum** workflows, or they can be installed on a group of managed instances using AWS Systems Manager.

Kernel Live Patching on Amazon Linux 2 is provided at no additional cost.

Topics

- [Supported configurations and prerequisites](#)
- [Work with Kernel Live Patching](#)
- [Limitations](#)

- [Frequently asked questions](#)

Supported configurations and prerequisites

Kernel Live Patching is supported on Amazon EC2 instances and [on-premises virtual machines](#) running Amazon Linux 2.

To use Kernel Live Patching on Amazon Linux 2, you must use:

- Kernel version 4.14 or 5.10 on the x86_64 architecture
- Kernel version 5.10 on the ARM64 architecture

Policy Requirements

To download packages from Amazon Linux repositories, Amazon Elastic Compute Cloud needs access to service-owned Amazon S3 buckets. If you are using an Amazon Virtual Private Cloud (VPC) endpoint for Amazon S3 in your environment, you need to ensure that your VPC endpoint policy allows access to those public buckets.

The table describes each of the Amazon S3 buckets that EC2 might need to access for Kernel Live Patching.

S3 bucket ARN	Description
arn:aws:s3:::packages. <i>region</i> .amazonaws.com/*	Amazon S3 bucket containing Amazon Linux AMI packages
arn:aws:s3:::repo. <i>region</i> .amazonaws.com/*	Amazon S3 bucket containing Amazon Linux AMI repositories
arn:aws:s3:::amazonlinux. <i>region</i> .amazonaws.com/*	Amazon S3 bucket containing Amazon Linux 2 repositories
arn:aws:s3:::amazonlinux-2-repos- <i>region</i> /*	Amazon S3 bucket containing Amazon Linux 2 repositories

S3 bucket ARN	Description
---------------	-------------

The following policy illustrates how to restrict access to identities and resources that belong to your organization and provide access to the Amazon S3 buckets required for Kernel Live Patching. Replace *region*, *principal-org-id* and *resource-org-id* with your organization's values.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "principal-org-id",
          "aws:ResourceOrgID": "resource-org-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToAmazonLinuxAMIRepositories",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::packages.region.amazonaws.com/*",
        "arn:aws:s3:::repo.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux-2-repos-region/*"
      ]
    }
  ]
}
```

```
]
}
```

Work with Kernel Live Patching

You can enable and use Kernel Live Patching on individual instances using the command line on the instance itself, or you can enable and use Kernel Live Patching on a group of managed instances using AWS Systems Manager.

The following sections explain how to enable and use Kernel Live Patching on individual instances using the command line.

For more information about enabling and using Kernel Live Patching on a group of managed instances, see [Use Kernel Live Patching on Amazon Linux 2 instances](#) in the *AWS Systems Manager User Guide*.

Topics

- [Enable Kernel Live Patching](#)
- [View the available kernel live patches](#)
- [Apply kernel live patches](#)
- [View the applied kernel live patches](#)
- [Disable Kernel Live Patching](#)

Enable Kernel Live Patching

Kernel Live Patching is disabled by default on Amazon Linux 2. To use live patching, you must install the **yum** plugin for Kernel Live Patching and enable the live patching functionality.

Prerequisites

Kernel Live Patching requires `binutils`. If you do not have `binutils` installed, install it using the following command:

```
$ sudo yum install binutils
```

To enable Kernel Live Patching

1. Kernel live patches are available for the following Amazon Linux 2 kernel versions:

- Kernel version 4.14 or 5.10 on the x86_64 architecture
- Kernel version 5.10 on the ARM64 architecture

To check your kernel version, run the following command.

```
$ sudo yum list kernel
```

2. If you already have a supported kernel version, skip this step. If you do not have a supported kernel version, run the following commands to update the kernel to the latest version and to reboot the instance.

```
$ sudo yum install -y kernel
```

```
$ sudo reboot
```

3. Install the **yum** plugin for Kernel Live Patching.

```
$ sudo yum install -y yum-plugin-kernel-livepatch
```

4. Enable the **yum** plugin for Kernel Live Patching.

```
$ sudo yum kernel-livepatch enable -y
```

This command also installs the latest version of the kernel live patch RPM from the configured repositories.

5. To confirm that the **yum** plugin for kernel live patching has installed successfully, run the following command.

```
$ rpm -qa | grep kernel-livepatch
```

When you enable Kernel Live Patching, an empty kernel live patch RPM is automatically applied. If Kernel Live Patching was successfully enabled, this command returns a list that includes the initial empty kernel live patch RPM. The following is example output.

```
yum-plugin-kernel-livepatch-1.0-0.11.amzn2.noarch  
kernel-livepatch-5.10.102-99.473-1.0-0.amzn2.x86_64
```

6. Install the **kpatch** package.

```
$ sudo yum install -y kpatch-runtime
```

7. Update the **kpatch** service if it was previously installed.

```
$ sudo yum update kpatch-runtime
```

8. Start the **kpatch** service. This service loads all of the kernel live patches upon initialization or at boot.

```
$ sudo systemctl enable kpatch.service
```

9. Enable the Kernel Live Patching topic in the Amazon Linux 2 Extras Library. This topic contains the kernel live patches.

```
$ sudo amazon-linux-extras enable livepatch
```

View the available kernel live patches

Amazon Linux security alerts are published to the Amazon Linux Security Center. For more information about the Amazon Linux 2 security alerts, which include alerts for kernel live patches, see the [Amazon Linux Security Center](#). Kernel live patches are prefixed with ALASLIVEPATCH. The Amazon Linux Security Center might not list kernel live patches that address bugs.

You can also discover the available kernel live patches for advisories and CVEs using the command line.

To list all available kernel live patches for advisories

Use the following command.

```
$ yum updateinfo list
```

The following shows example output.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-  
motd  
ALAS2LIVEPATCH-2020-002 important/Sec. kernel-  
livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
```

```
ALAS2LIVEPATCH-2020-005 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

To list all available kernel live patches for CVEs

Use the following command.

```
$ yum updateinfo list cves
```

The following shows example output.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motdamzn2-core/2/x86_64 | 2.4 kB 00:00:00
CVE-2019-15918 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2019-20096 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2020-8648 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

Apply kernel live patches

You apply kernel live patches using the **yum** package manager in the same way that you would apply regular updates. The **yum** plugin for Kernel Live Patching manages the kernel live patches that are to be applied and eliminates the need to reboot.

Tip

We recommend that you update your kernel regularly using Kernel Live Patching to ensure that it remains secure and up to date.

You can choose to apply a specific kernel live patch, or to apply any available kernel live patches along with your regular security updates.

To apply a specific kernel live patch

1. Get the kernel live patch version using one of the commands described in [View the available kernel live patches](#).
2. Apply the kernel live patch for your Amazon Linux 2 kernel.

```
$ sudo yum install kernel-livepatch-kernel_version.x86_64
```

For example, the following command applies a kernel live patch for Amazon Linux 2 kernel version 5.10.102-99.473.

```
$ sudo yum install kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
```

To apply any available kernel live patches along with your regular security updates

Use the following command.

```
$ sudo yum update --security
```

Omit the `--security` option to include bug fixes.

Important

- The kernel version is not updated after applying kernel live patches. The version is only updated to the new version after the instance is rebooted.
- An Amazon Linux 2 kernel receives kernel live patches for a period of three months. After the three month period has lapsed, no new kernel live patches are released for that kernel version. To continue to receive kernel live patches after the three-month period, you must reboot the instance to move to the new kernel version, which will then continue receiving kernel live patches for the next three months. To check the support window for your kernel version, run `yum kernel-livepatch supported`.

View the applied kernel live patches

To view the applied kernel live patches

Use the following command.

```
$ kpatch list
```

The command returns a list of the loaded and installed security update kernel live patches. The following is example output.

```
Loaded patch modules:
```



```
livepatch_cifs_lease_buffer_len [enabled]
livepatch_CVE_2019_20096 [enabled]
livepatch_CVE_2020_8648 [enabled]
```

Installed patch modules:

```
livepatch_cifs_lease_buffer_len (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2019_20096 (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2020_8648 (5.10.102-99.473.amzn2.x86_64)
```

Note

A single kernel live patch can include and install multiple live patches.

Disable Kernel Live Patching

If you no longer need to use Kernel Live Patching, you can disable it at any time.

To disable Kernel Live Patching

1. Remove the RPM packages for the applied kernel live patches.

```
$ sudo yum kernel-livepatch disable
```

2. Uninstall the **yum** plugin for Kernel Live Patching.

```
$ sudo yum remove yum-plugin-kernel-livepatch
```

3. Reboot the instance.

```
$ sudo reboot
```

Limitations

Kernel Live Patching has the following limitations:

- While applying a kernel live patch, you can't perform hibernation, use advanced debugging tools (such as SystemTap, kprobes, and eBPF-based tools), or access ftrace output files used by the Kernel Live Patching infrastructure.

Frequently asked questions

For frequently asked questions about Kernel Live Patching for Amazon Linux 2, see the [Amazon Linux 2 Kernel Live Patching FAQ](#).

User provided kernels

If you need a custom kernel on your Amazon EC2 instances, you can start with an AMI that is close to what you want, compile the custom kernel on your instance, and update the bootloader to point to the new kernel. This process varies depending on the virtualization type that your AMI uses. For more information, see [Linux AMI virtualization types](#).

Contents

- [HVM AMIs \(GRUB\)](#)
- [Paravirtual AMIs \(PV-GRUB\)](#)

HVM AMIs (GRUB)

HVM instance volumes are treated like actual physical disks. The boot process is similar to that of a bare metal operating system with a partitioned disk and bootloader, which enables it to work with all currently supported Linux distributions. The most common bootloader is GRUB or GRUB2.

By default, GRUB does not send its output to the instance console because it creates an extra boot delay. For more information, see [Instance console output](#). If you are installing a custom kernel, you should consider enabling GRUB output.

You don't need to specify a fallback kernel, but we recommend that you have a fallback when you test a new kernel. GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel enables the instance to boot even if the new kernel isn't found.

The legacy GRUB for Amazon Linux uses `/boot/grub/menu.lst`. GRUB2 for Amazon Linux 2 uses `/etc/default/grub`. For more information about updating the default kernel in the bootloader, see the documentation for your Linux distribution.

Paravirtual AMIs (PV-GRUB)

Amazon Machine Images that use paravirtual (PV) virtualization use a system called *PV-GRUB* during the boot process. PV-GRUB is a paravirtual bootloader that runs a patched version of GNU

GRUB 0.97. When you start an instance, PV-GRUB starts the boot process and then chain loads the kernel specified by your image's `menu.lst` file.

PV-GRUB understands standard `grub.conf` or `menu.lst` commands, which allows it to work with all currently supported Linux distributions. Older distributions such as Ubuntu 10.04 LTS, Oracle Enterprise Linux, or CentOS 5.x require a special "ec2" or "xen" kernel package, while newer distributions include the required drivers in the default kernel package.

Most modern paravirtual AMIs use a PV-GRUB AKI by default (including all of the paravirtual Linux AMIs available in the Amazon EC2 Launch Wizard Quick Start menu), so there are no additional steps that you need to take to use a different kernel on your instance, provided that the kernel you want to use is compatible with your distribution. The best way to run a custom kernel on your instance is to start with an AMI that is close to what you want and then to compile the custom kernel on your instance and modify the `menu.lst` file to boot with that kernel.

You can verify that the kernel image for an AMI is a PV-GRUB AKI. Run the following [describe-images](#) command (substituting your kernel image ID) and check whether the Name field starts with `pv-grub`:

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

Contents

- [Limitations of PV-GRUB](#)
- [Configure GRUB for paravirtual AMIs](#)
- [Amazon PV-GRUB Kernel Image IDs](#)
- [Update PV-GRUB](#)

Limitations of PV-GRUB

PV-GRUB has the following limitations:

- You can't use the 64-bit version of PV-GRUB to start a 32-bit kernel or vice versa.
- You can't specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.
- AWS has tested and verified that PV-GRUB works with these file system formats: EXT2, EXT3, EXT4, JFS, XFS, and ReiserFS. Other file system formats might not work.
- PV-GRUB can boot kernels compressed using the gzip, bzip2, lzo, and xz compression formats.

- Cluster AMIs don't support or need PV-GRUB, because they use full hardware virtualization (HVM). While paravirtual instances use PV-GRUB to boot, HVM instance volumes are treated like actual disks, and the boot process is similar to the boot process of a bare metal operating system with a partitioned disk and bootloader.
- PV-GRUB versions 1.03 and earlier don't support GPT partitioning; they support MBR partitioning only.
- If you plan to use a logical volume manager (LVM) with Amazon Elastic Block Store (Amazon EBS) volumes, you need a separate boot partition outside of the LVM. Then you can create logical volumes with the LVM.

Configure GRUB for paravirtual AMIs

To boot PV-GRUB, a GRUB menu `.lst` file must exist in the image; the most common location for this file is `/boot/grub/menu.lst`.

The following is an example of a `menu.lst` configuration file for booting an AMI with a PV-GRUB AKI. In this example, there are two kernel entries to choose from: Amazon Linux 2018.03 (the original kernel for this AMI), and Vanilla Linux 4.16.4 (a newer version of the Vanilla Linux kernel from <https://www.kernel.org/>). The Vanilla entry was copied from the original entry for this AMI, and the `kernel` and `initrd` paths were updated to the new locations. The `default 0` parameter points the bootloader to the first entry it sees (in this case, the Vanilla entry), and the `fallback 1` parameter points the bootloader to the next entry if there is a problem booting the first.

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.lst` file, but we recommend that you have a fallback when you test a new kernel. PV-GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel allows the instance to boot even if the new kernel isn't found.

PV-GRUB checks the following locations for `menu.lst`, using the first one it finds:

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,3)/boot/grub`
- `(hd0,3)/grub`

Note that PV-GRUB 1.03 and earlier only check one of the first two locations in this list.

Amazon PV-GRUB Kernel Image IDs

PV-GRUB AKIs are available in all Amazon EC2 regions, excluding Asia Pacific (Osaka). There are AKIs for both 32-bit and 64-bit architecture types. Most modern AMIs use a PV-GRUB AKI by default.

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Use the following [describe-images](#) command to get a list of the PV-GRUB AKIs for the current region:

```
aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-*.gz
```

PV-GRUB is the only AKI available in the `ap-southeast-2` Region. You should verify that any AMI you want to copy to this Region is using a version of PV-GRUB that is available in this Region.

The following are the current AKI IDs for each Region. Register new AMIs using an `hd0` AKI.

Note

We continue to provide hd00 AKIs for backward compatibility in Regions where they were previously available.

ap-northeast-1, Asia Pacific (Tokyo)

Image ID	Image Name
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1, Asia Pacific (Singapore) Region

Image ID	Image Name
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2, Asia Pacific (Sydney)

Image ID	Image Name
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1, Europe (Frankfurt)

Image ID	Image Name
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1, Europe (Ireland)

Image ID	Image Name
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1, South America (São Paulo)

Image ID	Image Name
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcfd	pv-grub-hd0_1.05-x86_64.gz

us-east-1, US East (N. Virginia)

Image ID	Image Name
aki-04206613	pv-grub-hd0_1.05-i386.gz
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

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

Image ID	Image Name
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1, US West (N. California)

Image ID	Image Name
aki-43cf8123	pv-grub-hd0_1.05-i386.gz

Image ID	Image Name
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2, US West (Oregon)

Image ID	Image Name
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

Update PV-GRUB

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Also, older versions of PV-GRUB are not available in all regions, so if you copy an AMI that uses an older version to a Region that does not support that version, you will be unable to boot instances launched from that AMI until you update the kernel image. Use the following procedures to check your instance's version of PV-GRUB and update it if necessary.

To check your PV-GRUB version

1. Find the kernel ID for your instance.

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute kernel --region region

{
  "InstanceId": "instance_id",
  "KernelId": "aki-70cb0e10"
}
```

The kernel ID for this instance is `aki-70cb0e10`.

2. View the version information of that kernel ID.

```
aws ec2 describe-images --image-ids aki-70cb0e10 --region region
```



```
{
  "Images": [
    {
      "VirtualizationType": "paravirtual",
      "Name": "pv-grub-hd0_1.05-x86_64.gz",
      ...
      "Description": "PV-GRUB release 1.05, 64-bit"
    }
  ]
}
```

This kernel image is PV-GRUB 1.05. If your PV-GRUB version is not the newest version (as shown in [Amazon PV-GRUB Kernel Image IDs](#)), you should update it using the following procedure.

To update your PV-GRUB version

If your instance is using an older version of PV-GRUB, you should update it to the latest version.

1. Identify the latest PV-GRUB AKI for your Region and processor architecture from [Amazon PV-GRUB Kernel Image IDs](#).
2. Stop your instance. Your instance must be stopped to modify the kernel image used.

```
aws ec2 stop-instances --instance-ids instance_id --region region
```

3. Modify the kernel image used for your instance.

```
aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --region region
```

4. Restart your instance.

```
aws ec2 start-instances --instance-ids instance_id --region region
```

Configure the Amazon Linux 2 MATE desktop connection

The [MATE desktop environment](#) is pre-installed and pre-configured in AMIs with the following description:

".NET Core *x.x*, Mono *x.xx*, PowerShell *x.x*, and MATE DE pre-installed to run your .NET applications on Amazon Linux 2 with Long Term Support (LTS)."

The environment provides an intuitive graphical user interface for administering Amazon Linux 2 instances with minimal use of the command line. The interface uses graphical representations, such as icons, windows, toolbars, folders, wallpapers, and desktop widgets. Built-in, GUI-based tools are available to perform common tasks. For example, there are tools for adding and removing software, applying updates, organizing files, launching programs, and monitoring system health.

Important

xrdp is the remote desktop software bundled in the AMI. By default, xrdp uses a self-signed TLS certificate to encrypt remote desktop sessions. Neither AWS nor the xrdp maintainers recommend using self-signed certificates in production. Instead, obtain a certificate from an appropriate certificate authority (CA) and install it on your instances. For more information about TLS configuration, see [TLS security layer](#) on the xrdp wiki.

Note

If you prefer to use a virtual network computing (VNC) service instead of xrdp, see the [How do I install a GUI on my Amazon EC2 instance running Amazon Linux 2](#) AWS Knowledge Center article.

Prerequisite

To run the commands shown in this topic, you must install the AWS Command Line Interface (AWS CLI) or AWS Tools for Windows PowerShell, and configure your AWS profile.

Options

1. Install the AWS CLI – For more information, see [Installing the AWS CLI](#) and [Configuration basics](#) in the *AWS Command Line Interface User Guide*.
2. Install the Tools for Windows PowerShell – For more information, see [Installing the AWS Tools for Windows PowerShell](#) and [Shared credentials](#) in the *AWS Tools for Windows PowerShell User Guide*.

Configure the RDP connection

Follow these steps to set up a Remote Desktop Protocol (RDP) connection from your local machine to an Amazon Linux 2 instance running the MATE desktop environment.

1. To get the ID of the AMI for Amazon Linux 2 that includes MATE in the AMI name, you can use the [describe-images](#) command from your local command line tool. If you have not installed the command line tools, you can perform the following query directly from an AWS CloudShell session. For information about how to launch a shell session from CloudShell, see [Getting started with AWS CloudShell](#). From the Amazon EC2 console, you can find the MATE-included AMI by launching an instance, and then entering MATE in the AMI search bar. The Amazon Linux 2 Quick Start with MATE pre-installed will appear in the search results.

```
aws ec2 describe-images --filters "Name=name,Values=amzn2*MATE*" --query
  "Images[*].[ImageId,Name,Description]"
[
  [
    "ami-0123example0abc12",
    "amzn2-x86_64-MATEDE_DOTNET-2020.12.04",
    ".NET Core 5.0, Mono 6.12, PowerShell 7.1, and MATE DE pre-installed to run
your .NET applications on Amazon Linux 2 with Long Term Support (LTS).",
  ],
  [
    "ami-0456example0def34",
    "amzn2-x86_64-MATEDE_DOTNET-2020.04.14",
    "Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop
Environment"
  ]
]
```

Choose the AMI that is appropriate for your use.

2. Launch an EC2 instance with the AMI that you located in the previous step. Configure the security group to allow for inbound TCP traffic to port 3389. For more information about configuring security groups, see [Security groups for your VPC](#). This configuration enables you to use an RDP client to connect to the instance.
3. Connect to the instance using [SSH](#).
4. Update the software and kernel on the instance.

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

After the update completes, reboot the instance to ensure that it is using the latest packages and libraries from the update; kernel updates are not loaded until a reboot occurs.

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

5. Reconnect to the instance and run the following command on your Linux instance to set the password for `ec2-user`.

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

6. Install the certificate and key.

If you already have a certificate and key, copy them to the `/etc/xrdp/` directory as follows:

- Certificate — `/etc/xrdp/cert.pem`
- Key — `/etc/xrdp/key.pem`

If you do not have a certificate and key, use the following command to generate them in the `/etc/xrdp` directory.

```
$ sudo openssl req -x509 -sha384 -newkey rsa:3072 -nodes -keyout /etc/xrdp/key.pem  
-out /etc/xrdp/cert.pem -days 365
```

Note

This command generates a certificate that is valid for 365 days.

7. Open an RDP client on the computer from which you will connect to the instance (for example, Remote Desktop Connection on a computer running Microsoft Windows). Enter `ec2-user` as the user name and enter the password that you set in the previous step.

To disable `xrdp` on your Amazon EC2 instance

You can disable `xrdp` at any time by running one of the following commands on your Linux instance. The following commands do not impact your ability to use MATE using an X11 server.

```
[ec2-user ~]$ sudo systemctl disable xrdp
```

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

To enable xrdp on your Amazon EC2 instance

To re-enable xrdp so that you can connect to your Amazon Linux 2 instance running the MATE desktop environment, run one of the following commands on your Linux instance.

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

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

AMI quotas

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,	1,000

Quota name	Description	Default quota per Region
	the number of accounts in the organization or OU does not count towards the quota.	

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/home>.
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

Note

For detailed instance type specifications, see [Specifications](#) in the *Amazon EC2 Instance Types Guide*. For pricing information, see [Amazon EC2 On-Demand Pricing](#).

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

- [What is Amazon EC2?](#)
- [Set up to use Amazon EC2](#)
- [Tutorial: Get started with Amazon EC2 Linux instances](#)
- [Instance lifecycle](#)

Before you launch a production environment, you need to answer the following questions.

Q. What instance type best meets my needs?

Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications. For more information, see [Amazon EC2 instance types](#).

Q. What purchasing option best meets my needs?

Amazon EC2 supports On-Demand Instances (the default), Spot Instances, and Reserved Instances. For more information, see [Instance purchasing options](#).

Q. Which type of root volume meets my needs?

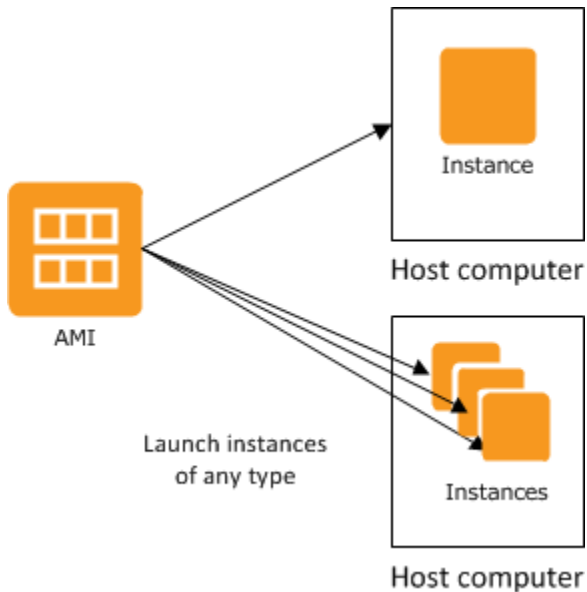
Each instance is backed by Amazon EBS or backed by instance store. Select an AMI based on which type of root volume you need. For more information, see [Storage for the root device](#).

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

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

Instances and AMIs

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



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

Instances

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

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

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

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

Storage for your instance

The root device for your instance contains the image used to boot the instance. The root device is either an Amazon Elastic Block Store (Amazon EBS) volume or an instance store volume. For more information, see [Amazon EC2 instance root volume](#).

Your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping. For more information, see [Block device mappings](#). After these volumes have been added to and mapped on your instance, they are available for you to mount and use. If your instance fails, or if your instance is stopped or terminated, the data on these volumes is lost; therefore, these volumes are best used for temporary data. To keep important data safe, you should use a replication strategy across multiple instances, or store your persistent data in Amazon S3 or Amazon EBS volumes. For more information, see [Storage options for your Amazon EC2 instances](#).

Security best practices

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

Stop and terminate instances

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

Stop an instance

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

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

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

Terminate an instance

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

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

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

Note

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*. For more information about Amazon EBS costs, see [Amazon EBS pricing](#).

AMIs

Amazon Web Services (AWS) publishes many [Amazon Machine Images \(AMIs\)](#) that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMI or AMIs; doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or a web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.

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

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

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

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.

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](#)
- [Hardware specifications](#)
- [AMI virtualization types](#)
- [Find an Amazon EC2 instance type](#)
- [Get recommendations for an instance type](#)
- [Change the instance type](#)
- [Burstable performance instances](#)

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. For more information, see [Instance types](#) in the *Amazon EC2 Instance Types Guide*.

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 the [Amazon EC2 Instance Types Guide](#).

For more information about features and use cases, see [Amazon EC2 Instance Types Details](#).

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 [Change the instance type](#).

Intel processor features

Amazon EC2 instances that run on Intel processors may include the following features. Not all of the following processor features are supported by all instance types. For detailed information about which features are available for each instance type, see [Amazon EC2 Instance types](#).

- **Intel AES New Instructions (AES-NI)** — Intel AES-NI encryption instruction set improves upon the original Advanced Encryption Standard (AES) algorithm to provide faster data protection and greater security. All current generation EC2 instances support this processor feature.
- **Intel Advanced Vector Extensions (Intel AVX, Intel AVX2, and Intel AVX-512)** — Intel AVX and Intel AVX2 are 256-bit, and Intel AVX-512 is a 512-bit instruction set extension designed for applications that are Floating Point (FP) intensive. Intel AVX instructions improve performance for applications like image and audio/video processing, scientific simulations, financial analytics, and 3D modeling and analysis. These features are only available on instances launched with HVM AMIs.
- **Intel Turbo Boost Technology** — Intel Turbo Boost Technology processors automatically run cores faster than the base operating frequency.
- **Intel Deep Learning Boost (Intel DL Boost)** — Accelerates AI deep learning use cases. The 2nd Gen Intel Xeon Scalable processors extend Intel AVX-512 with a new Vector Neural Network Instruction (VNNI/INT8) that significantly increases deep learning inference performance

over previous generation Intel Xeon Scalable processors (with FP32) for image recognition/segmentation, object detection, speech recognition, language translation, recommendation systems, reinforcement learning, and more. VNNI may not be compatible with all Linux distributions.

The following instances support VNNI: M5n, R5n, M5dn, M5zn, R5b, R5dn, D3, D3en, and C6i. C5 and C5d instances support VNNI for only 12xlarge, 24xlarge, and metal instances.

Confusion may result from industry naming conventions for 64-bit CPUs. Chip manufacturer Advanced Micro Devices (AMD) introduced the first commercially successful 64-bit architecture based on the Intel x86 instruction set. Consequently, the architecture is widely referred to as AMD64 regardless of the chip manufacturer. Windows and several Linux distributions follow this practice. This explains why the internal system information on an instance running Ubuntu or Windows displays the CPU architecture as AMD64 even though the instances are running on Intel hardware.

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, a fully-managed service that is the easiest way to get started with machine learning models. For more information, see [Get Started with SageMaker](#) in the *Amazon SageMaker 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 AMI 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.

AMI virtualization types

The virtualization type of your instance is determined by the AMI that you use to launch it. Current generation instance types support hardware virtual machine (HVM) only. Some previous generation instance types support paravirtual (PV) and some AWS Regions support PV instances. For more information, see [Linux AMI virtualization types](#).

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.

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.

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 “For existing workloads”](#).

Tasks

- [Find an instance type using the console](#)
- [Find an instance type using the AWS CLI](#)

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 instance using the new launch instance wizard](#).

Find an instance type using the AWS CLI

You can use AWS CLI commands for Amazon EC2 to find an instance type that meet your needs.

To find an instance type using the AWS CLI

1. If you have not done so already, install the AWS CLI For more information, see the [AWS Command Line Interface User Guide](#).
2. Use the [describe-instance-types](#) command to filter instance types based on instance attributes. For example, you can use the following command to display only 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" --query "InstanceTypes[*].
[InstanceType]" --output text | sort
```

3. Use the [describe-instance-type-offerings](#) command to filter instance types offered by location (Region or Zone). For example, you can use the following command to display the instance types offered in the specified 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
```

4. After locating the instance types that meet your needs, save the list so that you can use these instance types when you launch instances. For more information, see [Launching your instance](#) in the *AWS Command Line Interface User Guide*.

Get recommendations for an instance type

The following tools can help you select the optimal instance types for your new or existing workloads:

- **New workloads** – 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 to your new workloads.

- **Existing workloads** – AWS Compute Optimizer analyzes your existing instance specifications and utilization metrics. It then uses the compiled data to recommend which Amazon EC2 instance types are optimized for cost or performance, or both, for your existing workloads.

Get instance type recommendations:

- [Get instance type recommendations for a new workload](#)
- [Get instance type recommendations for an existing workload](#)

Get instance type recommendations for a new workload

Note

Powered by Amazon Bedrock: AWS implements [automated abuse detection](#). Because Amazon Q EC2 instance type selector is built on Amazon Bedrock, users can take full advantage of the controls implemented in Amazon Bedrock to enforce safety, security, and the responsible use of artificial intelligence (AI).

Amazon Q EC2 instance type selector is in preview release for Amazon EC2 and is subject to change.

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 to your new workloads.

With so many instance types available, it can be time-consuming and difficult finding the right instance types for your workload. By using the Amazon Q EC2 instance type selector, you can remain up to date with the latest instance types and achieve the best price-performance for your workloads.

This topic outlines how to get guidance and suggestions for EC2 instance types through 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 instance type recommendations for an existing workload](#).

Supported AWS Regions

Because Amazon Q EC2 instance type selector uses Amazon Q, it is supported in the same Regions in which Amazon Q is supported. For more information, see [Supported Regions for Amazon Q Developer](#) in the *Amazon Q Developer User Guide*.

Use Amazon Q EC2 instance type selector

In the Amazon EC2 console, you can choose the **Get advice** link to ask Amazon Q for instance type advice. Once you've specified your use case, workload type, CPU manufacturer preference, and price and performance priorities, Amazon Q opens to provide you with instance type suggestions. You can also go directly to Amazon Q to ask for instance type advice. For more information, see the [Amazon Q Developer User Guide](#).

The **Get advice** link appears on the following EC2 console pages:

- Amazon EC2 launch instance wizard
- Amazon EC2 launch template

Use the following instructions to get guidance and suggestions for EC2 instance types using the Amazon Q EC2 instance type selector in the Amazon EC2 console.

To get instance type advice using the Amazon Q EC2 instance type selector

1. Follow the procedure to [launch an instance](#) or [create a launch template](#).
2. To use the Amazon Q EC2 instance type selector to get instance type advice, do the following:
 - a. Next to **Instance type**, choose the **Get advice** link.
 - b. In the **Get advice on instance type selection from Amazon Q** window, specify your instance type requirements by selecting options from the drop-down lists. For **Use Case** and **Workload type**, you can select **Other**, and then enter your requirements.

Get advice on instance type selection from Amazon Q ✕

Tell us more about your requirements to generate instance type suggestions

We will use Amazon Q, a generative AI assistant, to generate instance type suggestions

Use Case	Workload type
<input type="text" value="Web Hosting"/>	<input type="text" value="Web/App Server"/>
Priority	CPU Manufacturers
<input type="text" value="Low cost"/>	<input type="text" value="No preference"/>

- c. Choose **Get instance type advice**.

Amazon Q opens with suggestions for instance types customized for your requirements.

- d. You can continue to chat in natural language with Amazon Q about more instance type requirements.
3. When you've decided on an instance type to use, in the launch instance wizard or launch template, for **Instance type**, select the instance type.
4. Complete the procedure to launch an instance or create a launch template.

Get instance type recommendations for an existing workload

AWS Compute Optimizer provides Amazon EC2 instance recommendations to help you improve performance, save money, or both. You can use these recommendations to decide whether to move to a new instance type.

To make recommendations, Compute Optimizer analyzes your existing instance specifications and utilization metrics. The compiled data is then used to recommend which Amazon EC2 instance types are best able to handle the existing workload. Recommendations are returned along with per-hour instance pricing.

This topic outlines how to view recommendations through the Amazon EC2 console. For more information, see the [AWS Compute Optimizer User Guide](#).

Note

To get recommendations from Compute Optimizer, you must first opt in to Compute Optimizer. For more information, see [Getting Started with AWS Compute Optimizer](#) in the *AWS Compute Optimizer User Guide*.

If you're looking for instance type recommendations for a *new* workload, use Amazon Q EC2 instance type selector. For more information, see [Get instance type recommendations for a new workload](#).

Contents

- [Limitations](#)
- [Findings](#)
- [View recommendations](#)
- [Considerations for evaluating recommendations](#)
- [Additional resources](#)

Limitations

Compute Optimizer currently generates recommendations for C, D, H, I, M, R, T, X, and z instance types. Other instance types are not considered by Compute Optimizer. If you're using other instance types, they will not be listed in the Compute Optimizer recommendations view. For more information about the supported and unsupported instance types, see [Amazon EC2 instance requirements](#) in the *AWS Compute Optimizer User Guide*.

Findings

Compute Optimizer classifies its findings for EC2 instances as follows:

- **Under-provisioned** – An EC2 instance is considered under-provisioned when at least one specification of your instance, such as CPU, memory, or network, does not meet the performance requirements of your workload. Under-provisioned EC2 instances might lead to poor application performance.

- **Over-provisioned** – An EC2 instance is considered over-provisioned when at least one specification of your instance, such as CPU, memory, or network, can be sized down while still meeting the performance requirements of your workload, and when no specification is under-provisioned. Over-provisioned EC2 instances might lead to unnecessary infrastructure cost.
- **Optimized** – An EC2 instance is considered optimized when all specifications of your instance, such as CPU, memory, and network, meet the performance requirements of your workload, and the instance is not over-provisioned. An optimized EC2 instance runs your workloads with optimal performance and infrastructure cost. For optimized instances, Compute Optimizer might sometimes recommend a new generation instance type.
- **None** – There are no recommendations for this instance. This might occur if you've been opted in to Compute Optimizer for less than 12 hours, or when the instance has been running for less than 30 hours, or when the instance type is not supported by Compute Optimizer. For more information, see [Limitations](#) in the previous section.

View recommendations

After you opt in to Compute Optimizer, you can view the findings that Compute Optimizer generates for your EC2 instances in the EC2 console. You can then access the Compute Optimizer console to view the recommendations. If you recently opted in, findings might not be reflected in the EC2 console for up to 12 hours.

To view a recommendation for an EC2 instance through the EC2 console


1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then choose the instance ID .
3. On the instance summary page, in the **AWS Compute Optimizer** banner near the bottom of the page, choose **View detail**.

The instance opens in Compute Optimizer, where it is labeled as the **Current** instance. Up to three different instance type recommendations, labeled **Option 1**, **Option 2**, and **Option 3**, are provided. The bottom half of the window shows recent CloudWatch metric data for the current instance: **CPU utilization**, **Memory utilization**, **Network in**, and **Network out**.

4. (Optional) In the Compute Optimizer console, choose settings



to change the visible columns in the table, or to view the public pricing information for a different purchasing option for the current and recommended instance types.

 **Note**


If you've purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.

Determine whether you want to use one of the recommendations. Decide whether to optimize for performance improvement, for cost reduction, or for a combination of the two. For more information, see [Viewing Resource Recommendations](#) in the *AWS Compute Optimizer User Guide*.

To view recommendations for all EC2 instances across all Regions through the Compute Optimizer console

1. Open the Compute Optimizer console at <https://console.aws.amazon.com/compute-optimizer/>.
2. Choose **View recommendations for all EC2 instances**.
3. You can perform the following actions on the recommendations page:
 - a. To filter recommendations to one or more AWS Regions, enter the name of the Region in the **Filter by one or more Regions** text box, or choose one or more Regions in the drop-down list that appears.
 - b. To view recommendations for resources in another account, choose **Account**, and then select a different account ID.

This option is available only if you are signed in to a management account of an organization, and you opted in all member accounts within the organization.

- c. To clear the selected filters, choose **Clear filters**.
- d. To change the purchasing option that is displayed for the current and recommended instance types, choose settings , and then choose **On-Demand Instances, Reserved Instances, standard 1-year no upfront**, or **Reserved Instances, standard 3-year no upfront**.
- e. To view details, such as additional recommendations and a comparison of utilization metrics, choose the finding (**Under-provisioned**, **Over-provisioned**, or **Optimized**) listed

next to the desired instance. For more information, see [Viewing Resource Details](#) in the *AWS Compute Optimizer User Guide*.

Considerations for evaluating recommendations

Before changing an instance type, consider the following:

- The recommendations don't forecast your usage. Recommendations are based on your historical usage over the most recent 14-day time period. Be sure to choose an instance type that is expected to meet your future resource needs.
- Focus on the graphed metrics to determine whether actual usage is lower than instance capacity. You can also view metric data (average, peak, percentile) in CloudWatch to further evaluate your EC2 instance recommendations. For example, notice how CPU percentage metrics change during the day and whether there are peaks that need to be accommodated. For more information, see [Viewing Available Metrics](#) in the *Amazon CloudWatch User Guide*.
- Compute Optimizer might supply recommendations for burstable performance instances, which are T3, T3a, and T2 instances. If you periodically burst above the baseline, make sure that you can continue to do so based on the vCPUs of the new instance type. For more information, see [Key concepts and definitions for burstable performance instances](#).
- 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.

There are other considerations when resizing an EC2 instance. For more information, see [Change the instance type](#).

Additional resources

For more information:

- [Amazon EC2 instance types](#)
- [AWS Compute Optimizer User Guide](#)

Change the instance type

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can resize your instance by changing its instance type. For example, if your `t2.micro` instance is too small for its workload, you can 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 instance type recommendations for an existing workload](#).

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 of an existing EBS-backed instance
EBS	Not compatible	Change the instance type by launching a new instance
Instance store	Not applicable	Change the instance type of an instance store-backed instance

Considerations for compatible instance types

Consider the following when changing the instance type of an existing instance:

- You must stop your Amazon EBS-backed 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, we release the address and give your instance a new public IPv4 address. If you require a public IPv4 address that does not change, use an [Elastic IP address](#).
- You can't change the instance type of a [Spot Instance](#).
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the scaling processes for the group while you're 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 [Instance volume limits](#).

Change the instance type of an existing EBS-backed instance

Use the following instructions to change the instance type of an EBS-backed instance if the instance type that you need is compatible with the instance's current configuration.

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. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Instances**.
4. Select the instance and choose **Instance state, Stop instance**. 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, Change instance type**. This option is grayed out if the instance state is not stopped.
6. 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: [Change the instance type by launching a new 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. Choose **Apply** to accept the new settings.
7. 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](#).

Change the instance type by launching a new instance

If the current configuration of your EBS-backed instance is incompatible with the new instance type that you want, then you can't change the instance type of the original instance. Instead, you must launch a new instance with a configuration that is compatible with the new instance type that you want, and then migrate your application to the new instance. For example, if you launched your original instance from a PV AMI, but want to change to a current generation instance type that is only supported by an HVM AMI, you'll need to launch a new instance from an HVM AMI. For information about how compatibility is determined, see [Compatibility for changing the instance type](#).

To migrate your application to a new instance, do the following:

- Back up the data on your original instance.
- Launch a new instance with a configuration that is compatible with the new instance type that you want, and attach any EBS volumes that were attached to your original instance.
- Install your application and any software on your new instance.
- Restore any data.
- If your original instance has an Elastic IP address, and you want to ensure that your users can continue uninterrupted to use the applications on your new instance, you must associate the Elastic IP address with your new instance. For more information, see [Elastic IP address](#).

To change the instance type for a new instance configuration

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Back up data that you need to keep, as follows:
 - For data on your instance store volumes, back up the data to persistent storage.
 - For data on your EBS volumes, create a snapshot of the volumes or detach the volumes from the instance so that you can attach them to the new instance later.
3. In the navigation pane, choose **Instances**.
4. Choose **Launch instances**. When you configure the instance, do the following:
 - a. Select an AMI that will support the instance type that you want. Note that 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're using an Elastic IP address, select the VPC that the original instance is currently running in.
 - d. If you want to allow the same traffic to reach the new instance, select the security group that is associated with the original instance.
 - e. 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 required, attach any new EBS volumes based on the snapshots that you created, or any EBS volumes that you detached from the original instance, 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 you are using 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 with which to associate the Elastic IP address.
 - 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**.

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.

For compatibility information for changing Windows instance types, see [Compatibility for changing the instance type](#) in the *User Guide for Windows Instances*.

Tip

You can use the `AWSSupport-MigrateXenToNitroLinux` runbook to migrate compatible 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.

Compatibility is determined in the following ways:

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 [Linux AMI 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 cards

Some instance types support multiple [network cards](#). You must select an instance type that supports the same number of network cards as the current instance type.

Enhanced networking

Instance types that support [enhanced networking](#) require the necessary drivers installed. For example, [instances built on the AWS Nitro System](#) 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 [instances built on the AWS Nitro System](#). 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`). Therefore, to mount file systems at boot time using `/etc/fstab`, you must use UUID/Label instead of device names.

Volumes limits

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 [Instance volume limits](#).

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.

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](#).

Change the instance type of an instance store-backed instance

An instance store-backed instance is an instance that has an instance store root volume. You can't change the instance type of an instance that has an instance store root volume. Instead, you must create an AMI from your instance, launch a new instance from this AMI and select the instance type that you want, and then migrate your application to the new instance. Note that the instance type that you want must be compatible with the AMI you create. For information about how compatibility is determined, see [Compatibility for changing the instance type](#).

To migrate your application to a new instance, do the following:

- Back up the data on your original instance.
- Create an AMI from your original instance.
- Launch a new instance from this AMI and select the instance type that you want.
- Install your application on the new instance.
- If your original instance has an Elastic IP address, and you want to ensure that your users can continue uninterrupted to use the applications on your new instance, you must associate the Elastic IP address with your new instance. For more information, see [Elastic IP address](#).

To change the instance type of an instance store-backed instance

1. Back up data that you need to keep, as follows:
 - For data on your instance store volumes, back up the data to persistent storage.
 - For data on your EBS volumes, create a snapshot of the volumes or detach the volume from the instance so that you can attach it to the new instance later.
2. Create an AMI from your instance by satisfying the prerequisites and following the procedures in [Create an instance store-backed Linux AMI](#). When you are finished creating an AMI from your instance, return to this procedure.
3. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
4. In the navigation pane, choose **AMIs**. From the filter lists, choose **Owned by me**, and select the image that you created in Step 2. Notice that **AMI name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

Note

If you do not see the AMI that you created in Step 2, make sure that you have selected the Region in which you created your AMI.

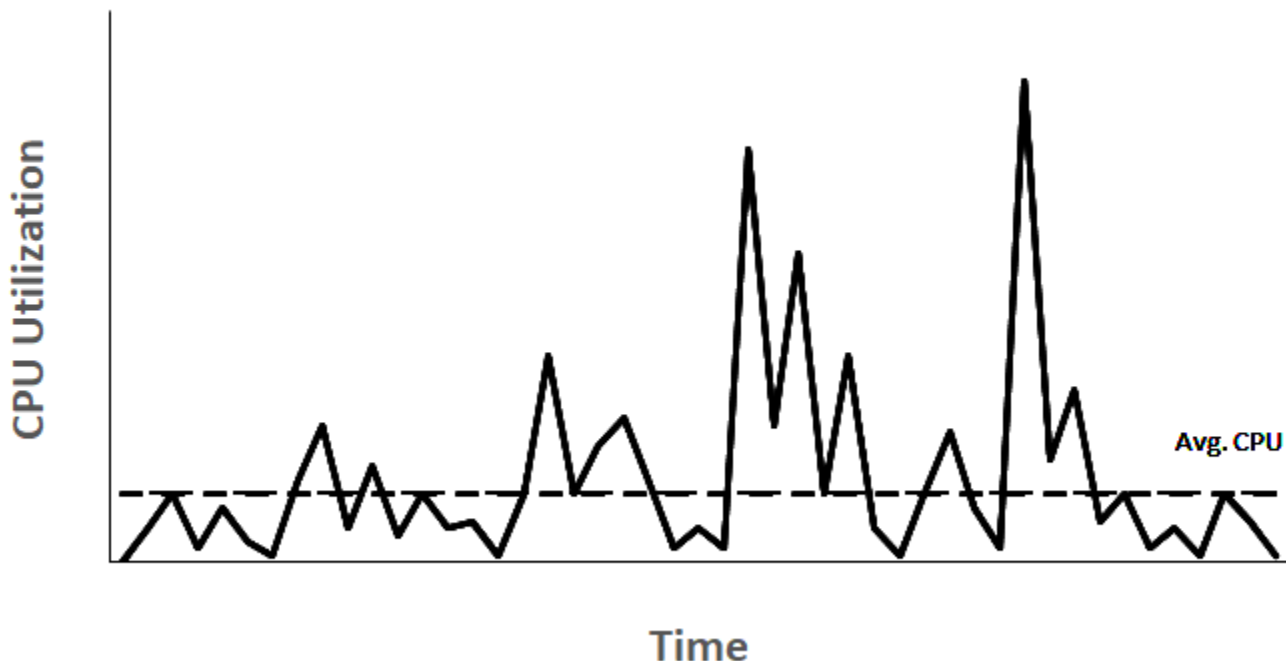
5. With the AMI selected, choose **Launch instance from image**. When you configure the instance, do the following:
 - a. Select the new instance type that you want. If the instance type that you want isn't available, then it's not compatible with the configuration of the AMI that you created. For more information, see [Compatibility for changing the instance type](#).
 - b. If you're using an Elastic IP address, select the VPC that the original instance is currently running in.
 - c. If you want to allow the same traffic to reach the new instance, select the security group that is associated 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.
6. If required, attach any new EBS volumes based on the snapshots that you created, or any EBS volumes that you detached from the original instance, to the new instance.
7. Install your application and any required software on the new instance.
8. If you are using 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 with which to associate the Elastic IP address.
 - f. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
 - g. Choose **Associate**.

- (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**.

Burstable performance instances

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

Many common workloads look like this



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

Type	Description	Processor family
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		
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 account is less than 12 months old, you can use a `t2.micro` instance for free (or a `t3.micro` instance in Regions where `t2.micro` is unavailable) within certain usage limits. For more information, see [AWS Free Tier](#).

Supported purchasing options for T instances

- On-Demand Instances
- Reserved Instances
- Dedicated Instances (T3 only)
- Dedicated Hosts (T3 only, in standard mode only)
- Spot Instances

For more information, see [Instance purchasing options](#).

Contents

- [Best practices](#)
- [Key concepts and definitions for burstable performance instances](#)
- [Unlimited mode for burstable performance instances](#)
- [Standard mode for burstable performance instances](#)

- [Work with burstable performance instances](#)
- [Monitor your CPU credits for burstable performance instances](#)

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 and definitions for burstable performance instances

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

Each burstable performance instance continuously earns 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](#)
- [Earn CPU credits](#)
- [CPU credit earn rate](#)
- [CPU credit accrual limit](#)
- [Accrued CPU credits life span](#)
- [Baseline utilization](#)

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: $\text{Baseline utilization \%} = (\text{number of credits earned} / \text{number of vCPUs}) / 60 \text{ minutes}$.

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) = $6 - 2 \text{ vCPUs} * 2\% \text{ CPU utilization} * 60 \text{ minutes} = 6 - 2.4 = 3.6$ 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	Standard (default), Unlimited	0 days (credits are lost when an instance stops)

Type	Type of CPU credits supported	Credit configuration modes	Accrued CPU credits lifespan between instance starts and stops
	credits (Unlimited mode only)		

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%

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs***	Baseline utilization per vCPU
t2.micro	6	144	1	10%
t2.small	12	288	1	20%
t2.medium	24	576	2	20%**
t2.large	36	864	2	30%**
t2.xlarge	54	1296	4	22.5%**
t2.2xlarge	81.6	1958.4	8	17%**
T3				
t3.nano	6	144	2	5%**
t3.micro	12	288	2	10%**
t3.small	24	576	2	20%**
t3.medium	24	576	2	20%**
t3.large	36	864	2	30%**
t3.xlarge	96	2304	4	40%**
t3.2xlarge	192	4608	8	40%**
T3a				
t3a.nano	6	144	2	5%**
t3a.micro	12	288	2	10%**
t3a.small	24	576	2	20%**
t3a.medium	24	576	2	20%**

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs***	Baseline utilization per vCPU
t3a.large	36	864	2	30%**
t3a.xlarge	96	2304	4	40%**
t3a.2xlarge	192	4608	8	40%**
T4g				
t4g.nano	6	144	2	5%**
t4g.micro	12	288	2	10%**
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](#).

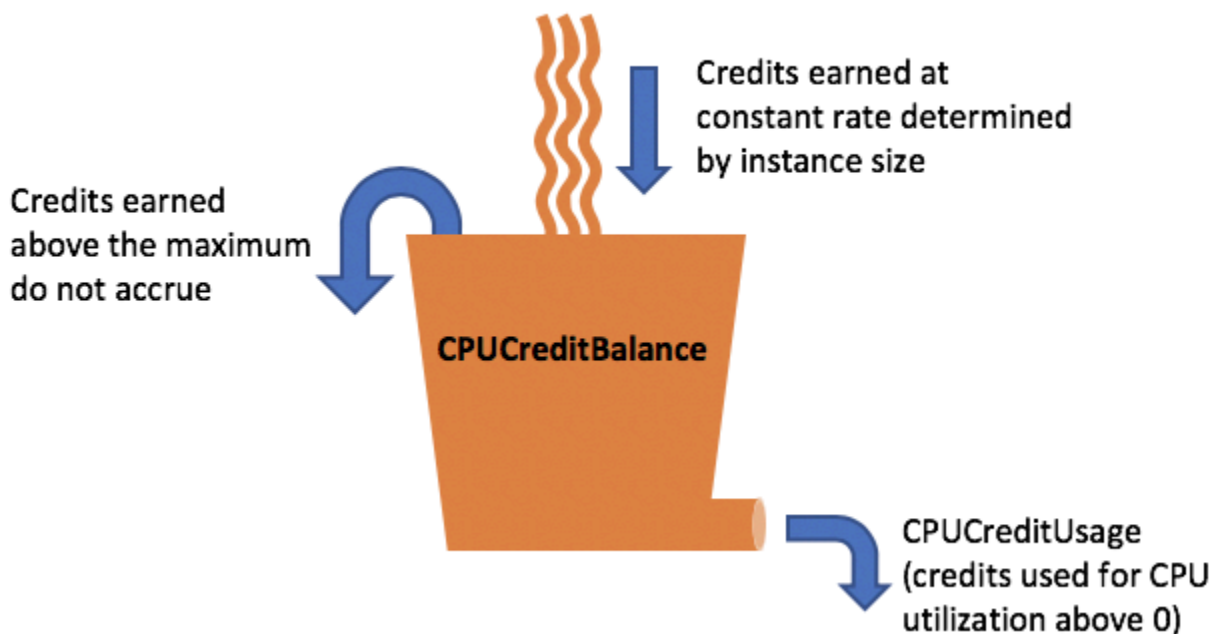
*** Each vCPU is a thread of either an Intel Xeon core or an AMD EPYC core, except for T2 and T4g instances.

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:

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

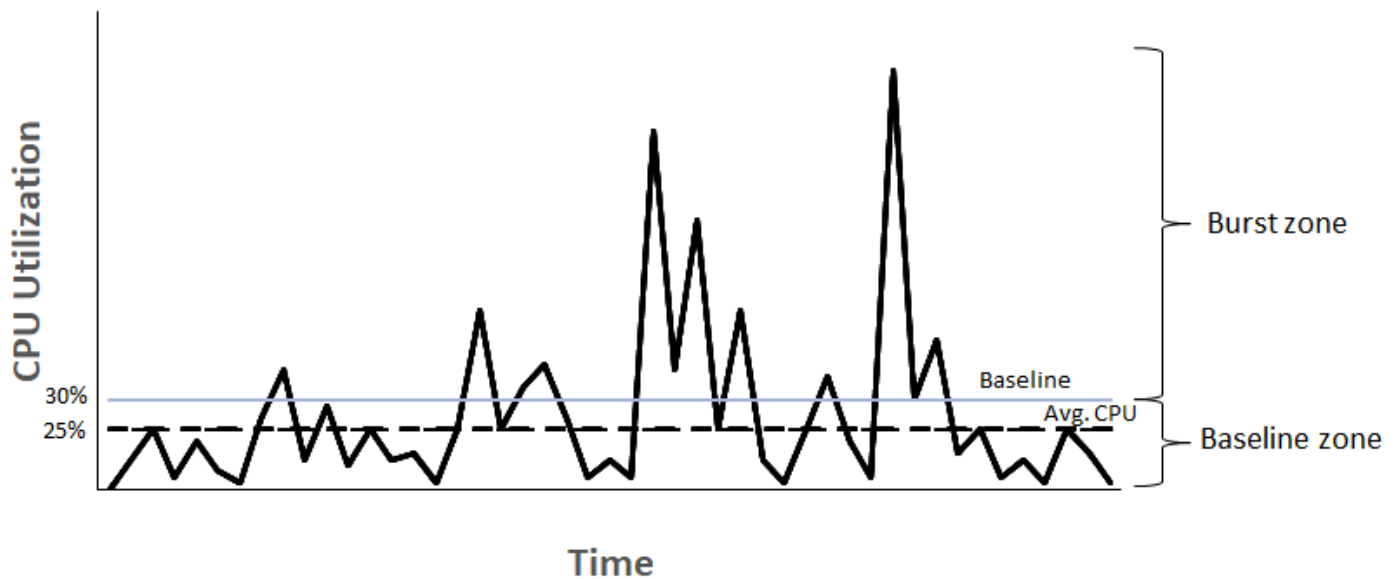
For example, a `t3.nano` instance, with 2 vCPUs, earns 6 credits per hour, resulting in a baseline utilization of 5%, which is calculated as follows:

$$(6 \text{ credits earned} / 2 \text{ vCPUs}) / 60 \text{ minutes} = 5\% \text{ baseline utilization}$$

A `t3.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.

Example of t3.large



Unlimited mode for burstable performance instances

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

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

If you use a `t2.micro` or `t3.micro` instance under the [AWS Free Tier](#) offer and use it in `unlimited` mode, charges might apply if your average utilization over a rolling 24-hour period exceeds the [baseline utilization](#) 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 [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](#)
 - [How Unlimited burstable performance instances work](#)
 - [When to use unlimited mode versus fixed CPU](#)
 - [Surplus credits can incur charges](#)
 - [No launch credits for T2 Unlimited instances](#)
 - [Enable unlimited mode](#)
 - [What happens to credits when switching between Unlimited and Standard](#)
 - [Monitor credit usage](#)
- [Unlimited mode examples](#)
 - [Example 1: Explain credit use with T3 Unlimited](#)
 - [Example 2: Explain credit use with T2 Unlimited](#)

Unlimited mode concepts

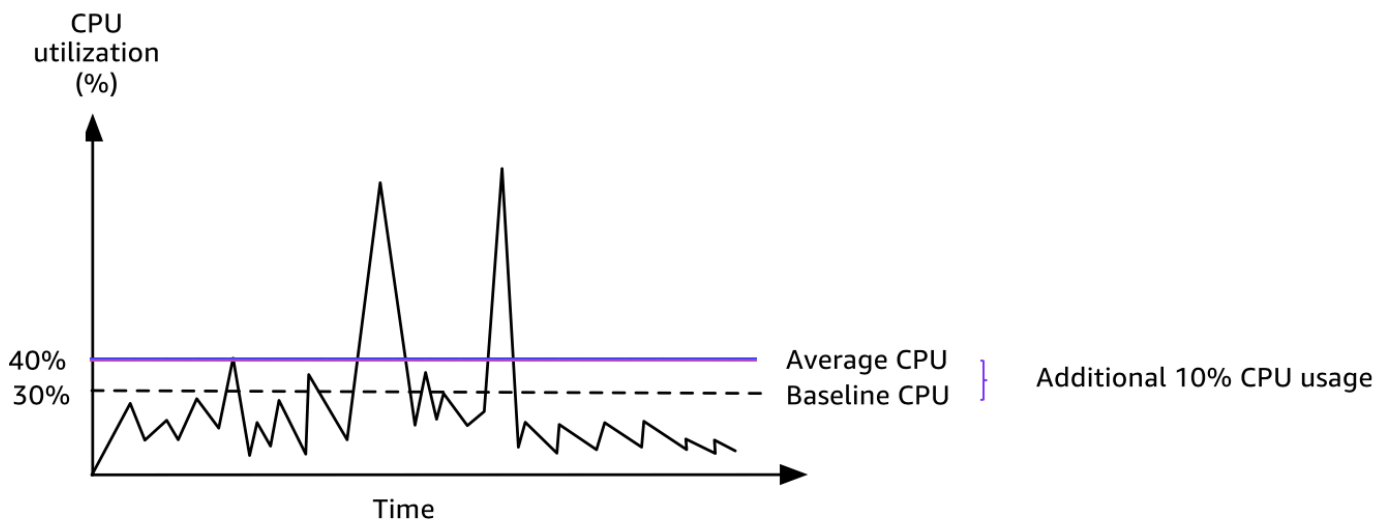
The unlimited mode is a credit configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance. You can [set unlimited as the default credit option](#) at the account level per AWS Region, per burstable performance instance family,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

How Unlimited burstable performance instances work

If a burstable performance instance configured as unlimited depletes its CPU credit balance, it can spend *surplus* credits to burst beyond the [baseline](#). 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](#).

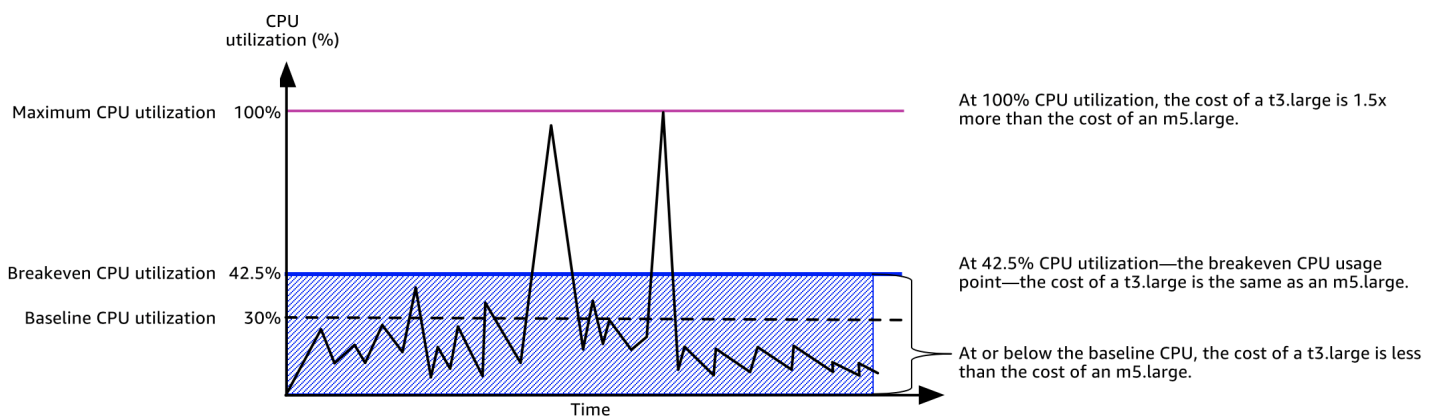
When to use unlimited mode versus fixed CPU

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	Addition burst minutes available	Addition 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	Additional CPU %	Breakeven CPU %
t3.large	2	\$0.0835	\$0.096	\$0.0125	30%	\$0.05	\$0.000833	15	12.5%	42.5%

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

The table provides the following information:

- Column A shows the instance type, t3.large.
- Column B shows the number of vCPUs for the t3.large.
- Column C shows the price of a t3.large per hour.
- Column D shows the price of an m5.large per hour.
- Column E shows the price difference between the t3.large and the m5.large.
- Column F shows the baseline utilization per vCPU of the t3.large, which is 30%. At the baseline, the hourly cost of the instance covers the cost of the CPU usage.
- Column G shows the [flat additional rate](#) per vCPU-hour that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column H shows the [flat additional rate](#) per vCPU-minute that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column I shows the number of additional minutes that the t3.large can burst per hour at 100% CPU while paying the same price per hour as an m5.large.
- Column J shows the additional CPU usage (in %) over baseline that the instance can burst while paying the same price per hour as an m5.large.

- Column K shows the breakeven CPU usage (in %) that the `t3.large` can burst without paying more than the `m5.large`. Anything above this, and the `t3.large` costs more than the `m5.large`.

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

T3 instance type	Breakeven CPU usage (in %) for T3 compared to M5
<code>t3.large</code>	42.5%
<code>t3.xlarge</code>	52.5%
<code>t3.2xlarge</code>	52.5%

Surplus credits can incur charges

If the average CPU utilization of an instance is at or below the baseline, the instance incurs no additional charges. Because an instance earns a [maximum number of credits](#) 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 support Unlimited mode. The Unlimited mode credit configuration enables T4g, T3a and T3 instances to use as much CPU as needed to burst beyond baseline and for as long as needed.

Enable unlimited mode

You can switch from unlimited to standard, and from standard to unlimited, at any time on a running or stopped instance. For more information, see [Launch a burstable performance instance as Unlimited or Standard](#) and [Modify 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 [Set the default credit specification for the 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 [View the credit specification of a burstable performance instance](#) and [View the default credit specification](#).

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 `CPU surplusCreditBalance` value is immediately charged.

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

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

Monitor credit usage

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

Unlimited mode examples

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

Examples

- [Example 1: Explain credit use with T3 Unlimited](#)
- [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.

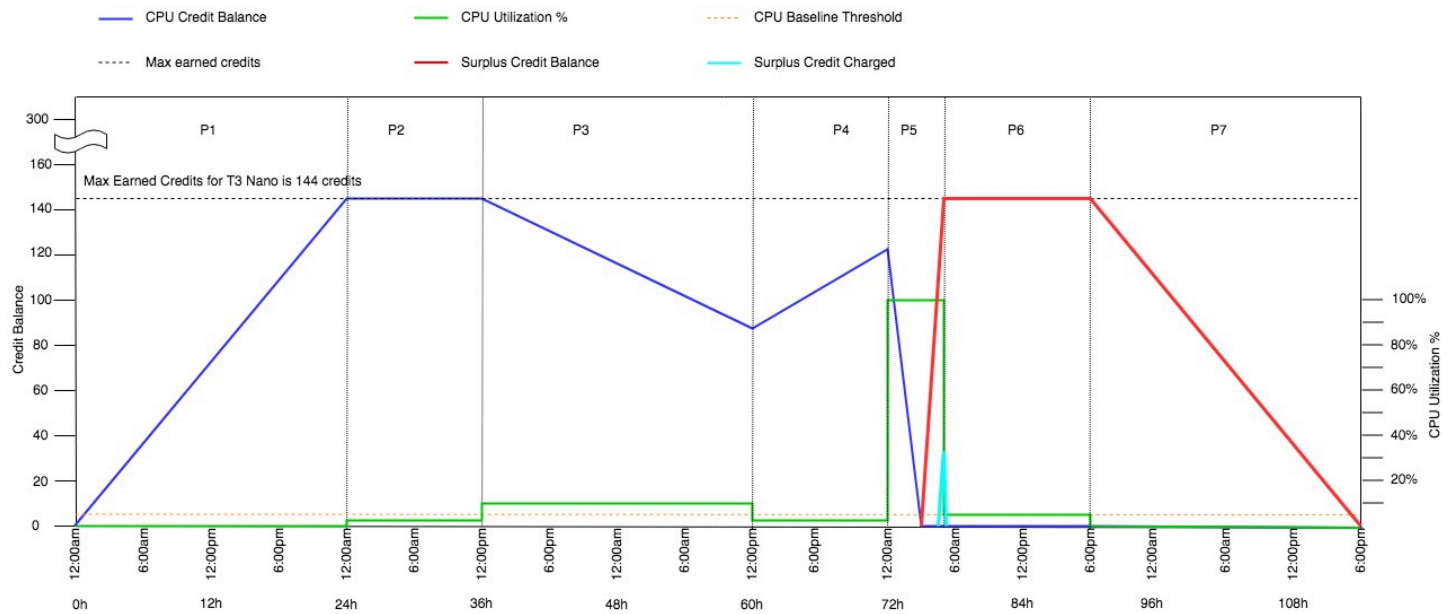
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

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.

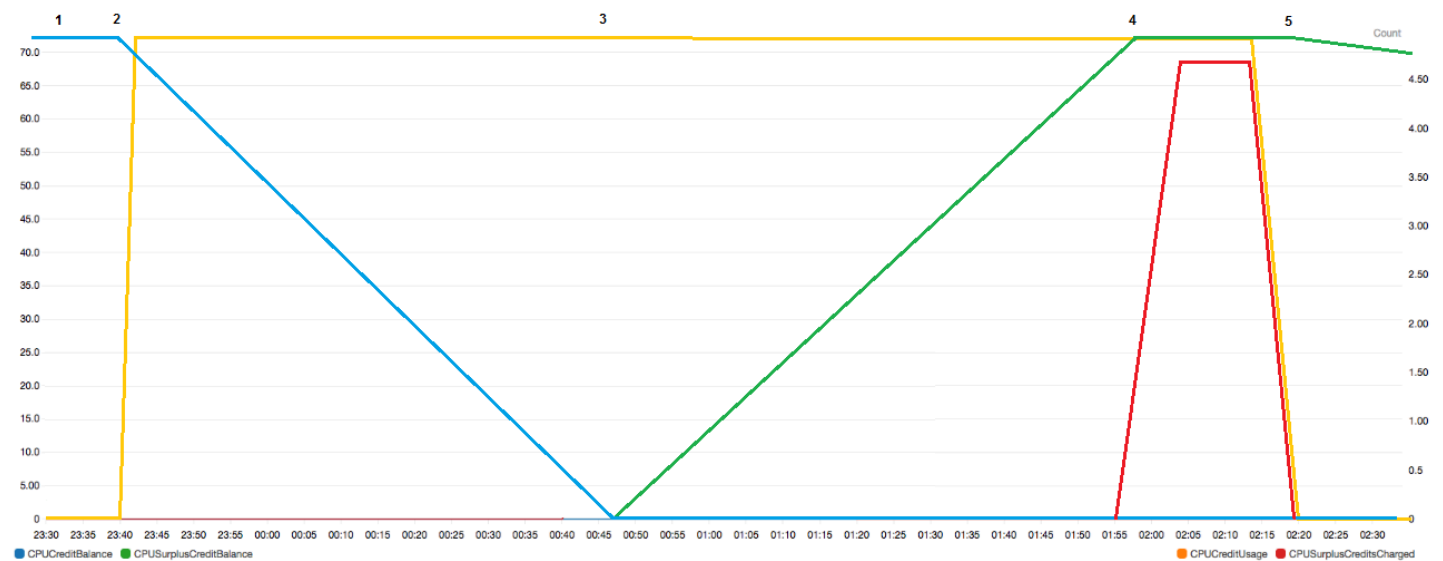
The intent of the example, illustrated by the following graph, is to show how an instance can burst using surplus credits even after it depletes its `CPUCreditBalance`. You can assume that, at the start of the time line in the graph, the instance has an accrued credit balance equal to the maximum number 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 `CPUcreditBalance` 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 `CPUcreditBalance`. After the `CPUcreditBalance` value reduces to 0, the instance starts to accrue earned credits in its `CPUcreditBalance` at 0.25 credits every 5 minutes.



Label	Details	Statistic	Period	Y Axis	Actions
CPUcreditBalance	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUcreditBalance	Maximum	5 Minutes	< >	
CPUcreditUsage	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUcreditUsage	Maximum	5 Minutes	< >	
CPUSurplusCreditBalance	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUSurplusCreditBalance	Maximum	5 Minutes	< >	
CPUSurplusCreditsCharged	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUSurplusCreditsCharged	Maximum	5 Minutes	< >	

Calculating the bill

Surplus credits cost \$0.05 per vCPU-hour. The instance spent approximately 25 surplus credits between 01:55 and 02:20, which is equivalent to 0.42 vCPU-hours.

Additional charges for this instance are 0.42 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

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

Standard mode for burstable performance instances

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

Contents

- [Standard mode concepts](#)
 - [How standard burstable performance instances work](#)
 - [Launch credits](#)
 - [Launch credit limits](#)
 - [Differences between launch credits and earned credits](#)
- [Standard mode examples](#)
 - [Example 1: Explain credit use with T3 Standard](#)
 - [Example 2: Explain credit use with T2 Standard](#)
 - [Period 1: 1 – 24 hours](#)
 - [Period 2: 25 – 36 hours](#)
 - [Period 3: 37 – 61 hours](#)
 - [Period 4: 62 – 72 hours](#)
 - [Period 5: 73 – 75 hours](#)
 - [Period 6: 76 – 90 hours](#)

- [Period 7: 91 – 96 hours](#)

Standard mode concepts

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

How standard burstable performance instances work

When a burstable performance instance configured as standard is in a running state, it continuously earns (at a millisecond-level resolution) a set rate of earned credits per hour. For T2 Standard, when the instance is stopped, it loses all its accrued credits, and its credit balance is reset to zero. When it is restarted, it receives a new set of launch credits, and begins to accrue earned credits. For 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	<p>T2 Standard instances get 30 launch credits per vCPU at launch or start.</p> <p>If a T2 instance is switched from <code>unlimited</code> to <code>standard</code>, it does not get launch credits at the time of switching.</p>	<p>Each T2 instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. For more information about the number of CPU credits earned per instance size, see the credit table.</p>
Credit earn limit	<p>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.</p>	<p>A T2 instance cannot accrue more credits than the CPU credit balance limit. If the CPU credit balance has reached its limit, any credits that are earned after the limit is reached are discarded. Launch credits do not count towards the limit. For more information about the CPU credit balance limit for each T2 instance size, see the credit table.</p>
Credit use	<p>Launch credits are spent first, before earned credits.</p>	<p>Earned credits are spent only after all launch credits are spent.</p>
Credit expiration	<p>When a T2 Standard instance is running, launch credits do not expire. When a T2 Standard instance stops or</p>	<p>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.</p>

	Launch credits	Earned credits
	is switched to T2 Unlimited, all launch 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

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 its limit, which is the maximum number of earned credits that can be accrued.

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

The following workflow references the numbered points on the graph:

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

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

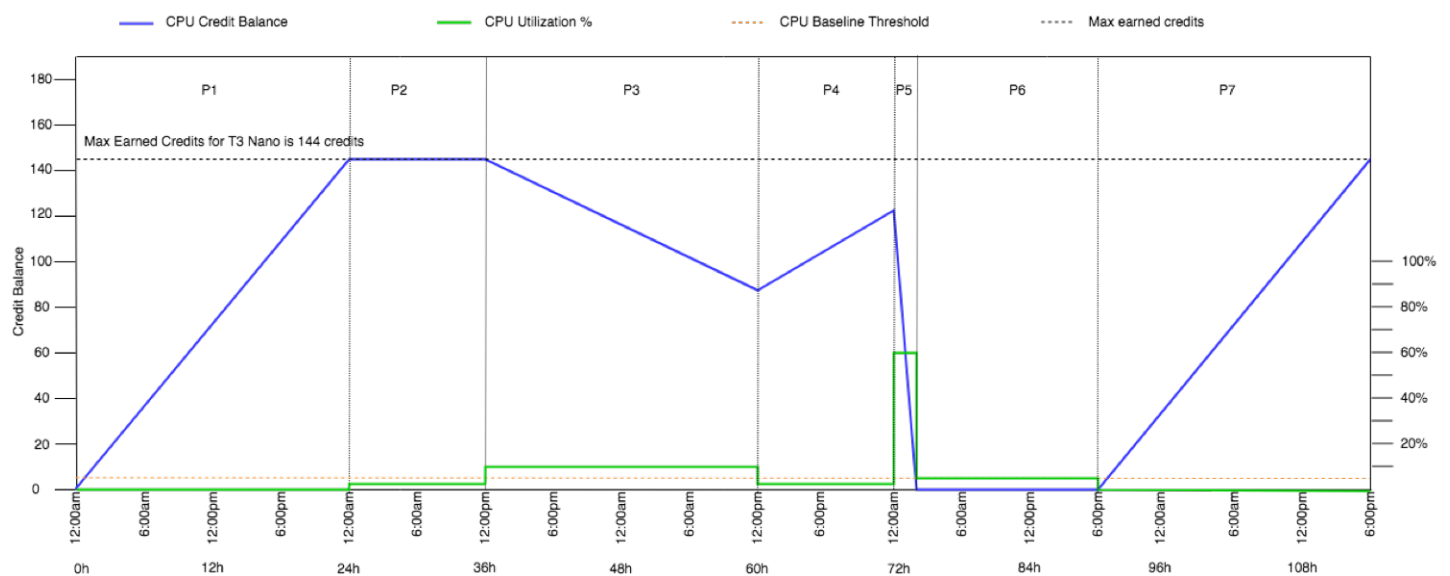
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the CPUcreditBalance value reduces to 86.4 credits.

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

P5 – For the next two hours, the instance bursts at 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.

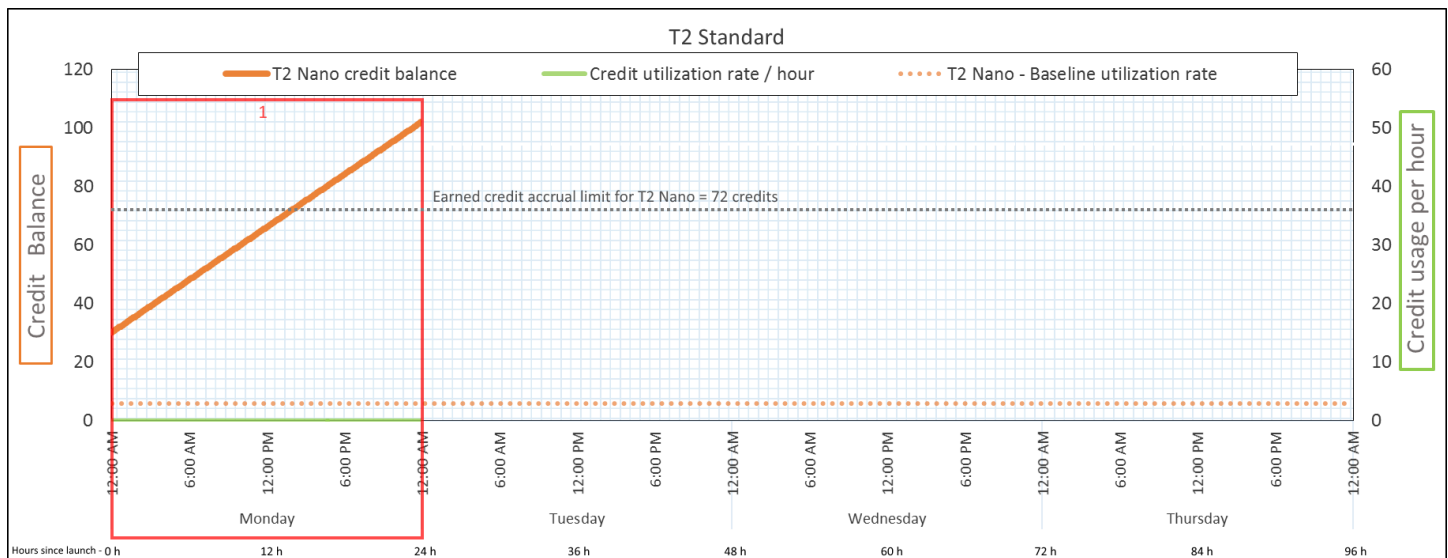
A `t2.nano` instance gets 30 launch credits when it is launched, and earns 72 credits every 24 hours. Its credit balance limit is 72 earned credits; launch credits do not count towards the limit. After the limit is reached, new credits that are earned are discarded. For more information about the number 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)

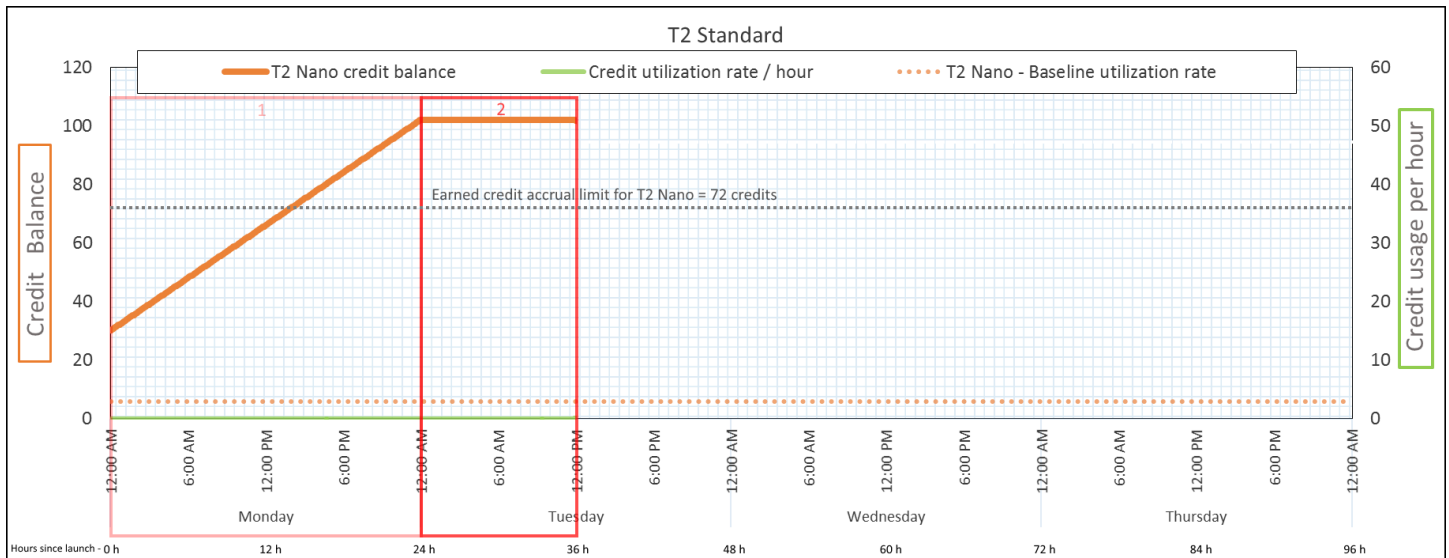
Conclusion

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.



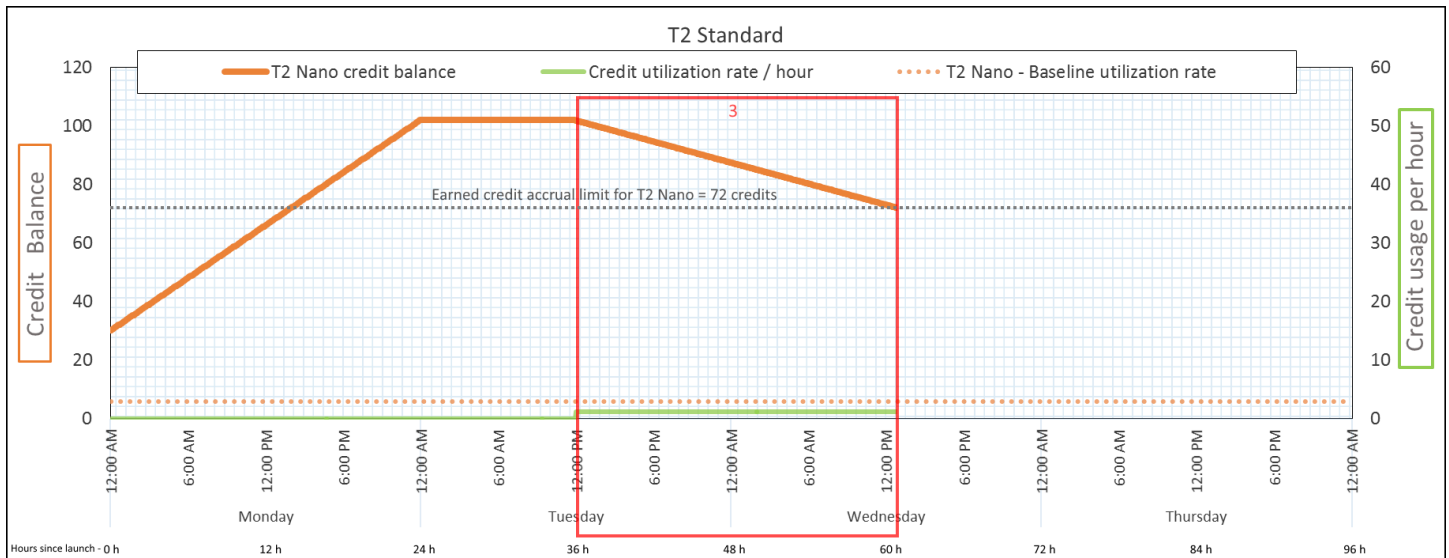
Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours (3 credits per hour)
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	102 credits (30 launch credits + 72 earned credits)—balance is unchanged

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)

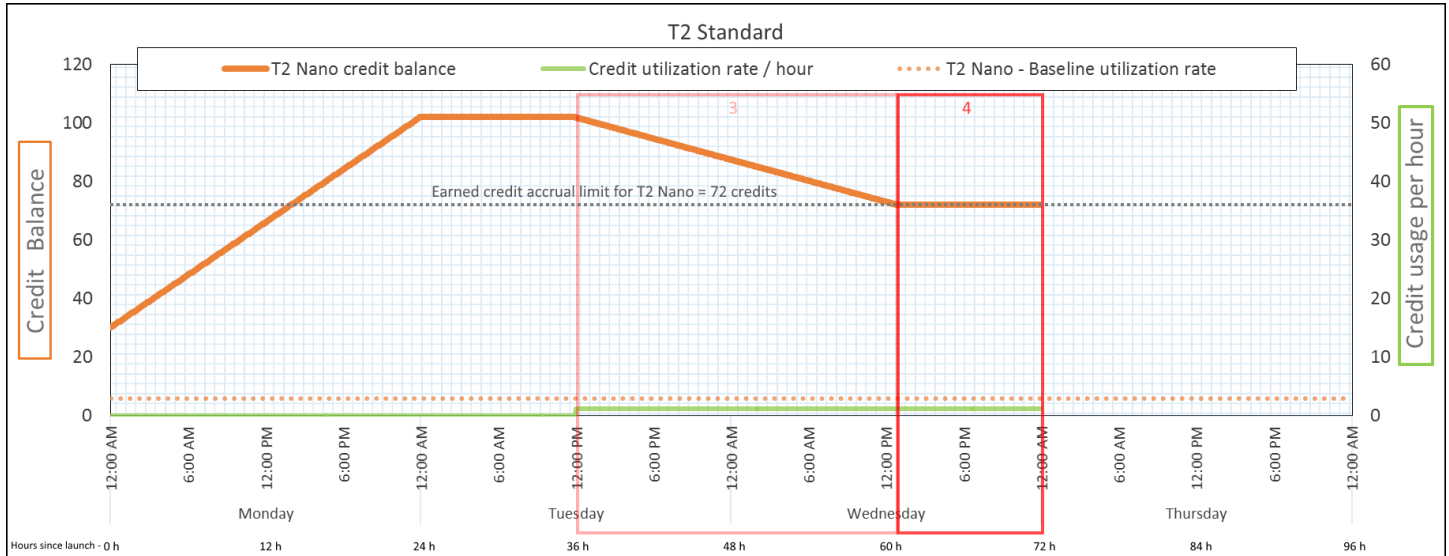
Conclusion

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.



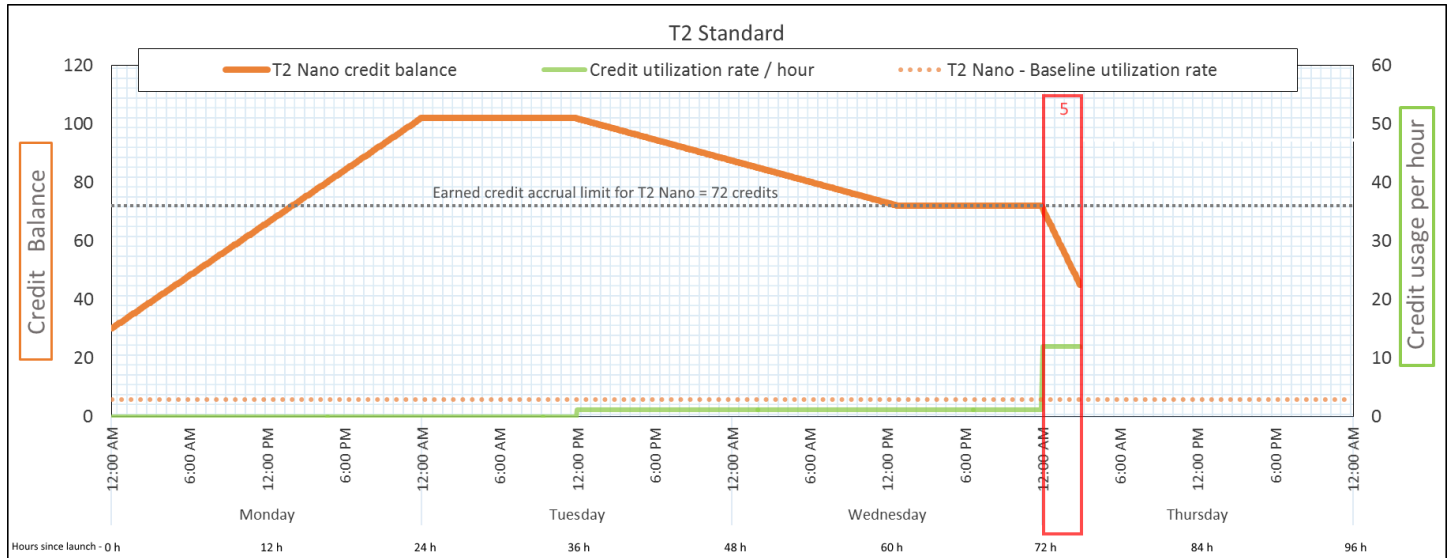
Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)— 13.2 credits over 11 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	43.2 credits per 24 hours (60% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72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

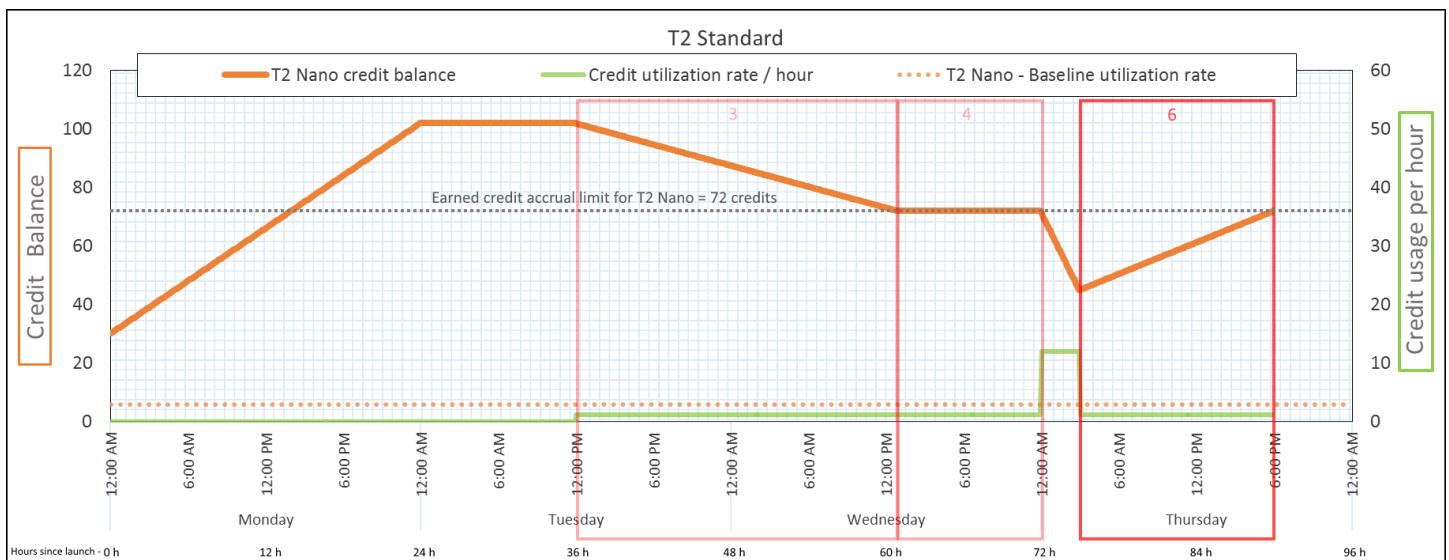
If an instance spends more credits than it earns, its credit balance decreases.

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)
----------------	--

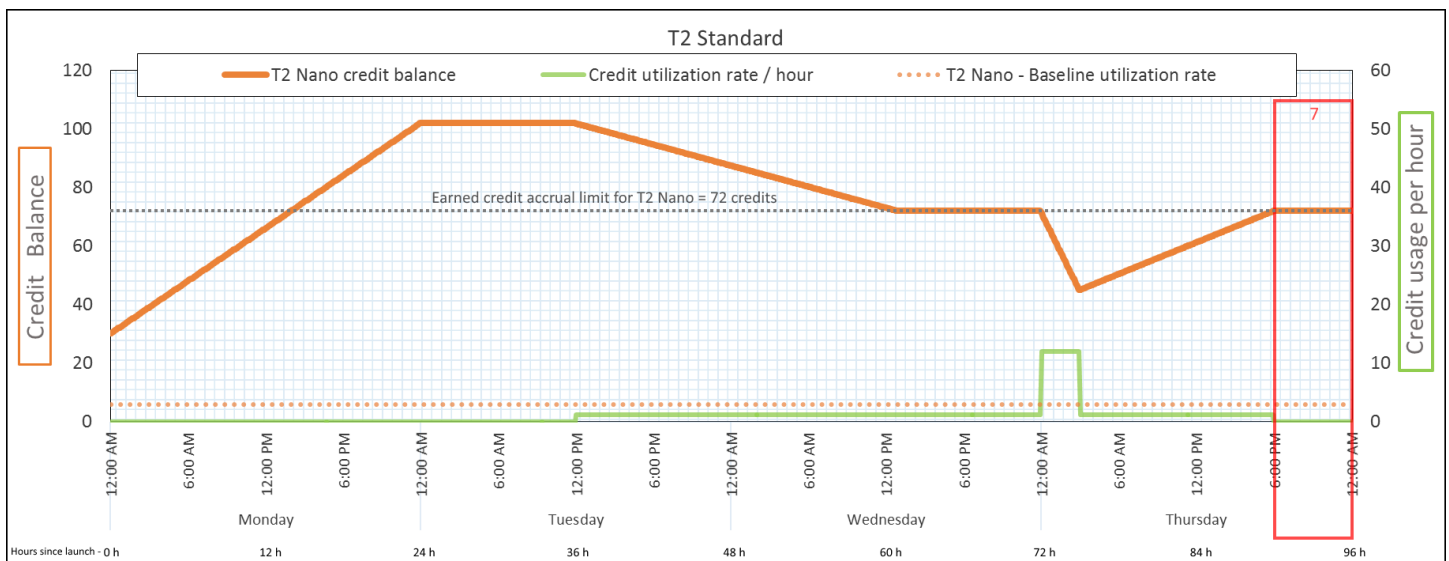
Conclusion

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](#).

Work with 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.

Contents

- [Launch a burstable performance instance as Unlimited or Standard](#)
- [Use an Auto Scaling group to launch a burstable performance instance as Unlimited](#)
- [View the credit specification of a burstable performance instance](#)
- [Modify the credit specification of a burstable performance instance](#)
- [Set the default credit specification for the account](#)
- [View the default credit specification](#)

Launch a burstable performance instance as Unlimited or Standard

You can launch your T instances as `unlimited` or `standard` using the Amazon EC2 console, an AWS SDK, a command line tool, or with an Auto Scaling group.

The following procedures describe how to use the EC2 console or the AWS CLI. For information about using an Auto Scaling group, see [Use an Auto Scaling group to launch a burstable performance instance as Unlimited](#).

Console

To launch a T instance as Unlimited or Standard

1. Follow the procedure to [launch an instance](#).
2. Under **Instance type**, select a T instance type.
3. Expand **Advanced details**, and for **Credit specification**, select a credit specification. If you do not make a selection, the default is used, which is standard for T2, and unlimited for T4g, T3a and T3.
4. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see [Launch an instance using the new launch instance wizard](#).

AWS CLI

To launch a T instance as Unlimited or Standard

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

- For T4g, T3a and T3, if you do not include the `--credit-specification` parameter, the instance launches as `unlimited` by default.
- For T2, if you do not include the `--credit-specification` parameter, the instance launches as `standard` by default.

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

Use an Auto Scaling group to launch a burstable performance instance as Unlimited

When 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`.

Note

`unlimited` mode is not supported for T3 instances that are launched on a Dedicated Host.

Console

To create a launch template that launches instances as Unlimited

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 launches instances as Unlimited

Use the [create-launch-template](#) command and specify `unlimited` as the credit specification.

- For T4g, T3a and T3, if you do not include the `CreditSpecification={CpuCredits=unlimited}` value, the instance launches as `unlimited` by default.
- For T2, if you do not include the `CreditSpecification={CpuCredits=unlimited}` value, the instance launches as `standard` by default.

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

Associate an Auto Scaling group with a launch template

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

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*.

AWS CLI

To create an Auto Scaling group using a launch template

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

Console

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 add a launch template to an existing Auto Scaling group

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

View the credit specification of a burstable performance instance

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

Console

To view the credit specification of a T instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Details** and view the **Credit specification** field. The value is either unlimited or standard.

AWS CLI

To describe the credit specification of a T instance

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

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

Example output

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

Modify the credit specification of a burstable performance instance

You can switch the credit specification of a running or stopped 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 modify the credit specification of a T instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance. To modify the credit specification for several instances at one time, select all applicable instances.
4. Choose **Actions, Instance settings, Change credit specification**. This option is enabled only if you selected a T instance.
5. To change the credit specification to `unlimited`, select the check box next to the instance ID. To change the credit specification to `standard`, clear the check box next to the instance ID.

AWS CLI

To modify the credit specification of a T instance

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

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

Example output

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

Set the default credit specification for the 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.

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 set the default credit specification at the account level per 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. 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 set the default credit specification at the account level (AWS CLI)

Use the [modify-default-credit-specification](#) command. Specify the AWS Region, instance family, and the default credit specification using the `--cpu-credits` parameter. Valid default credit specifications are `unlimited` and `standard`.

```
aws ec2 modify-default-credit-specification \  
  --region us-east-1 \  
  --instance-family t2 \  
  --cpu-credits unlimited
```

View the default credit specification

You can view the default credit specification of a T instance family at the account level per AWS Region.

Console

To view the default credit specification at the account level

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**.

AWS CLI

To view the default credit specification at the account level

Use the [get-default-credit-specification](#) command. Specify the AWS Region and instance family.

```
aws ec2 get-default-credit-specification --region us-east-1 --instance-family t2
```

Monitor your CPU credits for burstable performance 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 [List metrics using the console](#) and [List metrics using the AWS CLI](#).

Contents

- [Additional CloudWatch metrics for burstable performance instances](#)
- [Calculate CPU credit usage](#)

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 [List the available CloudWatch metrics 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%

Metric	Description
	<p>utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes).</p> <p>CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the Sum statistic instead of the Average statistic.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUCreditBalance	<p>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the <code>CPUCreditBalance</code> also includes the number of launch credits that have been accrued.</p> <p>Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits do not count towards the limit.</p> <p>The credits in the <code>CPUCreditBalance</code> are available for the instance to spend to burst beyond its baseline CPU utilization.</p> <p>When an instance is running, credits in the <code>CPUCreditBalance</code> do not expire. When a T4g, T3a or T3 instance stops, the <code>CPUCreditBalance</code> value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the <code>CPUCreditBalance</code> value does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p> <p>Units: Credits (vCPU-minutes)</p>

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

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

$$\text{CPUCreditBalance} = 2 + [0.5 - 1] = 1.5$$

Calculate CPU credit usage for Unlimited instances

When a burstable performance instance needs to burst above the baseline, it always spends accrued credits before spending surplus credits. When it depletes its accrued CPU credit balance, it can spend surplus credits to burst CPU for as long as it needs. When CPU utilization falls below the baseline, surplus credits are always paid down before the instance accrues earned credits.

We use the term `Adjusted balance` in the following equations to reflect the activity that occurs in this five-minute interval. We use this value to arrive at the values for the `CPUCreditBalance` and `CPUSurplusCreditBalance` CloudWatch metrics.

Example

$$\text{Adjusted balance} = [\text{prior CPUCreditBalance} - \text{prior CPUSurplusCreditBalance}] + [\text{Credits earned per hour} * (5/60) - \text{CPUCreditUsage}]$$

A value of 0 for `Adjusted balance` indicates that the instance spent all its earned credits for bursting, and no surplus credits were spent. As a result, both `CPUCreditBalance` and `CPUSurplusCreditBalance` are set to 0.

A positive `Adjusted balance` value indicates that the instance accrued earned credits, and previous surplus credits, if any, were paid down. As a result, the `Adjusted balance` value is assigned to `CPUCreditBalance`, and the `CPUSurplusCreditBalance` is set to 0. The instance size determines the [maximum number of credits](#) that it can accrue.

Example

$$\begin{aligned} \text{CPUCreditBalance} &= \min [\text{max earned credit balance}, \text{Adjusted balance}] \\ \text{CPUSurplusCreditBalance} &= 0 \end{aligned}$$

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.

Amazon EC2 Mac instances

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 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.

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.

Contents

- [Considerations](#)
- [Instance readiness](#)
- [EC2 macOS AMIs](#)
- [EC2 macOS Init](#)
- [Amazon EC2 System Monitor for macOS](#)
- [Related resources](#)
- [Launch a Mac instance](#)
- [Connect to your Mac instance](#)
- [Update the operating system and software on Mac instances](#)
- [Increase the size of an EBS volume on your Mac instance](#)
- [Stop and terminate your Mac instance](#)
- [Find supported macOS versions for your Amazon EC2 Mac Dedicated Host](#)
- [Subscribe to macOS AMI notifications](#)
- [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](#).
- 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)
- 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 General Purpose SSD (gp2 and gp3) and Provisioned IOPS SSD (io1 and io2) with Mac instances for optimal EBS performance.
- [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 Mac instances](#).
- When you stop or terminate a Mac instance, a scrubbing workflow is performed on the Dedicated Host. For more information, see [Stop and terminate your Mac instance](#).

Warning

Do not use FileVault. Enabling FileVault will result in the host failing to boot due to the partitions being 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 servers that host instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its 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-0123456789example \
  --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

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**, or **mac2-m2pro**. 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**, or **mac2-m2pro.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**, or **mac2-m2pro.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`, or `mac2-`

`m2pro.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`, or `mac2-m2pro.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](#) 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

You can connect to your Mac instance using SSH or a graphical user interface.

Connect to your instance using SSH

Important

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.

Amazon EC2 Mac instances do not allow remote root SSH by default. Password authentication is disabled to prevent brute-force password attacks. 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 connecting to your 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
```

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 user name 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 Mac instances

Warning

Installation of beta or preview macOS versions is only available on Amazon EC2 M1 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.

Steps to update software on x86 Mac instances and Apple silicon Mac instances.

- [Update software on x86 Mac instances](#)
- [Update software on Apple silicon 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
```

Update software on Apple silicon Mac instances

Considerations

Elastic Network Adapter (ENA) driver

Due to an update in the network driver configuration, ENA driver version 1.0.2 isn't compatible with macOS 13.3 or greater. 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. Download the ENA application into the `Applications` file using the following command.

```
[ec2-user ~]$ brew install amazon-ena-ethernet-dext
```

Troubleshooting tip

If you receive the warning `No available formula with the name amazon-ena-ethernet-dext`, run the following command.

```
[ec2-user ~]$ brew update
```

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.

Software update on Apple silicon Mac instances

On Apple silicon Mac instances, you must complete several steps to perform an in-place operating system update. First, access the internal disk of the instance using the GUI with a VNC (Virtual Network Computing) client. This procedure uses macOS Screen Sharing, the built in VNC client. Then, delegate ownership to the administrative user (`ec2-user`) by signing in as `aws-managed-user` on the Amazon EBS volume.

As you work through this procedure, you create two passwords. One password is for the administrative user (`ec2-user`) and the other password is for a special 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 and terminate your 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 user name `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](#).

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 and terminate your 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**.

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 and terminate your Mac instance](#).

For more information about launching Mac instances, see [Launch a Mac instance](#).

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**

For **Protocol**:

- b. **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

c. **SMS:**

For **Endpoint**, type a phone number that you can use to receive the notifications.

d. **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.

e. 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**.

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](#).

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
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
2024.04.12	<p>All AMIs</p> <ul style="list-style-type: none"> Updated Homebrew to 4.2.16-1

Release	Changes
	<ul style="list-style-type: none">Updated <code>aws-cli</code> to 2.15.36
	Release of macOS Sonoma 14.4.1 (all Mac instances)
	<ul style="list-style-type: none">Security content of macOS Sonoma 14.4.1
	Release of macOS Ventura 13.6.6 (all Mac instances)
	<ul style="list-style-type: none">Security content of macOS Ventura 13.6.6Updated Safari to 17.4.1<ul style="list-style-type: none">Security content of Safari 17.4.1
	For macOS Monterey (all Mac instances)
	<ul style="list-style-type: none">Updated Safari to 17.4.1<ul style="list-style-type: none">Security content of Safari 17.4.1

Amazon EBS–optimized instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated bandwidth to Amazon EBS. When attached to an EBS–optimized instance, General Purpose SSD (gp2 and gp3) volumes are designed to deliver at least 90% of their provisioned IOPS performance 99% of the time in a given year, and Provisioned IOPS SSD (io1 and io2) volumes are designed to deliver at least 90% of their provisioned IOPS performance 99.9% of the time in a given year. Both Throughput Optimized HDD (st1) and Cold HDD (sc1) deliver at least 90% of their expected throughput performance 99% of the time in a given year. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour. For more information, see [Amazon EBS volume types](#) in the *Amazon EBS User Guide*.

⚠ Important

An instance's EBS performance is bounded by the instance'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).

Contents

- [Supported instance types](#)
- [Get maximum performance](#)
- [View instances types that support EBS optimization](#)
- [Enable EBS optimization at launch](#)
- [Enable EBS optimization for an existing instance](#)

Supported instance types

The following tables show which instance types support EBS optimization. They include the dedicated bandwidth to Amazon EBS, the typical maximum aggregate throughput that can be achieved on that connection with a streaming read workload and 128 KiB I/O size, and the maximum IOPS the instance can support if you are using a 16 KiB I/O size.

Choose an EBS-optimized instance that provides more dedicated Amazon EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.

Contents

- [EBS optimized by default](#)
- [EBS optimization supported](#)

EBS optimized by default

The following tables list the instance types that support EBS optimization and EBS optimization is enabled by default. There is no need to enable EBS optimization and no effect if you disable EBS optimization.

Note

You can also view this information programmatically using the AWS CLI. For more information, see [View instances types that support EBS optimization](#).

Topics

- [General purpose](#)
- [Compute optimized](#)
- [Memory optimized](#)
- [Storage optimized](#)
- [Accelerated computing](#)
- [High-performance computing](#)

General purpose

Important

¹ These instances can support 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.

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
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

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.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
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

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)
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
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

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.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
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	

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.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
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

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.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
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

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)
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
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

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.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
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

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.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
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

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.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
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

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.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
m6id.24xlarge ²		30000		3750.0		120000
m6id.32xlarge ²		40000		5000.0		160000
m6id.meta1 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

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.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	
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

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.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
m6in.meta 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

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.8xlarge ²	10000	10000	1250.0	1250.0	40000	40000
m7a.12xlarge ²	15000	15000	1875.0	1875.0	60000	60000
m7a.16xlarge ²	20000	20000	2500.0	2500.0	80000	80000
m7a.24xlarge ²	30000	30000	3750.0	3750.0	120000	120000
m7a.32xlarge ²	40000	40000	5000.0	5000.0	160000	160000
m7a.48xlarge ²	40000	40000	5000.0	5000.0	240000	240000
m7a.metal-48xl ²	40000	40000	5000.0	5000.0	240000	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

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)
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
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

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.12xlarge ²	15000		1875.0		60000	
m7gd.16xlarge ²	20000		2500.0		80000	
m7gd.meta1 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
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	

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.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
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
mac1.metal ²		14000		1750.0		80000
mac2.metal ²		10000		1250.0		55000
mac2-m2.metal ²		8000		1000.0		55000

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)
mac2-m2pro.metal ²	8000		1000.0		55000	
t3.nano ¹	43	2085	5.38	260.62	250	11800
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

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)
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
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

Compute optimized

Important

¹ These instances can support 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.

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.meta 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.meta l 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

Memory optimized

Important

¹ These instances can support 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.

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

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)
r4.2xlarge 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
r5.16xlarge ₂		13600		1700.0		60000
r5.24xlarge 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)
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
r5ad.xlarge ₁	1085	2880	135.62	360.00	6000	16000
r5ad.2xlarge ₁	1580	2880	197.50	360.00	8333	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.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	
r5b.8xlarge ²	20000		2500.0		86667	
r5b.12xlarge ²	30000		3750.0		130000	

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.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	
r5d.24xlarge ²	19000		2375.0		80000	
r5d.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)
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.meta l ²		19000		2375.0		80000
r5n.large ¹	650	4750	81.25	593.75	3600	18750
r5n.xlarge ¹	1150	4750	143.75	593.75	6000	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.2xlarge ¹	2300	4750	287.50	593.75	12000	18750
r5n.4xlarge ²	4750	9500	593.75	1187.5	18750	37500
r5n.8xlarge ²	6800	13600	850.0	1700.0	30000	60000
r5n.12xlarge ²	9500	19000	1187.5	2375.0	40000	80000
r5n.16xlarge ²	13600	27200	1700.0	3400.0	60000	120000
r5n.24xlarge ²	19000	38000	2375.0	4750.0	80000	160000
r5n.metal ²	19000	38000	2375.0	4750.0	80000	160000
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
r6a.8xlarge ²	10000	10000	1250.0	1250.0	40000	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.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
r6g.8xlarge 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)
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
r6gd.16xlarge 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)
r6gd.meta l 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	
r6idn.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)
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
r6in.xlarge ¹	3125	25000	390.62	3125.00	12500	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.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
r6id.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)
r6id.4xlarge ¹	5000	10000	625.00	1250.00	20000	40000
r6id.8xlarge ²	10000	20000	1250.00	2500.00	40000	80000
r6id.12xlarge ²	15000	30000	1875.00	3750.00	60000	120000
r6id.16xlarge ²	20000	40000	2500.00	5000.00	80000	160000
r6id.24xlarge ²	30000	60000	3750.00	7500.00	120000	240000
r6id.32xlarge ²	40000	80000	5000.00	10000.00	160000	320000
r6id.metal ²	40000	80000	5000.00	10000.00	160000	320000
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
r7a.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)
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
r7g.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)
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
r7gd.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)
r7gd.8xlarge ²		10000		1250.0		40000
r7gd.12xlarge ²		15000		1875.0		60000
r7gd.16xlarge ²		20000		2500.0		80000
r7gd.meta l 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
r7i.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)
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	
r7iz.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)
r7iz.meta-l-16xl ²	20000		2500.0		80000	
r7iz.meta-l-32xl ²	40000		5000.0		160000	
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-12tb1.12xlarge ²	38000		4750.0		160000	
u-12tb1.metal ²	38000		4750.0		160000	
u-18tb1.12xlarge ²	38000		4750.0		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)
u-18tb1.metal ²		38000		4750.0		160000
u-24tb1.12xlarge ²		38000		4750.0		160000
u-24tb1.metal ²		38000		4750.0		160000
x1.16xlarge ²		7000		875.0		40000
x1.32xlarge 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
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

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.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	
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	

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.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	
x2iezn.8xlarge ²	12000		1500.0		55000	
x2iezn.12xlarge ²	19000		2375.0		80000	
x2iezn.metal ²	19000		2375.0		80000	
x1e.xlarge ²	500		62.5		3700	

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)
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	
z1d.large ¹	800	3170	100.00	396.25	3333	13333
z1d.xlarge ¹	1580	3170	197.50	396.25	6667	13333
z1d.2xlarge ²	3170		396.25		13333	
z1d.3xlarge ²	4750		593.75		20000	
z1d.6xlarge ²	9500		1187.5		40000	
z1d.12xlarge 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)
z1d.metal ²		19000		2375.0		80000

Storage optimized

Important

¹ These instances can support 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.

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

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)
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
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

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)
<code>h1.4xlarge²</code>	3500		437.5		20000	
<code>h1.8xlarge²</code>	7000		875.0		40000	
<code>h1.16xlarge²</code>	14000		1750.0		80000	
<code>i3.large²</code>	425		53.125		3000	
<code>i3.xlarge²</code>	850		106.25		6000	
<code>i3.2xlarge²</code>	1700		212.5		12000	
<code>i3.4xlarge²</code>	3500		437.5		16000	
<code>i3.8xlarge²</code>	7000		875.0		32500	
<code>i3.16xlarge²</code>	14000		1750.0		65000	
<code>i3.metal²</code>	19000		2375.0		80000	
<code>i3en.large¹</code>	576	4750	72.10	593.75	3000	20000
<code>i3en.xlarge¹</code>	1153	4750	144.20	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)
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
i3en.meta1 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

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.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
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
im4gn.large ¹	1250	10000	156.25	1250.00	5000	40000
im4gn.xlarge ¹	2500	10000	312.50	1250.00	10000	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)
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
is4gen.4xlarge ²	10000		1250.0		40000	
is4gen.8xlarge ²	20000		2500.0		80000	

Accelerated computing

Important

¹ These instances can support 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.

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)
dl1.24xlarge ²	19000		2375.0		80000	
dl2q.24xlarge ²	19000		2375.0		80000	
f1.2xlarge ²	1700		212.5		12000	
f1.4xlarge ²	3500		437.5		44000	
f1.16xlarge ²	14000		1750.0		75000	
g3.4xlarge ²	3500		437.5		20000	
g3.8xlarge ²	7000		875.0		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)
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	
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	

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)
g4dn.16xlarge ²	9500		1187.5		40000	
g4dn.meta1 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	
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

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)
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	
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	

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)
g6.24xlarge ²		30000		3750.0		120000
g6.48xlarge ²		60000		7500.0		240000
gr6.4xlarge ²		8000		1000.0		32000
gr6.8xlarge ²		16000		2000.0		64000
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

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)
inf2.48xlarge ²	60000		7500.0		240000	
p2.xlarge ²	750		93.75		6000	
p2.8xlarge ₂	5000		625.0		32500	
p2.16xlarge ²	10000		1250.0		65000	
p3.2xlarge ₂	1750		218.75		10000	
p3.8xlarge ₂	7000		875.0		40000	
p3.16xlarge ²	14000		1750.0		80000	
p3dn.24xlarge ²	19000		2375.0		80000	
p4d.24xlarge ²	19000		2375.0		80000	
p4de.24xlarge ²	19000		2375.0		80000	
p5.48xlarge 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)
trn1.2xlarge ¹	5000	20000	625.00	2500.00	16250	65000
trn1.32xlarge ²	80000		10000.0		260000	
trn1n.32xlarge 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	

High-performance computing

Important

¹ These instances can support 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.

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

EBS optimization supported

The following table lists the instance types that support EBS optimization but EBS optimization is not enabled by default. You can enable EBS optimization when you launch these instances or after they are running. Instances must have EBS optimization enabled to achieve the level

of performance described. When you enable EBS optimization for an instance that is not EBS-optimized by default, you pay an additional low, hourly fee for the dedicated capacity. For pricing information, see [EBS-Optimized Instances on the Amazon EC2 Pricing, On-Demand Pricing page](#).

Note

You can also view this information programmatically using the AWS CLI. For more information, see [View instances types that support EBS optimization](#).

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

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
r3.2xlarge	1000	125.0	8000
r3.4xlarge	2000	250.0	16000

The `i2.8xlarge`, `c3.8xlarge`, and `r3.8xlarge` instances do not have dedicated EBS bandwidth and therefore do not offer EBS optimization. On these instances, network traffic and Amazon EBS traffic share the same 10-gigabit network interface.

Get maximum performance

You can use the `EBSIOBalance%` and `EBSByteBalance%` metrics to help you determine whether your instances are sized correctly. You can view these metrics in the CloudWatch console and set an alarm that is triggered based on a threshold you specify. These metrics are expressed as a percentage. Instances with a consistently low balance percentage are candidates to size up. Instances where the balance percentage never drops below 100% are candidates for downsizing. For more information, see [Monitor your instances using CloudWatch](#).

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 performance listed in the table above. To get the maximum performance from an instance launched before February 26, 2020, stop and start it.
- `C5`, `C5d`, `C5n`, `M5`, `M5d`, `M5n`, `M5dn`, `R5`, `R5d`, `R5n`, `R5dn`, and `P3dn` instances launched after December 3, 2019 provide the maximum performance listed in the table above. To get the maximum performance from an instance launched before December 3, 2019, stop and start it.

- u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal instances launched after March 12, 2020 provide the performance in the table above. Instances of these types launched before March 12, 2020 might provide lower performance. To get the maximum performance from an instance launched before March 12, 2020, contact your account team to upgrade the instance at no additional cost.

View instances types that support EBS optimization

You can use the AWS CLI to view the instances types in the current Region that support EBS optimization.

To view the instance types that support EBS optimization and that have it enabled by default

Use the following [describe-instance-types](#) command.

```
aws ec2 describe-instance-types \
  --query 'InstanceTypes[].{InstanceType:InstanceType,"MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIOPS,"MaxThroughput(MB/s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
  --filters Name=ebs-info.ebs-optimized-support,Values=default --output=table
```

Example output for eu-west-1:

```
-----
|                                     DescribeInstanceTypes                                     |
|-----+-----+-----+-----+
| InstanceType | MaxBandwidth(Mb/s) | MaxIOPS | MaxThroughput(MB/s) |
|-----+-----+-----+-----+
| m5dn.8xlarge | 6800                | 30000  | 850.0                |
| m6gd.xlarge  | 4750                | 20000  | 593.75               |
| c4.4xlarge   | 2000                | 16000  | 250.0                |
| r4.16xlarge  | 14000              | 75000  | 1750.0               |
| m5ad.large   | 2880                | 16000  | 360.0                |
| ...
```

To view the instance types that support EBS optimization but do not have it enabled by default

Use the following [describe-instance-types](#) command.

```
aws ec2 describe-instance-types \
```

```
--query 'InstanceTypes[].{InstanceType:InstanceType,"MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIOPS,"MaxThroughput(MB/s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMbps}' \
--filters Name=ebs-info.ebs-optimized-support,Values=supported --output=table
```

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                |
+-----+-----+-----+-----+
```

Enable EBS optimization at launch

You can enable optimization for an instance by setting its attribute for EBS optimization.

To enable Amazon EBS optimization when launching an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, select an AMI.
4. In **Step 2: Choose an Instance Type**, select an instance type that is listed as supporting Amazon EBS optimization.

5. In **Step 3: Configure Instance Details**, complete the fields that you need and choose **Launch as EBS-optimized instance**. If the instance type that you selected in the previous step doesn't support Amazon EBS optimization, this option is not present. If the instance type that you selected is Amazon EBS-optimized by default, this option is selected and you can't deselect it.
6. Follow the directions to complete the wizard and launch your instance.

To enable EBS optimization when launching an instance using the command line

You can use one of the following commands with the corresponding option. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [run-instances](#) with `--ebs-optimized` (AWS CLI)
- [New-EC2Instance](#) with `-EbsOptimized` (AWS Tools for Windows PowerShell)

Enable EBS optimization for an existing instance

You can enable or disable optimization for an existing instance by modifying its Amazon EBS-optimized instance attribute. If the instance is running, you must stop it first.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

To enable EBS optimization for an existing instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. To stop the instance, choose **Actions, Instance state, Stop instance**. It can take a few minutes for the instance to stop.
4. With the instance still selected, choose **Actions, Instance settings, Change instance type**.
5. For **Change Instance Type**, do one of the following:
 - If the instance type of your instance is Amazon EBS-optimized by default, **EBS-optimized** is selected and you can't change it. You can choose **Cancel**, because Amazon EBS optimization is already enabled for the instance.

- If the instance type of your instance supports Amazon EBS optimization, choose **EBS-optimized** and then choose **Apply**.
 - If the instance type of your instance does not support Amazon EBS optimization, you can't choose **EBS-optimized**. You can select an instance type from **Instance type** that supports Amazon EBS optimization, choose **EBS-optimized**, and then choose **Apply**.
6. Choose **Instance state**, **Start instance**.

To enable EBS optimization for an existing instance using the command line

1. If the instance is running, use one of the following commands to stop it:
 - [stop-instances](#) (AWS CLI)
 - [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)
2. To enable EBS optimization, use one of the following commands with the corresponding option:
 - [modify-instance-attribute](#) with `--ebs-optimized` (AWS CLI)
 - [Edit-EC2InstanceAttribute](#) with `-EbsOptimized` (AWS Tools for Windows PowerShell)

Instance purchasing options

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

- [On-Demand Instances](#) – Pay, by the second, for the instances that you launch.
- [Savings Plans](#) – Reduce your Amazon EC2 costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years.
- [Reserved Instances](#) – Reduce your Amazon EC2 costs by making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years.
- [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 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](#).

For more information about Savings Plans, see the [Savings Plans User Guide](#).

Contents

- [Determine the instance lifecycle](#)
- [On-Demand Instances](#)
- [Reserved Instances](#)
- [Spot Instances](#)
- [Dedicated Hosts](#)
- [Dedicated Instances](#)
- [Capacity Reservations](#)

Determine the instance lifecycle

The lifecycle of an instance starts when it is launched and ends when it is terminated. The purchasing option that you choose affects the lifecycle of the instance. For example, an On-Demand Instance runs when you launch it and ends when you terminate it. A Spot Instance runs as long as capacity is available and your maximum price is higher than the Spot price.

Use one of the following methods to determine the lifecycle of an instance.

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Details** tab, under **Instance details**, find **Lifecycle**. If the value is `spot`, the instance is a Spot Instance. If the value is `normal`, the instance is either an On-Demand Instance or a Reserved Instance.

5. On the **Details** tab, under **Host and placement group**, find **Tenancy**. If the value is `host`, the instance is running on a Dedicated Host. If the value is `dedicated`, the instance is a Dedicated Instance.
6. (Optional) If you have purchased a Reserved Instance and want to verify that it is being applied, you can check the usage reports for Amazon EC2. For more information, see [Amazon EC2 usage reports](#).

To determine the instance lifecycle using the AWS CLI

Use the following [describe-instances](#) command:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

If the instance is running on a Dedicated Host, the output contains the following information:

```
"Tenancy": "host"
```

If the instance is a Dedicated Instance, the output contains the following information:

```
"Tenancy": "dedicated"
```

If the instance is a Spot Instance, the output contains the following information:

```
"InstanceLifecycle": "spot"
```

Otherwise, the output does not contain `InstanceLifecycle`.

On-Demand Instances

With On-Demand Instances, you pay for compute capacity by the 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](#).

Contents

- [Work with On-Demand Instances](#)
- [On-Demand Instance quotas](#)
 - [Monitor On-Demand Instance quotas and usage](#)
 - [Request a quota increase](#)
- [Query the prices of On-Demand Instances](#)

Work with On-Demand Instances

You can work with On-Demand Instances in the following ways:

- [Launch your instance](#)
- [Connect to your Linux instance](#)
- [Stop and start Amazon EC2 instances](#)
- [Hibernate your Amazon EC2 instance](#)
- [Reboot your instance](#)
- [Instance retirement](#)
- [Terminate Amazon EC2 instances](#)
- [Recover your instance](#)
- [Configure your Amazon Linux instance](#)
- [Identify EC2 Linux instances](#)

If you're new to Amazon EC2, see [Get started with Amazon EC2](#).

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.

We provide the following quota types for On-Demand Instances:

- Running On-Demand DL instances
- Running On-Demand F instances
- Running On-Demand G and VT instances
- Running On-Demand High Memory instances
- Running On-Demand HPC instances
- Running On-Demand Inf instances
- Running On-Demand P instances
- Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances
- Running On-Demand Trn instances
- Running On-Demand X instances

Quotas apply to running instances only. If your instance is pending, stopping, stopped, or hibernated, it does not count towards your quota.

Each quota type specifies the maximum number of vCPUs for one or more instance families. For information about the different instance families, generations, and sizes, see [Amazon EC2 Instance Types](#).

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

Important

We recommend Savings Plans over Reserved Instances. Savings 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](#).

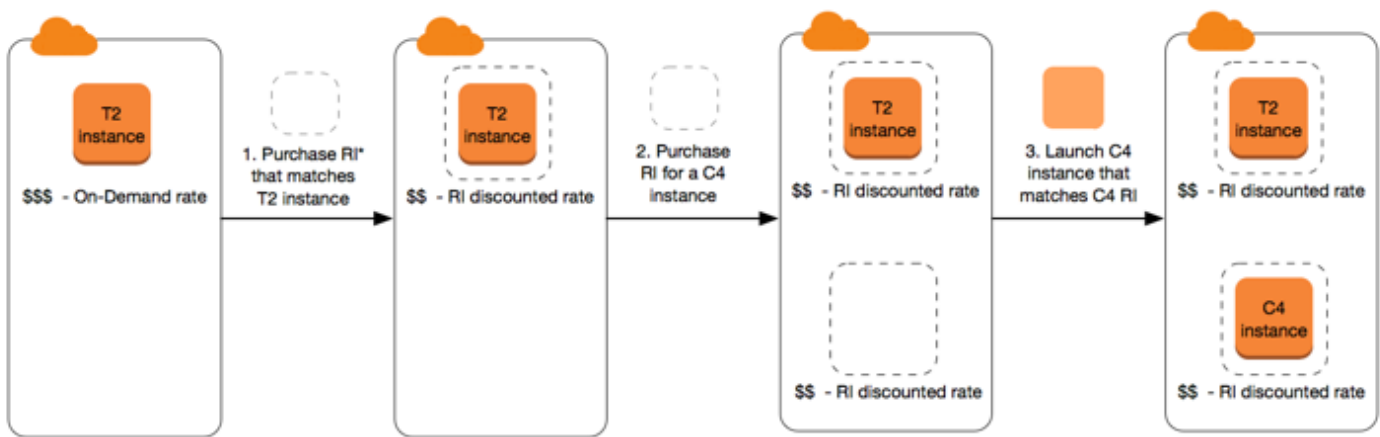
Reserved Instances provide you with significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. Reserved Instances are not physical instances, but rather a billing discount applied to the use of On-Demand Instances in your account. These On-Demand Instances must match certain attributes, such as instance type and Region, in order to benefit from the billing discount.

Reserved Instances topics

- [Reserved Instance overview](#)
- [Key variables that determine Reserved Instance pricing](#)
- [Regional and zonal Reserved Instances \(scope\)](#)
- [Types of Reserved Instances \(offering classes\)](#)
- [How Reserved Instances are applied](#)
- [Use your Reserved Instances](#)
- [How you are billed](#)
- [Buy Reserved Instances](#)
- [Sell in the Reserved Instance Marketplace](#)
- [Modify Reserved Instances](#)
- [Exchange Convertible Reserved Instances](#)
- [Reserved Instance quotas](#)

Reserved Instance overview

The following diagram shows a basic overview of purchasing and using Reserved Instances.



*RI = Reserved Instance

In this scenario, you have a running On-Demand Instance (T2) in your account, for which you're currently paying On-Demand rates. You purchase a Reserved Instance that matches the attributes of your running instance, and the billing benefit is immediately applied. Next, you purchase a Reserved Instance for a C4 instance. You do not have any running instances in your account that match the attributes of this Reserved Instance. In the final step, you launch an instance that matches the attributes of the C4 Reserved Instance, and the billing benefit is immediately applied.

Key variables that determine Reserved Instance pricing

The Reserved Instance pricing is determined by the following key variables.

Instance attributes

A Reserved Instance has four instance attributes that determine its price.

- **Instance type:** For example, `m4.large`. This is composed of the instance family (for example, `m4`) and the instance size (for example, `large`).
- **Region:** The Region in which the Reserved Instance is purchased.
- **Tenancy:** Whether your instance runs on shared (default) or single-tenant (dedicated) hardware. For more information, see [Dedicated Instances](#).
- **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 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 <i>not</i> 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 Instances 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 information, see Modify Reserved Instances .	Some attributes can be modified. For more information, see Modify Reserved Instances .
Exchange Reserved Instances	Can't be exchanged.	Can be exchanged during the term for another Convertible Reserved Instance with new attributes, including instance family, instance type, platform, scope, or tenancy. For more 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 Marketplace.	Can't be bought in the Reserved Instance Marketplace.

How Reserved Instances 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, generation, and attribute](#). 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, Inf1, and Inf2 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

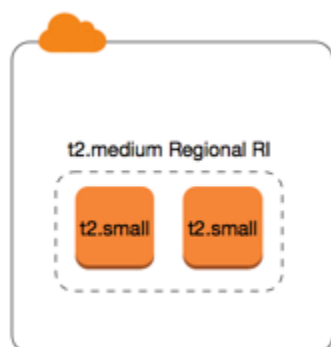
Instance size flexibility is determined by the normalization factor of the instance size. The discount applies either fully or partially to running instances of the same instance family, depending on the instance size of the reservation, in any Availability Zone in the Region. The only attributes that 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.

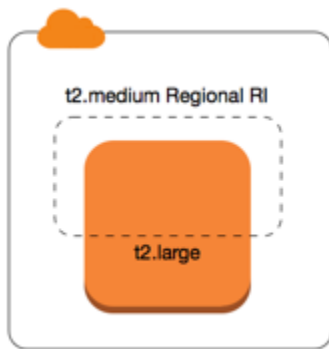
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96

Instance size	Normalization factor
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.

Instance size	Normalization factor
<code>a1.metal</code>	32
<code>m5zn.metal</code> <code>z1d.metal</code>	96
<code>c6g.metal</code> <code>c6gd.metal</code> <code>i3.metal</code> <code>m6g.metal</code> <code>m6gd.metal</code> <code>r6g.metal</code> <code>r6gd.metal</code> <code>x2gd.metal</code>	128
<code>c5n.metal</code>	144

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
u-*.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 [Reserved Instances in the Billing and Cost Management 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 [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 [Dedicated Instances](#).

For examples of how Reserved Instances are applied to your running On-Demand Instances, see [How Reserved Instances 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 your 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 you are billed

All Reserved Instances provide you with a discount compared to On-Demand pricing. With Reserved Instances, you pay for the entire term regardless of actual use. You can choose to pay for your Reserved Instance upfront, partially upfront, or monthly, depending on the [payment option](#) 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 under standard pricing guidelines. For information, see [AWS Free Tier](#).

Contents

- [Usage billing](#)
- [Viewing your bill](#)
- [Reserved Instances and consolidated billing](#)
- [Reserved Instance discount pricing tiers](#)

Usage billing

Reserved Instances are billed for every clock-hour during the term that you select, regardless of whether an instance is running. Each clock-hour starts on the hour (zero minutes and zero seconds past the hour) of a standard 24-hour clock. For example, 1:00:00 to 1:59:59 is one clock-hour. For more information about instance states, see [Instance lifecycle](#).

A Reserved Instance billing benefit can be applied to a running instance on a per-second basis. Per-second billing is available for instances using an open-source Linux distribution, such as Amazon Linux and Ubuntu. Per-hour billing is used for commercial Linux distributions, such as Red Hat Enterprise Linux and SUSE Linux Enterprise Server.

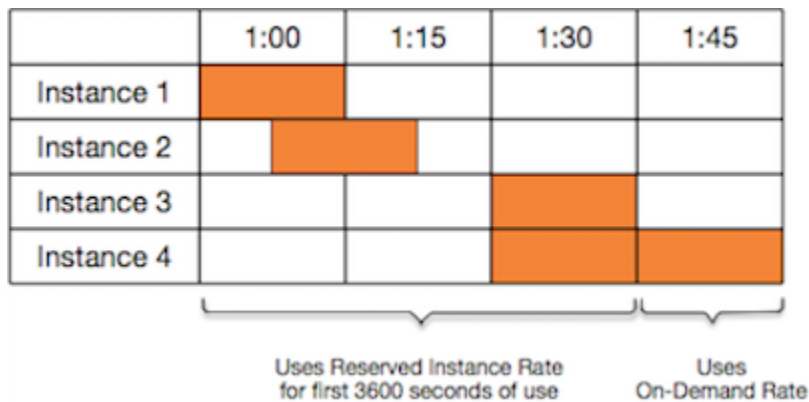
A Reserved Instance billing benefit can apply to a maximum of 3600 seconds (one hour) of instance usage per clock-hour. You can run multiple instances concurrently, but can only receive the benefit of the Reserved Instance discount for a total of 3600 seconds per clock-hour; instance usage that exceeds 3600 seconds in a clock-hour is billed at the On-Demand rate.

For example, if you purchase one `m4.xlarge` Reserved Instance and run four `m4.xlarge` instances concurrently for one hour, one instance is charged at one hour of Reserved Instance usage and the other three instances are charged at three hours of On-Demand usage.

However, if you purchase one `m4.xlarge` Reserved Instance and run four `m4.xlarge` instances for 15 minutes (900 seconds) each within the same hour, the total running time for the instances is one hour, which results in one hour of Reserved Instance usage and 0 hours of On-Demand usage.

	1:00	1:15	1:30	1:45
Instance 1				
Instance 2				
Instance 3				
Instance 4				

If multiple eligible instances are running concurrently, the Reserved Instance billing benefit is applied to all the instances at the same time up to a maximum of 3600 seconds in a clock-hour; thereafter, On-Demand rates apply.



Cost Explorer on the [Billing and Cost Management](#) console enables you to analyze the savings against running On-Demand Instances. The [Reserved Instances FAQ](#) includes an example of a list value calculation.

If you close your AWS account, On-Demand billing for your resources stops. However, if you have any Reserved Instances in your account, you continue to receive a bill for these until they expire.

Viewing your bill

You can find out about the charges and fees to your account by viewing the [AWS Billing and Cost Management](#) console.

- The **Dashboard** displays a spend summary for your account.
- On the **Bills** page, under **Details** expand the **Elastic Compute Cloud** section and the Region to get billing information about your Reserved Instances.

You can view the charges online, or you can download a CSV file.

You can also track your Reserved Instance utilization using the AWS Cost and Usage Report. For more information, see [Reserved Instances](#) under Cost and Usage Report in the *AWS Billing User Guide*.

Reserved Instances and consolidated billing

The pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account. The instance usage across all member accounts is aggregated in the payer account every month. This is typically useful for companies in which there are different functional teams or groups; then, the normal Reserved Instance logic is applied to calculate the bill. For more information, see [Consolidated billing for AWS Organizations](#).

If you close the account that purchased the Reserved Instance, the payer account is charged for the Reserved Instance until the Reserved Instance expires. After the closed account is permanently deleted in 90 days, the member accounts no longer benefit from the Reserved Instance billing discount.

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 [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 Standard, SQL Server Web, and SQL Server Enterprise.
- Pricing tier discounts only apply to purchases made from AWS. They do not apply to purchases of third-party Reserved Instances.
- Discount pricing tiers are currently not applicable to Convertible Reserved Instance purchases.

Topics

- [Calculate Reserved Instance pricing discounts](#)
- [Buy with a discount tier](#)
- [Crossing pricing tiers](#)
- [Consolidated billing for pricing tiers](#)

Calculate Reserved Instance pricing discounts

You can determine the pricing tier for your account by calculating the list value for all of your Reserved Instances in a Region. Multiply the hourly recurring price for each reservation by the total number of hours for the term and add the undiscounted upfront price (also known as the fixed price) at the time of purchase. Because the list value is based on undiscounted (public) pricing, it is not affected if you qualify for a volume discount or if the price drops after you buy your Reserved Instances.


$$\text{List value} = \text{fixed price} + (\text{undiscounted recurring hourly price} * \text{hours in term})$$

For example, for a 1-year Partial Upfront t2.small Reserved Instance, assume the upfront price is \$60.00 and the hourly rate is \$0.007. This provides a list value of \$121.32.

$$121.32 = 60.00 + (0.007 * 8760)$$


New console

To view the fixed price values for Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. To display the **Upfront price** column, choose settings () in the top-right corner, turn on **Upfront price**, and choose **Confirm**.

Old console

To view the fixed price values for Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. To display the **Upfront Price** column, choose settings () in the top-right corner, select **Upfront Price**, and choose **Close**.

To view the fixed price values for Reserved Instances using the command line

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)
- [DescribeReservedInstances](#) (Amazon EC2 API)

Buy with a discount tier

When you buy Reserved Instances, Amazon EC2 automatically applies any discounts to the part of your purchase that falls within a discount pricing tier. You don't need to do anything differently, and you can buy Reserved Instances using any of the Amazon EC2 tools. For more information, see [Buy Reserved Instances](#).

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

To purchase a Reserved Instance, search for *Reserved Instance offerings* from AWS and third-party sellers, adjusting your search parameters until you find the exact match that you're looking for.

When you search for Reserved Instances to buy, you receive a quote on the cost of the returned offerings. When you proceed with the purchase, AWS automatically places a limit price on the purchase price. The total cost of your Reserved Instances won't exceed the amount that you were quoted.

If the price rises or changes for any reason, the purchase is not completed. When you are purchasing a third-party seller's Reserved Instance from the 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.

Note

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 [Example Policies for Working With the AWS CLI or an AWS SDK](#) and [Example Policies for Working in the Amazon EC2 Console](#).

Topics

- [Choosing a platform](#)
- [Queue your purchase](#)
- [Buy Standard Reserved Instances](#)
- [Buy Convertible Reserved Instances](#)
- [Buy from the Reserved Instance Marketplace](#)
- [View your Reserved Instances](#)
- [Cancel a queued purchase](#)
- [Renew a Reserved Instance](#)

Choosing a platform

Amazon EC2 supports the following Linux platforms for Reserved Instances:

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux
- Red Hat Enterprise Linux with HA

When you purchase a Reserved Instance, you must choose an offering for a *platform* that represents the operating system for your instance.

- For SUSE Linux and RHEL distributions, you must choose offerings for those specific platforms, i.e., for the **SUSE Linux** or **Red Hat Enterprise Linux** platforms.
- For all other Linux distributions (including Ubuntu), choose an offering for the **Linux/UNIX** platform.
- If you bring your existing RHEL subscription, you must choose an offering for the **Linux/UNIX** platform, not an offering for the **Red Hat Enterprise Linux** platform.

Note

Ubuntu Pro is not available as a Reserved Instance. For significant savings compared to On-Demand Instance pricing, we recommend that you use Ubuntu Pro with Savings Plans. For more information, see the [Savings Plans User Guide](#).

Important

If you plan to purchase a Reserved Instance to apply to an On-Demand Instance that was launched from an AWS Marketplace AMI, first check the `PlatformDetails` field of the AMI. The `PlatformDetails` field indicates which Reserved Instance to purchase. The platform details of the AMI must match the platform of the Reserved Instance, otherwise the Reserved Instance will not be applied to the On-Demand Instance. For information about how to view the platform details of the AMI, see [Understand AMI billing information](#).

For information about the supported platforms for Windows, see [Choosing a platform](#) in the *Amazon EC2 User Guide for Windows Instances*.

Queue your purchase

By default, when you purchase a Reserved Instance, the purchase is made immediately. Alternatively, you can queue your purchases for a future date and time. For example, you can queue a purchase for around the time that an existing Reserved Instance expires. This can help you ensure that you have uninterrupted coverage.

You can queue purchases for regional Reserved Instances, but not zonal Reserved Instances or Reserved Instances from other sellers. You can queue a purchase up to three years in advance. On

the scheduled date and time, the purchase is made using the default payment method. After the payment is successful, the billing benefit is applied.

You can view your queued purchases in the Amazon EC2 console. The status of a queued purchase is **queued**. You can cancel a queued purchase any time before its scheduled time. For details, see [Cancel a queued purchase](#).

Buy Standard Reserved Instances

You can buy Standard Reserved Instances in a specific Availability Zone and get a capacity reservation. Alternatively, you can forego the capacity reservation and purchase a regional Standard Reserved Instance.

New console

To buy Standard Reserved Instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering class**, choose **Standard** to display Standard Reserved Instances.
4. To purchase a capacity reservation, toggle on **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. When you toggle on this setting, the **Availability Zone** field appears.

To purchase a regional Reserved Instance, toggle off this setting. When you toggle off this setting, the **Availability Zone** field disappears.

5. Select other configurations as needed, and then choose **Search**.
6. For each Reserved Instance that you want to purchase, enter the desired quantity, and choose **Add to cart**.

To purchase a Standard Reserved Instance from the Reserved Instance Marketplace, look for **3rd party** in the **Seller** column in the search results. The **Term** column displays non-standard terms. For more information, see [Buy from the Reserved Instance Marketplace](#).


7. To see a summary of the Reserved Instances that you selected, choose **View cart**.
8. If **Order on** is **Now**, the purchase is completed immediately after you choose **Order all**. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.

9. To complete the order, choose **Order all**.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose **Close**.

The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from Payment-pending to Active. When the Reserved Instance is Active, it is ready to use.

 **Note**

If the status goes to Retired, AWS might not have received your payment.

Old console

To buy Standard Reserved Instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering Class**, choose **Standard** to display Standard Reserved Instances.
4. To purchase a capacity reservation, choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. To purchase a regional Reserved Instance, leave the check box unselected.
5. Select other configurations as needed and choose **Search**.

To purchase a Standard Reserved Instance from the Reserved Instance Marketplace, look for **3rd Party** in the **Seller** column in the search results. The **Term** column displays non-standard terms.


6. For each Reserved Instance that you want to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of the Reserved Instances that you selected, choose **View Cart**.

8. If **Order On** is **Now**, the purchase is completed immediately. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.
9. To complete the order, choose **Order**.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose **Close**.

The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from `payment-pending` to `active`. When the Reserved Instance is `active`, it is ready to use.

 **Note**

If the status goes to `retired`, AWS might not have received your payment.

To buy a Standard Reserved Instance using the AWS CLI

1. Find available Reserved Instances using the [describe-reserved-instances-offerings](#) command. Specify `standard` for the `--offering-class` parameter to return only Standard Reserved Instances. You can apply additional parameters to narrow your results. For example, if you want to purchase a regional `t2.large` Reserved Instance with a default tenancy for Linux/UNIX for a 1-year term only:

```
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 \  
  --filters Name=marketplace,Values=marketplace
```

```
--instance-type t2.large \  
--offering-class standard \  
--product-description "Linux/UNIX" \  
--instance-tenancy default \  
--filters Name=marketplace,Values=true
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. Use the [purchase-reserved-instances-offering](#) command to buy your Reserved Instance. You must specify the Reserved Instance offering ID you obtained the previous step and you must specify the number of instances for the reservation.

```
aws ec2 purchase-reserved-instances-offering \  
--reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \  
--instance-count 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. Use the [describe-reserved-instances](#) command to get the status of your Reserved Instance.

```
aws ec2 describe-reserved-instances
```

Alternatively, use the following AWS Tools for Windows PowerShell commands:

- [Get-EC2ReservedInstancesOffering](#)
- [New-EC2ReservedInstance](#)
- [Get-EC2ReservedInstance](#)

After the purchase is complete, if you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have a suitable running instance, launch an instance and

ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see [Use your Reserved Instances](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances are applied](#).

Buy Convertible Reserved Instances

You can buy Convertible Reserved Instances in a specific Availability Zone and get a capacity reservation. Alternatively, you can forego the capacity reservation and purchase a regional Convertible Reserved Instance.

New console

To buy Convertible Reserved Instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering class**, choose **Convertible** to display Convertible Reserved Instances.
4. To purchase a capacity reservation, toggle on **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. When you toggle on this setting, the **Availability Zone** field appears.

To purchase a regional Reserved Instance, toggle off this setting. When you toggle off this setting, the **Availability Zone** field disappears.

5. Select other configurations as needed and choose **Search**.
6. For each Convertible Reserved Instance that you want to purchase, enter the quantity, and choose **Add to cart**.
7. To see a summary of your selection, choose **View cart**.
8. If **Order on** is **Now**, the purchase is completed immediately after you choose **Order all**. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.
9. To complete the order, choose **Order all**.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose **Close**.

The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from **Payment-pending** to **Active**. When the Reserved Instance is **Active**, it is ready to use.

Note

If the status goes to **Retired**, AWS might not have received your payment.

Old console


To buy **Convertible Reserved Instances** using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering Class**, choose **Convertible** to display Convertible Reserved Instances.
4. To purchase a capacity reservation, choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. To purchase a regional Reserved Instance, leave the check box unselected.
5. Select other configurations as needed and choose **Search**.
6. For each Convertible Reserved Instance that you want to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of your selection, choose **View Cart**.
8. If **Order On** is **Now**, the purchase is completed immediately. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.
9. To complete the order, choose **Order**.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose **Close**.

The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from `payment-pending` to `active`. When the Reserved Instance is active, it is ready to use.

 **Note**

If the status goes to `retired`, AWS might not have received your payment.

To buy a Convertible Reserved Instance using the AWS CLI

1. Find available Reserved Instances using the [describe-reserved-instances-offerings](#) command. Specify `convertible` for the `--offering-class` parameter to return only Convertible Reserved Instances. You can apply additional parameters to narrow your results; for example, if you want to purchase a regional `t2.large` Reserved Instance with a default tenancy for Linux/UNIX:

```
aws ec2 describe-reserved-instances-offerings \  
  --instance-type t2.large \  
  --offering-class convertible \  
  --product-description "Linux/UNIX" \  
  --instance-tenancy default \  
  --filters Name=scope,Values=Region
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. Use the [purchase-reserved-instances-offering](#) command to buy your Reserved Instance. You must specify the Reserved Instance offering ID you obtained the previous step and you must specify the number of instances for the reservation.

```
aws ec2 purchase-reserved-instances-offering \  
  --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \  
  --instance-count 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. Use the [describe-reserved-instances](#) command to get the status of your Reserved Instance.

```
aws ec2 describe-reserved-instances
```

Alternatively, use the following AWS Tools for Windows PowerShell commands:

- [Get-EC2ReservedInstancesOffering](#)
- [New-EC2ReservedInstance](#)
- [Get-EC2ReservedInstance](#)

If you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have a suitable running instance, launch an instance and ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see [Use your Reserved Instances](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances are applied](#).

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 [AWS Support](#).

View your Reserved Instances

You can view the Reserved Instances you've purchased using the Amazon EC2 console, or a command line tool.

To view your Reserved Instances in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Your queued, active, and retired Reserved Instances are listed. The **State** column displays the state.
4. If you are a seller in the Reserved Instance Marketplace, the **My Listings** tab displays the status of a reservation that's listed in the [Reserved Instance Marketplace](#). For more information, see [Reserved Instance listing states](#).

To view your Reserved Instances using the command line

- [describe-reserved-instances](#) (AWS CLI)

- [Get-EC2ReservedInstance](#) (Tools for Windows PowerShell)

Cancel a queued purchase

You can queue a purchase up to three years in advance. You can cancel a queued purchase any time before its scheduled time.

New console

To cancel a queued purchase

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select one or more Reserved Instances.
4. Choose **Actions, Delete queued Reserved Instances**.
5. When prompted for confirmation, choose **Delete**, and then **Close**.

Old console

To cancel a queued purchase

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select one or more Reserved Instances.
4. Choose **Actions, Delete Queued Reserved Instances**.
5. When prompted for confirmation, choose **Yes, Delete**.

To cancel a queued purchase using the command line

- [delete-queued-reserved-instances](#) (AWS CLI)
- [Remove-EC2QueuedReservedInstance](#) (Tools for Windows PowerShell)

Renew a Reserved Instance

You can renew a Reserved Instance before it is scheduled to expire. Renewing a Reserved Instance queues the purchase of a Reserved Instance with the same configuration until the current Reserved Instance expires.

New console

To renew 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**.

Old 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**.

Sell in the Reserved Instance Marketplace

The Reserved Instance Marketplace is a platform that supports the sale of third-party and AWS customers' unused Standard Reserved Instances, which vary in term lengths and pricing options. For example, you might want to sell Reserved Instances after moving instances to a new AWS Region, changing to a new instance type, ending projects before the term expiration, when your business needs change, or if you have unneeded capacity.

As soon as you list your Reserved Instances in the Reserved Instance Marketplace, they are available for potential buyers to find. All Reserved Instances are grouped according to the duration of the term remaining and the hourly price.

To fulfill a buyer's request to purchase a third-party seller's Reserved Instance via the EC2 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](#)
- [Register as a seller](#)
- [Bank account for disbursement](#)
- [Tax information](#)
- [Price your Reserved Instances](#)
- [List your Reserved Instances](#)
- [Reserved Instance listing states](#)
- [Lifecycle of a listing](#)
- [After your Reserved Instance is sold](#)
- [Getting paid](#)
- [Information shared with the buyer](#)

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 cannot 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 cannot sell a Standard Reserved Instance in a Region that is [disabled by default](#).
- The minimum price allowed in the Reserved Instance Marketplace is \$0.00.
- You can sell No Upfront, Partial Upfront, or All Upfront Reserved Instances in the Reserved Instance Marketplace 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 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 Internet Services Private Limited (AISPL) customers can't sell Reserved Instances in the Reserved Instance Marketplace even if they have a US bank account. For more information, see [What are the differences between AWS accounts and AISPL accounts?](#)

Register as a seller

Note

Only the AWS account root user can register an account as a seller.

To sell in the Reserved Instance Marketplace, you must first register as a seller. During registration, you provide the following information:

- **Bank information**—AWS must have your bank information in order to disburse funds collected when you sell your reservations. The bank you specify must have a US address. For more information, see [Bank account for disbursement](#).
- **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 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 a 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.
 - **You can sell up to \$50,000 in Reserved Instances.**
 - **You can sell up to 5,000 Reserved Instances.**

These limits typically can't be increased, but will be evaluated on a case-by-case basis if requested. To request a limit increase, complete the [Service limit increase](#) form. For **Limit type**, choose **EC2 Reserved Instance Sales**.

- **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.

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select the Reserved Instances to list, and choose **Actions, Sell Reserved Instances**.
4. On the **Configure Your Reserved Instance Listing** page, set the number of instances to sell and the upfront price for the remaining term in the relevant columns. See how the value of your reservation changes over the remainder of the term by selecting the arrow next to the **Months Remaining** column.
5. If you are an advanced user and you want to customize the pricing, you can enter different values for the subsequent months. To return to the default linear price drop, choose **Reset**.
6. Choose **Continue** when you are finished configuring your listing.
7. Confirm the details of your listing, on the **Confirm Your Reserved Instance Listing** page and if you're satisfied, choose **List Reserved Instance**.

To view your listings in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select the Reserved Instance that you've listed and choose the **My Listings** tab near the bottom of the page.

To manage Reserved Instances in the Reserved Instance Marketplace using the AWS CLI

1. Get a list of your Reserved Instances by using the [describe-reserved-instances](#) command.
2. Note the ID of the Reserved Instance you want to list and call [create-reserved-instances-listing](#). You must specify the ID of the Reserved Instance, the number of instances, and the pricing schedule.
3. To view your listing, use the [describe-reserved-instances-listings](#) command.
4. To cancel your listing, use the [cancel-reserved-instances-listings](#) command.

Reserved Instance listing states

Listing State on the **My Listings** tab of the Reserved Instances page displays the current status of your listings:

The information displayed by **Listing State** is about the status of your listing in the Reserved Instance Marketplace. It is different from the status information that is displayed by the **State** column in the **Reserved Instances** page. This **State** information is about your reservation.

- **active**—The listing is available for purchase.
- **canceled**—The listing is canceled and isn't available for purchase in the Reserved Instance Marketplace.
- **closed**—The Reserved Instance is not listed. A Reserved Instance might be closed because the sale of the listing was completed.

Lifecycle of a listing

When all the instances in your listing are matched and sold, the **My Listings** tab shows that the **Total instance count** matches the count listed under **Sold**. Also, there are no **Available** instances left for your listing, and its **Status** is closed.

When only a portion of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the listing ID and the listing that it represents, which now has fewer reservations for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as closed.

For example, you create a listing *Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample* with a listing count of 5.

The **My Listings** tab in the **Reserved Instance** console page displays the listing this way:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 0
- Available = 5
- Status = active

A buyer purchases two of the reservations, which leaves a count of three reservations still available for sale. Because of this partial sale, AWS creates a new reservation with a count of three to represent the remaining reservations that are still for sale.

This is how your listing looks in the **My Listings** tab:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 2
- Available = 3
- Status = active

If you cancel your listing and a portion of that listing has already sold, the cancelation is not effective on the portion that has been sold. Only the unsold portion of the listing is no longer available in the Reserved Instance Marketplace.

After your Reserved Instance is sold

When your Reserved Instance is sold, AWS sends you an email notification. Each day that there is any kind of activity, you receive one email notification capturing all the activities of the day. Activities can include when you create or sell a listing, or when AWS sends funds to your account.

To track the status of a Reserved Instance listing in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation page, choose **Reserved Instances**.

3. Choose the **My Listings** tab.

The **My Listings** tab contains the **Listing State** value. It also contains information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and canceled.

You can also use the [describe-reserved-instances-listings](#) command with the appropriate filter to obtain information about your listings.

Getting paid

As soon as AWS receives funds from the buyer, a message is sent to the registered owner account email for the sold Reserved Instance.

AWS sends an Automated Clearing House (ACH) wire transfer to your specified bank account. Typically, this transfer occurs between one to three days after your Reserved Instance has been sold. Disbursements take place once a day. You will receive an email with a disbursement report after the funds are released. Keep in mind that you can't receive disbursements until AWS receives verification from your bank. This can take up to two weeks.

The Reserved Instance that you sold continues to appear when you describe your Reserved Instances.

You receive a cash disbursement for your Reserved Instances through a wire transfer directly into your bank account. AWS charges a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the Reserved Instance Marketplace.

Information shared with the buyer

When you sell in the Reserved Instance Marketplace, AWS shares your company's legal name on the buyer's statement in accordance with US regulations. In addition, if the buyer calls AWS Support because the buyer needs to contact you for an invoice or for some other tax-related reason, AWS might need to provide the buyer with your email address so that the buyer can contact you directly.

For similar reasons, the buyer's ZIP code and country information are provided to the seller in the disbursement report. As a seller, you might need this information to accompany any necessary transaction taxes that you remit to the government (such as sales tax and value-added tax).

AWS cannot offer tax advice, but if your tax specialist determines that you need specific additional information, [contact AWS Support](#).

Modify Reserved Instances

When your needs change, you can modify your Standard or Convertible Reserved Instances and continue to benefit from the billing benefit. You can modify attributes such as the Availability Zone, 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

- [Requirements and restrictions for modification](#)
- [Support for modifying instance sizes](#)
- [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

Modifiable attribute	Supported platforms	Limitations and considerations
		<p>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.</p> <p>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 words, from regional to zonal), you lose Availability Zone flexibility and instance size flexibility (if applicable).</p> <p>For more information, see How Reserved Instances are applied.</p>

Modifiable attribute	Supported platforms	Limitations and considerations
<p>Change the instance size within the same instance family and generation</p>	<p>Linux/UNIX only</p> <p>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 Server Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web.</p>	<p>The reservation must use default tenancy. Some instance families are not supported, because there are no other sizes available. For more information, see Support for modifying instance sizes.</p>

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

- The Reserved Instance is not a G4, G4ad, G4dn, G5, G5g, Inf1, or Inf2 instance.

Support for modifying instance sizes

You can modify the instance size of a Reserved Instance if the following requirements are met.

Requirements

- The platform is Linux/UNIX.
- You must select another instance size in the same [instance family](#) (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't modify the instance size of Reserved Instances for the following instances, because each of the following has only one size:
 - `t1.micro`
- You can't modify the instance size of Reserved Instances for the following combinations of instance family, generation, and attribute:
 - G4ad
 - G4dn
 - G5
 - G5g
 - Inf1
 - Inf2
- The original and new Reserved Instance must have the same instance size footprint.

Contents

- [Instance size footprint](#)
- [Normalization factors for bare metal instances](#)

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.

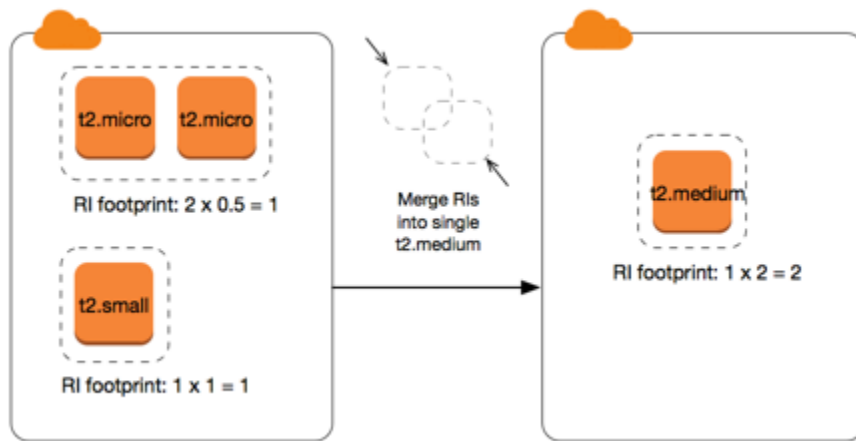
To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. In the Amazon EC2 console, the normalization factor is measured in units. The following table describes the normalization factor for the instance sizes in an instance family. 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

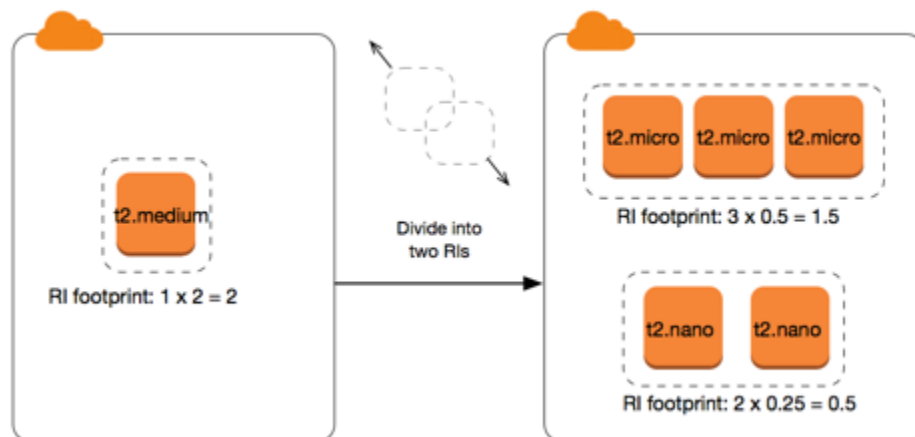
Instance size	Normalization factor
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
112xlarge	896

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 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
u-*.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.

New console

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Actions, Modify Reserved Instances**.

Note

If your Reserved Instances are not in the active state or cannot be modified, **Modify Reserved Instances** is disabled.

3. The first entry in the modification table displays attributes of the selected Reserved Instances, and at least one target configuration beneath it. The **Units** column displays the total instance size footprint. Choose **Add** for each new configuration to add. Modify the attributes as needed for each configuration.
 - **Scope:** Choose whether the configuration applies to an Availability Zone or to the whole Region.
 - **Availability Zone:** Choose the required Availability Zone. Not applicable for regional Reserved Instances.
 - **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

Old console

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Actions, Modify Reserved Instances**.

Note

If your Reserved Instances are not in the active state or cannot be modified, **Modify Reserved Instances** is disabled.

3. The first entry in the modification table displays attributes of selected Reserved Instances, and at least one target configuration beneath it. The **Units** column displays the total instance size footprint. Choose **Add** for each new configuration to add. Modify the attributes as needed for each configuration, and then choose **Continue**:
 - **Scope**: Choose whether the configuration applies to an Availability Zone or to the whole Region.
 - **Availability Zone**: Choose the required Availability Zone. Not applicable for regional Reserved Instances.
 - **Instance Type**: Select the required instance type. The combined configurations must equal the instance size footprint of your original configurations.
 - **Count**: Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose **Add**, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can

change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.

4. To confirm your modification choices when you finish specifying your target configurations, choose **Submit Modifications**.
5. You can determine the status of your modification request by looking at the **State** column in the Reserved Instances screen. The following are the possible states.
 - **active (*pending modification*)** — Transition state for original Reserved Instances
 - **retired (*pending modification*)** — Transition state for original Reserved Instances while new Reserved Instances are being created
 - **retired** — Reserved Instances successfully modified and replaced
 - **active** — One of the following:
 - New Reserved Instances created from a successful modification request
 - Original Reserved Instances after a failed modification request

To modify your Reserved Instances using the command line

1. To modify your Reserved Instances, you can use one of the following commands:
 - [modify-reserved-instances](#) (AWS CLI)
 - [Edit-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)
2. To get the status of your modification request (processing, fulfilled, or failed), use one of the following commands:
 - [describe-reserved-instances-modifications](#) (AWS CLI)
 - [Get-EC2ReservedInstancesModification](#) (AWS Tools for Windows PowerShell)

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](#)

- [Calculate Convertible Reserved Instances exchanges](#)
- [Merge Convertible Reserved Instances](#)
- [Exchange a portion of a Convertible Reserved Instance](#)
- [Submit exchange requests](#)

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:

$$3.5 = 3 \text{ whole Convertible Reserved Instances} + 1 \text{ additional Convertible Reserved Instance}$$

The fourth Convertible Reserved Instance has the same end date as the other three. If you are exchanging Partial or All Upfront Convertible Reserved Instances, you pay the true-up cost for the fourth reservation. If the remaining upfront cost of your Convertible Reserved Instances is \$500, and the new reservation would normally cost \$600 on a prorated basis, you are charged \$100.

$$\$600 \text{ prorated upfront cost of new reservations} - \$500 \text{ remaining upfront cost of old reservations} = \$100 \text{ difference}$$

Merge Convertible Reserved Instances

If you merge two or more Convertible Reserved Instances, the term of the new Convertible Reserved Instance must be the same as the 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

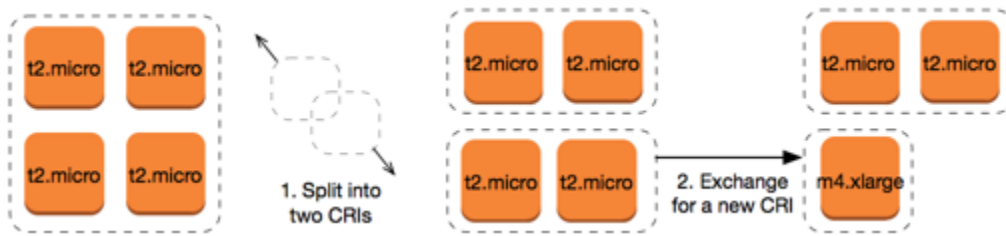
You can use the modification process to split your Convertible Reserved Instance into smaller reservations, and then exchange one or more of the new reservations for a new Convertible Reserved Instance. The following examples demonstrate how you can do this.

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 using the Amazon EC2 console or a command line tool.

Exchange a Convertible Reserved Instance using the console

You can search for Convertible Reserved Instances offerings and select your new configuration from the choices provided.

New console

To exchange Convertible Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Reserved Instances**, select the Convertible Reserved Instances to exchange, and choose **Actions, Exchange Reserved Instance**.
3. Select the attributes of the desired configuration, and choose **Find offering**.
4. Select a new Convertible Reserved Instance. At the bottom of the screen, you can view the number of Reserved Instances that you receive for the exchange, and any additional costs.
5. When you have selected a Convertible Reserved Instance that meets your needs, choose **Review**.
6. Choose **Exchange**, and then **Close**.

Old console

To exchange Convertible Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Reserved Instances**, select the Convertible Reserved Instances to exchange, and choose **Actions, Exchange Reserved Instance**.
3. Select the attributes of the desired configuration, and choose **Find Offering**.
4. Select a new Convertible Reserved Instance. The **Instance Count** column displays the number of Reserved Instances that you receive for the exchange. When you have selected a Convertible Reserved Instance that meets your needs, choose **Exchange**.

The Reserved Instances that were exchanged are retired, and the new Reserved Instances are displayed in the Amazon EC2 console. This process can take a few minutes to propagate.

Exchange a Convertible Reserved Instance using the command line interface

To exchange a Convertible Reserved Instance, first find a new Convertible Reserved Instance that meets your needs:

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (Tools for Windows PowerShell)

Get a quote for the exchange, which includes the number of Reserved Instances you get from the exchange, and the true-up cost for the exchange:

- [get-reserved-instances-exchange-quote](#) (AWS CLI)
- [GetEC2-ReservedInstancesExchangeQuote](#) (Tools for Windows PowerShell)

Finally, perform the exchange:

- [accept-reserved-instances-exchange-quote](#) (AWS CLI)
- [Confirm-EC2ReservedInstancesExchangeQuote](#) (Tools for Windows PowerShell)

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)

Quotas apply to running instances only. If your instance is pending, stopping, stopped, or hibernated, it does 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.

The screenshot displays the Amazon EC2 console interface for Reserved Instances. At the top, it shows 'Reserved Instances (32)' with a search bar and a 'Purchase Reserved Instances' button. Below this is a table with columns: Instance type, Scope, Availability, Instance count, Start, Expires, and Offering class. Two rows are visible, both for 't3.micro' instances. The first row is selected, and its 'Instance count' value of '10' is highlighted with a red box. The second row has an 'Instance count' of '4'. Below the table, a summary bar indicates '1 Reserved Instance selected'. The 'Details' view is expanded, showing various attributes for the selected instance. The 'Instance count' field in this detail view is also highlighted with a red box and shows a value of '10'.

Instance type	Scope	Availability	Instance count	Start	Expires	Offering cl...
t3.micro	Region	-	10	August 27, 2022, 15:29 (UTC+2:00)	August 27, 2023, 15:29 (UTC+2:00)	Standard
t3.micro	Region	-	4	November 8, 2021, 14:19 (UTC+2:00)	November 8, 2022, 14:19 (UTC+2:00)	Standard

Reserved Instance ID: 2fbf16dd-98b6-4a3a-955f-83f87790f04b			
Instance type t3.micro	Scope Region	Instance count 10	Availability Zone -
Start August 27, 2022, 15:29 (UTC+2:00)	Platform Linux/UNIX	Expires August 27, 2023, 15:29 (UTC+2:00)	Term 1 year
Payment option All upfront	Time left around 50 weeks 6 days	Upfront price \$59.00	Offering class Standard
Usage price \$0.00	State Active	Hourly charges \$0.00	Tenancy Default

AWS CLI

To view the number of Reserved Instances you have purchased

Use the [describe-reserved-instances](#) CLI command and specify the ID of the Reserved Instance configuration.

```
aws ec2 describe-reserved-instances \
  --reserved-instances-ids 2fbf16dd-98b6-4a3a-955f-83f87790f04b \
  --output table
```

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 |  2fbf16dd-98b6-4a3a-955f-83f87790f04b |  ||
||  Scope             |  Region   |  ||
||  Start             |  2022-08-27T13:29:45.938000+00:00 |  ||
||  State             |  active   |  ||
||  UsagePrice        |  0.0     |  ||
|+-----+-----+-----+-----+-----+-----+|
|||                               RecurringCharges                               ||| | | |
||+-----+-----+-----+-----+-----+-----+||
|||  Amount            |           |  0.0 |  |||
|||  Frequency         |           |  Hourly |  |||
||+-----+-----+-----+-----+-----+-----+||
```

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 [Instance purchasing options](#).

Topics

- [Concepts](#)
- [How to get started](#)
- [Related services](#)
- [Pricing and savings](#)

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.

Key differences between Spot Instances and On-Demand Instances

The following table lists the key differences between Spot Instances and [On-Demand Instances](#).

	Spot Instances	On-Demand Instances
Launch time	Can only be launched immediately if the Spot 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).

How to get started

The first thing that you need to do is get set up to use Amazon EC2. It can also be helpful to have experience launching On-Demand Instances before launching Spot Instances.

Get up and running

- [Set up to use Amazon EC2](#)
- [Tutorial: Get started with Amazon EC2 Linux instances](#)

Spot basics

- [How Spot Instances work](#)

Working with Spot Instances

- [Create a Spot Instance request](#)
- [Get request status information](#)
- [Spot Instance interruptions](#)

Related services

You can provision Spot Instances directly using Amazon EC2. You can also provision Spot Instances using other services in AWS. For more information, see the following documentation.

Amazon EC2 Auto Scaling and Spot Instances

You can create launch templates or configurations so that Amazon EC2 Auto Scaling can launch Spot Instances. For more information, see [Requesting Spot Instances for fault-tolerant and flexible applications](#) and [Auto Scaling groups with multiple instance types and purchase options](#) in the *Amazon EC2 Auto Scaling User Guide*.

Amazon EMR and Spot Instances

There are scenarios where it can be useful to run Spot Instances in an Amazon EMR cluster. For more information, see [Spot Instances](#) and [When Should You Use Spot Instances](#) in the *Amazon EMR Management Guide*.

AWS CloudFormation templates

AWS CloudFormation enables you to create and manage a collection of AWS resources using a template in JSON format. For more information, see [EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration](#).

AWS SDK for Java

You can use the Java programming language to manage your Spot Instances. For more information, see [Tutorial: Amazon EC2 Spot Instances](#) and [Tutorial: Advanced Amazon EC2 Spot Request Management](#).

AWS SDK for .NET

You can use the .NET programming environment to manage your Spot Instances. For more information, see [Tutorial: Amazon EC2 Spot Instances](#).

Pricing and savings

You pay the Spot price for Spot Instances, which is set by Amazon EC2 and adjusted gradually based on the long-term supply of and demand for Spot Instances. 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](#).

View prices

To view the current (updated every five minutes) lowest Spot price per AWS Region and instance type, see the [Amazon EC2 Spot Instances Pricing](#) page.

To view the Spot price history for the past three months, use the Amazon EC2 console or the [describe-spot-price-history](#) command (AWS CLI). For more information, see [Spot Instance pricing history](#).

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 EC2 Spot

Amazon EC2 Spot Instances are spare EC2 compute capacity in the AWS Cloud that are available to you at savings of up to 90% off compared to On-Demand prices. The only difference between On-Demand Instances and Spot Instances is that Spot Instances can be interrupted by Amazon EC2, with two minutes of notification, when Amazon EC2 needs the capacity back.

Spot Instances are recommended for stateless, fault-tolerant, flexible applications. For example, Spot Instances work well for big data, containerized workloads, CI/CD, stateless web servers, high performance computing (HPC), and rendering workloads.

While running, Spot Instances are exactly the same as On-Demand Instances. However, Spot does not guarantee that you can keep your running instances long enough to finish your workloads. Spot also does not guarantee that you can get immediate availability of the instances that you are looking for, or that you can always get the aggregate capacity that you requested. Moreover, Spot Instance interruptions and capacity can change over time because Spot Instance availability varies based on supply and demand, and past performance isn't a guarantee of future results.

Spot Instances are not suitable for workloads that are inflexible, stateful, fault-intolerant, or tightly coupled between instance nodes. They're also not recommended for workloads that are intolerant of occasional periods when the target capacity is not completely available. We strongly warn against using Spot Instances for these workloads or attempting to fail-over to On-Demand Instances to handle interruptions.

Regardless of whether you're an experienced Spot user or new to Spot Instances, if you are currently experiencing issues with Spot Instance interruptions or availability, we recommend that you follow these best practices to have the best experience using the Spot service.

Spot best practices

- [Prepare individual instances for interruptions](#)
- [Be flexible about instance types 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 proactive capacity rebalancing](#)
- [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. For more information, see [Use proactive capacity rebalancing](#).

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 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 about allocation strategies, see [Spot Instances](#) in the *Amazon EC2 Auto Scaling User Guide* and [When workloads have a high cost of interruption](#) in this user guide.

Use proactive capacity rebalancing

Capacity Rebalancing helps you maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running Spot Instance receives the two-minute Spot Instance interruption notice. When Capacity Rebalancing is enabled, Auto Scaling or EC2 Fleet attempts to proactively replace Spot Instances that have received a rebalance recommendation, providing the opportunity to rebalance your workload to new Spot Instances that are not at elevated risk of interruption.

Capacity Rebalancing complements the price-capacity-optimized allocation strategy (which is designed to help find the most optimal spare capacity) and the mixed instances policy (which is designed to enhance availability by deploying instances across multiple instance types running in multiple Availability Zones).

For more information, see [Capacity Rebalancing](#).

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, AWS Elastic Beanstalk, and Amazon GameLift. 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	<ul style="list-style-type: none"> You need multiple instances with either a single configuration or a mixed configuration. 	Create an Auto Scaling group that manages the lifecycle of your instances while maintaining the desired number of instances. Supports horizontal	Yes

API	When to use?	Use case	Should I use this API?
	<p>You want to automate the lifecycle management through a configurable API.</p>	<p>scaling (adding more instances) between specified minimum and maximum limits.</p>	
<p>CreateFleet</p>	<ul style="list-style-type: none"> 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 fleet. 	<p>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-optimized, capacity-optimized, or diversified.</p>	<p>Yes – in instant mode if you don't need auto scaling</p>

API	When to use?	Use case	Should I use this API?
RunInstances	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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](#) AWS CLI command. For more information, see [Create a Spot Instance request](#).
- 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 request](#).

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*).

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](#).

You can stop, start, reboot, or terminate an Amazon EBS-backed Spot Instance. The Spot service can stop, terminate, or hibernate a Spot Instance when it interrupts it.

Contents

- [Launch Spot Instances in a launch group](#)
- [Launch Spot Instances in an Availability Zone group](#)

- [Launch Spot Instances in a VPC](#)

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.

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.

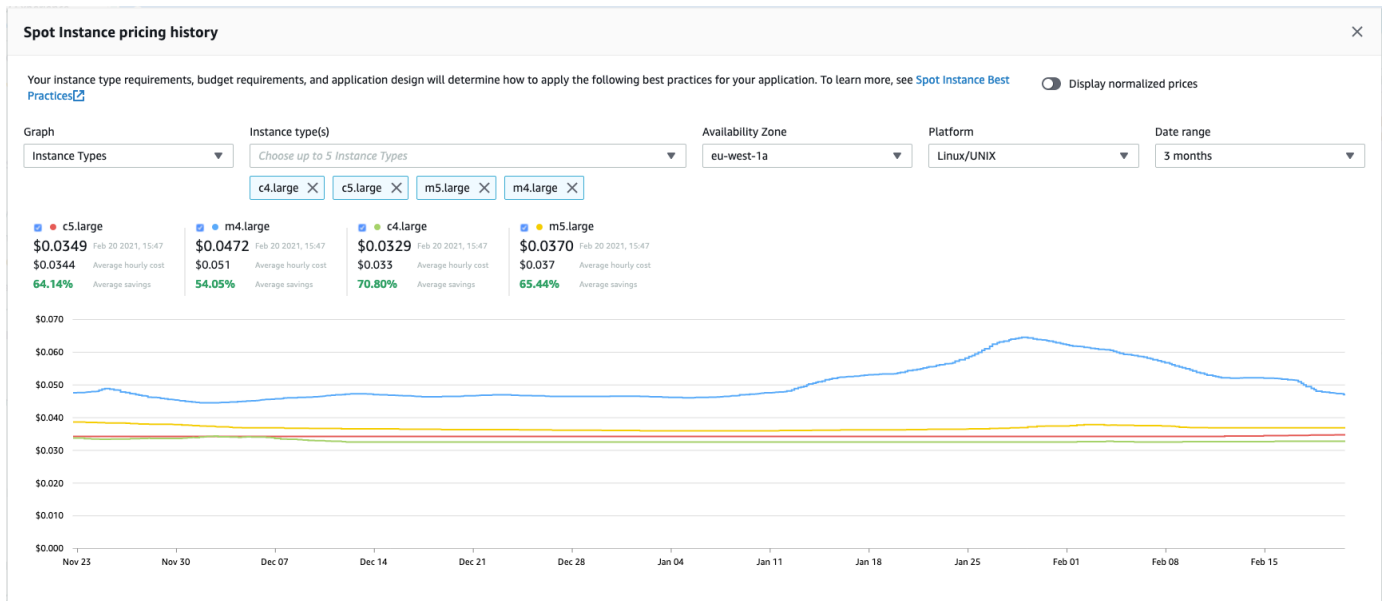
To view the current Spot prices

For the *current* Spot Instance prices, see [Amazon EC2 Spot Instances Pricing](#).

To view the Spot price history (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Choose **Pricing history**.
4. For **Graph**, choose to compare the price history by **Availability Zones** or by **Instance Types**.
 - If you choose **Availability Zones**, then choose the **Instance type**, operating system (**Platform**), and **Date range** for which to view the price history.
 - If you choose **Instance Types**, then choose up to five **Instance type(s)**, the **Availability Zone**, operating system (**Platform**), and **Date range** for which to view the price history.

The following screenshot shows a price comparison for different instance types.



5. 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**.

To view the Spot price history using the command line

You can use one of the following commands. For more information, see [Access Amazon EC2](#).

- [describe-spot-price-history](#) (AWS CLI)
- [Get-EC2SpotPriceHistory](#) (AWS Tools for Windows PowerShell)

Savings from purchasing Spot Instances

You can view the usage and savings information for Spot Instances at the per-fleet level, or for all running Spot Instances. At the per-fleet level, the usage and savings information includes all instances launched and terminated by the fleet. You can view this information from the last hour or the last three days.

The following screenshot from the **Savings** section shows the Spot usage and savings information for a Spot Fleet.

Spot usage and savings

4	266	700	\$9.55	\$2.99	69%
Spot Instances	vCPU-hours	Mem(GiB)-hours	On-Demand total	Spot total	Savings
				\$0.0112	\$0.0043
				Average cost per vCPU-hour	Average cost per mem(GiB)-hour

Details

Instance type	vCPU hours	Mem(GiB)-hours	On-Demand total	Savings
t3.medium (1)	2 vCPU hours	4 mem(GiB)-hours	\$0.01 total	70% savings
m4.large (1)	144 vCPU hours	576 mem(GiB)-hours	\$2.52 total	68% savings
t2.micro (2)	120 vCPU hours	120 mem(GiB)-hours	\$0.46 total	70% savings

You can view the following usage and savings information:

- **Spot Instances** – The number of Spot Instances launched and terminated by the Spot Fleet. When viewing the savings summary, the number represents all your running Spot Instances.
- **vCPU-hours** – The number of vCPU hours used across all the Spot Instances for the selected time frame.
- **Mem(GiB)-hours** – The number of GiB hours used across all the Spot Instances for the selected time frame.
- **On-Demand total** – The total amount you would've paid for the selected time frame had you launched these instances as On-Demand Instances.
- **Spot total** – The total amount to pay for the selected time frame.
- **Savings** – The percentage that you are saving by not paying the On-Demand price.
- **Average cost per vCPU-hour** – The average hourly cost of using the vCPUs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per vCPU-hour = Spot total / vCPU-hours.**
- **Average cost per mem(GiB)-hour** – The average hourly cost of using the GiBs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per mem(GiB)-hour = Spot total / Mem(GiB)-hours.**

- **Details** table – The different instance types (the number of instances per instance type is in parentheses) that comprise the Spot Fleet. When viewing the savings summary, these comprise all your running Spot Instances.

Savings information can only be viewed using the Amazon EC2 console.

To view the savings information for a Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. Select the ID of a Spot Fleet request and scroll to the **Savings** section.

Alternatively, select the check box next to the Spot Fleet request ID and choose the **Savings** tab.

4. By default, the page displays usage and savings information for the last three days. You can choose **last hour** or the **last three days**. For Spot Fleets that were launched less than an hour ago, the page shows the estimated savings for the hour.

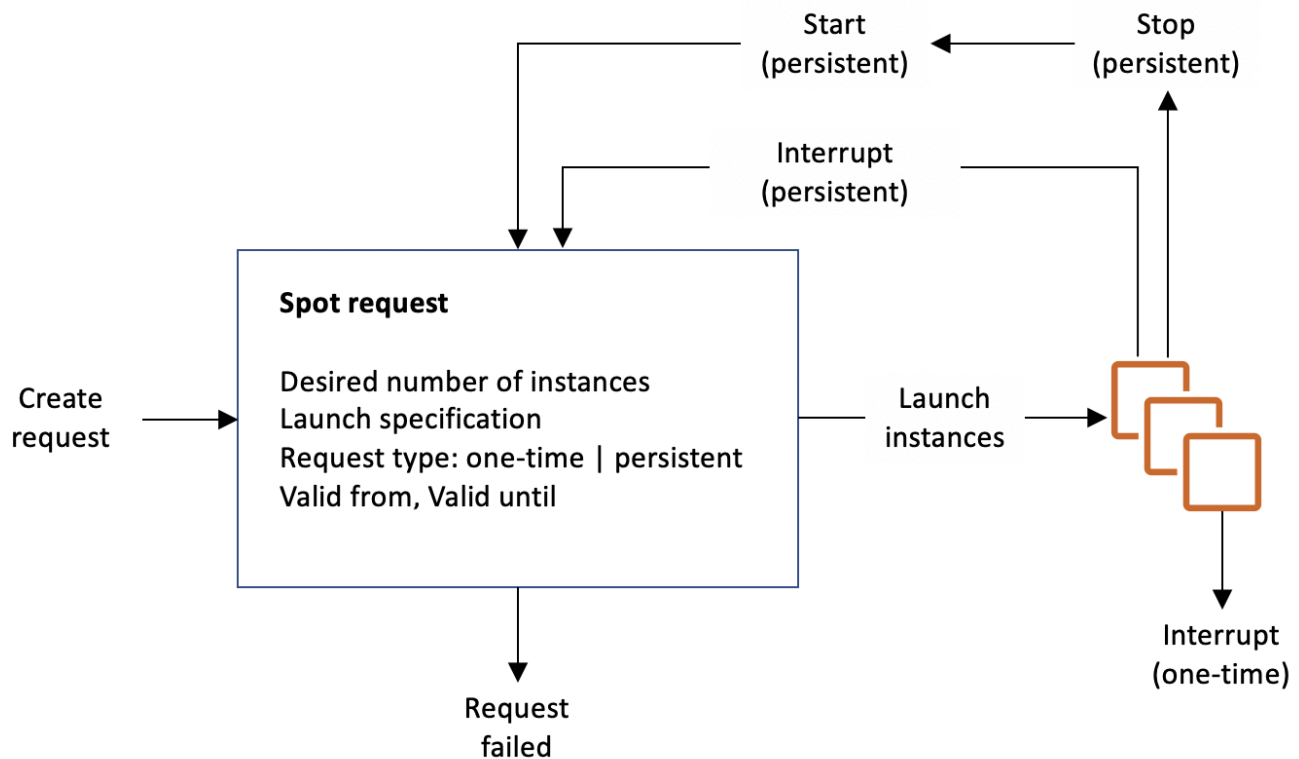
To view the savings information for all running Spot Instances (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. Choose **Savings summary**.

Work with Spot Instances

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.

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

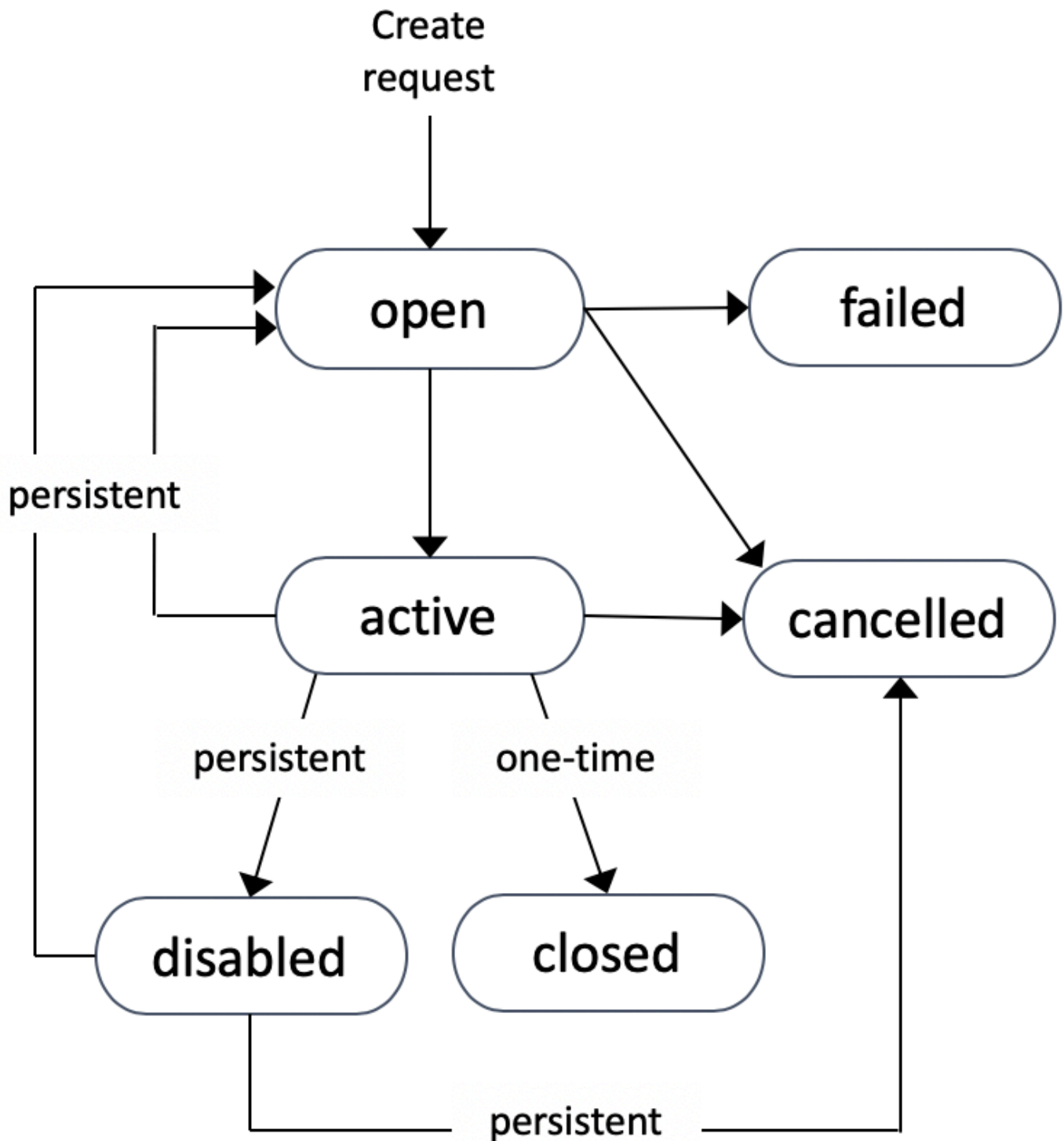
- [Spot Instance request states](#)
- [Specify a tenancy for your Spot Instances](#)
- [Service-linked role for Spot Instance requests](#)
- [Create a Spot Instance request](#)
- [Find running Spot Instances](#)
- [Tag Spot Instance requests](#)
- [Cancel a Spot Instance request](#)
- [Stop a Spot Instance](#)
- [Start a Spot Instance](#)
- [Terminate a Spot Instance](#)
- [Spot Instance request example launch specifications](#)

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 [Spot request status](#).

Specify a tenancy for your Spot Instances

You can run a Spot Instance on single-tenant hardware. Dedicated Spot Instances are physically isolated from instances that belong to other AWS accounts. For more information, see [Dedicated Instances](#) and the [Amazon EC2 Dedicated Instances](#) product page.

To run a Dedicated Spot Instance, do one of the following:

- Specify a tenancy of `dedicated` when you create the Spot Instance request. For more information, see [Create a Spot Instance request](#).
- Request a Spot Instance in a VPC with an instance tenancy of `dedicated`. For more information, see [Create a VPC with a dedicated instance tenancy](#). You cannot request a Spot Instance with a tenancy of `default` if you request it in a VPC with an instance tenancy of `dedicated`.

All instance families support Dedicated Spot Instances except T instances. For each supported instance family, only the largest instance size or metal size supports Dedicated Spot Instances.

Service-linked role for Spot Instance requests

Amazon EC2 uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service. Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2Spot** to launch and manage Spot Instances on your behalf.

Permissions granted by `AWSServiceRoleForEC2Spot`

Amazon EC2 uses `AWSServiceRoleForEC2Spot` to complete the following actions:

- `ec2:DescribeInstances` – Describe Spot Instances
- `ec2:StopInstances` – Stop Spot Instances
- `ec2:StartInstances` – Start Spot Instances

Create the service-linked role

Under most circumstances, you don't need to manually create a service-linked role. Amazon EC2 creates the `AWSServiceRoleForEC2Spot` service-linked role the first time you request a Spot Instance using the console.

If you had an active Spot Instance request before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the `AWSServiceRoleForEC2Spot` role in your AWS account. For more information, see [A New Role Appeared in My Account](#) in the *IAM User Guide*.

If you use the AWS CLI or an API to request a Spot Instance, you must first ensure that this role exists.

To create `AWSServiceRoleForEC2Spot` using the console

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. On the **Select type of trusted entity** page, choose **EC2, EC2 - Spot Instances, Next: Permissions**.
5. On the next page, choose **Next:Review**.
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"
```

Create a Spot Instance request

You can use the [launch instance wizard](#) in the Amazon EC2 console or the [run-instances](#) AWS CLI 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 [Spot request status](#).

New console

To create a Spot Instance request using the launch instance wizard

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** check box.
 - 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**.

- **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 [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 Amazon 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 [Interruption behavior](#).

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 instance launch issues](#).

Old console

To create a Spot Instance request 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, select a Region.
3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI. For more information, see [Step 1: Choose an Amazon Machine Image \(AMI\)](#).
5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch, and then choose Next: Configure Instance Details. For more information, see [Step 2: Choose an Instance Type](#).
6. On the **Configure Instance Details** page, configure the Spot Instance request as follows:
 - **Number of instances:** Enter the number of instances to launch.

Note

Amazon EC2 creates a separate request for each Spot Instance.

- (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).
- **Purchasing option:** Choose **Request Spot instances** to launch a Spot Instance. When you choose this option, the following fields appear.
- **Current price:** The current Spot price in each Availability Zone is displayed for the instance type that you selected.
- (Optional) **Maximum price:** You can leave the field empty, or you can specify the maximum amount you're willing to pay.

⚠ Warning

If you specify a maximum price, your instances will be interrupted more frequently than if you leave the field empty.

- If you specify a maximum price that is less than the 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.
- If you leave the field empty, you'll pay the current Spot price.
- **Persistent request:** Choose **Persistent request** to resubmit the Spot Instance request if your Spot Instance is interrupted.
- **Interruption behavior:** By default, the Spot service terminates a Spot Instance when it is interrupted. If you choose **Persistent request**, you can then specify that the Spot service stops or hibernates your Spot Instance when it's interrupted. For more information, see [Interruption behavior](#).
- (Optional) **Request valid to:** Choose **Edit** to specify when the Spot Instance request expires.

For more information about configuring your Spot Instance, see [Step 3: Configure Instance Details](#).

7. The AMI you selected includes one or more volumes of storage, including the root device volume. On the **Add Storage** page, you can specify additional volumes to attach to the instance by choosing **Add New Volume**. For more information, see [Step 4: Add Storage](#).
8. On the **Add Tags** page, specify [tags](#) by providing key and value combinations. For more information, see [Step 5: Add Tags](#).
9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see

[Amazon EC2 security groups for Linux instances.](#)) Select or create a security group, and then choose **Review and Launch**. For more information, see [Step 6: Configure Security Group](#).

10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link. When you are ready, choose **Launch**. For more information, see [Step 7: Review Instance Launch and Select Key Pair](#).
11. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair that you created when getting set up. For more information, see [Amazon EC2 key pairs and Amazon EC2 instances](#).

⚠ Important

If you choose the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

12. To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

If the instance fails to launch or the state immediately goes to terminated instead of running, see [Troubleshoot instance launch issues](#).

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.


```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 5 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --instance-market-options file://spot-options.json
```

The following is the data structure to specify in the JSON file for `--instance-market-options`. You can also specify `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).

Find running 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.

To find running Spot Instances (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**. You can see both Spot Instance requests and Spot Fleet requests. If a Spot Instance request has been fulfilled, **Capacity** is the ID of the Spot Instance. For a Spot Fleet, **Capacity** indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot Fleet, choose the expand arrow, or select the fleet and choose **Instances**.

Note

For Spot Instance requests that are created by a Spot Fleet, the requests are not tagged instantly with the system tag that indicates the Spot Fleet to which they belong, and for a period of time may appear separate from Spot Fleet request.

Alternatively, in the navigation pane, choose **Instances**. In the top right corner, choose the settings icon (



), and then under **Attribute columns**, select **Instance lifecycle**. For each instance, **Instance lifecycle** is either `normal`, `spot`, or `scheduled`.

To find running Spot Instances (AWS CLI)

To enumerate your Spot Instances, use the [describe-spot-instance-requests](#) command with the `--query` option.

```
aws ec2 describe-spot-instance-requests \
  --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

The following is example output:

```
[
  {
    "ID": "i-1234567890abcdef0"
  },
  {
    "ID": "i-0598c7d356eba48d7"
  }
]
```

Alternatively, you can enumerate your Spot Instances using the [describe-instances](#) command with the `--filters` option.

```
aws ec2 describe-instances \
  --filters "Name=instance-lifecycle,Values=spot"
```

To describe a single Spot Instance instance, use the [describe-spot-instance-requests](#) command with the `--spot-instance-request-ids` option.

```
aws ec2 describe-spot-instance-requests \
  --spot-instance-request-ids sir-08b93456
```

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](#)
- [Tag a new Spot Instance request](#)
- [Tag an existing Spot Instance request](#)

- [View Spot Instance request tags](#)

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLaunchInstances",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:image/*",
```



```

        "arn:aws:ec2:us-east-1:*:subnet/*",
        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
        "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
    ]
},
{
    "Sid": "TagSpotInstanceRequests",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*"
}
]
}

```

Note

When you use the `RunInstances` action to create Spot Instance requests and tag the Spot Instance requests on create, you need to be aware of how Amazon EC2 evaluates the `spot-instances-request` resource in the `RunInstances` statement.

The `spot-instances-request` resource is evaluated in the IAM policy as follows:

- If you don't tag a Spot Instance request on create, Amazon EC2 does not evaluate the `spot-instances-request` resource in the `RunInstances` statement.
- If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the `RunInstances` statement.

Therefore, for the `spot-instances-request` resource, the following rules apply to the IAM policy:

- If you use `RunInstances` to create a Spot Instance request and you don't intend to tag the Spot Instance request on create, you don't need to explicitly allow the `spot-instances-request` resource; the call will succeed.
- If you use `RunInstances` to create a Spot Instance request and intend to tag the Spot Instance request on create, you must include the `spot-instances-request` resource in the `RunInstances` allow statement, otherwise the call will fail.

- If you use `RunInstances` to create a Spot Instance request and intend to tag the Spot Instance request on create, you must specify the `spot-instances-request` resource or include a `*` wildcard in the `CreateTags` allow statement, otherwise the call will fail.

For example IAM policies, including policies that are not supported for Spot Instance requests, see [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.

```
{
  "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

To tag a new Spot Instance request using the console

1. Follow the [Create a Spot Instance request](#) procedure.
2. To add a tag, on the **Add Tags** page, choose **Add Tag**, and enter the key and value for the tag. Choose **Add another tag** for each additional tag.

For each tag, you can tag the Spot Instance request, the Spot Instances, and the volumes with the same tag. To tag all three, ensure that **Instances**, **Volumes**, and **Spot Instance Requests** are selected. To tag only one or two, ensure that the resources you want to tag are selected, and the other resources are cleared.

3. Complete the required fields to create a Spot Instance request, and then choose **Launch**. For more information, see [Create a Spot Instance request](#).

To tag a new Spot Instance request using the AWS CLI

To tag a Spot Instance request when you create it, configure the Spot Instance request configuration as follows:

- Specify the tags for the Spot Instance request using the `--tag-specification` parameter.
- For `ResourceType`, specify `spot-instances-request`. If you specify another value, the Spot Instance request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the Spot Instance request is tagged with two tags: `Key=Environment` and `Value=Production`, and `Key=Cost-Center` and `Value=123`.

```
aws ec2 request-spot-instances \  
  --instance-count 5 \  
  --type "one-time" \  
  --launch-specification file://specification.json \  
  --tag-specification 'ResourceType=spot-instances-  
request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]'
```

Tag an existing Spot Instance request

To tag an existing Spot Instance request using the console

After you have created a Spot Instance request, you can add tags to the Spot Instance request using the console.

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

1. In the navigation pane, choose **Spot Requests**.
2. Select your Spot Instance request.

3. Choose the **Tags** tab and choose **Create Tag**.

To tag an existing Spot Instance using the console

After your Spot Instance request has launched your Spot Instance, you can add tags to the instance using the console. For more information, see [Add and delete tags on an individual resource](#).

To tag an existing Spot Instance request or Spot Instance using the AWS CLI

Use the [create-tags](#) command to tag existing resources. In the following example, the existing Spot Instance request and the Spot Instance are tagged with Key=purpose and Value=test.

```
aws ec2 create-tags \  
  --resources sir-08b93456 i-1234567890abcdef0 \  
  --tags Key=purpose,Value=test
```

View Spot Instance request tags

To view Spot Instance request tags using the console

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

1. In the navigation pane, choose **Spot Requests**.
2. Select your Spot Instance request and choose the **Tags** tab.

To describe Spot Instance request tags

Use the [describe-tags](#) command to view the tags for the specified resource. In the following example, you describe the tags for the specified request.

```
aws ec2 describe-tags \  
  --filters "Name=resource-id,Values=sir-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{  
  "Tags": [  
    {  
      "Key": "Environment",  
      "ResourceId": "sir-11112222-3333-4444-5555-66666EXAMPLE",  
      "ResourceType": "spot-instances-request",  
      "Value": "Production"  
    },  
  ],  
}
```

```

    {
      "Key": "Another key",
      "ResourceId": "sir-11112222-3333-4444-5555-66666EXAMPLE",
      "ResourceType": "spot-instances-request",
      "Value": "Another value"
    }
  ]
}

```

You can also view the tags of a Spot Instance request by describing the Spot Instance request.

Use the [describe-spot-instance-requests](#) command to view the configuration of the specified Spot Instance request, which includes any tags that were specified for the request.

```

aws ec2 describe-spot-instance-requests \
  --spot-instance-request-ids sir-11112222-3333-4444-5555-66666EXAMPLE

```

```

{
  "SpotInstanceRequests": [
    {
      "CreateTime": "2020-06-24T14:22:11+00:00",
      "InstanceId": "i-1234567890EXAMPLE",
      "LaunchSpecification": {
        "SecurityGroups": [
          {
            "GroupName": "launch-wizard-6",
            "GroupId": "sg-1234567890EXAMPLE"
          }
        ],
        "BlockDeviceMappings": [
          {
            "DeviceName": "/dev/xvda",
            "Ebs": {
              "DeleteOnTermination": true,
              "VolumeSize": 8,
              "VolumeType": "gp2"
            }
          }
        ],
        "ImageId": "ami-1234567890EXAMPLE",
        "InstanceType": "t2.micro",
        "KeyName": "my-key-pair",
        "NetworkInterfaces": [

```

```
        {
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "SubnetId": "subnet-11122233"
        }
    ],
    "Placement": {
        "AvailabilityZone": "eu-west-1c",
        "Tenancy": "default"
    },
    "Monitoring": {
        "Enabled": false
    }
},
"LaunchedAvailabilityZone": "eu-west-1c",
"ProductDescription": "Linux/UNIX",
"SpotInstanceRequestId": "sir-1234567890EXAMPLE",
"SpotPrice": "0.012600",
"State": "active",
"Status": {
    "Code": "fulfilled",
    "Message": "Your spot request is fulfilled.",
    "UpdateTime": "2020-06-25T18:30:21+00:00"
},
"Tags": [
    {
        "Key": "Environment",
        "Value": "Production"
    },
    {
        "Key": "Another key",
        "Value": "Another value"
    }
],
"Type": "one-time",
"InstanceInterruptionBehavior": "terminate"
}
]
```

Cancel a Spot Instance request

If you no longer want your Spot Instance request, you can cancel it. You can only cancel Spot Instance requests that are open, active, or disabled.

- Your Spot Instance request is open when your request has not yet been fulfilled and no instances have been launched.
- Your Spot Instance request is active when your request has been fulfilled and Spot Instances have launched as a result.
- Your Spot Instance request is disabled when you stop your Spot Instance.

If your Spot Instance request is active and has an associated running Spot Instance, canceling the request does not terminate the instance. For more information about terminating a Spot Instance, see [Terminate a Spot Instance](#).

To cancel a Spot Instance request (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests** and select the Spot Instance request.
3. Choose **Actions, Cancel request**.
4. (Optional) If you are finished with the associated Spot Instances, you can terminate them. In the **Cancel Spot request** dialog box, select **Terminate instances**, and then choose **Confirm**.

To cancel a Spot Instance request (AWS CLI)

- Use the [cancel-spot-instance-requests](#) command to cancel the specified Spot Instance request.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Stop a Spot Instance

If you don't need your Spot Instances now, but you want to restart them later without losing the data persisted in the Amazon EBS volume, you can stop them. The steps for stopping a Spot Instance are similar to the steps for stopping an On-Demand Instance.

Note

While a Spot Instance is stopped, you can modify some of its instance attributes, but not the instance type.

We don't charge usage for a stopped Spot Instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes.

Limitations

- You can only stop a Spot Instance if the Spot Instance was launched from a persistent Spot Instance request.
- You can't stop a Spot Instance if the associated Spot Instance request is cancelled. When the Spot Instance request is cancelled, you can only terminate the Spot Instance.
- You can't stop a Spot Instance if it is part of a fleet or launch group, or Availability Zone group.

Console**To stop a Spot Instance (console)**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Instance state**, **Stop instance**.
4. When prompted for confirmation, choose **Stop**.

AWS CLI**To stop a Spot Instance (AWS CLI)**

- Use the [stop-instances](#) command to manually stop one or more Spot Instances.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```


Start a Spot Instance

You can start a Spot Instance that you previously stopped. The steps for starting a Spot Instance are similar to the steps for starting an On-Demand Instance.

Prerequisites

You can only start a Spot Instance if:

- You manually stopped the Spot Instance.
- The Spot Instance is an EBS-backed instance.
- Spot Instance capacity is available.
- The Spot price is lower than your maximum price.

Limitations

- You can't start a Spot Instance if it is part of fleet or launch group, or Availability Zone group.

Console

To start a Spot Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Instance state**, **Start instance**.

AWS CLI

To start a Spot Instance (AWS CLI)

- Use the [start-instances](#) command to manually start one or more Spot Instances.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Terminate a Spot Instance

If you terminate a running or stopped Spot Instance that was launched by a persistent Spot Instance request, the Spot Instance request transitions to the open state so that a new Spot

Instance can be launched. To ensure that no new Spot Instance is launched, you must first cancel the Spot Instance request.

If you cancel an active Spot Instance request that has a running Spot Instance, the running Spot Instance is not automatically terminated; you must manually terminate the Spot Instance.

If you cancel a disabled Spot Instance request that has a stopped Spot Instance, the stopped Spot Instance is automatically terminated by the Amazon EC2 Spot service. There might be a short lag between when you cancel the Spot Instance request and when the Spot service terminates the Spot Instance.

For information about canceling a Spot Instance request, see [Cancel a Spot Instance request](#).

Console

To manually terminate a Spot Instance using the console

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Instances**.
4. To confirm that the instance is a Spot Instance, check that **spot** appears in the **Instance lifecycle** column.
5. Select the instance, and choose **Instance state**, **Terminate instance**.
6. Choose **Terminate** when prompted for confirmation.

AWS CLI

To manually terminate a Spot Instance using the AWS CLI

- Use the [terminate-instances](#) command to manually terminate Spot Instances.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

Spot Instance request example launch specifications

The following examples show launch configurations that you can use with the [request-spot-instances](#) command to create a Spot Instance request. For more information, see [Create a Spot Instance request](#).

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"
  }
}
```

Spot request status

To help you track your Spot Instance requests and plan your use of Spot Instances, use the request status provided by Amazon EC2. For example, the request status can provide the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

At each step of the process—also called the Spot request *lifecycle*—specific events determine successive request states.

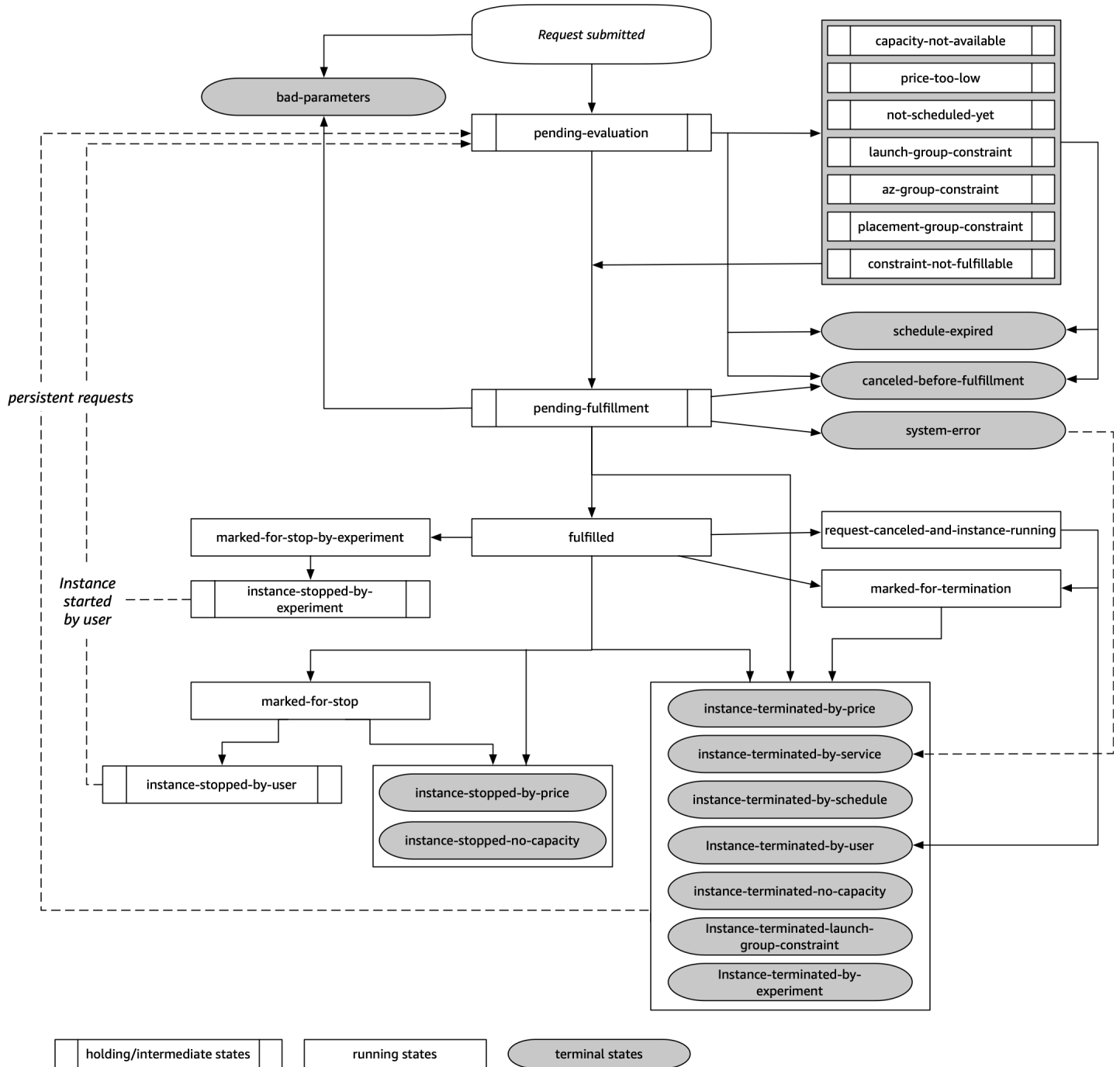
Contents

- [Lifecycle of a Spot request](#)
- [Get request status information](#)
- [Spot request status codes](#)

• [EC2 Spot Instance Request Fulfillment event](#)

Lifecycle of a Spot request

The following diagram shows you the paths that your Spot request can follow throughout its lifecycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot Instance.



Pending evaluation

As soon as you create a Spot Instance request, it goes into the pending-evaluation state unless one or more request parameters are not valid (bad-parameters).

Status code	Request state	Instance state
pending-evaluation	open	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

Status code	Request state	Instance state
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
cancel-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 †

Status code	Request state	Instance state
instance-terminated-no-capacity	closed (one-time), open (persistent)	terminated
instance-terminate-d-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`.

Persistent requests

When your Spot Instances are terminated (either by you or Amazon EC2), if the Spot request is a persistent request, it returns to the `pending-evaluation` state and then Amazon EC2 can launch a new Spot Instance when the constraints are met.

Get request status information

You can get request status information using the AWS Management Console or a command line tool.

To get request status information (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests** and select the Spot request.
3. To check the status, on the **Description** tab, check the **Status** field.

To get request status information using the command line

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

- [describe-spot-instance-requests](#) (AWS CLI)
- [Get-EC2SpotInstanceRequest](#) (AWS Tools for Windows PowerShell)

Spot request status codes

Spot request status information is composed of a status code, the update time, and a status message. Together, these help you determine the disposition of your Spot request.

The following are the Spot request status codes:

`az-group-constraint`

Amazon EC2 cannot launch all the instances you requested in the same Availability Zone.

`bad-parameters`

One or more parameters for your Spot request are not valid (for example, the AMI you specified does not exist). The status message indicates which parameter is not valid.

`canceled-before-fulfillment`

The user canceled the Spot request before it was fulfilled.

`capacity-not-available`

There is not enough capacity available for the instances that you requested.

`constraint-not-fulfillable`

The Spot request can't be fulfilled because one or more constraints are not valid (for example, the Availability Zone does not exist). The status message indicates which constraint is not valid.

`fulfilled`

The Spot request is active, and Amazon EC2 is launching your Spot Instances.

`instance-stopped-by-price`

Your instance was stopped because the Spot price exceeded your maximum price.

instance-stopped-by-user

Your instance was stopped because a user stopped the instance or ran the shutdown command from the instance.

instance-stopped-no-capacity

Your instance was stopped due to EC2 capacity management needs.

instance-terminated-by-price

Your instance was terminated because the Spot price exceeded your maximum price. If your request is persistent, the process restarts, so your request is pending evaluation.

instance-terminated-by-schedule

Your Spot Instance was terminated at the end of its scheduled duration.

instance-terminated-by-service

Your instance was terminated from a stopped state.

instance-terminated-by-user or spot-instance-terminated-by-user

You terminated a Spot Instance that had been fulfilled, so the request state is closed (unless it's a persistent request) and the instance state is terminated.

instance-terminated-launch-group-constraint

One or more of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.

instance-terminated-no-capacity

Your instance was terminated due to standard capacity management processes.

launch-group-constraint

Amazon EC2 cannot launch all the instances that you requested at the same time. All instances in a launch group are started and terminated together.

limit-exceeded

The limit on the number of EBS volumes or total volume storage was exceeded. For more information about these limits and how to request an increase, see [Amazon EBS Limits](#) in the *Amazon Web Services General Reference*.

marked-for-stop

The Spot Instance is marked for stopping.

marked-for-termination

The Spot Instance is marked for termination.

not-scheduled-yet

The Spot request is not evaluated until the scheduled date.

pending-evaluation

After you make a Spot Instance request, it goes into the `pending-evaluation` state while the system evaluates the parameters of your request.

pending-fulfillment

Amazon EC2 is trying to provision your Spot Instances.

placement-group-constraint

The Spot request can't be fulfilled yet because a Spot Instance can't be added to the placement group at this time.

price-too-low

The request can't be fulfilled yet because your maximum price is below the Spot price. In this case, no instance is launched and your request remains open.

request-canceled-and-instance-running

You canceled the Spot request while the Spot Instances are still running. The request is cancelled, but the instances remain running.

schedule-expired

The Spot request expired because it was not fulfilled before the specified date.

system-error

There was an unexpected system error. If this is a recurring issue, please contact AWS Support for assistance.

EC2 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-1a2b3c4d",
    "instance-id": "i-1234567890abcdef0"
  }
}
```

For more information, see the [Amazon EventBridge User Guide](#).

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.

Topics

- [Rebalance actions you can take](#)
- [Monitor rebalance recommendation signals](#)
- [Services that use the rebalance recommendation signal](#)

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 [Capacity Rebalancing](#) for EC2 Fleet and [Capacity Rebalancing](#) for Spot Fleet in this user guide.

Monitor rebalance recommendation signals

You can monitor the rebalance recommendation signal so that, when it is emitted, you can take the actions that are specified in the preceding section. The rebalance recommendation signal is made available as an event that is sent to Amazon EventBridge (formerly known as Amazon CloudWatch Events) and as instance metadata on the Spot Instance.

Monitor rebalance recommendation signals:

- [Use Amazon EventBridge](#)
- [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

```
[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/recommendations/rebalance
```

IMDSv1

```
[ec2-user ~]$ curl 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*
- [Capacity Rebalancing](#) in the EC2 Fleet topic in this user guide
- [Capacity Rebalancing](#) in the Spot Fleet topic in this user guide

Spot Instance interruptions

You can launch Spot Instances on spare EC2 capacity for steep discounts in exchange for returning them when Amazon EC2 needs the capacity back. When Amazon EC2 reclaims a Spot Instance, we call this event a *Spot Instance interruption*.

When Amazon EC2 interrupts a Spot Instance, it either terminates, stops, or hibernates the instance, depending on what you specified when you created the Spot request.

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.

An On-Demand Instance specified in an EC2 Fleet or Spot Fleet cannot be interrupted.

Contents

- [Reasons for interruption](#)
- [Interruption behavior](#)
- [Stop interrupted Spot Instances](#)
- [Hibernate interrupted Spot Instances](#)
- [Terminate interrupted Spot Instances](#)
- [Prepare for interruptions](#)
- [Initiate a Spot Instance interruption](#)
- [Spot Instance interruption notices](#)
- [Find interrupted Spot Instances](#)
- [Determine whether Amazon EC2 terminated a Spot Instance](#)

- [Billing for interrupted Spot Instances](#)

Reasons for interruption

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.

You can see the historical interruption rates for your instance type in the [Spot Instance Advisor](#).

Interruption behavior

You can specify that Amazon EC2 must do one of the following when it interrupts a Spot Instance:

- [Stop interrupted Spot Instances](#)
- [Hibernate interrupted Spot Instances](#)
- [Terminate interrupted Spot Instances](#) (this is the default behavior)

Specify the interruption behavior

You can specify the interruption behavior when you create a Spot request. If you do not specify an interruption behavior, the default is that Amazon EC2 terminates Spot Instances when they are interrupted.

The way in which you specify the interruption behavior is different depending on how you request Spot Instances.

- If you request Spot Instances using the [launch instance wizard](#), you can specify the interruption behavior as follows: In the launch instance wizard, expand **Advanced details** and select the **Request Spot Instances** check box. Choose **Customize**. From **Interruption behavior**, choose an interruption behavior. If the interruption behavior is hibernation, you can alternatively choose **Enable for Stop - Hibernate behavior**.
- If you request Spot Instances using the [run-instances](#) CLI, you can specify the interruption behavior as follows: In the request configuration, (`--instance-market-options`), for `InstanceInterruptionBehavior`, specify an interruption behavior. If the interruption behavior is hibernate, you can alternatively enable hibernation using the `--hibernation-options Configured=true` parameter.
- If you configure Spot Instances in a [launch template](#), you can specify the interruption behavior as follows: In the launch template, expand **Advanced details** and select the **Request Spot Instances** check box. Choose **Customize** and then, from **Interruption behavior**, choose an interruption behavior.
- If you request Spot Instances using the [Spot console](#), you can specify the interruption behavior as follows: Select the **Maintain target capacity** check box and then, from **Interruption behavior**, choose an interruption behavior.
- If you configure Spot Instances in the request configuration when using the [create-fleet](#) CLI, you can specify the interruption behavior as follows: For `InstanceInterruptionBehavior`, specify an interruption behavior.
- If you configure Spot Instances in the request configuration when using the [request-spot-fleet](#) CLI, you can specify the interruption behavior as follows: For `InstanceInterruptionBehavior`, specify an interruption behavior.
- If you configure Spot Instances using the [request-spot-instances](#) CLI, you can specify the interruption behavior as follows: For `--instance-interruption-behavior`, specify an interruption behavior.

Note

We strongly discourage using the [request-spot-fleet](#) and [request-spot-instances](#) commands to request Spot Instances because they are legacy APIs with no planned investment. For more information, see [Which is the best Spot request method to use?](#)

Stop interrupted Spot Instances

You can specify that Amazon EC2 stops your Spot Instances when they are interrupted. For more information, see [Specify the interruption behavior](#).

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).
- For Spot Instances launched by an EC2 Fleet or Spot Fleet of type maintain: After a Spot Instance is interrupted, Amazon EC2 launches a replacement instance to maintain the target capacity. Amazon EC2 finds the best Spot capacity pools based on the specified allocation strategy (`lowestPrice`, `diversified`, or `InstancePoolsToUseCount`); it does not prioritize the pool with the earlier stopped instance. Later, if the allocation strategy leads to a pool containing the earlier stopped instance, Amazon EC2 restarts the stopped instance to meet the target capacity.

For example, consider a Spot Fleet with the `lowestPrice` allocation strategy. At initial launch, a `c3.large` pool meets the `lowestPrice` criteria for the launch specification. Later, when the `c3.large` instances are interrupted, Amazon EC2 stops the instances and replenishes capacity from another pool that fits the `lowestPrice` strategy. This time, the pool happens to be a `c4.large` pool and Amazon EC2 launches `c4.large` instances to meet the target capacity. Similarly, Spot Fleet could move to a `c5.large` pool the next time. In each of these transitions, Amazon EC2 does not prioritize pools with earlier stopped instances, but rather prioritizes purely on the specified allocation strategy. The `lowestPrice` strategy can lead back to pools with earlier stopped instances. For example, if instances are interrupted in the `c5.large` pool and the `lowestPrice` strategy leads it back to the `c3.large` or `c4.large` pools, the earlier stopped instances are restarted to fulfill target capacity.

- While a Spot Instance is stopped, you can modify some of its instance attributes, but not the instance type. If you detach or delete an EBS volume, it is not attached when the Spot Instance is started. If you detach the root volume and 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 reboot, stop, hibernate, and terminate](#).

Prerequisites

To stop an interrupted Spot Instance, the following prerequisites must be in place:

Spot request type

Spot Instance request type – Must be persistent. You can't specify a launch group in the Spot Instance request.

EC2 Fleet or Spot Fleet request type – Must be maintain.

Root volume type

Must be an EBS volume, not an instance store volume.

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 [Specify the interruption behavior](#).

Prepare for 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).

Topics

- [Initiate a Spot Instance interruption](#)
- [Verify the Spot Instance interruption](#)
- [Quotas](#)

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
Target SpotInstances for <code>aws:ec2:send-spot-instance-interruptions</code>	Each supported Region: 5	Yes	The maximum number of Spot Instances that <code>aws:ec2:send-spot-instance-interruptions</code> 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 a 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 notice

When Amazon EC2 is going to interrupt your Spot Instance, it emits an event two minutes prior to the actual interruption (except for hibernation, which gets the interruption notice, but not two minutes in advance, because hibernation begins immediately). This event can be detected by Amazon 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

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.

```
[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/spot/instance-action
```

The `instance-action` item specifies the action and the approximate time, in UTC, when the action will occur.

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

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.

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" `
```



```
[ec2-user ~]$ if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo termination_scheduled; fi
```

The `termination-time` item specifies the approximate time in UTC when the instance will receive the shutdown signal. 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

In the console, the **Instances** pane displays all instances, including 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`.

To find an interrupted Spot 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. 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`

- For additional details about the reason for the interruption, check the Spot request status code. For more information, see [the section called "Spot request status"](#).

To find interrupted Spot Instances using the AWS CLI

You can list your interrupted Spot Instances using the [describe-instances](#) command with the `--filters` parameter. To list only the instance IDs in the output, include the `--query` parameter.

If the instance interruption behavior is to terminate the Spot Instances, use the following command:

```
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 instance interruption behavior is to stop the Spot Instances, use the following command:

```
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"
```

Determine whether Amazon EC2 terminated a Spot Instance

If a Spot Instance is terminated, you can use CloudTrail to see whether Amazon EC2 terminated the Spot Instance. In AWS CloudTrail, the event name `BidEvictedEvent` indicates that Amazon EC2 terminated the Spot Instance.

To view `BidEvictedEvent` events in CloudTrail

- Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
- In the navigation pane, choose **Event history**.
- In the filter drop-down, choose **Event name**, and then in the filter field to the right, enter **BidEvictedEvent**.
- Choose **BidEvictedEvent** in the resulting list to view its details. Under **Event record**, you can find the instance ID.

For more information about using CloudTrail, see [Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail](#).

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 interrupted 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.

Other charges

If your running Spot Instance incurs charges for other services, such as for data transfer, Elastic IP addresses, or usage of other AWS managed services, you will be billed for their usage. This is regardless of who interrupts the Spot Instance or when it was interrupted. Even if you're not charged for Spot Instance usage when Amazon EC2 interrupts your Spot Instance in the first hour, you can incur other charges.

For more information about other charges, see [Amazon EC2 On-Demand Pricing](#).

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.

Benefits

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.

Topics

- [Costs](#)
- [How Spot placement score works](#)
- [Limitations](#)
- [Required IAM permission](#)
- [Calculate a Spot placement score](#)
- [Example configurations](#)

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 [Creating an Auto Scaling group with a set of requirements on the instance types used](#), [Attribute-based instance type selection for EC2 Fleet](#), and [Attribute-based instance type selection for 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.

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.

Required IAM permission

By default, IAM identities (users, roles, or groups) don't have permission to use the Spot placement score feature. To allow IAM identities to use the Spot placement score feature, 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.

```
{
  "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 [Creating 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 [Creating 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 a Spot placement score

You can calculate a Spot placement score by using the Amazon EC2 console or the AWS CLI.

Topics

- [Calculate a Spot placement score by specifying instance attributes \(console\)](#)
- [Calculate a Spot placement score by specifying instance types \(console\)](#)
- [Calculate the Spot placement score \(AWS CLI\)](#)

Calculate a Spot placement score by specifying instance attributes (console)

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 **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](#) in the *Amazon EC2 Command Line Reference*.
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** check box. 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 a Spot placement score by specifying instance types (console)

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 **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** check box. 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 the Spot placement score (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
```

Expected 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](#) in the *Amazon EC2 Command Line Reference*.

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](#) in the *Amazon EC2 Command Line Reference*.

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
      }
    }
  }
}
```

Spot Instance data feed

To help you understand the charges for your Spot Instances, Amazon EC2 provides a data feed that describes your Spot Instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

Data feed files arrive in your bucket typically once an hour, and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before they are delivered to your bucket. Amazon EC2 can write multiple files for a given hour of usage where files are large (for example, when file contents for the hour exceed 50 MB before compression).

Note

You can create only one Spot Instance data feed per AWS account. If you don't have a Spot Instance running during a certain hour, you don't receive a data feed file for that hour.

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](#)
- [Amazon S3 bucket requirements](#)
- [Subscribe to your Spot Instance data feed](#)
- [Describe your Spot Instance data feed](#)
- [View the data in your data feed](#)
- [Delete your Spot Instance data feed](#)

Data feed file name and format

The Spot Instance data feed file name uses the following format (with the date and hour in UTC):

```
bucket-name.s3.amazonaws.com/optional-prefix/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz
```

For example, if your bucket name is **my-bucket-name** and your prefix is **my-prefix**, your file names are similar to the following:

```
my-bucket-name.s3.amazonaws.com/my-prefix/111122223333.2023-12-09-07.001.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.
UsageType	The type of usage and instance type being charged for. For <code>m1.small</code> Spot Instances, this field is set to <code>SpotUsage</code> . For all other instance types, this field is set to <code>SpotUsage: {instance-type}</code> . For example, <code>SpotUsage:c1.medium</code> .
Operation	The product being charged for. For Linux Spot Instances, this field is set to <code>RunInstances</code> . For Windows Spot Instances, this field is set to <code>RunInstances:0002</code> . Spot usage is grouped according to Availability Zone.
InstanceID	The ID of the Spot Instance that generated this instance usage.
MyBidID	The ID for the Spot Instance request that generated this instance usage.
MyMaxPrice	The maximum price specified for this Spot request.
MarketPrice	The Spot price at the time specified in the <code>Timestamp</code> field.
Charge	The price charged for this instance usage.
Version	The 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*.

Note

If you update the ACL and remove the permissions for the AWS data feed account, the data feed files cannot be written to the bucket. You must resubscribe to the data feed to receive the data feed files.

- Each data feed file has its own ACL (separate from the ACL for the bucket). The bucket owner has `FULL_CONTROL` permission to the data files. The AWS data feed account has read and write permissions.
- If you applied disabled ACLs for your buckets, add a bucket policy that allows users with full control to write to the bucket. For more information, see [Review and update bucket policies](#).
- 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.
- You must use a customer managed key if you encrypt your Amazon S3 bucket using server-side encryption with a AWS KMS key stored in AWS Key Management Service (SSE-KMS). For more information, see [Amazon S3 bucket server-side encryption](#) in the *Amazon CloudWatch Logs User Guide*.

Note

For Spot Instance data feed, the resource that generates the S3 files is no longer Amazon CloudWatch Logs. Therefore, you must remove the `aws:SourceArn` section from the S3 bucket permission policy and from the KMS policy.

Subscribe to your Spot Instance data feed

To subscribe to your data feed, use the [create-spot-datafeed-subscription](#) command.

```
aws ec2 create-spot-datafeed-subscription \  
  --bucket my-bucket-name \  
  [--prefix my-prefix]
```

Example output

```
{  
  "SpotDatafeedSubscription": {  
    "OwnerId": "111122223333",  
    "Bucket": "my-bucket-name",  
    "Prefix": "my-prefix",  
    "State": "Active"  
  }  
}
```

Describe your Spot Instance data feed

To describe your data feed subscription, use the [describe-spot-datafeed-subscription](#) command.

```
aws ec2 describe-spot-datafeed-subscription
```

Example output

```
{  
  "SpotDatafeedSubscription": {  
    "OwnerId": "123456789012",  
    "Prefix": "spotdata",  
    "Bucket": "my-s3-bucket",  
    "State": "Active"  
  }  
}
```

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 the folder that you specify.

```
aws s3 sync s3://my-s3-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

To delete your data feed, use the [delete-spot-datafeed-subscription](#) command.

```
aws ec2 delete-spot-datafeed-subscription
```

Spot Instance quotas

There are quotas for the number of running Spot Instances and pending Spot Instance requests per AWS account per Region. Once 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:

- All DL Spot Instance Requests
- All F Spot Instance Requests
- All G and VT Spot Instance Requests
- All Inf Spot Instance Requests
- All P Spot Instance Requests
- All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests
- All Trn Spot Instance Requests
- All X Spot Instance Requests

Each quota type specifies the maximum number of vCPUs for one or more instance families. For information about the different instance families, generations, and sizes, see [Amazon EC2 Instance Types](#).

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).

Tasks

- [Monitor Spot Instance quotas and usage](#)
- [Request a quota increase](#)

Monitor Spot Instance quotas and usage

You can view and manage your Spot Instance quotas using the following:

- The Amazon EC2 [Services quotas page](#) in the Service Quotas console
- The [get-service-quota](#) AWS CLI

For more information, see [Amazon EC2 service quotas](#) in the *Amazon EC2 User Guide for Linux Instances* and [Viewing service quotas](#) in the *Service Quotas User Guide*.

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*.

Request a quota increase

Even though Amazon EC2 automatically increases 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](#).

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.

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 [Work with shared Dedicated Hosts](#).

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 [Understand auto-placement and 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
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

	Dedicated Host	Dedicated Instance
Targeted instance placement	Provides additional visibility and control over how instances are placed on a physical server	Not supported
Automatic instance recovery	Supported. For more information, see Host recovery .	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 [Dedicated Instances](#).

Contents

- [Instance capacity configurations](#)
- [Bring your own license](#)
- [Pricing and billing](#)
- [Burstable T3 instances on Dedicated Hosts](#)
- [Dedicated Hosts restrictions](#)
- [Work with Dedicated Hosts](#)
- [Work with shared Dedicated Hosts](#)
- [Dedicated Hosts on AWS Outposts](#)
- [Host recovery](#)
- [Host maintenance](#)
- [Track configuration changes](#)

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](#)
- [Multiple instance type support](#)

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 any combination of R5 instance types, such as `r5.large`, `r5.xlarge`, `r5.2xlarge`, and `r5.4xlarge`, on that host, up to the host's physical core capacity.

The following instance families support Dedicated Hosts with multiple instance type support:

- **General purpose:** A1, M5, M5n, M6i, and T3
- **Compute optimized:** C5, C5n, and C6i
- **Memory optimized:** R5, R5n, and R6i

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 can run any combination of R5 instances as long as they do not exceed 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 size. 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 sizes can then be used to fill the host to core capacity. For the supported number of instance sizes for each instance family, see the [Dedicated Hosts Configuration Table](#).

The following table shows example instance type combinations:

Instance family	Example instance size combinations
R5	<ul style="list-style-type: none"> • Example 1: 4 x <code>r5.4xlarge</code> + 4 x <code>r5.2xlarge</code> • Example 2: 1 x <code>r5.12xlarge</code> + 1 x <code>r5.4xlarge</code> + 1 x <code>r5.2xlarge</code> + 5 x <code>r5.xlarge</code> + 2 x <code>r5.large</code>
C5	<ul style="list-style-type: none"> • Example 1: 1 x <code>c5.9xlarge</code> + 2 x <code>c5.4xlarge</code> + 1 x <code>c5.xlarge</code> • Example 2: 4 x <code>c5.4xlarge</code> + 1 x <code>c5.xlarge</code> + 2 x <code>c5.large</code>
M5	<ul style="list-style-type: none"> • Example 1: 4 x <code>m5.4xlarge</code> + 4 x <code>m5.2xlarge</code>

Instance family	Example instance size combinations	
	<ul style="list-style-type: none">• 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:

- With N-type Dedicated Hosts, such as C5n, M5n, and R5n, you can't mix smaller instance sizes (2xlarge and smaller) with larger instance sizes (4xlarge and larger, including metal). If you require smaller and larger instance sizes on N-type Dedicated Hosts at the same time, you must allocate separate hosts for the smaller and larger instance sizes.
- We recommend that you launch larger instance types first, and then fill the remaining instance capacity with smaller instance types as needed.

Bring your own license

Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses. When you bring your own license, you are responsible for managing your own licenses. However, Amazon EC2 has features that help you maintain license compliance, such as instance affinity and targeted placement.

These are the general steps to follow in order to bring your own volume licensed machine image into Amazon EC2.

1. Verify that the license terms controlling the use of your machine images allow usage in a virtualized cloud environment.
2. After you have verified that your machine image can be used within Amazon EC2, import it using VM Import/Export. For information about how to import your machine image, see the [VM Import/Export User Guide](#).
3. After you import your machine image, you can launch instances from it onto active Dedicated Hosts in your account.

4. When you run these instances, depending on the operating system, you might be required to activate these instances against your own KMS server.

Note

To track how your images are used in AWS, enable host recording in AWS Config. You can use AWS Config to record configuration changes to a Dedicated Host and use the output as a data source for license reporting. For more information, see [Track configuration changes](#).

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 Dedicated Hosts](#).

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](#).

Burstable T3 instances on 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.

Instance 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.nano	2	0.5	5%	5	Up to 2,085	192
t3.micro	2	1	10%	5	Up to 2,085	192
t3.small	2	2	20%	5	Up to 2,085	192
t3.medium	2	4	20%	5	Up to 2,085	192
t3.large	2	8	30%	5	2,780	96
t3.xlarge	4	16	40%	5	2,780	48
t3.2xlarge	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](#).

Dedicated Hosts restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions:

- To run RHEL, SUSE Linux, and SQL Server on Dedicated Hosts, you must bring your own AMIs. RHEL, SUSE Linux, and SQL Server AMIs that are offered by AWS or that are available on AWS Marketplace can't be used with Dedicated Hosts. For more information on how to create your own AMI, see [Bring your own license](#).

This restriction does not apply to hosts allocated for high memory instances (`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, and `u-24tb1.metal`). RHEL and SUSE Linux AMIs that are offered by AWS or that are available on AWS Marketplace can be used with these hosts.

- 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](#).
- The instances that run on a Dedicated Host can only be launched in a VPC.
- Auto Scaling groups are supported when using a launch template that specifies a host resource group. For more information, see [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 cannot 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`.

Work with Dedicated Hosts

To use a Dedicated Host, you first allocate hosts for use in your account. You then launch instances onto the hosts by specifying *host* tenancy for the instance. You must select a specific host for the instance to launch on to, or you can allow it to launch on to any host that has auto-placement enabled and matches its instance type. When an instance is stopped and restarted, the *Host affinity* setting determines whether it's restarted on the same, or a different, host.

If you no longer need an On-Demand host, you can stop the instances running on the host, direct them to launch on a different host, and then *release* the host.

Dedicated Hosts are also integrated with AWS License Manager. With License Manager, you can create a host resource group, which is a collection of Dedicated Hosts that are managed as a single entity. When creating a host resource group, you specify the host management preferences, such as auto-allocate and auto-release, for the Dedicated Hosts. This allows you to launch instances onto Dedicated Hosts without manually allocating and managing those hosts. For more information, see [Host Resource Groups](#) in the *AWS License Manager User Guide*.

Contents

- [Allocate Dedicated Hosts](#)
- [Launch instances onto a Dedicated Host](#)
- [Launch instances into a host resource group](#)
- [Understand auto-placement and affinity](#)
- [Modify Dedicated Host auto-placement](#)
- [Modify the supported instance types](#)
- [Modify instance tenancy and affinity](#)
- [View Dedicated Hosts](#)
- [Tag Dedicated Hosts](#)
- [Monitor Dedicated Hosts](#)
- [Release Dedicated Hosts](#)
- [Purchase Dedicated Host Reservations](#)
- [View Dedicated Host reservations](#)
- [Tag Dedicated Host Reservations](#)

Allocate Dedicated Hosts

To begin using Dedicated Hosts, you must allocate Dedicated Hosts in your account using the Amazon EC2 console or the command line tools. After you allocate the Dedicated Host, the Dedicated Host capacity is made available in your account immediately and you can start launching instances onto the Dedicated Host.

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 [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 [Understand auto-placement and affinity](#).

7. To enable host recovery for the Dedicated Host, for **Host recovery**, choose **Enable**. For more information, see [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](#) AWS CLI command. The following command allocates a Dedicated Host that supports multiple instance types from the m5 instance family in us-east-1a Availability Zone. The host also has host recovery enabled and it has auto-placement disabled.

```
aws ec2 allocate-hosts --instance-family "m5" --availability-zone "us-east-1a" --auto-placement "off" --host-recovery "on" --quantity 1
```

The following command allocates a Dedicated Host that supports *untargeted* m4.large instance launches in the eu-west-1a Availability Zone, enables host recovery, and applies a tag with a key of purpose and a value of production.

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a" --auto-placement "on" --host-recovery "on" --quantity 1 --tag-specifications 'ResourceType=dedicated-host,Tags=[{Key=purpose,Value=production}]'
```

PowerShell

To allocate a Dedicated Host

Use the [New-EC2Host](#) AWS Tools for Windows PowerShell command. The following command allocates a Dedicated Host that supports multiple instance types from the m5 instance family in us-east-1a Availability Zone. The host also has host recovery enabled and it has auto-placement disabled.

```
PS C:\> New-EC2Host -InstanceFamily m5 -AvailabilityZone us-east-1a -AutoPlacement Off -HostRecovery On -Quantity 1
```

The following commands allocate a Dedicated Host that supports *untargeted* `m4.large` instance launches in the `eu-west-1a` Availability Zone, enable host recovery, and apply a tag with a key of `purpose` and a value of `production`.

The `TagSpecification` parameter used to tag a Dedicated Host on creation requires an object that specifies the type of resource to be tagged, the tag key, and the tag value. The following commands create the required object.

```
PS C:\> $tag = @{ Key="purpose"; Value="production" }
PS C:\> $tagspec = new-object Amazon.EC2.Model.TagSpecification
PS C:\> $tagspec.ResourceType = "dedicated-host"
PS C:\> $tagspec.Tags.Add($tag)
```

The following command allocates the Dedicated Host and applies the tag specified in the `$tagspec` object.

```
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -
AutoPlacement On -HostRecovery On -Quantity 1 -TagSpecification $tagspec
```

Launch instances onto a Dedicated Host

After you have allocated a Dedicated Host, you can launch instances onto it. You can't launch instances with host tenancy if you do not have active Dedicated Hosts with enough available capacity for the instance type that you are launching.

Tip

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](#).

You can launch an instance onto a Dedicated Host using the following methods.

Console


To launch an instance onto a specific Dedicated Host from the Dedicated Hosts page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, select a host and choose **Actions, Launch Instance(s) onto host**.
4. In the **Application and OS Images** section, select an AMI from the list.

 **Note**

SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.

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**, 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 [Understand auto-placement and 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 [Launch an instance using defined 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.

Note

SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.

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**, 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 [Understand auto-placement and affinity](#).

7. Configure the remaining instance options as needed. For more information, see [Launch an instance using defined parameters](#).
8. Choose **Launch instance**.

AWS CLI

To launch an instance onto a Dedicated Host

Use the [run-instances](#) AWS CLI command and specify the instance affinity, tenancy, and host in the `Placement` request parameter.

PowerShell

To launch an instance onto a Dedicated Host

Use the [New-EC2Instance](#) AWS Tools for Windows PowerShell command and specify the instance affinity, tenancy, and host in the `Placement` request parameter.

Launch instances into a host resource group

When you launch an instance into a host resource group that has a Dedicated Host with available instance capacity, Amazon EC2 launches the instance onto that host. If the host resource group does not have a host with available instance capacity, Amazon EC2 automatically allocates a new host in the host resource group, and then launches the instance onto that host. For more information, see [Host Resource Groups](#) in the *AWS License Manager User Guide*.

Requirements and limits

- You must associate a core- or socket-based license configuration with the AMI.
- You can't use SQL Server, SUSE, or RHEL AMIs provided by Amazon EC2 with Dedicated Hosts.
- You can't target a specific host by choosing a host ID, and you can't enable instance affinity when launching an instance into a host resource group.


You can launch an instance into a host resource group using the following methods.

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.

 **Note**

SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.

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 [Understand auto-placement and affinity](#).

7. Configure the remaining instance options as needed. For more information, see [Launch an instance using defined parameters](#).
8. Choose **Launch instance**.

AWS CLI

To launch an instance into a host resource group

Use the [run-instances](#) AWS CLI command, and in the Placement request parameter, omit the Tenancy option and specify the host resource group ARN.

PowerShell

To launch an instance into a host resource group

Use the [New-EC2Instance](#) AWS Tools for Windows PowerShell command, and in the `Placement` request parameter, omit the `Tenancy` option and specify the host resource group ARN.

Understand auto-placement and affinity

Placement control for Dedicated Hosts happens on both the instance level and host level.

Auto-placement

Auto-placement is configured at the host level. It allows you to manage whether instances that you launch are launched onto a specific host, or onto any available host that has matching configurations.

When the auto-placement of a Dedicated Host is *disabled*, it only accepts *Host* tenancy instance launches that specify its unique host ID. This is the default setting for new Dedicated Hosts.

When the auto-placement of a Dedicated Host is *enabled*, it accepts any untargeted instance launches that match its instance type configuration.

When launching an instance, you need to configure its tenancy. Launching an instance onto a Dedicated Host without providing a specific `HostId` enables it to launch on any Dedicated Host that has auto-placement *enabled* and that matches its instance type.

Host affinity

Host affinity is configured at the instance level. It establishes a launch relationship between an instance and a Dedicated Host.

When affinity is set to `Host`, an instance launched onto a specific host always restarts on the same host if stopped. This applies to both targeted and untargeted launches.

When affinity is set to `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).

Modify Dedicated Host auto-placement

You can modify the auto-placement settings of a Dedicated Host after you have allocated it to your AWS account, using one of the following methods.

Console

To modify the auto-placement of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select a host and choose **Actions, Modify host**.
4. For **Instance auto-placement**, choose **Enable** to enable auto-placement, or clear **Enable** to disable auto-placement. For more information, see [Understand auto-placement and affinity](#).
5. Choose **Save**.

AWS CLI

To modify the auto-placement of a Dedicated Host

Use the [modify-hosts](#) AWS CLI command. The following example enables auto-placement for the specified Dedicated Host.

```
aws ec2 modify-hosts --auto-placement on --host-ids h-012a3456b7890cdef
```

PowerShell

To modify the auto-placement of a Dedicated Host

Use the [Edit-EC2Host](#) AWS Tools for Windows PowerShell command. The following example enables auto-placement for the specified Dedicated Host.

```
PS C:\> Edit-EC2Host --AutoPlacement 1 --HostId h-012a3456b7890cdef
```

Modify the supported instance types

Support for multiple instance types on the same Dedicated Host is available for the following instance families: C5, M5, R5, C5n, R5n, M5n, and T3. Other instance families support only a single instance type on the same Dedicated Host.

You can allocate a Dedicated Host using the following methods.

You can modify a Dedicated Host to change the instance types that it supports. If it currently supports a single instance type, you can modify it to support multiple instance types within that instance family. Similarly, if it currently supports multiple instance types, you can modify it to support a specific instance type only.

To modify a Dedicated Host to support multiple instance types, you must first stop all running instances on the host. The modification takes approximately 10 minutes to complete. The Dedicated Host transitions to the pending state while the modification is in progress. You can't start stopped instances or launch new instances on the Dedicated Host while it is in the pending state.

To modify a Dedicated Host that supports multiple instance types to support only a single instance type, the host must either have no running instances, or the running instances must be of the instance type that you want the host to support. For example, to modify a host that supports multiple instance types in the m5 instance family to support only m5.large instances, the Dedicated Host must either have no running instances, or it must have only m5.large instances running on it.

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.

You can modify the supported instance types using one of the following methods.

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](#) AWS CLI command.

The following command modifies a Dedicated Host to support multiple instance types within the m5 instance family.

```
aws ec2 modify-hosts --instance-family m5 --host-ids h-012a3456b7890cdef
```

The following command modifies a Dedicated Host to support m5.xlarge instances only.

```
aws ec2 modify-hosts --instance-type m5.xlarge --instance-family --host-ids h-012a3456b7890cdef
```

PowerShell

To modify the supported instance types for a Dedicated Host

Use the [Edit-EC2Host](#) AWS Tools for Windows PowerShell command.

The following command modifies a Dedicated Host to support multiple instance types within the m5 instance family.

```
PS C:\> Edit-EC2Host --InstanceFamily m5 --HostId h-012a3456b7890cdef
```


The following command modifies a Dedicated Host to support `m5.xlarge` instances only.

```
PS C:\> Edit-EC2Host --InstanceType m5.xlarge --HostId h-012a3456b7890cdef
```

Modify instance tenancy and affinity

You can change the tenancy of an instance 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.

You can modify an instance's tenancy and affinity using the following methods.

Console

To modify instance tenancy or affinity

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances**, and select the instance to modify.
3. Choose **Instance state, Stop**.
4. 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 [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 [Understand auto-placement and affinity](#).

6. Choose **Save**.

AWS CLI

To modify instance tenancy or affinity

Use the [modify-instance-placement](#) AWS CLI command. The following example changes the specified instance's affinity from default to host, and specifies the Dedicated Host that the instance has affinity with.

```
aws ec2 modify-instance-placement --instance-id i-1234567890abcdef0 --affinity host --tenancy host --host-id h-012a3456b7890cdef
```

PowerShell

To modify instance tenancy or affinity

Use the [Edit-EC2InstancePlacement](#) AWS Tools for Windows PowerShell command. The following example changes the specified instance's affinity from default to host, and specifies the Dedicated Host that the instance has affinity with.

```
PS C:\> Edit-EC2InstancePlacement -InstanceId i-1234567890abcdef0 -Affinity host -Tenancy host -HostId h-012a3456b7890cdef
```

View Dedicated Hosts

You can view details about a Dedicated Host and the individual instances on it using the following methods.

Console

To view the details of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. On the **Dedicated Hosts** page, select a host.
4. For information about the host, choose **Details**.

Available vCPUs indicates the vCPUs that are available on the Dedicated Host for new instance launches. For example, a Dedicated Host that supports multiple instance types within the c5 instance family, and that has no instances running on it, has 72 available vCPUs. This means that you can launch different combinations of instance types onto the Dedicated Host to consume the 72 available vCPUs.

For information about instances running on the host, choose **Running instances**.

AWS CLI

To view the capacity of a Dedicated Host

Use the [describe-hosts](#) AWS CLI command.

The following example uses the [describe-hosts](#) (AWS CLI) command to view the available instance capacity for a Dedicated Host that supports multiple instance types within the c5 instance family. The Dedicated Host already has two c5.4xlarge instances and four c5.2xlarge instances running on it.

```
$ aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

```
"AvailableInstanceCapacity": [  
  { "AvailableCapacity": 2,  
    "InstanceType": "c5.xlarge",  
    "TotalCapacity": 18 },  
  { "AvailableCapacity": 4,
```

```
    "InstanceType": "c5.large",  
    "TotalCapacity": 36 }  
  ],  
  "AvailableVCpus": 8
```

PowerShell

To view the instance capacity of a Dedicated Host

Use the [Get-EC2Host](#) AWS Tools for Windows PowerShell command.

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

Tag Dedicated Hosts

You can assign custom tags to your existing Dedicated Hosts to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host based on the custom tags that you assigned. Dedicated Host tags can also be used for cost allocation tracking.

You can also apply tags to Dedicated Hosts at the time of creation. For more information, see [Allocate Dedicated Hosts](#).

You can tag a Dedicated Host using the following methods.

Console

To tag a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host to tag, and then choose **Actions, Manage tags**.
4. In the **Manage tags** screen, choose **Add tag**, and then specify the key and value for the tag.
5. (Optional) Choose **Add tag** to add additional tags to the Dedicated Host.
6. Choose **Save changes**.

AWS CLI

To tag a Dedicated Host

Use the [create-tags](#) AWS CLI command.

The following command tags the specified Dedicated Host with Owner=TeamA.

```
aws ec2 create-tags --resources h-abc12345678909876 --tags Key=Owner,Value=TeamA
```

PowerShell

To tag a Dedicated Host

Use the [New-EC2Tag](#) AWS Tools for Windows PowerShell command.

The New-EC2Tag command needs a Tag object, which specifies the key and value pair to be used for the Dedicated Host tag. The following commands create a Tag object named \$tag, with a key and value pair of Owner and TeamA respectively.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag  
PS C:\> $tag.Key = "Owner"  
PS C:\> $tag.Value = "TeamA"
```

The following command tags the specified Dedicated Host with the \$tag object.

```
PS C:\> New-EC2Tag -Resource h-abc12345678909876 -Tag $tag
```

Monitor Dedicated Hosts

Amazon EC2 constantly monitors the state of your Dedicated Hosts. Updates are communicated on the Amazon EC2 console. You can view information about a Dedicated Host using the following methods.

Console

To view the state of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Locate the Dedicated Host in the list and review the value in the **State** column.

AWS CLI

To view the state of a Dedicated Host

Use the [describe-hosts](#) AWS CLI command and then review the state property in the hostSet response element.

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

To view the state of a Dedicated Host

Use the [Get-EC2Host](#) AWS Tools for Windows PowerShell command and then review the state property in the hostSet response element.

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

The following table explains the possible Dedicated Host states.

State	Description
available	AWS hasn't detected an issue with the Dedicated Host. No maintenance or repairs are scheduled. Instances can be launched onto this Dedicated Host.
released	The Dedicated Host has been released. The host ID is no longer in use. Released hosts can't be reused.
under-assessment	AWS is exploring a possible issue with the Dedicated Host. If action must be taken, you are notified via the AWS Management Console or email. Instances can't be launched onto a Dedicated Host in this state.
pending	The Dedicated Host cannot be used for new instance launches. It is either being modified to support multiple instance types , 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

State	Description
	d 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.

Release Dedicated Hosts

Any running instances on the Dedicated Host must be stopped before you can release the host. These instances can be migrated to other Dedicated Hosts in your account so that you can continue to use them. These steps apply only to On-Demand Dedicated Hosts.

You can release a Dedicated Host using the following methods.

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](#) AWS CLI command.

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

To release a Dedicated Host

Use the [Remove-EC2Hosts](#) AWS Tools for Windows PowerShell command.

```
PS C:\> Remove-EC2Hosts -HostId h-012a3456b7890cdef
```

After you release a Dedicated Host, you can't reuse the same host or host ID again, and you are no longer charged On-Demand billing rates for it. The state of the Dedicated Host is changed to released, and you are not able to launch any instances onto that host.

Note

If you have recently released Dedicated Hosts, it can take some time for them to stop counting towards your limit. During this time, you might experience `LimitExceeded` errors when trying to allocate new Dedicated Hosts. If this is the case, try allocating new hosts again after a few minutes.

The instances that were stopped are still available for use and are listed on the **Instances** page. They retain their host tenancy setting.

Purchase Dedicated Host Reservations

You can purchase reservations using the following methods:

Console

To purchase reservations

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts, Dedicated Host Reservations, Purchase Dedicated Host Reservation**.
3. On the **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](#) AWS CLI command to list the available offerings that match your needs. The following example lists the offerings that support instances in the m4 instance family and have a one-year term.

Note

The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-  
family,Values=m4 --max-duration 31536000
```

The command returns a list of offerings that match your criteria. Note the `offeringId` of the offering to purchase.

2. Use the [purchase-host-reservation](#) AWS CLI command to purchase the offering and provide the `offeringId` noted in the previous step. The following example purchases the specified reservation and associates it with a specific Dedicated Host that is already allocated in the AWS account, and it applies a tag with a key of `purpose` and a value of `production`.

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --  
host-id-set h-013abcd2a00cbd123 --tag-specifications 'ResourceType=host-  
reservation,Tags={Key=purpose,Value=production}'
```

PowerShell

To purchase reservations

1. Use the [Get-EC2HostReservationOffering](#) AWS Tools for Windows PowerShell command to list the available offerings that match your needs. The following examples list the offerings that support instances in the m4 instance family and have a one-year term.

Note

The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

```
PS C:\> $filter = @{"Name"="instance-family"; Value="m4"}
```

```
PS C:\> Get-EC2HostReservationOffering -filter $filter -MaxDuration 31536000
```

The command returns a list of offerings that match your criteria. Note the `offeringId` of the offering to purchase.

2. Use the [New-EC2HostReservation](#) AWS Tools for Windows PowerShell command to purchase the offering and provide the `offeringId` noted in the previous step. The following example purchases the specified reservation and associates it with a specific Dedicated Host that is already allocated in the AWS account.

```
PS C:\> New-EC2HostReservation -OfferingId hro-03f707bf363b6b324 -  
HostIdSet h-013abcd2a00cbd123
```

View Dedicated Host reservations

You can view information about the Dedicated Hosts that are associated with your reservation, including:

- The term of the reservation
- The payment option
- The start and end dates

You can view details of your Dedicated Host reservations using the following methods.

Console

To view the details of a Dedicated Host reservation

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, choose **Dedicated Host Reservations**, and then select the reservation from the list provided.
4. Choose **Details** for information about the reservation.
5. Choose **Hosts** for information about the Dedicated Hosts with which the reservation is associated.

AWS CLI

To view the details of a Dedicated Host reservation

Use the [describe-host-reservations](#) AWS CLI command.

```
aws ec2 describe-host-reservations
```

PowerShell

To view the details of a Dedicated Host reservation

Use the [Get-EC2HostReservation](#) AWS Tools for Windows PowerShell command.

```
PS C:\> Get-EC2HostReservation
```

Tag Dedicated Host Reservations

You can assign custom tags to your Dedicated Host Reservations to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host Reservation based on the custom tags that you assigned.

You can tag a Dedicated Host Reservation using the command line tools only.

AWS CLI

To tag a Dedicated Host Reservation

Use the [create-tags](#) AWS CLI command.

```
aws ec2 create-tags --resources hr-1234563a4ffc669ae --tags Key=Owner,Value=TeamA
```

PowerShell

To tag a Dedicated Host Reservation

Use the [New-EC2Tag](#) AWS Tools for Windows PowerShell command.

The `New-EC2Tag` command needs a `Tag` parameter, which specifies the key and value pair to be used for the Dedicated Host Reservation tag. The following commands create the `Tag` parameter.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource hr-1234563a4ffc669ae -Tag $tag
```

Work with shared Dedicated Hosts

Dedicated Host sharing enables Dedicated Host owners to share their Dedicated Hosts with other AWS accounts or within an AWS organization. This enables you to create and manage Dedicated Hosts centrally, and share the Dedicated Host across multiple AWS accounts or within your AWS organization.

In this model, the AWS account that owns the Dedicated Host (*owner*) shares it with other AWS accounts (*consumers*). Consumers can launch instances onto Dedicated Hosts that are shared with them in the same way that they would launch instances onto Dedicated Hosts that they allocate in their own account. The owner is responsible for managing the Dedicated Host and the instances that they launch onto it. Owners can't modify instances that consumers launch onto shared Dedicated Hosts. Consumers are responsible for managing the instances that they launch onto Dedicated Hosts shared with them. Consumers can't view or modify instances owned by other consumers or by the Dedicated Host owner, and they can't modify Dedicated Hosts that are shared with them.

A Dedicated Host owner can share a Dedicated Host with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization
- Its entire AWS organization

Contents

- [Prerequisites for sharing Dedicated Hosts](#)
- [Limitations for sharing Dedicated Hosts](#)
- [Related services](#)
- [Share across Availability Zones](#)
- [Share a Dedicated Host](#)
- [Unshare a shared Dedicated Host](#)
- [Identify a shared Dedicated Host](#)
- [View instances running on a shared Dedicated Host](#)
- [Shared Dedicated Host permissions](#)
- [Billing and metering](#)
- [Dedicated Host limits](#)
- [Host recovery and Dedicated Host sharing](#)

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>.
2. The Availability Zone IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

Share a Dedicated Host

When an owner shares a Dedicated Host, it enables consumers to launch instances on the host. Consumers can launch as many instances onto the shared host as its available capacity allows.

Important


Note that you are responsible for ensuring that you have appropriate license rights to share any BYOL licenses on your Dedicated Hosts.

If you share a Dedicated Host with auto-placement enabled, keep the following in mind as it could lead to unintended Dedicated Host usage:

- If consumers launch instances with Dedicated Host tenancy and they do not have capacity on a Dedicated Host that they own in their account, the instance is automatically launched onto the shared Dedicated Host.

To share a Dedicated Host, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. You can add the Dedicated Host to an existing resource, or you can add it to a new resource share.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are automatically granted access to the shared Dedicated Host. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Dedicated Host after accepting the invitation.

 **Note**

After you share a Dedicated Host, it could take a few minutes for consumers to have access to it.

You can share a Dedicated Host that you own by using one of the following methods.

Amazon EC2 console

To share a Dedicated Host that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Choose the Dedicated Host to share and choose **Actions, Share host**.
4. Select the resource share to which to add the Dedicated Host and choose **Share host**.

It could take a few minutes for consumers to get access to the shared host.

AWS RAM console

To share a Dedicated Host that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

AWS CLI

To share a Dedicated Host that you own using the AWS CLI

Use the [create-resource-share](#) command.

Unshare a shared Dedicated Host

The Dedicated Host owner can unshare a shared Dedicated Host at any time. When you unshare a shared Dedicated Host, the following rules apply:

- Consumers with whom the Dedicated Host was shared can no longer launch new instances onto it.
- Instances owned by consumers that were running on the Dedicated Host at the time of unsharing continue to run but are scheduled for [retirement](#). Consumers receive retirement notifications for the instances and they have two weeks to take action on the notifications. However, if the Dedicated Host is reshared with the consumer within the retirement notice period, the instance retirements are cancelled.

To unshare a shared Dedicated Host that you own, you must remove it from the resource share. You can do this by using one of the following methods.

Amazon EC2 console

To unshare a shared Dedicated Host that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Choose the Dedicated Host to unshare and choose the **Sharing** tab.
4. The **Sharing** tab lists the resource shares to which the Dedicated Host has been added. Select the resource share from which to remove the Dedicated Host and choose **Remove host from resource share**.

AWS RAM console

To unshare a shared Dedicated Host that you own using the AWS RAM console

See [Updating a Resource Share](#) in the *AWS RAM User Guide*.

Command line

To unshare a shared Dedicated Host that you own using the AWS CLI

Use the [disassociate-resource-share](#) command.

Identify a shared Dedicated Host

Owners and consumers can identify shared Dedicated Hosts using one of the following methods.

Amazon EC2 console

To identify a shared Dedicated Host using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**. The screen lists Dedicated Hosts that you own and Dedicated Hosts that are shared with you. The **Owner** column shows the AWS account ID of the Dedicated Host owner.

Command line

To identify a shared Dedicated Host using the AWS CLI

Use the [describe-hosts](#) command. The command returns the Dedicated Hosts that you own and Dedicated Hosts that are shared with you.

View instances running on a shared Dedicated Host

Owners and consumers can view the instances running on a shared Dedicated Host at any time using one of the following methods.

Amazon EC2 console

To view the instances running on a shared Dedicated Host using the Amazon EC2 console

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

2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host for which to view the instances and choose **Instances**. The tab lists the instances that are running on the host. Owners see all of the instances running on the host, including instances launched by consumers. Consumers only see running instances that they launched onto the host. The **Owner** column shows the AWS account ID of the account that launched the instance.

Command line

To view the instances running on a shared Dedicated Host using the AWS CLI

Use the [describe-hosts](#) command. The command returns the instances running on each Dedicated Host. Owners see all of the instances running on the host. Consumers only see running instances that they launched on the shared hosts. `InstanceOwnerId` shows the AWS account ID of the instance owner.

Shared Dedicated Host permissions

Permissions for owners

Owners are responsible for managing their shared Dedicated Hosts and the instances that they launch onto them. Owners can view all instances running on the shared Dedicated Host, including those launched by consumers. However, owners can't take any action on running instances that were launched by consumers.

Permissions for consumers

Consumers are responsible for managing the instances that they launch onto a shared Dedicated Host. Consumers can't modify the shared Dedicated Host in any way, and they can't view or modify instances that were launched by other consumers or the Dedicated Host owner.

Billing and metering

There are no additional charges for sharing Dedicated Hosts.

Owners are billed for Dedicated Hosts that they share. Consumers are not billed for instances that they launch onto shared Dedicated Hosts.

Dedicated Host Reservations continue to provide billing discounts for shared Dedicated Hosts. Only Dedicated Host owners can purchase Dedicated Host Reservations for shared Dedicated Hosts that they own.

Dedicated Host limits

Shared Dedicated Hosts count towards the owner's Dedicated Hosts limits only. Consumer's Dedicated Hosts limits are not affected by Dedicated Hosts that have been shared with them. Similarly, instances that consumers launch onto shared Dedicated Hosts do not count towards their instance limits.

Host recovery and Dedicated Host sharing

Host recovery recovers instances launched by the Dedicated Host owner and the consumers with whom it has been shared. The replacement Dedicated Host is allocated to the owner's account. It is added to the same resource shares as the original Dedicated Host, and it is shared with the same consumers.

For more information, see [Host recovery](#).

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](#)
- [Supported features](#)
- [Considerations](#)
- [Allocate and use a Dedicated Host on an Outpost](#)

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 [Instance capacity configurations](#).
- Dedicated Hosts on Outposts support auto-placement and targeted instance launches. For more information, see [Understand auto-placement and affinity](#).
- Dedicated Hosts on Outposts support host affinity. For more information, see [Understand auto-placement and affinity](#).
- Dedicated Hosts on Outposts support sharing with AWS RAM. For more information, see [Work with shared Dedicated Hosts](#).

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 and use a Dedicated Host on an Outpost

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:

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 Dedicated Hosts](#).

Note


Availability Zone and **Outpost ARN** should be pre-populated with the Availability Zone and ARN of the selected Outpost.

4. Choose **Allocate**.

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 Dedicated Hosts](#).
7. Choose **Allocate**.

AWS CLI

Use the [allocate-hosts](#) AWS CLI 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
```

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 instances onto a Dedicated Host](#).

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

- [Host recovery basics](#)
- [Supported instance types](#)
- [Configure host recovery](#)
- [Host recovery states](#)
- [Manually recover unsupported instances](#)
- [Related services](#)
- [Pricing](#)

Host recovery basics

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*.

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, P2, P3, R3, R4, R5, R5b, R5n, R6g, R6i, T3, X1, X1e, X2iezn, u-6tb1, u-9tb1, u-12tb1, u-18tb1, and u-24tb1.

To recover instances that are not supported, see [Manually recover unsupported instances](#).

Note

Dedicated Host auto recovery of supported metal [instance types](#) will take longer to detect and recover from than non-metal instance types.

Configure host recovery

You can configure host recovery at the time of Dedicated Host allocation, or after allocation using the Amazon EC2 console or AWS Command Line Interface (CLI).

Contents

- [Enable host recovery](#)
- [Disable host recovery](#)
- [View the host recovery configuration](#)

Enable host recovery

You can enable host recovery at the time of Dedicated Host allocation or after allocation.

For more information about enabling host recovery at the time of Dedicated Host allocation, see [Allocate Dedicated Hosts](#).

To enable host recovery after allocation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host for which to enable host recovery, and then choose **Actions, Modify Host Recovery**.
4. For **Host recovery**, choose **Enable**, and then choose **Save**.

To enable host recovery after allocation using the AWS CLI

Use the [modify-hosts](#) command and specify the `host-recovery` parameter.

```
$ aws ec2 modify-hosts --host-recovery on --host-ids h-012a3456b7890cdef
```

Disable host recovery

You can disable host recovery at any time after the Dedicated Host has been allocated.

To disable host recovery after allocation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host for which to disable host recovery, and then choose **Actions, Modify Host Recovery**.
4. For **Host recovery**, choose **Disable**, and then choose **Save**.

To disable host recovery after allocation using the AWS CLI

Use the [modify-hosts](#) command and specify the `host-recovery` parameter.

```
$ aws ec2 modify-hosts --host-recovery off --host-ids h-012a3456b7890cdef
```

View the host recovery configuration

You can view the host recovery configuration for a Dedicated Host at any time.

To view the host recovery configuration for a Dedicated Host using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host, and in the **Description** tab, review the **Host Recovery** field.

To view the host recovery configuration for a Dedicated Host using the AWS CLI

Use the [describe-hosts](#) command.

```
$ aws ec2 describe-hosts --host-ids h-012a3456b7890cdef
```

The `HostRecovery` response element indicates whether host recovery is enabled or disabled.

Host recovery states

When a Dedicated Host failure is detected, the impaired Dedicated Host enters the `under-assessment` state, and all of the instances enter the `impaired` state. You can't launch instances on to the impaired Dedicated Host while it is in the `under-assessment` state.

After the replacement Dedicated Host is allocated, it enters the `pending` state. It remains in this state until the host recovery process is complete. You can't launch instances on to the replacement Dedicated Host while it is in the `pending` state. Recovered instances on the replacement Dedicated Host remain in the `impaired` state during the recovery process.

After the host recovery is complete, the replacement Dedicated Host enters the `available` state, and the recovered instances return to the `running` state. You can launch instances on to the replacement Dedicated Host after it enters the `available` state. The original impaired Dedicated Host is permanently released and it enters the `released-permanent-failure` state.

If the impaired Dedicated Host has instances that do not support host recovery, such as instances with instance store-backed volumes, the Dedicated Host is not released. Instead, it is marked for retirement and enters the `permanent-failure` state.

Manually recover unsupported instances

Host recovery does not support recovering instances that use instance store volumes. Follow the instructions below to manually recover any of your instances that could not be automatically recovered.

Warning

Data on instance store volumes is lost when an instance is stopped, hibernated, or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. To protect data from instance store volumes, back it up to persistent storage before the instance is stopped or terminated.

Manually recover EBS-backed instances

For EBS-backed instances that could not be automatically recovered, we recommend that you manually stop and start the instances to recover them onto a new Dedicated Host. For more

information about stopping your instance, and about the changes that occur in your instance configuration when it's stopped, see [Stop and start 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.

Related services

Dedicated Host integrates with the following services:

- **AWS License Manager**—Tracks licenses across your Amazon EC2 Dedicated Hosts (supported only in Regions in which AWS License Manager is available). For more information, see the [AWS License Manager User Guide](#).

Pricing

There are no additional charges for using host recovery, but the usual Dedicated Host charges apply. For more information, see [Amazon EC2 Dedicated Hosts Pricing](#).

As soon as host recovery is initiated, you are no longer billed for the impaired Dedicated Host. Billing for the replacement Dedicated Host begins only after it enters the available state.

If the impaired Dedicated Host was billed using the On-Demand rate, the replacement Dedicated Host is also billed using the On-Demand rate. If the impaired Dedicated Host had an active Dedicated Host Reservation, it is transferred to the replacement Dedicated Host.

Host maintenance

With host maintenance, your Amazon EC2 instances on the degraded Dedicated Host are automatically rebooted on a replacement Dedicated Host during a scheduled maintenance event. This helps reduce application downtime and offloads the undifferentiated heavy-lift of maintenance to AWS. Host maintenance is also performed for planned and routine Amazon EC2 maintenance.

Host maintenance is supported on all new Dedicated Host allocations made through Amazon EC2 console. For any Dedicated Host in your AWS account or any new Dedicated Host allocated through [AllocateHosts](#) API, you can configure host maintenance for supported Dedicated Hosts. For more information, see [the section called “Configuring host maintenance”](#).

Contents

- [Host maintenance basics](#)
- [Host maintenance versus host recovery](#)
- [Supported instance types](#)
- [Instances on Dedicated Host](#)
- [Configuring host maintenance](#)
- [Maintenance event](#)
- [Host maintenance states](#)
- [Related services](#)
- [Pricing](#)

Host maintenance basics

When degradation is detected on a Dedicated Host, a new Dedicated Host is allocated. Degradation can be caused by degradation of the underlying hardware or detection of certain problematic conditions. Your instances on the degraded Dedicated Host are scheduled to be automatically rebooted on the replacement Dedicated Host.

The replacement Dedicated Host receives a new host ID, but retains the same attributes as the original Dedicated Host. These attributes include the following.

- Auto placement settings
- Availability Zone
- Reservation
- Host affinity
- Host maintenance settings
- Host recovery settings
- Instance type
- Tags

Host maintenance is available in all AWS Regions for all supported Dedicated Hosts. For more information about Dedicated Hosts where host maintenance is not supported, see [the section called “Limitations”](#).

Your degraded Dedicated Host is released after all of your instances have been rebooted to a new Dedicated Host or stopped. You can access your instances on the degraded Dedicated Host before the scheduled maintenance event, but launching instances on the degraded Dedicated Host is not supported.

You can use the replacement Dedicated Host to launch new instances on the host before the scheduled maintenance event. However, some instance capacity on the replacement host is reserved for the instances that must be migrated from the degraded host. You can't launch new instances into this reserved capacity. For more information, see [the section called “Instances on Dedicated Host”](#).

Limitations

- Host maintenance is not supported in AWS Outposts, AWS Local Zones, and AWS Wavelength Zones.
- Host maintenance cannot 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 only supported on specific instance types. For more information, see [the section called “Supported instance types”](#).

Host maintenance versus host recovery

The following table shows the main differences between host recovery and host maintenance.

	Host recovery	Host maintenance
Accessibility	Unreachable	Reachable
State	under-assessment	permanent-failure
Action	Recovery is immediate	Maintenance is scheduled
Scheduling flexibility	Cannot be rescheduled	Can be rescheduled

	Host recovery	Host maintenance
Host Resource Group	Supported	Not supported

For more information about host recovery, see [Host recovery](#).

Supported instance types

Host maintenance is supported for the following instance families:

- General purpose: A1 | M4 | M5 | M5a | M5n | M5zn | M6a | M6g | M6i | M6in | M7a | M7g | M7i | T3
- Compute optimized: C4 | C5 | C5a | C5n | C6a | C6g | C6gn | C6i | C6in | C7g | C7gn | C7i
- Memory optimized: R4 | R5 | R5a | R5b | R5n | R6a | R6g | R6i | R6in | R7a | R7g | R7iz | u-12tb1 | u-18tb1 | u-24tb1 | u-3tb1 | u-6tb1 | u-9tb1 | X2iezn
- Accelerated computing: G3 | G5g | Inf1 | P2 | P3

Instances on Dedicated Host

Amazon EC2 automatically reserves capacity on the replacement host for the instances that will be automatically migrated from the degraded host. Amazon EC2 does not reserve capacity on the replacement host for instances that can't be automatically migrated, such as instances with instance store root volumes. The reserved capacity can't be used to launch new instances.

Note

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.

If you manually stop an instance on the degraded host that can be automatically migrated, the capacity that was reserved for that instance on the replacement host is released and becomes available for use.

During the scheduled maintenance event, the instances on the degraded host are rebooted and migrated into the reserved capacity on the replacement Dedicated Host. The migrated instances retain the same attributes as the ones on your degraded host, including the following.

- Amazon EBS volume attachments
- Elastic IP addresses
- Instance ID
- Instance metadata
- Private IP address

You can **stop and start an instance** on the degraded host at any point before the scheduled maintenance event is initiated. Doing this reboots your instance on to another host, and your instance won't undergo scheduled maintenance. You must update your instance's host affinity to the new host where you want to reboot your instance. If you stop all instances on the degraded host before the maintenance event is initiated, the degraded host is released and the maintenance event is canceled. For more information, see [Stop and start Amazon EC2 instances](#).

 **Note**

The data on any local store volume is not preserved when you stop and start your instance.

Instances with an **instance store volume** as the root device are terminated after the specified termination date. Any data on the instance store volumes is deleted when the instances are terminated. Terminated instances are permanently deleted, and cannot be started again. For instances with instance store volumes as the root device, we recommend launching replacement instances on a different Dedicated Host using the most recent Amazon Machine Image, and migrating all available data to the replacement instances before the specified termination date. For more information, see [Instance retirement](#).

Instances that **cannot be rebooted** automatically are stopped after the specified date. You can start these instances again on a different host. Instances using an Amazon EBS volume as a root device continue to use the same Amazon EBS volume after being started on a new host.

You can set the **order of instance reboot** by rescheduling the start time of an instance reboot in <https://console.aws.amazon.com/ec2/>.

Configuring host maintenance

You can configure host maintenance for all supported Dedicated Hosts via AWS Management Console or AWS CLI. See the following table for more details.

AWS Management Console

To enable host maintenance for your Dedicated Host using AWS Management Console.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host > **Actions** > **Modify host**.
4. Select *on* in the **Host maintenance** field.

To disable host maintenance for your Dedicated Host using AWS Management Console.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host > **Actions** > **Modify host**.
4. Select *off* in the **Host maintenance** field.

To view the host maintenance configuration for your Dedicated Host using AWS Management Console.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host, and in the **Description** tab, review the **Host maintenance** field.

AWS CLI

To enable or disable host maintenance for your new Dedicated Host during allocation using AWS CLI.

Use the [allocate-hosts](#) command.

Enable

```
aws ec2 allocate-hosts --region us-east-1 --quantity 1 --instance-type m3.large --availability-zone us-east-1b --host-maintenance on
```

Disable

```
aws ec2 allocate-hosts --region us-east-1 --quantity 1 --instance-type m3.large --availability-zone us-east-1b --host-maintenance off
```

To enable or disable host maintenance for your existing Dedicated Host using AWS CLI.

Use the [modify-hosts](#) command.

Enable

```
aws ec2 modify-hosts --region us-east-1 --host-maintenance on --host-ids h-0d123456bbf78910d
```

Disable

```
aws ec2 modify-hosts --region us-east-1 --host-maintenance off --host-ids h-0d123456bbf78910d
```

To view the host maintenance configuration for your Dedicated Host using AWS CLI.

Use the [describe-hosts](#) command.

```
aws ec2 describe-hosts --region us-east-1 --host-ids h-0d123456bbf78910d
```

Note

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.

Maintenance event

On detection of degradation, a maintenance event is scheduled 14 days later, to reboot your instances on a new Dedicated Host. You receive an email notification providing details about the degraded host, scheduled maintenance event, and maintenance time slots. For more information, see [View scheduled events](#).

You can reschedule the maintenance event for any day up to seven days after the date of the scheduled event. For more information about rescheduling, see [Reschedule a scheduled event](#).

The maintenance event usually takes a few minutes to complete. In the rare case of unsuccessful event, you receive an email notification to evict the instances on the degraded host within a specified time frame.

Host maintenance states

Your Dedicated Host is set to `permanent-failure` state when degradation is detected. You cannot launch instances on a Dedicated Host in the state of `permanent-failure`. On completion of maintenance event, the degraded host is released and put in the state of `released`, `permanent-failure`.

After detecting degradation on a Dedicated Host and before scheduling a maintenance event, host maintenance automatically allocates a replacement Dedicated Host in your account. This replacement host stays in a `pending` state until a maintenance event is scheduled. Once the maintenance event is scheduled, the replacement Dedicated Host moves to the `available` state.

You can use the replacement Dedicated Host to launch new instances on the host before the scheduled maintenance event. However, some instance capacity on the replacement host is reserved for the instances that must be migrated from the degraded host. You can't launch new instances into this reserved capacity. For more information, see [the section called "Instances on Dedicated Host"](#).

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 maintenance 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.

Track configuration changes

You can use AWS Config to record configuration changes for Dedicated Hosts, and for instances that are launched, stopped, or terminated on them. You can then use the information captured by AWS Config as a data source for license reporting.

AWS Config records configuration information for Dedicated Hosts and instances individually, and pairs this information through relationships. There are three reporting conditions:

- **AWS Config recording status**—When **On**, AWS Config is recording one or more AWS resource types, which can include Dedicated Hosts and Dedicated Instances. To capture the information required for license reporting, verify that hosts and instances are being recorded with the following fields.
- **Host recording status**—When **Enabled**, the configuration information for Dedicated Hosts is recorded.
- **Instance recording status**—When **Enabled**, the configuration information for Dedicated Instances is recorded.

If any of these three conditions are disabled, the icon in the **Edit Config Recording** button is red. To derive the full benefit of this tool, ensure that all three recording methods are enabled. When all three are enabled, the icon is green. To edit the settings, choose **Edit Config Recording**. You are directed to the **Set up AWS Config** page in the AWS Config console, where you can set up AWS Config and start recording for your hosts, instances, and other supported resource types. For more information, see [Setting up AWS Config using the Console](#) in the *AWS Config Developer Guide*.

Note

AWS Config records your resources after it discovers them, which might take several minutes.

After AWS Config starts recording configuration changes to your hosts and instances, you can get the configuration history of any host that you have allocated or released and any instance that you

have launched, stopped, or terminated. For example, at any point in the configuration history of a Dedicated Host, you can look up how many instances are launched on that host, along with the number of sockets and cores on the host. For any of those instances, you can also look up the ID of its Amazon Machine Image (AMI). You can use this information to report on licensing for your own server-bound software that is licensed per-socket or per-core.

You can view configuration histories in any of the following ways:

- By using the AWS Config console. For each recorded resource, you can view a timeline page, which provides a history of configuration details. To view this page, choose the gray icon in the **Config Timeline** column of the **Dedicated Hosts** page. For more information, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- By running AWS CLI commands. First, you can use the [list-discovered-resources](#) command to get a list of all hosts and instances. Then, you can use the [get-resource-config-history](#) command to get the configuration details of a host or instance for a specific time interval. For more information, see [View Configuration Details Using the CLI](#) in the *AWS Config Developer Guide*.
- By using the AWS Config API in your applications. First, you can use the [ListDiscoveredResources](#) action to get a list of all hosts and instances. Then, you can use the [GetResourceConfigHistory](#) action to get the configuration details of a host or instance for a specific time interval.

For example, to get a list of all of your Dedicated Hosts from AWS Config, run a CLI command such as the following.

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Host
```

To obtain the configuration history of a Dedicated Host from AWS Config, run a CLI command such as the following.

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --resource-id i-1234567890abcdef0
```

To manage AWS Config settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, choose **Edit Config Recording**.
3. In the AWS Config console, follow the steps provided to turn on recording. For more information, see [Setting up AWS Config using the Console](#).

For more information, see [Viewing Configuration Details in the AWS Config Console](#).

To activate AWS Config using the command line or API

- AWS CLI: [Viewing Configuration Details \(AWS CLI\)](#) in the *AWS Config Developer Guide*.
- Amazon EC2 API: [GetResourceConfigHistory](#).

Dedicated Instances

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.

	Dedicated Host	Dedicated Instance
Instance capacity sharing	Can share instance capacity with other accounts.	Not supported
Billing	Per-host billing	Per-instance billing
Visibility of sockets, cores, and host ID	Provides visibility of the number of sockets and physical cores	No visibility
Host and instance affinity	Allows you to consistently deploy your instances to the same physical server over time	Not supported
Targeted instance placement	Provides additional visibility and control over how instances are placed on a physical server	Not supported
Automatic instance recovery	Supported. For more information, see Host recovery .	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 [Dedicated Hosts](#).

Topics

- [Dedicated Instance basics](#)
- [Supported features](#)
- [Dedicated Instances limitations](#)
- [Pricing for Dedicated Instances](#)
- [Work with Dedicated Instances](#)

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 [Create a VPC with a dedicated instance 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](#).

Supported features

Dedicated Instances support the following features and AWS service integrations:

Topics

- [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](#) and [On-Demand Capacity Reservations](#).

When you purchase a Dedicated Reserved Instance, you are purchasing the capacity to launch a Dedicated Instance into a VPC at a much reduced usage fee; the price break in the usage charge applies only if you launch an instance with dedicated tenancy. When you purchase a Reserved Instance with default tenancy, it applies only to a running instance with default tenancy; it 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 [Launching Auto Scaling Instances in a VPC](#) 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 [Recover your instance](#).

Dedicated Spot Instances

You can run a Dedicated Spot Instance by specifying a tenancy of `dedicated` when you create a Spot Instance request. For more information, see [Specify a tenancy for your Spot Instances](#).

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 product page](#).

Work with Dedicated Instances

You can create a VPC with an instance tenancy of `dedicated` to ensure that all instances launched into the VPC are Dedicated Instances. Alternatively, you can specify the tenancy of the instance during launch.

Topics

- [Create a VPC with a dedicated instance tenancy](#)
- [Launch Dedicated Instances into a VPC](#)
- [Display tenancy information](#)
- [Change the tenancy of an instance](#)
- [Change the tenancy of a VPC](#)

Create a VPC with a dedicated instance 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`, the instance will always run as a Dedicated Instance on hardware that's dedicated for your use.

For more information about creating a VPC and choosing the tenancy options, see [Create a VPC](#) in the *Amazon VPC User Guide*.

Launch Dedicated Instances into a VPC

You can launch a Dedicated Instance using the Amazon EC2 launch instance wizard.

Console

To launch a Dedicated Instance into a default tenancy VPC using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. 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.

Note

Ensure that you choose an instance type that's supported as a Dedicated Instance. For more information, see [Amazon EC2 Dedicated Instances](#).

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 [Launch an instance using defined parameters](#).
8. Choose **Launch instance**.

Command line

To set the tenancy option for an instance during launch using the command line

- [run-instances](#) (AWS CLI)


- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

For more information about launching an instance with a tenancy of host, see [Launch instances onto a Dedicated Host](#).


Display tenancy information

Console

To display tenancy information for your VPC using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Check the instance tenancy of your VPC in the **Tenancy** column.
4. If the **Tenancy** column isn't displayed, choose settings () in the top-right corner, turn on **Tenancy**, and choose **Confirm**.

To display tenancy information for your instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Check the tenancy of your instance in the **Tenancy** column.
4. If the **Tenancy** column isn't displayed, do one of the following:
 - Choose settings () in the top-right corner, turn on **Tenancy**, and choose **Confirm**.
 - Select the instance. On the **Details** tab near the bottom of the page, under **Host and placement group**, check the value for **Tenancy**.

Command line

To describe the tenancy of your VPC using the command line

- [describe-vpcs](#) (AWS CLI)

- [Get-EC2Vpc](#) (AWS Tools for Windows PowerShell)

To describe the tenancy of your instance using the command line

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

To describe the tenancy value of a Reserved Instance using the command line

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)

To describe the tenancy value of a Reserved Instance offering using the command line

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (AWS Tools for Windows PowerShell)

Change the tenancy of an 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.

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. 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 change the tenancy of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select your instance.
3. Choose **Instance state, Stop instance, Stop**.
4. Choose **Actions, Instance settings, Modify instance placement**.
5. For **Tenancy**, choose whether to run your instance on dedicated hardware or on a Dedicated Host. Choose **Save**.

Command line

To modify the tenancy value of an instance using the command line

- [modify-instance-placement](#) (AWS CLI)
- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

Change the tenancy of a VPC

You can change the instance tenancy of a VPC from `dedicated` to `default` after you create it. Modifying the instance tenancy of the VPC does not affect the tenancy of any existing instances in the VPC. The next time you launch an instance in the VPC, it has a tenancy of `default`, unless you specify otherwise during launch.

Note

You cannot change the instance tenancy of a VPC from `default` to `dedicated` after it is created.

You can modify the instance tenancy of a VPC using the AWS CLI, an AWS SDK, or the Amazon EC2 API only.

Command line

To modify the instance tenancy attribute of a VPC using the AWS CLI

Use the [modify-vpc-tenancy](#) command and specify the ID of the VPC and instance tenancy value. The only supported value is default.

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

Capacity Reservations

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 are running business-critical workloads that 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.

On-Demand Capacity Reservations

On-Demand Capacity Reservations enable you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. Capacity Reservations mitigate against the risk of being unable to get On-Demand capacity in case there are capacity constraints. 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.

You can create Capacity Reservations at any time, without entering into a one-year or three-year term commitment. The capacity becomes available and billing starts as soon as the Capacity Reservation is provisioned in your account. When you no longer need the capacity assurance, cancel the Capacity Reservation to release the capacity and to stop incurring charges. You can also use the billing discounts offered by Savings Plans and Regional Reserved Instances to reduce the cost of a Capacity Reservation.

When you create a Capacity Reservation, you specify:

- The Availability Zone in which to reserve the capacity
- The number of instances for which to reserve capacity
- The instance attributes, including the instance type, tenancy, and platform/OS

Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.

Contents

- [Differences between Capacity Reservations, Reserved Instances, and Savings Plans](#)

- [Supported platforms](#)
- [Quotas](#)
- [Limitations](#)
- [Capacity Reservation pricing and billing](#)
- [Work with Capacity Reservations](#)
- [Work with Capacity Reservation groups](#)
- [Capacity Reservations in cluster placement groups](#)
- [Capacity Reservations in Local Zones](#)
- [Capacity Reservations in Wavelength Zones](#)
- [Capacity Reservations on AWS Outposts](#)
- [Work with shared Capacity Reservations](#)
- [Capacity Reservation Fleets](#)
- [Monitoring Capacity Reservations](#)

Differences between Capacity Reservations, Reserved Instances, and Savings Plans

The following table highlights key differences between Capacity Reservations, Reserved Instances, and Savings Plans:

	Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
Term	No commitment required. Can be created and canceled as needed.	Requires a fixed one-year or three-year commitment		
Capacity benefit	Capacity reserved in a specific Availability Zone.	No capacity reserved.		
Billing discount	No billing discount. †	Provides a billing discount.		

	Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
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](#)
- [Savings Plans User Guide](#)

Supported platforms

You must create the Capacity Reservation with the correct platform to ensure that it properly matches with your instances. Capacity Reservations support the following platforms:

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- 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

When you purchase a Capacity Reservation, you must specify the *platform* that represents the operating system for your instance.

- For SUSE Linux and RHEL distributions, excluding BYOL, you must choose the specific platform. For example, the **SUSE Linux** or **Red Hat Enterprise Linux** platform.
- For all other Linux distributions (including Ubuntu), choose the **Linux/UNIX** platform.
- If you bring your existing RHEL subscription (BYOL), you must choose the **Linux/UNIX** platform.

For more information about the supported Windows platforms, see [Supported platforms](#) in the *Amazon EC2 User Guide for Windows Instances*.

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.

Quotas apply to running instances only. If your instance is pending, stopping, stopped, or hibernated, it does not count towards your 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 [Work with 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.
- Capacity Reservations do not ensure that a hibernated instance can resume after you try to start it.

Capacity Reservation pricing and billing

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.

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](#)
- [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](#) in the *AWS Billing and Cost Management User Guide*.

Work with Capacity Reservations

To start using Capacity Reservations, you create the capacity reservation in the required Availability Zone. Then, you can launch instances into the reserved capacity, view its capacity utilization in real time, and increase or decrease its capacity as needed.

By default, Capacity Reservations automatically match new instances and running instances that have matching attributes (instance type, platform, and Availability Zone). This means that any instance with matching attributes automatically runs in the Capacity Reservation. However, you can also target a Capacity Reservation for specific workloads. This enables you to explicitly control which instances are allowed to run in that reserved capacity.

You can specify how the reservation ends. You can choose to cancel the Capacity Reservation or end it automatically at a specified time. If you specify an end time, the Capacity Reservation is canceled within an hour of the specified time. For example, if you specify 5/31/2019, 13:30:55, the Capacity Reservation is guaranteed to end between 13:30:55 and 14:30:55 on 5/31/2019. After a reservation ends, you can no longer target instances to the Capacity Reservation. Instances running in the reserved capacity continue to run uninterrupted. If instances targeting a Capacity Reservation are stopped, you cannot restart them until you remove their Capacity Reservation targeting preference or configure them to target a different Capacity Reservation.

Contents

- [Create a Capacity Reservation](#)
- [Launch instances into an existing Capacity Reservation](#)
- [Modify a Capacity Reservation](#)
- [Modify an instance's Capacity Reservation settings](#)
- [View a Capacity Reservation](#)
- [Cancel a Capacity Reservation](#)

Create a Capacity Reservation

If your request to create a Capacity Reservation succeeds, the capacity is available immediately. The capacity remains reserved for your use as long as the Capacity Reservation is active, and you can launch instances into it at any time. If the Capacity Reservation is open, new instances and existing instances that have matching attributes automatically run in the capacity of the Capacity Reservation. If the Capacity Reservation is targeted, instances must specifically target it to run in the reserved capacity.

Your request to create a Capacity Reservation could fail if one of the following is true:

- Amazon EC2 does not have sufficient capacity to fulfill the request. Either try again at a later time, try a different Availability Zone, or try a smaller 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](#).

To create a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations**, and then choose **Create Capacity Reservation**.
3. On the Create a Capacity Reservation page, configure the following settings in the **Instance details** section. The instance type, platform, and Availability Zone of the instances that you launch must match the instance type, platform, and Availability Zone that you specify here or the Capacity Reservation is not applied. For example, if an open Capacity Reservation doesn't match, an instance launch that targets that Capacity Reservation explicitly will fail.
 - a. **Instance Type**—The type of instance to launch into the reserved capacity.
 - b. **Launch EBS-optimized instances**—Specify whether to reserve the capacity for EBS-optimized instances. This option is selected by default for some instance types. For more information, see [the section called "EBS optimization"](#).
 - c. **Platform**—The operating system for your instances. For more information, see [Supported platforms](#). For more information about the supported Windows platforms, see [Supported platforms](#) in the *Amazon EC2 User Guide for Windows Instances*.
 - d. **Availability Zone**—The Availability Zone in which to reserve the capacity.
 - e. **Tenancy**—Specify whether to run on shared hardware (default) or a dedicated instance.
 - f. **(Optional) Placement group ARN** —The ARN of the cluster placement group in which to create the Capacity Reservation.

For more information, see [Capacity Reservations in cluster placement groups](#).

- g. **Quantity**—The number of instances for which to reserve capacity. If you specify a quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the request is denied.

4. Configure the following settings in the **Reservation details** section:
 - a. **Reservation Ends**—Choose one of the following options:
 - **Manually**—Reserve the capacity until you explicitly cancel it.
 - **Specific time**—Cancel the capacity reservation automatically at the specified date and time.
 - b. **Instance eligibility**—Choose one of the following options:
 - **open**—(Default) The Capacity Reservation matches any instance that has matching attributes (instance type, platform, and Availability Zone). If you launch an instance with matching attributes, it is placed into the reserved capacity automatically.
 - **targeted**—The Capacity Reservation only accepts instances that have matching attributes (instance type, platform, and Availability Zone), and that explicitly target the reservation.
5. Choose **Request reservation**.

To create a Capacity Reservation using the AWS CLI

Use the [create-capacity-reservation](#) command. For more information, see [Supported platforms](#). For more information about the supported Windows platforms, see [Supported platforms](#) in the *Amazon EC2 User Guide for Windows Instances*.

For example, the following command creates a Capacity Reservation that reserves capacity for three `m5.2xlarge` instances running Red Hat Enterprise Linux AMIs in the `us-east-1a` Availability Zone.

```
aws ec2 create-capacity-reservation --instance-type m5.2xlarge --instance-platform Red Hat Enterprise Linux --availability-zone us-east-1a --instance-count 3
```

Launch instances into an existing Capacity Reservation

When you launch an instance, you can specify whether to launch the instance into any open Capacity Reservation, into a specific Capacity Reservation, or into a group of Capacity Reservations. You can only launch an instance into a Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and sufficient capacity. Alternatively, you can configure the instance to avoid running in a Capacity Reservation, even if you have an open Capacity Reservation that has matching attributes and available capacity.

Launching an instance into a Capacity Reservation reduces its available capacity by the number of instances launched. For example, if you launch three instances, the available capacity of the Capacity Reservation is reduced by three.

To launch instances into an existing Capacity Reservation using the console

1. 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.
 - **Target by ID** – Launches the instances into the selected Capacity Reservation. If the selected Capacity Reservation does not have sufficient capacity for the number of instances you selected, the instance launch fails.
 - **Target by group** – Launches the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group. If the selected group does not have a Capacity Reservation with matching attributes and available capacity, the instances launch into On-Demand capacity.
3. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see [Launch an instance using the new launch instance wizard](#).

To launch an instance into an existing Capacity Reservation using the AWS CLI

Use the [run-instances](#) command and specify the `--capacity-reservation-specification` parameter.

The following example launches a `t2.micro` instance into any open Capacity Reservation that has matching attributes and available capacity:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro
--key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-
specification CapacityReservationPreference=open
```

The following example launches a `t2.micro` instance into a targeted Capacity Reservation:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro
--key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-
specification CapacityReservationTarget={CapacityReservationId=cr-a1234567}
```

The following example launches a `t2.micro` instance into a Capacity Reservation group:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1
--instance-type t2.micro --key-name MyKeyPair --subnet-
id subnet-1234567890abcdef1 --capacity-reservation-specification
CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-1:123456789012:group/my-cr-group}
```

Modify a Capacity Reservation

You can change the attributes of an active Capacity Reservation after you have created it. You cannot modify a Capacity Reservation after it has expired or after you have explicitly canceled it.

When modifying a Capacity Reservation, you can only increase or decrease the quantity and change the way in which it is released. You cannot change the instance type, EBS optimization, platform, Availability Zone, or instance eligibility of a Capacity Reservation. If you need to modify any of these attributes, we recommend that you cancel the reservation, and then create a new one with the required attributes.

If you specify a new quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the update fails.

To modify a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations**, select the Capacity Reservation to modify, and then choose **Edit**.

3. Modify the **Quantity** or **Reservation ends** options as needed, and choose **Save changes**.

To modify a Capacity Reservation using the AWS CLI

Use the [modify-capacity-reservation](#) command:

For example, the following command modifies a Capacity Reservation to reserve capacity for eight instances.

```
aws ec2 modify-capacity-reservation --capacity-reservation-id cr-1234567890abcdef0 --instance-count 8
```

Modify an instance's Capacity Reservation settings

You can modify the following Capacity Reservation settings for a stopped instance at any time:

- Start in any Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and available capacity.
- Start the instance in a specific Capacity Reservation.
- Start in any Capacity Reservation that has matching attributes and available capacity in a Capacity Reservation group
- Prevent the instance from starting in a Capacity Reservation.

To modify an instance's Capacity Reservation settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** and select the instance to modify. Stop the instance if it is not already stopped.
3. Choose **Actions, Modify Capacity Reservation Settings**.
4. For **Capacity Reservation**, choose one of the following options:
 - **Open** — Launches the instances into any Capacity Reservation that has matching attributes and sufficient capacity for the number of instances you selected. If there is no matching Capacity Reservation with sufficient capacity, the instance uses On-Demand capacity.
 - **None** — Prevents the instances from launching into a Capacity Reservation. The instances run in On-Demand capacity.

- **Specify Capacity Reservation** — Launches the instances into the selected Capacity Reservation. If the selected Capacity Reservation does not have sufficient capacity for the number of instances you selected, the instance launch fails.
- **Specify Capacity Reservation group** — Launches the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group. If the selected group does not have a Capacity Reservation with matching attributes and available capacity, the instances launch into On-Demand capacity.

To modify an instance's Capacity Reservation settings using the AWS CLI

Use the [modify-instance-capacity-reservation-attributes](#) command.

For example, the following command changes an instance's Capacity Reservation setting to open or none.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id i-1234567890abcdef0 --capacity-reservation-specification CapacityReservationPreference=none|open
```

For example, the following command modifies an instance to target a specific Capacity Reservation.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id i-1234567890abcdef0 --capacity-reservation-specification CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

For example, the following command modifies an instance to target a specific Capacity Reservation group.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id i-1234567890abcdef0 --capacity-reservation-specification CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-groups:us-west-1:123456789012:group/my-cr-group}
```

View a Capacity Reservation

Capacity Reservations have the following possible states:

- **active**—The capacity is available for use.

- **expired**—The Capacity Reservation expired automatically at the date and time specified in your reservation request. The reserved capacity is no longer available for your use.
- **cancelled**—The Capacity Reservation was canceled. The reserved capacity is no longer available for your use.
- **pending**—The Capacity Reservation request was successful but the capacity provisioning is still pending.
- **failed**—The Capacity Reservation request has failed. A request can fail due to request parameters that are not valid, capacity constraints, or instance limit constraints. You can view a failed request for 60 minutes.

Note

Due to the [eventual consistency](#) model followed by the Amazon EC2 APIs, after you create a Capacity Reservation, it can take up to 5 minutes for the console and the [describe-capacity-reservations](#) response to indicate that the Capacity Reservation is in the active state. During this time, the console and the `describe-capacity-reservations` response might indicate that the Capacity Reservation is in the pending state. However, the Capacity Reservation might already be available for use and you can attempt to launch instances into it.

To view your Capacity Reservations using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations** and select a Capacity Reservation to view.
3. Choose **View launched instances for this reservation**.

To view your Capacity Reservations using the AWS CLI

Use the [describe-capacity-reservations](#) command:

For example, the following command describes all Capacity Reservations.

```
aws ec2 describe-capacity-reservations
```

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"
    }
  ]
}
```

Cancel a Capacity Reservation

You can cancel a Capacity Reservation at any time if you no longer need the reserved capacity. When you cancel a Capacity Reservation, the capacity is released immediately, and it is no longer reserved for your use.

You can cancel empty Capacity Reservations and Capacity Reservations that have running instances. If you cancel a Capacity Reservation that has running instances, the instances continue to run normally outside of the capacity reservation at standard On-Demand Instance rates or at a discounted rate if you have a matching Savings Plan or Regional Reserved Instance.

After you cancel a Capacity Reservation, instances that target it can no longer launch. Modify these instances so that they either target a different Capacity Reservation, launch into any open Capacity Reservation with matching attributes and sufficient capacity, or avoid launching into a Capacity Reservation. For more information, see [Modify an instance's Capacity Reservation settings](#).

To cancel a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations** and select the Capacity Reservation to cancel.
3. Choose **Cancel reservation**, **Cancel reservation**.

To cancel a Capacity Reservation using the AWS CLI

Use the [cancel-capacity-reservation](#) command:

For example, the following command cancels a Capacity Reservation with an ID of `cr-1234567890abcdef0`.

```
aws ec2 cancel-capacity-reservation --capacity-reservation-id cr-1234567890abcdef0
```

Work with Capacity Reservation groups

You can use AWS Resource Groups to create logical collections of Capacity Reservations, called *resource groups*. A resource group is a logical grouping of AWS resources that are all in the same AWS Region. 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, and Availability Zone) 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, and Availability Zone) and available capacity. If the group does not have a Capacity Reservation with matching attributes and available capacity, the instances run using On-Demand capacity. If a matching Capacity Reservation is added to the targeted group at a later stage, the instance is automatically matched with and moved into its reserved capacity.

To prevent unintended use of Capacity Reservations in a group, configure the Capacity Reservations in the group to accept only instances that explicitly target the capacity reservation. To do this, set **Instance eligibility to targeted** (old console) or **Only instances that specify this reservation** (new console) when creating the Capacity Reservation using the Amazon EC2 console. When using the AWS CLI, specify `--instance-match-criteria targeted` when creating the Capacity Reservation. Doing this ensures that only instances that explicitly target the group, or a Capacity Reservation in the group, can run in the group.

If a Capacity Reservation in a group is canceled or expires while it has running instances, the instances are automatically moved to another Capacity Reservation in the group that has matching attributes and available capacity. If there are no remaining Capacity Reservations in the group that have matching attributes and available capacity, the instances run in On-Demand capacity. If a matching Capacity Reservation is added to the targeted group at a later stage, the instance is automatically moved into its reserved capacity.

Topics

- [Create a Capacity Reservation group](#)
- [Add a Capacity Reservation to a group](#)
- [View Capacity Reservations in a group](#)
- [View the groups to which a Capacity Reservation belongs](#)
- [Remove a Capacity Reservation from a group](#)
- [Delete a Capacity Reservation group](#)

Create a Capacity Reservation group

To create a group for Capacity Reservations

Use the [create-group](#) AWS CLI command. For name, provide a descriptive name for the group, and for configuration, specify two Type request parameters:

- `AWS::EC2::CapacityReservationPool` to ensure that the resource group can be targeted for instance launches
- `AWS::ResourceGroups::Generic` with `allowed-resource-types` set to `AWS::EC2::CapacityReservation` to ensure that the resource group accepts Capacity Reservations only

For example, the following command creates a group named `MyCRGroup`.

```
$ aws resource-groups create-group --name MyCRGroup --configuration
'{"Type":"AWS::EC2::CapacityReservationPool"}'
'{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-
types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

The following shows example output.

```
{
  "GroupConfiguration": {
    "Status": "UPDATE_COMPLETE",
    "Configuration": [
      {
        "Type": "AWS::EC2::CapacityReservationPool"
      },
      {
        "Type": "AWS::ResourceGroups::Generic",
        "Parameters": [
          {
            "Values": [
              "AWS::EC2::CapacityReservation"
            ],
            "Name": "allowed-resource-types"
          }
        ]
      }
    ]
  },
  "Group": {
    "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",
    "Name": "MyCRGroup"
  }
}
```

```
}  
}
```

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.

To add a Capacity Reservation to a group

Use the [group-resources](#) AWS CLI command. For `group`, specify the name of the group to which to add the Capacity Reservations, and for `resources`, specify ARNs of the Capacity Reservations to add. To add multiple Capacity Reservations, separate the ARNs with a space. To get the ARNs of the Capacity Reservations to add, use the [describe-capacity-reservations](#) AWS CLI command and specify the IDs of the Capacity Reservations.

For example, the following command adds two Capacity Reservations to a group named `MyCRGroup`.

```
$ aws resource-groups group-resources --group MyCRGroup --resource-arns arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1 arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

The following shows example output.

```
{  
  "Failed": [],  
  "Succeeded": [  
    "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",  
    "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"  
  ]  
}
```

View Capacity Reservations in a group

To view the Capacity Reservations in a specific group

Use the [list-group-resources](#) AWS CLI command. For `group`, specify the name of the group.

For example, the following command lists the Capacity Reservations in a group named `MyCRGroup`.

```
$ aws resource-groups list-group-resources --group MyCRGroup
```

The following shows example output.

```
{
  "QueryErrors": [],
  "ResourceIdentifiers": [
    {
      "ResourceType": "AWS::EC2::CapacityReservation",
      "ResourceArn": "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1"
    },
    {
      "ResourceType": "AWS::EC2::CapacityReservation",
      "ResourceArn": "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/
cr-54321abcdef567890"
    }
  ]
}
```

Note

The command output includes Capacity Reservations that you own and Capacity Reservations that are shared with you.

View the groups to which a Capacity Reservation belongs

AWS CLI

To view the groups to which a specific Capacity Reservation has been added

Use the [get-groups-for-capacity-reservation](#) AWS CLI command.

For example, the following command lists the groups to which Capacity Reservation `cr-1234567890abcdef1` has been added.

```
$ aws ec2 get-groups-for-capacity-reservation --capacity-reservation-
id cr-1234567890abcdef1
```

The following shows example output.

```
{
  "CapacityReservationGroups": [
    {
      "OwnerId": "123456789012",
      "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/
MyCRGroup"
    }
  ]
}
```

Note

If you specify a Capacity Reservation that is shared with you, the command returns only Capacity Reservation groups that you own.

Amazon EC2 console

To view the groups to which a specific Capacity Reservation has been added

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**, select the Capacity Reservation to view, and then choose **View**.

The groups to which the Capacity Reservation has been added are listed in the **Groups** card.

Note

If you choose a Capacity Reservation that is shared with you, the console displays only Capacity Reservation groups that you own.

Remove a Capacity Reservation from a group

To remove a Capacity Reservation from a group

Use the [ungroup-resources](#) AWS CLI command. For `group`, specify the ARN of the group from which to remove the Capacity Reservation, and for `resources` specify the ARNs of the Capacity

Reservations to remove. To remove multiple Capacity Reservations, separate the ARNs with a space.

The following example removes two Capacity Reservations from a group named MyCRGroup.

```
$ aws resource-groups ungroup-resources --group MyCRGroup --resource-arns arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

The following shows example output.

```
{
  "Failed": [],
  "Succeeded": [
    "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd",
    "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
  ]
}
```

Delete a Capacity Reservation group

To delete a group

Use the [delete-group](#) AWS CLI command. For group, provide the name of the group to delete.

For example, the following command deletes a group named MyCRGroup.

```
$ aws resource-groups delete-group --group MyCRGroup
```

The following shows example output.

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",
    "Name": "MyCRGroup"
  }
}
```

Capacity Reservations in 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.

Topics

- [Limitations](#)
- [Work with 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't share Capacity Reservations that have been created in a cluster placement group.
- 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. For information about how to find the ARN of your existing cluster placement group, see [View placement group information](#).

Topics

- [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 the 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. For information about how to find the ARN of your existing cluster placement group, see [View placement group information](#).

You can create the cluster placement group using one of the following methods.

Console**To create a cluster placement group using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**, 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 using the AWS CLI

Use the [create-placement-group](#) command. For `--group-name`, specify a descriptive name for the placement group, and for `--strategy`, specify `cluster`.

The following example creates a placement group named `MyPG` that uses the `cluster` placement strategy.

```
$ aws ec2 create-placement-group \  
  --group-name MyPG \  
  --strategy cluster
```

Make a note of the placement group ARN returned in the command 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. For more information, see [Create a 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. For more information, see [Cluster placement group rules and limitations](#).
- 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](#).

You can create the Capacity Reservation in the cluster placement group using one of the following methods.

Console

To create a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations**, and then choose **Create Capacity Reservation**.
3. On the **Create a Capacity Reservation** page, 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 using the AWS CLI

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 az \
  --instance-count quantity \
  --placement-group-arn placement_group_ARN
```

For more information, see [Create a Capacity Reservation](#).

Step 3: Launch instances into the cluster placement group

You launch an instance into a Capacity Reservation in a cluster placement group in the same way that you launch an instance into any Capacity Reservation. However, you must also specify the ARN of the cluster placement group in which to launch the instance. For more information, see [Create a Capacity Reservation](#).

Considerations

- If the Capacity Reservation is open, you do not need 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.
- 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.
- If the Capacity Reservation is in a Capacity Reservation group, you must specify the target Capacity Reservation group in addition to the cluster placement group in the request. For more information, see [Work with Capacity Reservation groups](#).

You can launch an instance into a Capacity Reservation in a cluster placement group using one of the following methods.

Console

To launch instances into an existing Capacity Reservation using the console

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 instance using the new launch instance wizard](#).

For more information, see [Launch instances into an existing Capacity Reservation](#).

AWS CLI

To launch instances into an existing Capacity Reservation using the AWS CLI

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.

The following command launches an instance into a targeted Capacity Reservation in a cluster placement group.

```
$ aws ec2 run-instances \  
  --image-id ami_id \  
  --count quantity \  
  --instance-type instance_type \  
  --key-name key_pair_name \  
  --subnet-id subnetid \  
  --capacity-reservation-specification  
  CapacityReservationTarget={CapacityReservationId=capacity_reservation_id} \  
  --placement "GroupName=cluster_placement_group_name"
```

For more information, see [Launch instances into an existing Capacity Reservation](#).

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 [Opt in to Local Zones](#).
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-w11-bos-w1z-1`.

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

Considerations

You can't use Capacity Reservation groups in a Wavelength Zone.

To use a Capacity Reservation in a Wavelength Zone

1. Enable the Wavelength Zone for use in your AWS account. For more information, see [Enable Wavelength Zones](#) in the *Amazon EC2 User Guide for Linux Instances*.
2. Create a Capacity Reservation in the Wavelength Zone. For **Availability Zone**, choose the Wavelength. The Wavelength is represented by an AWS Region code followed by an identifier that indicates the location, for example `us-east-1-w11-bos-w1z-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-w11-bos-w1z-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 [Work with 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*.

Work with shared Capacity Reservations

Capacity Reservation sharing enables Capacity Reservation owners to share their reserved capacity with other AWS accounts or within an AWS organization. This enables you to create and manage Capacity Reservations centrally, and share the reserved capacity across multiple AWS accounts or within your AWS organization.

In this model, the AWS account that owns the Capacity Reservation (owner) shares it with other AWS accounts (consumers). Consumers can launch instances into Capacity Reservations that are shared with them in the same way that they launch instances into Capacity Reservations that they own in their own account. The Capacity Reservation owner is responsible for managing the Capacity Reservation and the instances that they launch into it. Owners cannot modify instances that consumers launch into Capacity Reservations that they have shared. Consumers are responsible for managing the instances that they launch into Capacity Reservations shared with them. Consumers cannot view or modify instances owned by other consumers or by the Capacity Reservation owner.

A Capacity Reservation owner can share a Capacity Reservation with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization
- Its entire AWS organization

Contents

- [Prerequisites for sharing Capacity Reservations](#)

- [Related services](#)
- [Share across Availability Zones](#)
- [Share a Capacity Reservation](#)
- [Stop sharing a Capacity Reservation](#)
- [Identify and view a shared Capacity Reservation](#)
- [View shared Capacity Reservation usage](#)
- [Shared Capacity Reservation permissions](#)
- [Billing and metering](#)
- [Instance limits](#)

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>.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

Share a Capacity Reservation

When you share a Capacity Reservation that you own with other AWS accounts, you enable them to launch instances into your reserved capacity. If you share an open Capacity Reservation, keep the following in mind as it could lead to unintended Capacity Reservation usage:

- If consumers have running instances that match the attributes of the Capacity Reservation, have the `CapacityReservationPreference` parameter set to `open`, and are not yet running in reserved capacity, they automatically use the shared Capacity Reservation.
- If consumers launch instances that have matching attributes (instance type, platform, and Availability Zone) and have the `CapacityReservationPreference` parameter set to `open`, they automatically launch into the shared Capacity Reservation.

To share a Capacity Reservation, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share a Capacity Reservation using the Amazon EC2 console, you add it to an existing resource share. To add the Capacity Reservation to a new resource share, you must create the resource share using the [AWS RAM console](#).

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are granted access to the shared Capacity Reservation if the [prerequisites 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.

You can share a Capacity Reservation that you own using the Amazon EC2 console, AWS RAM console, or the AWS CLI.

To share a Capacity Reservation that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**.
3. Choose the Capacity Reservation to share and choose **Actions, Share reservation**.
4. Select the resource share to which to add the Capacity Reservation and choose **Share Capacity Reservation**.

It could take a few minutes for consumers to get access to the shared Capacity Reservation.

To share a Capacity Reservation that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

To share a Capacity Reservation that you own using the AWS CLI

Use the [create-resource-share](#) command.

Stop sharing a Capacity Reservation

The Capacity Reservation owner can stop sharing a Capacity Reservation at any time. The following rules apply:

- Instances owned by consumers that were running in the shared capacity at the time sharing stops continue to run normally outside of the reserved capacity, and the capacity is restored to the Capacity Reservation subject to Amazon EC2 capacity availability.
- Consumers with whom the Capacity Reservation was shared can no longer launch new instances into the reserved capacity.

To stop sharing a Capacity Reservation that you own, you must remove it from the resource share. You can do this using the Amazon EC2 console, AWS RAM console, or the AWS CLI.

To stop sharing a Capacity Reservation that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**.
3. Select the Capacity Reservation and choose the **Sharing** tab.
4. The **Sharing** tab lists the resource shares to which the Capacity Reservation has been added. Select the resource share from which to remove the Capacity Reservation and choose **Remove from resource share**.

To stop sharing a Capacity Reservation that you own using the AWS RAM console

See [Updating a Resource Share](#) in the *AWS RAM User Guide*.

To stop sharing a Capacity Reservation that you own using the AWS CLI

Use the [disassociate-resource-share](#) command.

Identify and view a shared Capacity Reservation

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 AWS CLI. 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 instance will launch into On-Demand capacity.

Owners and consumers can identify and view shared Capacity Reservations using the Amazon EC2 console and AWS CLI.

To identify a shared Capacity Reservation using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**. The screen lists Capacity Reservations that you own and Capacity Reservations that are shared with you. The **Owner** column shows the AWS account ID of the Capacity Reservation owner. (me) next to the AWS account ID indicates that you are the owner.

To identify a shared Capacity Reservation using the AWS CLI

Use the [describe-capacity-reservations](#) command. The command returns the Capacity Reservations that you own and Capacity Reservations that are shared with you. OwnerId shows the AWS account ID of the Capacity Reservation owner.

View shared Capacity Reservation usage

The owner of a shared Capacity Reservation can view its usage at any time using the Amazon EC2 console and the AWS CLI.

To view Capacity Reservation usage using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**.
3. Select the Capacity Reservation for which to view the usage and choose the **Usage** tab.

The **AWS account ID** column shows the account IDs of the consumers currently using the Capacity Reservation. The **Launched instances** column shows the number of instances each consumer currently has running in the reserved capacity.

To view Capacity Reservation usage using the AWS CLI

Use the [get-capacity-reservation-usage](#) command. AccountId shows the account ID of the account using the Capacity Reservation. UsedInstanceCount shows the number of instances the consumer currently has running in the reserved capacity.

Shared Capacity Reservation permissions

Permissions for owners

Owners are responsible for managing and canceling their shared Capacity Reservations. Owners cannot modify instances running in the shared Capacity Reservation that are owned by other accounts. Owners remain responsible for managing instances that they launch into the shared Capacity Reservation.

Permissions for consumers

Consumers are responsible for managing their instances that are running the shared Capacity Reservation. Consumers cannot modify the shared Capacity Reservation in any way, and they cannot view or modify instances that are owned by other consumers or the Capacity Reservation owner.

Billing and metering

There are no additional charges for sharing Capacity Reservations.

The Capacity Reservation owner is billed for instances that they run inside the Capacity Reservation and for unused reserved capacity. Consumers are billed for the instances that they run inside the shared Capacity Reservation.

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 they 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](#)
- [Considerations](#)
- [Pricing](#)
- [Capacity Reservation Fleet concepts](#)
- [Work with Capacity Reservation Fleets](#)
- [Example Capacity Reservation Fleet configurations](#)
- [Using service-linked roles for Capacity Reservation Fleet](#)

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, and Availability Zone) automatically run in the Capacity Reservations created by a Fleet. Capacity Reservation Fleets do not support target 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

This topic describes some of the concepts of Capacity Reservation Fleets.

Topics

- [Total target capacity](#)
- [Allocation strategy](#)
- [Instance type weight](#)
- [Instance type priority](#)

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 ($384 \text{ total target capacity} / 64 \text{ instance type weight} = 6$ 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.xlarge instances or 384 m5.xlarge 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.

Work with Capacity Reservation Fleets

Topics

- [Before you begin](#)
- [Capacity Reservation Fleet states](#)
- [Create a Capacity Reservation Fleet](#)
- [View a Capacity Reservation Fleet](#)
- [Modify a Capacity Reservation Fleet](#)
- [Cancel a Capacity Reservation Fleet](#)

Before you begin

Before you create a Capacity Reservation Fleet:

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.

Capacity Reservation Fleet states

A Capacity Reservation Fleet can be in one of the following states:

- **submitted** — The Capacity Reservation Fleet request has been submitted and Amazon EC2 is preparing to create the Capacity Reservations.
- **modifying** — The Capacity Reservation Fleet is being modified. The Fleet remains in this state until the modification is complete.
- **active** — The Capacity Reservation Fleet has fulfilled its total target capacity and it is attempting to maintain this capacity. The Fleet remains in this state until it is modified or deleted.
- **partially_fulfilled** — The Capacity Reservation Fleet has partially fulfilled its total target capacity. There is insufficient Amazon EC2 capacity to fulfill the total target capacity. The Fleet is attempting to asynchronously fulfill its total target capacity.
- **expiring** — The Capacity Reservation Fleet has reached its end date and it is in the process of expiring. One or more of its Capacity Reservations might still be active.
- **expired** — The Capacity Reservation Fleet has reached its end date. The Fleet and its Capacity Reservations are expired. The Fleet can't create new Capacity Reservations.
- **cancelling** — The Capacity Reservation Fleet is in the process of being cancelled. One or more of its Capacity Reservations might still be active.
- **cancelled** — The Capacity Reservation Fleet has been manually cancelled. The Fleet and its Capacity Reservations are cancelled and the Fleet can't create new Capacity Reservations.
- **failed** — The Capacity Reservation Fleet failed to reserve capacity for the specified instance types.

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.

You can create a Capacity Reservation Fleet using the command line only.

To create a Capacity Reservation Fleet

Use the [create-capacity-reservation-fleet](#) AWS CLI command.

```
$ aws ec2 create-capacity-reservation-fleet \
--total-target-capacity capacity_units \
--allocation-strategy prioritized \
--instance-match-criteria open \
--tenancy dedicated/default \
--end-date yyyy-mm-ddThh:mm:ss.000Z \
--instance-type-specifications file://instanceTypeSpecification.json
```

The following is the contents of `instanceTypeSpecification.json`.

```
[
  {
    "InstanceType": "instance_type",
    "InstancePlatform": "platform",
    "Weight": instance_type_weight,
    "AvailabilityZone": "availability_zone",
    "AvailabilityZoneId" : "az_id",
    "EbsOptimized": true/false,
    "Priority" : instance_type_priority
  }
]
```

Expected output.

```
{
  "Status": "status",
  "TotalFulfilledCapacity": fulfilled_capacity,
```

```
"CapacityReservationFleetId": "cr_fleet_id",
"TotalTargetCapacity": capacity_units
}
```

Example

```
$ 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
```

instanceTypeSpecification.json

```
[
  {
    "InstanceType": "m5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "Weight": 3.0,
    "AvailabilityZone": "us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

Example output.

```
{
  "Status": "submitted",
  "TotalFulfilledCapacity": 0.0,
  "CapacityReservationFleetId": "crf-abcdef01234567890",
  "TotalTargetCapacity": 24
}
```

View a Capacity Reservation Fleet

You can view configuration and capacity information for a Capacity Reservation Fleet at any time. Viewing a Fleet also provides details about the individual Capacity Reservations that are inside the Fleet.

You can view a Capacity Reservation Fleet using the command line only.

To view a Capacity Reservation Fleet

Use the [describe-capacity-reservation-fleets](#) AWS CLI command.

```
$ aws ec2 describe-capacity-reservation-fleets \
--capacity-reservation-fleet-ids cr_fleet_ids
```

Expected output

```
{
  "CapacityReservationFleets": [
    {
      "Status": "status",
      "EndDate": "yyyy-mm-ddThh:mm:ss.000Z",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "CapacityReservationFleetId": "cr_fleet_id",
      "Tenancy": "dedicated/default",
      "InstanceTypeSpecifications": [
        {
          "CapacityReservationId": "cr1_id",
          "AvailabilityZone": "cr1_availability_zone",
          "FulfilledCapacity": cr1_used_capacity,
          "Weight": cr1_instance_type_weight,
          "CreateDate": "yyyy-mm-ddThh:mm:ss.000Z",
          "InstancePlatform": "cr1_platform",
          "TotalInstanceCount": cr1_number of instances,
          "Priority": cr1_instance_type_priority,
          "EbsOptimized": true/false,
          "InstanceType": "cr1_instance_type"
        },
        {
          "CapacityReservationId": "cr2_id",
          "AvailabilityZone": "cr2_availability_zone",
          "FulfilledCapacity": cr2_used_capacity,
          "Weight": cr2_instance_type_weight,
          "CreateDate": "yyyy-mm-ddThh:mm:ss.000Z",
          "InstancePlatform": "cr2_platform",
          "TotalInstanceCount": cr2_number of instances,
          "Priority": cr2_instance_type_priority,
          "EbsOptimized": true/false,

```

```

        "InstanceType": "cr2_instance_type"
      },
    ],
    "TotalTargetCapacity": total_target_capacity,
    "TotalFulfilledCapacity": total_target_capacity,
    "CreateTime": "yyyy-mm-ddThh:mm:ss.000Z",
    "AllocationStrategy": "prioritized"
  }
]
}

```

Example

```

$ aws ec2 describe-capacity-reservation-fleets \
--capacity-reservation-fleet-ids crf-abcdef01234567890

```

Example output

```

{
  "CapacityReservationFleets": [
    {
      "Status": "active",
      "EndDate": "2021-12-31T23:59:59.000Z",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "CapacityReservationFleetId": "crf-abcdef01234567890",
      "Tenancy": "default",
      "InstanceTypeSpecifications": [
        {
          "CapacityReservationId": "cr-1234567890abcdef0",
          "AvailabilityZone": "us-east-1a",
          "FulfilledCapacity": 5.0,
          "Weight": 1.0,
          "CreateDate": "2021-07-02T08:34:33.398Z",
          "InstancePlatform": "Linux/UNIX",
          "TotalInstanceCount": 5,
          "Priority": 1,
          "EbsOptimized": true,
          "InstanceType": "m5.xlarge"
        }
      ],
      "TotalTargetCapacity": 5,
      "TotalFulfilledCapacity": 5.0,
    }
  ]
}

```



```

        "CreateTime": "2021-07-02T08:34:33.397Z",
        "AllocationStrategy": "prioritized"
    }
]
}

```

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.

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 modify a Capacity Reservation Fleet using the command line only.

To modify a Capacity Reservation Fleet

Use the [modify-capacity-reservation-fleet](#) AWS CLI command.

Note

You can't specify `--end-date` and `--remove-end-date` in the same command.

```

$ aws ec2 modify-capacity-reservation-fleet \
  --capacity-reservation-fleet-id cr_fleet_ids \
  --total-target-capacity capacity_units \
  --end-date yyyy-mm-ddThh:mm:ss.000Z \
  --remove-end-date

```

Expected output

```
{
```

```
"Return": true
}
```

Example: Modify total target capacity

```
$ aws ec2 modify-capacity-reservation-fleet \  
--capacity-reservation-fleet-id crf-01234567890abcdef \  
--total-target-capacity 160
```

Example: Modify end date

```
$ aws ec2 modify-capacity-reservation-fleet \  
--capacity-reservation-fleet-id crf-01234567890abcdef \  
--end-date 2021-07-04T23:59:59.000Z
```

Example: Remove end date

```
$ aws ec2 modify-capacity-reservation-fleet \  
--capacity-reservation-fleet-id crf-01234567890abcdef \  
--remove-end-date
```

Example output

```
{  
  "Return": true  
}
```

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 cancelled and the instances that were previously running in the reserved capacity continue to run normally in shared capacity.

You can cancel a Capacity Reservation Fleet using the command line only.

To cancel a Capacity Reservation Fleet

Use the [cancel-capacity-reservation-fleet](#) AWS CLI command.

```
$ aws ec2 cancel-capacity-reservation-fleets \
--capacity-reservation-fleet-ids cr_fleet_ids
```

Expected output

```
{
  "SuccessfulFleetCancellations": [
    {
      "CurrentFleetState": "state",
      "PreviousFleetState": "state",
      "CapacityReservationFleetId": "cr_fleet_id_1"
    },
    {
      "CurrentFleetState": "state",
      "PreviousFleetState": "state",
      "CapacityReservationFleetId": "cr_fleet_id_2"
    }
  ],
  "FailedFleetCancellations": [
    {
      "CapacityReservationFleetId": "cr_fleet_id_3",
      "CancelCapacityReservationFleetError": [
        {
          "Code": "code",
          "Message": "message"
        }
      ]
    }
  ]
}
```

Example: Successful cancellation

```
$ aws ec2 cancel-capacity-reservation-fleets \
--capacity-reservation-fleet-ids crf-abcdef01234567890
```

Example output

```
{
  "SuccessfulFleetCancellations": [
    {
      "CurrentFleetState": "cancelling",
```

```

        "PreviousFleetState": "active",
        "CapacityReservationFleetId": "crf-abcdef01234567890"
    }
],
"FailedFleetCancellations": []
}

```

Example Capacity Reservation Fleet configurations

Topics

- [Example 1: Reserve capacity based on vCPUs](#)

Example 1: Reserve capacity based on vCPUs

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 that were previously created by 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` policy, which includes the following permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeCapacityReservations",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateCapacityReservation",
        "ec2:CancelCapacityReservation",
        "ec2:ModifyCapacityReservation"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:capacity-reservation/*"
      ],
      "Condition": {
        "StringLike": {
          "ec2:CapacityReservationFleet": "arn:aws:ec2:*:*:capacity-
reservation-fleet/crf-*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:capacity-reservation/*"
      ],
      "Condition": {
```

```
        "StringEquals": {
            "ec2:CreateAction": "CreateCapacityReservation"
        }
    }
}
```

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 [Editing a Service-Linked Role](#) 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 [Deleting 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](#).

Monitoring Capacity Reservations

You can use the following features to monitor your Capacity Reservations:

Topics

- [Monitor Capacity Reservations using CloudWatch metrics](#)
- [Monitor Capacity Reservations using EventBridge](#)
- [Utilization notifications](#)

Monitor Capacity Reservations using 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.

On-Demand Capacity Reservations send metric data to CloudWatch every five minutes. Metrics are not supported for Capacity Reservations that are active for less than five minutes.

For more information about viewing metrics in the CloudWatch console, see [Using Amazon CloudWatch Metrics](#). For more information about creating alarms, see [Creating Amazon CloudWatch Alarms](#).

Contents

- [Capacity Reservation usage metrics](#)
- [Capacity Reservation metric dimensions](#)
- [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.

Dimension	Description
CapacityReservationId	This globally unique dimension filters the data you request for the identified capacity reservation only.

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 the metric dimension **By Capacity Reservation**. Metrics will be grouped by CapacityReservationId.
6. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric.

To view Capacity Reservation metrics (AWS CLI)

Use the following [list-metrics](#) command:

```
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

Monitor Capacity Reservations using EventBridge

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).

Contents

- [Events](#)
- [Create an EventBridge rule](#)

Events

AWS Health sends the following events when capacity usage for a Capacity Reservation is below 20 percent.

Events

- [AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION](#)
- [AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY](#)

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**.

Utilization 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 Reservations using EventBridge](#).

Capacity Blocks for ML

Capacity Blocks for ML allow you to reserve highly sought-after GPU 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, nonblocking networking.

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.

Capacity Blocks are currently available for p5.48xlarge and p4d.24xlarge instances. The p5.48xlarge instances are available in the US East (Ohio) and US East (N. Virginia) Regions. The p4d.24xlarge instances are available in the US East (Ohio) and the US West (Oregon) Regions. You can reserve a Capacity Block with a reservation start time up to eight weeks in the future.

You can use Capacity Blocks to reserve p5 and p4d instances with the following reservation duration and instance quantity options.

- Reservation durations for **1-day increments up 14 days total**
- Reservation instance quantity options of **1, 2, 4, 8, 16, 32, or 64 instances**

To reserve a Capacity Block, you start by specifying your capacity needs, including the instance type, the number of instances, 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 before the end time of the Capacity Block. With 30 minutes 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. The last 30 minutes of the reservation are not charged in the price of the Capacity Block. We emit an event through EventBridge 10 minutes before the termination process begins. For more information, see [Monitor Capacity Blocks with EventBridge](#).

Topics

- [Supported platforms](#)
- [Considerations](#)

- [Related resources](#)
- [Capacity Blocks pricing and billing](#)
- [Work with Capacity Blocks](#)
- [Monitor Capacity Blocks](#)

Supported platforms

Capacity Blocks for ML currently support `p5.48xlarge` and `p4d.24xlarge` instances with default tenancy. 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.

- Capacity Blocks start and 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 modifications and cancellations aren't allowed.
- Capacity Blocks can't be shared across AWS accounts or within your AWS Organization.
- Capacity Blocks can't be used in a capacity reservation group.
- The total number of instances that can be reserved in Capacity Blocks across all accounts in your AWS Organization can't exceed 64 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](#).
- Capacity Blocks currently can't be used with Amazon EKS managed node groups or Karpenter. For more information about how to create an Amazon EKS self-managed node group, see [Capacity Blocks for ML](#) in the Amazon EKS User Guide.

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 into 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

Topics

- [Pricing](#)
- [Billing](#)

Pricing

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 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 within 12 hours, 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 [Pricing](#). 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 than 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.

Work with Capacity Blocks

To start using Capacity Blocks, you first find and purchase an available Capacity Block that matches your reservation size, duration, and timing needs. Then, when the reservation begins, you can use the Capacity Block by launching instances that target the reservation ID. Thirty minutes before the reservation expires, we begin to terminate any instances that are still running in the Capacity Block.

Capacity Blocks are delivered as targeted Capacity Reservations in a single Availability Zone. To run instances in a Capacity Block, you must specify the reservation ID when launching your instances. If you stop instances on your own and the Capacity Block expires, you can't restart them until you target another Capacity Block in the active state.

By default, Capacity Blocks deliver low-latency, high-throughput network connectivity between the instances inside the Capacity Block, so there is no need to use a cluster placement group with a Capacity Block.

Topics

- [Prerequisites](#)
- [Find and purchase Capacity Blocks](#)
- [Launch instances into Capacity Blocks](#)
- [View Capacity Blocks](#)

Prerequisites


You must use the corresponding AWS Region for the instance type that you want to use. For more information, see [Regions](#).

Capacity Blocks with p5.48xlarge instances are available in following AWS Regions.

Region name	Region code
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1

Capacity Blocks with p4d.24xlarge instances are available in following AWS Regions.

Region name	Region code
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2

 **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 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 select one of the following options.

- **For reservation duration** – Up to 14 days in 1-day increments
- **For instance count** – 1, 2, 4, 8, 16, 32, or 64 instances

If a Capacity Block is available that matches your specifications, we return the details of a single Capacity Block offering. 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 [Pricing](#).

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](#).

You can use one of the following methods to find and purchase a Capacity Block.


Console

To find and purchase a Capacity Block using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, select 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, Purchase Capacity Blocks**.
4. 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](#).
5. Under **Total capacity**, select the number of instances you want to reserve.
6. Under **Duration**, enter the number of days you need the reservation for.
7. Under **Date range to search for Capacity Blocks**, enter the earliest possible start date and latest acceptable end date for your reservation.
8. Choose **Find Capacity Blocks**.
9. 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 lowest priced Capacity Block offering available is shown. To view other Capacity Block offerings, adjust your search inputs and choose **Find Capacity Blocks** again.

10. When you find a Capacity Block offering that you want to purchase, choose **Next**.
11. (Optional) On the **Add tags** page, choose **Add new tag**.
12. The **Review and purchase** page lists the start and end date, duration, total number of instances, and price.

 **Note**

Capacity Blocks can't be modified or canceled after you reserve them.

13. In the popup window **Purchase a Capacity Block**, type confirm, then choose **Purchase**.

AWS CLI

To find a Capacity Block using the AWS CLI

Use the `describe-capacity-block-offerings` command.

The following example searches for a Capacity Block that has 16 p5.48xlarge instances with a date range starting 2023-08-14 and ending 2023-10-22 with a duration of 48 hours. The instance count must be an integer from a predefined set of options 1, 2, 4, 8, 16, 32, 64. The capacity duration must be an integer that is a multiple of 24 between 24 and 336 indicating the number of days in hours.

```
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 48
```

To purchase a Capacity Block using the AWS CLI

Use the `purchase-capacity-block` command and specify the offering ID of the Capacity Block that you want to purchase and the instance platform.

```
aws ec2 purchase-capacity-block \  
--capacity-block-offering-id cbr-0123456789abcdefg \  
--instance-platform Linux/UNIX
```

Launch instances into 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](#).

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.

The following steps explain how to launch instances into a Capacity Block in the `active` state using the AWS Management Console or the AWS CLI.

For information about how to set up an EKS node group to automatically use a Capacity Block when it begins, see [Capacity Blocks for ML](#) in the Amazon EKS User Guide.

For information about how to launch instances into a Capacity Block using EC2 Fleet, see [Tutorial: Launch instances into Capacity Blocks](#).

For information about how to create a Launch Template targeting a Capacity Block, see [Launch an instance from a launch template](#).

You can use one of the following methods to launch instances into a Capacity Block.

Console

To launch instances into a Capacity Block using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, select the Region for your Capacity Block reservation.
3. From the Amazon EC2 console dashboard, choose **Launch instance**.
4. (Optional) Under **Name and tags**, you can name your instance and tag the instance. For information about tags, see [Tag your Amazon EC2 resources](#)
5. Under **Application and OS Images**, select an Amazon Machine Image (AMI).
6. Under **Instance type**, select the instance type that matches your Capacity Block reservation.
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](#).
8. Under **Network settings**, use the default settings, or choose **Edit** to configure the network settings as necessary.

Important

Your instance can't launch in a subnet in a different Availability Zone from the Availability Zone where your Capacity Block is located.

9. Under **Advanced details**, configure the instance as follows.
 - a. Under **Purchasing option (market type)**, select **Capacity Blocks**.
 - b. Under **Capacity Reservation**, select **Target by ID**.
 - c. Select the Capacity Reservation ID of your Capacity Block reservation.
10. On the **Summary** panel, for **Number of instances**, enter the number of instances to launch.
11. Choose **Launch instance**.

AWS CLI

To launch instances into a Capacity Block using the AWS CLI

- Use the `run-instances` command and specify a `MarketType` of `capacity-block` in the `instance-market-options` structure. You must also specify the `capacity-reservation-specification` parameter.

The following example launches a single `p5.48xlarge` instance into an active Capacity Block that has matching attributes and available capacity.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 \  
  --instance-type p5.48xlarge --key-name MyKeyPair \  
  --subnet-id subnet-1234567890abcdef1 \  
  --instance-market-options MarketType='capacity-block' \  
  --capacity-reservation-specification \  
  CapacityReservationTarget={CapacityReservationId=cr-a1234567}
```

View Capacity Blocks

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.


You can use one of the following methods to view your Capacity Block reservation.

Console

To view Capacity Blocks using the console

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 using the AWS CLI

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 using `capacity-block` for the `capacity-reservation-type` parameter.

For example, the following command describes one or more of your Capacity Block reservations in your current AWS Region.

```
aws ec2 describe-capacity-reservations --reservation-type capacity-block
```

Example output.

```
{
  "CapacityReservations": [
    {
      "CapacityReservationId": "cr-12345678",
      "EndDateType": "limited",
      "ReservationType": "capacity-block"
    }
  ]
}
```

```
"AvailabilityZone": "eu-east-2a",
  "InstanceMatchCriteria": "targeted",
  "EphemeralStorage": false,
  "CreateDate": "2023-11-29T14:22:45Z",
  "StartDate": "2023-12-15T12:00:00Z",
  "EndDate": "2023-08-19T12:00:00Z",
  "AvailableInstanceCount": 0,
  "InstancePlatform": "Linux/UNIX",
  "TotalInstanceCount": 16,
  "State": "payment-pending",
  "Tenancy": "default",
  "EbsOptimized": true,
  "InstanceType": "p5.48xlarge"
},
...
```

Monitor Capacity Blocks

Topics

- [Monitor Capacity Blocks with EventBridge](#)
- [Logging Capacity Blocks API calls with AWS CloudTrail](#)

Monitor Capacity Blocks with 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 (N. Virginia):123456789012:instance/
i-1234567890abcdef0"
  }
  {
    "accountId": "123456789012",
    "type": "AWS::EC2::Instance",
    "ARN": "arn:aws::ec2:US East (N. Virginia):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 boto3/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:US East (N. Virginia):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 (N. Virginia):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 (N. Virginia):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 (N. Virginia):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,
}

```

```

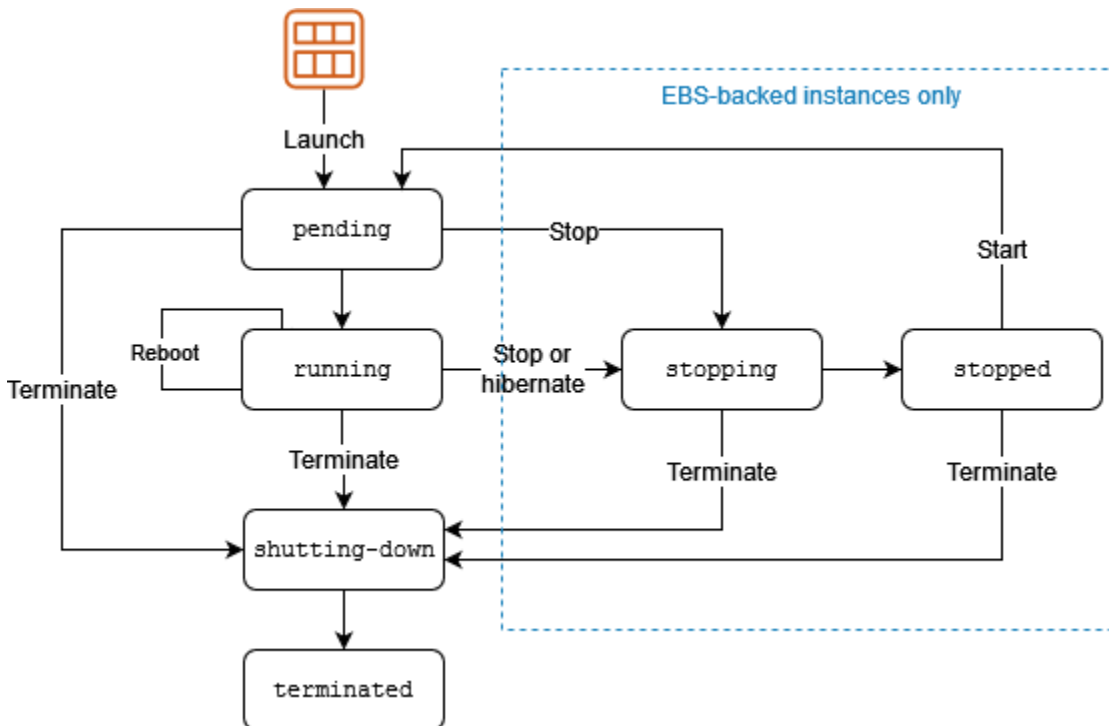
"resources": [
  {
    "ARN": "arn:aws:ec2:US East (N. Virginia):123456789012:capacity-reservation/
cr-12345678",
    "accountId": "123456789012",
    "type": "AWS::EC2::CapacityReservation"
  }
],
"eventType": "AwsServiceEvent",
"recipientAccountId": "123456789012",
"serviceEventDetails": {
  "capacityReservationId": "cr-12345678",
  "capacityReservationState": "expired"
}
}

```

Instance lifecycle


An Amazon EC2 instance transitions through different states from the moment you launch it through to its termination.

The following illustration represents the transitions between instance states. Notice that you can't stop and start an instance store-backed instance. For more information about instance store-backed instances, see [Storage for the root device](#).



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 state. An instance enters the pending state when it is launched or when it is started after being in the stopped state.	Not billed
running	The instance is running and ready for use.	Billed
stopping	The instance is preparing to be stopped.	Not billed
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

Instance state	Description	Instance usage billing
		according to their payment option. For more information, see Reserved Instances

Contents

- [Instance launch](#)
- [Instance stop and start \(Amazon EBS-backed instances only\)](#)
- [Instance hibernate \(Amazon EBS-backed instances only\)](#)
- [Instance reboot](#)
- [Instance termination](#)
- [Differences between reboot, stop, hibernate, and terminate](#)
- [Launch your instance](#)
- [Stop and start Amazon EC2 instances](#)
- [Hibernate your Amazon EC2 instance](#)
- [Reboot your instance](#)
- [Terminate Amazon EC2 instances](#)
- [Instance retirement](#)
- [Recover your instance](#)

Instance launch

When you launch an instance, it enters the pending state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the running state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance transitions to the running state, you're billed for each second, with a one-minute minimum, that you keep the instance running, even if the instance remains idle and you don't connect to it.

For more information, see [Launch your instance](#) and [Connect to your Linux instance](#).

Instance stop and start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the stopping state, and then the stopped state. 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](#).

Instance hibernate (Amazon EBS-backed instances only)

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you start your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.

When you hibernate your instance, it enters the stopping state, and then the stopped state. We don't charge usage for a hibernated instance when it is in the stopped state, but we do charge while it is in the stopping state, unlike when you [stop an instance](#) 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](#).

Instance reboot

You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing period; per second billing continues without a further one-minute minimum charge.

For more information, see [Reboot your instance](#).

Instance termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to shutting-down or terminated, you stop incurring charges for that instance.

If you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You can also describe a terminated instance using the CLI and API. Resources (such as tags) are gradually disassociated from the terminated instance, therefore may

no longer be visible on the terminated instance after a short while. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate shutdown from within the instance itself (for example, by using the **shutdown** command on Linux). The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see [Terminate Amazon EC2 instances](#).

Differences between reboot, stop, hibernate, and terminate

The following table summarizes the key differences between rebooting, stopping, hibernating, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	We move the instance to a new host computer (though in some cases, it remains on the current host).	We move the instance to a new host computer (though in some cases, it remains on the current host).	None
Private and public IPv4 addresses	These addresses stay the same	The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP	The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address,	None

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
		address, which doesn't change during a stop/start.	which doesn't change during a stop/start.	
Elastic IP addresses (IPv4)	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address is disassociated from the instance
IPv6 address	The 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
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

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
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 the instance initiated shutdown behavior](#).

Launch your instance

An instance is a virtual server in the AWS Cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#). You can use the free tier to launch and use a `t2.micro` instance for free for 12 months (in Regions where `t2.micro` is unavailable, you can use a `t3.micro` instance under the free tier). If

you launch an instance that is not within the free tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see [Amazon EC2 pricing](#).

You can launch an instance using the following methods.

Method	Documentation
[Amazon EC2 console] Use the launch instance wizard to specify the launch parameters.	Launch an instance using the old launch instance wizard
[Amazon EC2 console] Create a launch template and launch the instance from the launch template.	Launch an instance from a launch template
[Amazon EC2 console] Use an existing instance as the base.	Launch an instance using parameters from an existing instance
[Amazon EC2 console] Use an AMI that you purchased from the AWS Marketplace.	Launch an AWS Marketplace instance
[AWS CLI] Use an AMI that you select.	Using Amazon EC2 through the AWS CLI
[AWS Tools for Windows PowerShell] Use an AMI that you select.	Amazon EC2 from the AWS Tools for Windows PowerShell
[AWS CLI] Use EC2 Fleet to provision capacity across different EC2 instance types and Availability Zones, and across On-Demand Instance, Reserved Instance, and Spot Instance purchase models.	EC2 Fleet
[AWS CloudFormation] Use a AWS CloudFormation template to specify an instance.	AWS::EC2::Instance in the <i>AWS CloudFormation User Guide</i>
[AWS SDK] Use a language-specific AWS SDK to launch an instance.	AWS SDK for .NET AWS SDK for C++ AWS SDK for Go AWS SDK for Java

Method	Documentation
	AWS SDK for JavaScript
	AWS SDK for PHP V3
	AWS SDK for Python
	AWS SDK for Ruby V3

Note

To launch an EC2 instance into an IPv6-only subnet, you must use [Instances built on the AWS Nitro System](#).

Note

When launching an IPv6-only instance, it is possible that DHCPv6 may not immediately provide the instance with the IPv6 DNS name server. During this initial delay, the instance may not be able to resolve public domains.

For instances running on Amazon Linux 2, if you want to immediately update the `/etc/resolv.conf` file with the IPv6 DNS name server, run the following [cloud-init directive](#) at launch:

```
#cloud-config
bootcmd:
- /usr/bin/sed -i -E 's,^nameserver\s+[\.:digit:]]+$,nameserver
fd00:ec2::253,' /etc/resolv.conf
```

Another option is to change the configuration file and re-image your AMI so that the file has the IPv6 DNS name server address immediately on booting.

When you launch your instance, you can launch your instance in a subnet that is associated with one of the following resources:

- An Availability Zone - This option is the default.

- A Local Zone - To launch an instance in a Local Zone, you must opt in to the Local Zone, and then create a subnet in the zone. For more information, see [Local Zones](#)
- A Wavelength Zone - To launch an instance in a Wavelength Zone, you must opt in to the Wavelength Zone, and then create a subnet in the zone. For information about how to launch an instance in a Wavelength Zone, see [Get started with AWS Wavelength](#) in the *AWS Wavelength Developer Guide*.
- An Outpost - To launch an instance in an Outpost, you must create an Outpost. For information about how to create an Outpost, see [Get Started with AWS Outposts](#) in the *AWS Outposts User Guide*.

After you launch your instance, you can connect to it and use it. To begin, the instance state is pending. When the instance state is running, the instance has started booting. There might be a short time before you can connect to the instance. Note that bare metal instance types might take longer to launch.

The instance receives a public DNS name that you can use to contact the instance from the internet. The instance also receives a private DNS name that other instances within the same VPC can use to contact the instance. For more information about connecting to your instance, see [Connect to your Linux instance](#).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate Amazon EC2 instances](#).

Launch an instance using the new launch instance wizard

You can launch an instance using the new launch instance wizard. The launch instance wizard specifies the launch parameters that are required for launching an instance. Where the launch instance wizard provides a default value, you can accept the default or specify your own value. If you accept the default values, then it's possible to launch an instance by selecting only a key pair.

Before you launch your instance, be sure that you are set up. For more information, see [Set up to use Amazon EC2](#).

Important

When you launch an instance that's not within the [AWS Free Tier](#), you are charged for the time that the instance is running, even if it remains idle.

Topics

- [Quickly launch an instance](#)
- [Launch an instance using defined parameters](#)
- [Launch an instance using the old launch instance wizard](#)

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. For information about all of the parameters in the launch instance wizard, see [Launch an instance using defined parameters](#).

To quickly launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current AWS Region is displayed (for example, US East (Ohio)). Select a Region in which to launch the instance. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations](#).
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

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. The parameters are grouped in the launch instance wizard. The following instructions take you through each parameter group.

Parameters for instance configuration

- [Initiate instance launch](#)
- [Name and tags](#)

- [Application and OS Images \(Amazon Machine Image\)](#)
- [Instance type](#)
- [Key pair \(login\)](#)
- [Network settings](#)
- [Configure storage](#)
- [Advanced details](#)
- [Summary](#)

Initiate instance launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current AWS Region is displayed (for example, US East (Ohio)). Select a Region in which to launch the instance. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations](#).
3. From the Amazon EC2 console dashboard, choose **Launch instance**.

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.

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 is eligible for the 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 [Use a Systems Manager parameter to find an AMI](#).
- 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 AWS Marketplace instance](#). 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 [Storage for the root device](#).
- Check the **Virtualization** type listed for each AMI. Notice which AMIs are the type that you need: either **hvm** or **paravirtual**. For example, some instance types require HVM. For more information, see [Linux AMI virtualization types](#).
- 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 [Boot modes](#).
- Choose an AMI that meets your needs, and then choose **Select**.

Warning when changing the AMI

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.

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](#).

- For **Instance type**, select the instance type for the instance.

Free Tier – If your AWS account is 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). If an instance type is eligible under the Free Tier, it is labeled **Free tier eligible**. For more information about t2.micro and t3.micro, see [Burstable performance instances](#).

- **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 Amazon Q EC2 instance type selector. For more information, see [Get instance type recommendations for a new workload](#).

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

Configure the network settings, as necessary.

- **VPC:** Choose an existing VPC for your instance. You can choose the default VPC or a VPC that you created. For more information, see [the section called "Virtual private clouds"](#).
- **Subnet:** You can launch an instance in a subnet associated with an Availability Zone, Local Zone, Wavelength Zone, or Outpost.

To launch the instance in an Availability Zone, select the subnet in which to launch your instance. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the launch instance wizard and choose the Refresh icon to load your subnet in the list.

To launch the instance in an IPv6-only subnet, the instance must be [built on the Nitro System](#).

To launch the instance in a Local Zone, select a subnet that you created in the Local Zone.

To launch an instance in an Outpost, select a subnet in a VPC that you associated with the Outpost.

- **Auto-assign Public IP:** Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address, and instances in a nondefault subnet

don't. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see [Public IPv4 addresses](#).

- **Firewall (security groups):** Use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. For more information about security groups, see [Amazon EC2 security groups for Linux instances](#).

If you add a network interface, you must specify the same security group in the network interface.

Select or create a security group as follows:

- To select an existing security group for your VPC, choose **Select existing security group**, and select your security group from **Common security groups**.
- To create a new security group for your VPC, choose **Create security group**. The launch instance wizard automatically defines the **launch-wizard-x** security group and provides the following check boxes for quickly adding security group rules:

Allow SSH traffic from – Creates an inbound rule to allow you to connect to your instance over SSH (port 22). Specify whether the traffic comes from **Anywhere**, **Custom**, or **My IP**.

Allow HTTPS traffic from the internet – Creates an inbound rule that opens port 443 (HTTPS) to allow internet traffic from anywhere. If your instance will be a web server, you'll need this rule.

Allow HTTP traffic from the internet – Creates an inbound rule that opens port 80 (HTTP) to allow internet traffic from anywhere. If your instance will be a web server, you'll need this rule.

You can edit these rules and add rules to suit your needs.

To edit or add a rule, choose **Edit** (at top right). To add a rule, choose **Add security group rule**. For **Type**, select the network traffic type. The **Protocol** field is automatically filled in with the protocol to open to network traffic. For **Source type**, select the source type. To let the launch instance wizard add your computer's public IP address, choose **My IP**. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

⚠ Warning

Rules that enable all IP addresses ($0.0.0.0/0$) to access your instance over SSH or RDP are acceptable if you are briefly launching a test instance and will stop or terminate it soon, but are unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

- **Advanced network configuration** – Available only if you choose a subnet.

Network interface

- **Device 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.
- **Network interface:** Select **New interface** to let Amazon EC2 create a new interface, or select an existing, available network interface.
- **Description:** (Optional) A description for the new network interface.
- **Subnet:** The subnet in which to create the new network interface. For the primary network interface (`eth0`), this is the subnet in which the instance is launched. If you've entered an existing network interface for `eth0`, the instance is launched in the subnet in which the network interface is located.
- **Security groups:** One or more security groups in your VPC with which to associate the network interface.
- **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:** One or more additional private IPv4 addresses from the range of your subnet. Choose **Manually assign** and enter an IP address. Choose **Add IP** to add another IP address. Alternatively, choose **Automatically assign** to let Amazon EC2 choose one for you, and enter a value to indicate the number of IP addresses to add.
- (IPv6-only) **IPv6 IPs:** An IPv6 address from the range of the subnet. Choose **Manually assign** and enter an IP address. Choose **Add IP** to add another IP address. Alternatively, choose **Automatically assign** to let Amazon EC2 choose one for you, and enter a value to indicate the number of IP addresses to add.
- **IPv4 Prefixes:** The IPv4 prefixes for the network interface.
- **IPv6 Prefixes:** The IPv6 prefixes for the network interface.

- (Dual-stack and IPv6-only) **Assign Primary IPv6 IP:** (Optional) If you're launching an instance into a dual-stack or IPv6-only subnet, you have the option to **Assign Primary IPv6 IP**. Assigning a primary IPv6 address enables you to avoid disrupting traffic to instances or ENIs. Choose **Enable** if this instance relies on its IPv6 address not changing. When you launch the instance, 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 cannot 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.
- **Delete on termination:** Whether the network interface is deleted when the instance is deleted.
- **Elastic Fabric Adapter:** Indicates whether the network interface is an Elastic Fabric Adapter. For more information, see [Elastic Fabric Adapter](#).
- **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 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 does not include ENA Express configuration for the instance unless you select **Enable** or **Disable**.

Choose **Add network interface** to add additional network interfaces. Additional network interfaces can reside in a different subnet of the same VPC or in a subnet in a different VPC that you own (as long as the subnet is in the same Availability Zone as your instance). If you choose to add an additional network interface residing in another VPC subnet, you'll see the **Multi-VPC subnets** option when you select a subnet. If you select a subnet in another VPC, the **Multi-VPC** label appears next to the network interface you've added. This enables you to create multi-homed instances across VPCs with different networking and security configurations. Note that if you attach an additional ENI from another VPC, you must choose a security group for the ENI from that VPC.

For more information, see [Elastic network interfaces](#). If you specify more than one network interface, your instance cannot receive a public IPv4 address. Additionally, if you specify an existing network interface for eth0, you cannot override the subnet's public IPv4 setting using

Auto-assign Public IP. For more information, see [Assign a public IPv4 address during instance launch](#).

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.

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 [Amazon EC2 instance store](#) 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. If you have selected an AMI and instance that are eligible for the free tier, keep in mind that to stay within the free tier, you must stay under 30 GiB of total storage.
- **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.


- **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](#). For more information about mounting an Amazon FSx file system, see [Use Amazon FSx with Amazon EC2](#)

Advanced details

For **Advanced details**, expand the section to view the fields and specify any additional parameters for the instance.

- **Purchasing option:** Choose **Request Spot Instances** to request Spot Instances at the Spot price, capped at the On-Demand price, and choose **Customize** to change the default Spot Instance settings. 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 [Create a Spot Instance request](#).
- **Domain join directory:** Select the AWS Directory Service directory (domain) to which your Linux instance is joined after launch. If you select a domain, you must select an IAM role with the required permissions. For more information, see [Seamlessly join a Linux EC2 instance to your AWS Managed Microsoft AD directory](#).
- **IAM instance profile:** Select an AWS Identity and Access Management (IAM) instance profile to associate with the instance. 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 [Amazon EC2 instance hostname types](#).
- **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 [Amazon EC2 instance hostname types](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Change the 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 [Enable 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](#).
- **Elastic inference:** An elastic inference accelerator to attach to your EC2 CPU instance. For more information, see [Working with Amazon Elastic Inference](#) in the *Amazon Elastic Inference Developer Guide*.

 **Note**

Starting April 15, 2023, AWS will not onboard new customers to Amazon Elastic Inference (EI), and will help current customers migrate their workloads to options that offer better price and performance. After April 15, 2023, new customers will not be able to launch instances with Amazon EI accelerators in Amazon SageMaker, Amazon ECS, or Amazon EC2. However, customers who have used Amazon EI at least once during the past 30-day period are considered current customers and will be able to continue using the service.

- **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 name:** 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](#).
- **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"](#).
- **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 [Dedicated Instances](#) and [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*.
- **Metadata accessible:** You can enable or disable access to the instance metadata. For more information, see [Configure instance metadata options for new instances](#).
- **Metadata transport:** Enable the instance to reach the link local IMDSv2 IPv6 address (fd00:ec2::254) to retrieve instance metadata. This option is only available if you are launching [instances built on the AWS Nitro System](#) into an [IPv6-only subnet](#). For more information about retrieving instance metadata, see [Retrieve instance metadata](#).
- **Metadata version:** If you enable access to the instance metadata, you can choose to require the use of Instance Metadata Service Version 2 when requesting instance metadata. For more information, see [Configure instance metadata options for new instances](#).
- **Metadata response hop limit:** If you enable instance metadata, you can set the allowable number of network hops for the metadata token. For more information, see [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 [Allow 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, see [Run commands on your Linux instance at launch](#).

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 Auto Scaling instances](#) 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**.

If the instance fails to launch or the state immediately goes to terminated instead of running, see [Troubleshoot instance launch issues](#).

(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 an instance using the old launch instance wizard

You can launch an instance using the old launch instance wizard only if your Region supports the old launch experience. The launch instance wizard specifies all the launch parameters required for launching an instance. Where the launch instance wizard provides a default value, you can accept the default or specify your own value. You must specify an AMI and a key pair to launch an instance.

For the instructions to use the *new* launch instance wizard, see [Launch an instance using the new launch instance wizard](#).

Before you launch your instance, be sure that you are set up. For more information, see [Set up to use Amazon EC2](#).

Important

When you launch an instance that's not within the [AWS Free Tier](#), you are charged for the time that the instance is running, even if it remains idle.

Steps to launch an instance:

- [Initiate instance launch](#)
- [Step 1: Choose an Amazon Machine Image \(AMI\)](#)
- [Step 2: Choose an Instance Type](#)
- [Step 3: Configure Instance Details](#)
- [Step 4: Add Storage](#)
- [Step 5: Add Tags](#)
- [Step 6: Configure Security Group](#)
- [Step 7: Review Instance Launch and Select Key Pair](#)

Initiate instance launch

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

2. In the navigation bar at the top of the screen, the current Region is displayed (for example, US East (Ohio)). Select a Region for the instance that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations](#).
3. From the Amazon EC2 console dashboard, choose **Launch instance**.

Step 1: Choose an Amazon Machine Image (AMI)

When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server, such as Linux, Apache, and your website.

When you launch an instance, you can either select an AMI from the list, or you can select a Systems Manager parameter that points to an AMI ID. For more information, see [Using a Systems Manager parameter to find an AMI](#).

On the **Choose an Amazon Machine Image (AMI)** page, use one of two options to choose an AMI. Either [search the list of AMIs](#), or [search by Systems Manager parameter](#).

By searching the list of AMIs

1. Select the type of AMI to use in the left pane:

Quick Start

A selection of popular AMIs to help you get started quickly. To select an AMI that is eligible for the free tier, choose **Free tier only** in the left pane. These AMIs are marked **Free tier eligible**.

My AMIs

The private AMIs that you own, or private AMIs that have been shared with you. To view AMIs that are shared with you, choose **Shared with me** in the left pane.

AWS Marketplace

An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see [Launch an AWS Marketplace instance](#).

Community AMIs

The AMIs that AWS community members have made available for others to use. To filter the list of AMIs by operating system, choose the appropriate check box under **Operating system**. You can also filter by architecture and root device type.

2. Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need, either `ebs` (backed by Amazon EBS) or `instance-store` (backed by instance store). For more information, see [Storage for the root device](#).
3. Check the **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM. For more information, see [Linux AMI virtualization types](#).
4. Check the **Boot mode** listed for each AMI. Notice which AMIs use the boot mode that you need, either `legacy-bios` or `uefi`. For more information, see [Boot modes](#).
5. Choose an AMI that meets your needs, and then choose **Select**.

By Systems Manager parameter

1. Choose **Search by Systems Manager parameter** (at top right).
2. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.
3. Choose **Search**. The AMIs that match the AMI ID appear in the list.
4. Select the AMI from the list, and choose **Select**.

Step 2: Choose an Instance Type

On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see [Amazon EC2 instance types](#).

To remain eligible for the free tier, choose the **t2.micro** instance type (or the **t3.micro** instance type in Regions where **t2.micro** is unavailable). If an instance type is eligible under the Free Tier, it is labeled **Free tier eligible**. For more information about `t2.micro` and `t3.micro`, see [Burstable performance instances](#).

By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, choose **All generations** from the filter list.

Note

To set up an instance quickly for testing purposes, choose **Review and Launch** to accept the default configuration settings, and launch your instance. Otherwise, to configure your instance further, choose **Next: Configure Instance Details**.

Step 3: Configure Instance Details

On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then choose **Next: Add Storage**:

- **Number of instances:** Enter the number of instances to launch.

Tip

To ensure faster instance launches, break up large requests into smaller batches. For example, create five separate launch requests for 100 instances each instead of one launch request for 500 instances.

- (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).

Note

If Amazon EC2 Auto Scaling marks an instance that is in an Auto Scaling group as unhealthy, the instance is automatically scheduled for replacement where it is terminated and another is launched, and you lose your data on the original instance. An instance is marked as unhealthy if you stop or reboot the instance, or if another event marks the instance as unhealthy. For more information, see [Health checks for Auto Scaling instances](#) in the *Amazon EC2 Auto Scaling User Guide*.

- **Purchasing option:** Choose **Request Spot instances** to launch a Spot Instance. This adds and removes options from this page. You can optionally set your maximum price (not recommended), and optionally change the request type, interruption behavior, and request validity. For more information, see [Create a Spot Instance request](#).
- **Network:** Select the VPC or to create a new VPC, choose **Create new VPC** to go the Amazon VPC console. When you have finished, return to the launch instance wizard and choose **Refresh** to load your VPC in the list.
- **Subnet:** You can launch an instance in a subnet associated with an Availability Zone, Local Zone, Wavelength Zone or Outpost.

To launch the instance in an Availability Zone, select the subnet into which to launch your instance. You can select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose **Refresh** to load your subnet in the list.

To launch the instance in a Local Zone, select a subnet that you created in the Local Zone.

To launch an instance in an Outpost, select a subnet in a VPC that you associated with an Outpost.

- **Auto-assign Public IP:** Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address and instances in a nondefault subnet don't. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see [Public IPv4 addresses](#).
- **Auto-assign IPv6 IP:** Specify whether your instance receives an IPv6 address from the range of the subnet. Select **Enable** or **Disable** to override the subnet's default setting. This option is only available if you've associated an IPv6 CIDR block with your VPC and subnet. For more information, see [Add an IPv6 CIDR block to your VPC](#) in the *Amazon VPC User Guide*.
- **Hostname type:** Select whether the guest OS hostname of the instance will include the resource name or the IP name. For more information, see [Amazon EC2 instance hostname types](#).
- **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 [Amazon EC2 instance hostname types](#).
- **Domain join directory:** Select the AWS Directory Service directory (domain) to which your Linux instance is joined after launch. If you select a domain, you must select an IAM role with the required permissions. For more information, see [Seamlessly join a Linux EC2 instance to your AWS Managed Microsoft AD directory](#).

- **Placement group:** A placement group determines the placement strategy of your instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see [Placement groups](#).
- **Capacity Reservation:** Specify whether to launch the instance into shared capacity, any open Capacity Reservation, a specific Capacity Reservation, or a Capacity Reservation group. For more information, see [Launch instances into an existing Capacity Reservation](#).
- **IAM role:** Select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see [IAM roles for Amazon EC2](#).
- **CPU options:** Choose **Specify CPU options** to specify a custom number of vCPUs during launch. Set the number of CPU cores and threads per core. For more information, see [Optimize CPU options](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Change the instance initiated shutdown behavior](#).
- **Stop - Hibernate behavior:** To enable hibernation, select this check box. This option is only available if your instance meets the hibernation prerequisites. For more information, see [Hibernate your Amazon EC2 instance](#).
- **Enable termination protection:** To prevent accidental termination, select this check box. For more information, see [Enable termination protection](#).
- **Enable stop protection:** To prevent accidental stopping, select this check box. For more information, see [Enable stop protection](#).
- **Monitoring:** Select this check box to turn on detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitor your instances using CloudWatch](#).
- **EBS-optimized instance:** An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply. For more information, see [Amazon EBS-optimized instances](#).
- **Tenancy:** If you are launching your instance into a VPC, you can choose to run your instance on isolated, dedicated hardware (**Dedicated**) or on a Dedicated Host (**Dedicated host**). Additional charges may apply. For more information, see [Dedicated Instances](#) and [Dedicated Hosts](#).
- **T2/T3 Unlimited:** Select this check box to enable applications to burst beyond the baseline for as long as needed. Additional charges may apply. For more information, see [Burstable performance instances](#).

- **File systems:** To create a new file system to mount to your instance, choose **Create new file system**, enter a name for the new file system, and then choose **Create**. The file system is created using Amazon EFS Quick Create, which applies the service recommended settings. The security groups required to enable access to the file system are automatically created and attached to the instance and the mount targets of the file system. You can also choose to manually create and attach the required security groups. To mount one or more existing Amazon EFS file systems to your instance, choose **Add file system** and then choose the file systems to mount and the mount points to use. For more information, see [Use Amazon EFS with Amazon EC2](#).
- **Network interfaces:** If you selected a specific subnet, you can specify up to two network interfaces for your instance:
 - For **Network Interface**, select **New network interface** to let AWS create a new interface, or select an existing, available network interface.
 - For **Primary IP**, enter a private IPv4 address from the range of your subnet, or leave **Auto-assign** to let AWS choose a private IPv4 address for you.
 - For **Secondary IP addresses**, choose **Add IP** to assign more than one private IPv4 address to the selected network interface.
 - (IPv6-only) For **IPv6 IPs**, choose **Add IP**, and enter an IPv6 address from the range of the subnet, or leave **Auto-assign** to let AWS choose one for you.
 - **Network Card Index:** The index of the network card. The primary network interface must be assigned to network card index **0**. Some instance types support multiple network cards.
 - Choose **Add Device** to add a secondary network interface. A secondary network interface can reside in a different subnet of the VPC, provided it's in the same Availability Zone as your instance.

For more information, see [Elastic network interfaces](#). If you specify more than one network interface, your instance cannot receive a public IPv4 address. Additionally, if you specify an existing network interface for eth0, you cannot override the subnet's public IPv4 setting using **Auto-assign Public IP**. For more information, see [Assign a public IPv4 address during instance launch](#).

- **Kernel ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific kernel.
- **RAM disk ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.

- **Enclave:** Select **Enable** to enable the instance for AWS Nitro Enclaves. For more information, see [What is AWS Nitro Enclaves?](#) in the *AWS Nitro Enclaves User Guide*.
- **Metadata accessible:** You can enable or disable access to the Instance Metadata Service (IMDS). For more information, see [Use IMDSv2](#).
- **Metadata transport:** Enable the instance to reach the link local IMDSv2 IPv6 address (fd00:ec2::254) to retrieve instance metadata. This option is only available if you are launching [instances built on the AWS Nitro System](#) into an [IPv6-only subnet](#). For more information about retrieving instance metadata, see [Retrieve instance metadata](#).
- **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 token 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 [Use IMDSv2](#).
- **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.

Step 4: Add Storage

The AMI you selected includes one or more volumes of storage, including the root device volume. On the **Add Storage** page, you can specify additional volumes to attach to the instance by choosing **Add New Volume**. Configure each volume as follows, and then choose **Next: Add Tags**.

- **Type:** Select instance store or Amazon EBS volumes to associate with your instance. The types of volume available in the list depend on the instance type you've chosen. For more information, see [Amazon EC2 instance store](#) and [Amazon EBS volumes](#).
- **Device:** Select from the list of available device names for the volume.
- **Snapshot:** Enter the name or ID of the snapshot from which to restore a volume. You can also search for available shared and public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.
- **Size:** For EBS volumes, you can specify a storage size. Even if you have selected an AMI and instance that are eligible for the free tier, to stay within the free tier, you must stay under 30 GiB of total storage.
- **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, select this check box to delete the volume when the instance is terminated. For more information, see [Preserve data when an instance is terminated](#).
- **Encrypted:** If the instance type supports EBS encryption, you can specify the encryption state of the volume. If you have enabled encryption by default in this Region, the default customer managed key is selected for you. You can select a different key or disable encryption. For more information, see [Amazon EBS encryption](#) in the *Amazon EBS User Guide*.

Step 5: Add Tags

On the **Add Tags** page, specify [tags](#) by providing key and value combinations. You can tag the instance, the volumes, or both. For Spot Instances, you can tag the Spot Instance request only. Choose **Add another tag** to add more than one tag to your resources. Choose **Next: Configure Security Group** when you are done.

Step 6: Configure Security Group

On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see [Amazon EC2 security groups for Linux instances](#).) Select or create a security group as follows, and then choose **Review and Launch**.

- To select an existing security group, choose **Select an existing security group**, and select your security group. You can't edit the rules of an existing security group, but you can copy them to a new group by choosing **Copy to new**. Then you can add rules as described in the next step.
- To create a new security group, choose **Create a new security group**. The wizard automatically defines the **launch-wizard-x** security group and creates an inbound rule to allow you to connect to your instance over SSH (port 22).
- You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow internet traffic.

To add a rule, choose **Add Rule**, select the protocol to open to network traffic, and then specify the source. Choose **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

⚠ Warning

Rules that enable all IP addresses ($0.0.0.0/0$) to access your instance over SSH or RDP are acceptable for this short exercise, but are unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

Step 7: Review Instance Launch and Select Key Pair

On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.

When you are ready, choose **Launch**.

In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up. For more information, see [Amazon EC2 key pairs and Amazon EC2 instances](#).

⚠ Important

If you choose the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

(Optional) You can create a status check alarm for the instance (additional fees may apply). On the confirmation screen, choose **Create status check alarms** and follow the directions. Status check alarms can also be created after you launch the instance. For more information, see [Create and edit status check alarms](#).

If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see [Troubleshoot instance launch issues](#).

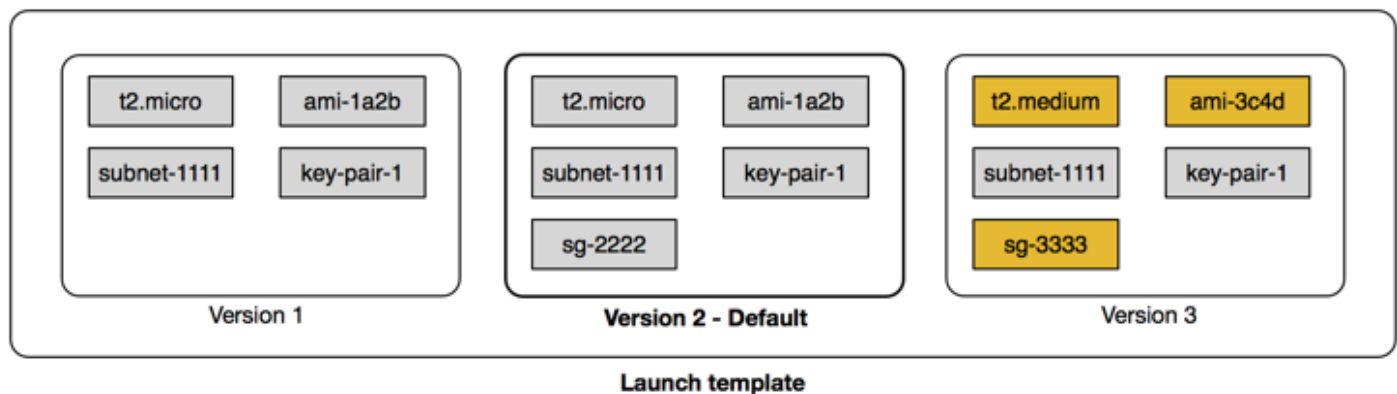
Launch an instance from a launch template

You can use a *launch template* to store instance launch parameters so that you do not have to specify them every time you launch an instance. For example, you can create a launch template

with 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 do not specify a version, the default version is used. You can set any version of the launch template as the default version—by default, it's the first version of the launch template.

The following diagram shows a launch template with three versions. The first version specifies the instance type, AMI ID, subnet, and key pair to use to launch the instance. The second version is based on the first version and also specifies a security group for the instance. The third version uses different values for some of the parameters. Version 2 is set as the default version. If you launched an instance from this launch template, the launch parameters from version 2 would be used if no other version were specified.



Contents

- [Launch template restrictions](#)
- [Control access to launch templates with IAM permissions](#)
- [Use launch templates to control launching instances](#)
- [Create a launch template](#)
- [Modify a launch template \(manage launch template versions\)](#)
- [Delete a launch template](#)
- [Launch instances from a launch template](#)

Launch template restrictions

The following rules 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 request to launch an instance includes all the required parameters. For example, if your launch template does not include an AMI ID, you must specify both the launch template and an AMI ID when you launch an instance.
- **Parameters not validated** – Launch template parameters are not fully validated when you create the launch template. If you specify incorrect values for parameters, or if you do not use supported parameter combinations, no instances can launch using this launch template. Ensure that you specify the correct values for the parameters and that you use supported parameter combinations. For example, to launch an instance in a placement group, you must specify a supported instance type.
- **Tags** – You can tag a launch template, but you cannot 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 cannot specify the version number yourself.

Control access to launch templates with IAM permissions

You can use IAM permissions to control which launch template actions users can perform, such as viewing, creating, or deleting launch templates.

When you grant users permission to create launch templates and launch template versions, you can't use resource-level permissions to restrict the resources that they can specify in a launch template. Therefore, ensure that you grant permissions to create launch templates and launch template versions only to appropriate administrators.

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

- [ec2:CreateLaunchTemplate](#)
- [ec2:DescribeLaunchTemplates](#)
- [ec2:DescribeLaunchTemplateVersions](#)
- [ec2>DeleteLaunchTemplate](#)
- [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 resources during creation](#).

```
{
  "Sid": "IAMPolicyForTaggingLaunchTemplatesOnCreation",
  "Action": "ec2:CreateTags",
  "Effect": "Allow",
  "Resource": "arn:aws:ec2:region:account-id: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 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 all launch templates in the account.

```
{
  "Sid": "IAMPolicyForDescribingLaunchTemplates",
  "Action": "ec2:DescribeLaunchTemplates",
  "Effect": "Allow",
  "Resource": "*"
}
```

ec2:DescribeLaunchTemplateVersions

Principals who 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 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 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 uses 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:region:account-id: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:region:account-id: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 launch templates to control launching instances

You can specify 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 launch an instance. When you launch an instance using a launch template, you can override parameters that are specified in the launch template. Or, you can specify additional parameters that are not in the launch template.

Note

You cannot remove launch template parameters during launch (for example, you 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 permissions 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 cannot use resource-level permissions to

control which resources users can specify in the launch template. To restrict the resources that are used to launch an instance, ensure that you grant permissions to create launch templates and launch template versions only to appropriate administrators.

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 a launch template

Create a launch template using parameters that you define, or use an existing launch template or an instance as the basis for a new launch template.

Tasks

- [Create a launch template from 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 from parameters

To create a launch template, you must specify the launch template name and at least one instance configuration parameter.

Console directions

To create a launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. The launch template parameters are grouped. For details about each group see the sections below.
4. 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.
5. When you're ready to create your launch template, choose **Create launch template**.

Launch template name, description, and tags

1. For **Launch template name**, enter a descriptive name for the launch template.
2. For **Template version description**, provide a brief description of this version of the launch template.
3. To [tag](#) the launch template on creation, expand **Template tags**, choose **Add tag**, and then enter a tag key and value pair. Choose **Add 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 [Resource tags](#).

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 template 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 is eligible for the 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 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 [Use a Systems Manager parameter to find an AMI](#).
- To specify a Systems Manager parameter that will resolve to an AMI at the time an instance is launched from the launch template, choose the arrow button to the right of the search bar, and then choose **Specify custom value/Systems Manager parameter**. For more information, see [Use a Systems Manager parameter instead of an AMI ID](#).
- 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 AWS Marketplace instance](#). 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 [Storage for the root device](#).
- Check the **Virtualization** type listed for each AMI. Notice which AMIs are the type that you need: either **hvm** or **paravirtual**. For example, some instance types require HVM. For more information, see [Linux AMI virtualization types](#).
- 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 [Boot modes](#).
- Choose an AMI that meets your needs, and then choose **Select**.

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 the [Amazon EC2 instance types](#).

For **Instance type**, you can either select an instance type, or you can specify instance attributes and let Amazon EC2 identify the instance types with those attributes.

 **Note**

Specifying instance attributes is supported only when using Auto Scaling groups, EC2 Fleet, and Spot Fleet to launch instances. For more information, see [Creating an Auto Scaling group using attribute-based instance type selection](#), [Attribute-based instance type selection for EC2 Fleet](#), and [Attribute-based instance type selection for Spot Fleet](#).

If you plan to use the launch template in the [launch instance wizard](#) or with the [RunInstances API](#), you must select an instance type.

- **Instance type:** Ensure that the instance type is compatible with the AMI that you've specified. For more information, see [Amazon EC2 instance types](#).
- **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 Amazon Q EC2 instance type selector. For more information, see [Get instance type recommendations for a new workload](#).
- **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)

The key pair for the instance.

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](#).

Network settings

Configure the network settings, as necessary.

- **Subnet:** You can launch an instance in a subnet associated with an Availability Zone, Local Zone, Wavelength Zone, or Outpost.

To launch the instance in an Availability Zone, select the subnet in which to launch your instance. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose the Refresh icon to load your subnet in the list.

To launch the instance in a Local Zone, select a subnet that you created in the Local Zone.

To launch an instance in an Outpost, select a subnet in a VPC that you associated with the Outpost.

- **Firewall (security groups):** Use one or more security groups to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. For more information about security groups, see [Amazon EC2 security groups for Linux instances](#).

If you add a network interface, you must specify the same security groups in the network interface.

Select or create a security group as follows:

- To select an existing security group, choose **Select existing security group**, and select your security group from **Common security groups**.
- To create a new security group, choose **Create security group**.

You can add rules to suit your needs. For example, if your instance will be a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow internet traffic.

To add a rule, choose **Add security group rule**. For **Type**, select the network traffic type. The **Protocol** field is automatically filled in with the protocol to open to network traffic. For **Source**

type, select the source type. To let the launch template add your computer's public IP address, choose **My IP**. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Warning

Rules that enable all IP addresses ($0.0.0.0/0$) to access your instance over SSH or RDP are acceptable if you are briefly launching a test instance and will stop or terminate it soon, but are unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

- **Advanced network configuration**

Network interface

- **Device index:** The device number for the network interface, for example, `eth0` for the primary network interface. If you leave the field blank, AWS creates the primary network interface.
- **Network interface:** Select **New interface** to let Amazon EC2 create a new interface, or select an existing, available network interface.
- **Description:** (Optional) A description for the new network interface.
- **Subnet:** The subnet in which to create the new network interface. For the primary network interface (`eth0`), this is the subnet in which the instance is launched. If you've entered an existing network interface for `eth0`, the instance is launched in the subnet in which the network interface is located.
- **Security groups:** One or more security groups in your VPC with which to associate the network interface.
- **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:** One or more additional private IPv4 addresses from the range of your subnet. Choose **Manually assign** and enter an IP address. Choose **Add IP** to add another IP address. Alternatively, choose **Automatically assign** to let Amazon EC2 choose one for you, and enter a value to indicate the number of IP addresses to add.

- (IPv6-only) **IPv6 IPs**: An IPv6 address from the range of the subnet. Choose **Manually assign** and enter an IP address. Choose **Add IP** to add another IP address. Alternatively, choose **Automatically assign** to let Amazon EC2 choose one for you, and enter a value to indicate the number of IP addresses to add.
- **IPv4 Prefixes**: The IPv4 prefixes for the network interface.
- **IPv6 Prefixes**: The IPv6 prefixes for the network interface.
- (Optional) **Assign Primary IPv6 IP**: If you're launching an instance into a dual-stack or IPv6-only subnet, you have the option to **Assign Primary IPv6 IP**. Assigning a primary IPv6 address enables you to avoid disrupting traffic to instances or ENIs. Choose **Enable** if this instance relies on its IPv6 address not changing. When you launch the instance, 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 cannot 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.
- **Delete on termination**: Whether the network interface is deleted when the instance is deleted.
- **Elastic Fabric Adapter**: Indicates whether the network interface is an Elastic Fabric Adapter. For more information, see [Elastic Fabric Adapter](#).
- **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 template does not include ENA Express configuration for the instance unless you select **Enable** or **Disable**.
- **ENA Express UDP**: If you've enabled ENA Express, you can optionally use it for UDP traffic. The launch template does not include ENA Express configuration for the instance unless you select **Enable** or **Disable**.

Choose **Add network interface** to add more network interfaces. The number of network interfaces that you can add depends on the number that is supported by the selected instance type. Additional network interfaces can reside in a different subnet of the same VPC or in a subnet in a different VPC that you own (as long as the subnet is in the same Availability Zone as

your instance). If you select a subnet in another VPC, the **Multi-VPC** label appears next to the network interface you've added. This enables you to create multi-homed instances across VPCs with different networking and security configurations. Note that if you attach an additional ENI from another VPC, you must choose a security group for the ENI from that VPC.

For more information, see [Elastic network interfaces](#). If you specify more than one network interface, your instance cannot receive a public IPv4 address. Additionally, if you specify an existing network interface for eth0, you cannot override the subnet's public IPv4 setting using **Auto-assign Public IP**. For more information, see [Assign a public IPv4 address during instance launch](#).

Configure storage

If you specify an AMI for the launch template, the AMI includes one or more volumes of storage, including the root volume (**Volume 1 (AMI Root)**). 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 volume. To specify all volume parameters, choose the **Advanced** view (at top right of the card).

To add a new volume, choose **Add new volume**.

By using the **Advanced** view, you can configure each volume as follows:

- **Storage type:** The type of volume (EBS or ephemeral) to associate with your instance. The instance store (ephemeral) volume type is only available if you select an instance type that supports it. For more information, see [Amazon EC2 instance store](#) and [Amazon EBS volumes](#).
- **Device name:** Select from the list of available device names for the volume.
- **Snapshot:** Select the snapshot from which to create 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. If you have selected an AMI and instance that are eligible for the free tier, keep in mind that to stay within the free tier, you must stay under 30 GiB of total storage.
- **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 (io1 and io2) or General Purpose SSD (gp3) volume type, then you can enter the number of I/O operations per second (IOPS) that the

volume can support. This is required for io1, io2, and gp3 volumes. It is not supported for gp2, st1, sc1, or standard volumes. If you omit this parameter for the launch template, you must specify a value for it when you launch an instance from the launch template.

- **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.

Resource tags

To [tag](#) the resources that are created when an instance is launched, under **Resource tags**, 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
- Spot Instance requests
- Network interfaces

Note

To tag the launch template itself, you must specify the tags under **Template tags**. For more information, see [Launch template name, description, and tags](#).

Advanced details

For **Advanced details**, expand the section to view the fields and specify any additional parameters for the instance.

- **Purchasing option:** Choose **Request Spot Instances** to request Spot Instances at the Spot price, capped at the On-Demand price, and choose **Customize** to change the default Spot Instance settings. 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, EC2 launches an On-Demand Instance by default. For more information, see [Spot Instances](#).
- **IAM instance profile:** Select an AWS Identity and Access Management (IAM) instance profile to associate with the instance. 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 [Amazon EC2 instance hostname types](#).
- **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 [Amazon EC2 instance hostname types](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Change the instance initiated shutdown behavior](#).
- **Stop - Hibernate behavior:** To enable hibernation, choose **Enable**. This field is only valid for instances that meet the hibernation prerequisites. For more information, see [Hibernate your Amazon EC2 instance](#).
- **Termination protection:** To prevent accidental termination, choose **Enable**. For more information, see [Enable termination protection](#).
- **Stop protection:** To prevent accidental stopping, choose **Enable**. For more information, see [Enable stop protection](#).
- **Detailed CloudWatch monitoring:** Choose **Enable** to enable detailed monitoring of the instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitor your instances using CloudWatch](#).
- **Elastic inference:** An elastic inference accelerator to attach to your EC2 CPU instance. For more information, see [Working with Amazon Elastic Inference](#) in the *Amazon Elastic Inference Developer Guide*.

Note

Starting April 15, 2023, AWS will not onboard new customers to Amazon Elastic Inference (EI), and will help current customers migrate their workloads to options that offer better price and performance. After April 15, 2023, new customers will not be able to launch instances with Amazon EI accelerators in Amazon SageMaker, Amazon ECS, or Amazon EC2. However, customers who have used Amazon EI at least once during the past 30-day period are considered current customers and will be able to continue using the service.

- **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 name:** 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 can be launched in a placement group. For more information, see [Placement groups](#).
- **EBS-optimized instance:** Select **Enable** to provide additional, dedicated capacity for Amazon EBS I/O. Not all instance types support this feature. Additional charges apply. For more information, see [the section called "EBS optimization"](#).
- **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 [Dedicated Instances](#) and [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:** Choose **Specify CPU options** to specify a custom number of vCPUs during launch. Set the number of CPU cores and threads per core. For more information, see [Optimize CPU options](#).
- **Metadata transport:** You can enable or disable the access method to the Instance Metadata Service (IMDS) that's available for this EC2 instance based on the IP address type (IPv4, IPv6, or IPv4 and IPv6) of the instance. For more information, see [Retrieve instance metadata](#).
- **Metadata accessible:** You can enable or disable access to the IMDS. For more information, see [Configure instance metadata options for new instances](#).
- **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 instance's tags from its metadata. If you do not include this setting in the template, by default, access to the tags in instance metadata is not allowed. For more information, see [Allow 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, see [Run commands on your Linux instance at launch](#).

AWS CLI example

The following example uses the [create-launch-template](#) command to create a launch template with the specified name and instance configuration.

```
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-7b16de0c"
  }],
  "ImageId": "ami-8c1be5f6",
  "InstanceType": "r4.4xlarge",
  "TagSpecifications": [{
    "ResourceType": "instance",
    "Tags": [{
      "Key": "Name",
      "Value": "webserver"
    }]
  }],
  "CpuOptions": {
    "CoreCount": 4,
    "ThreadsPerCore": 2
  }
}
```

The following is example output.

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-01238c059e3466abc",
    "LaunchTemplateName": "TemplateForWebServer",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-27T09:13:24.000Z"
  }
}
```

AWS Tools for Windows PowerShell example

The following example uses the [New-EC2LaunchTemplate](#) cmdlet to create a launch template with the specified name and instance configuration.

```
$launchTemplateData = [Amazon.EC2.Model.RequestLaunchTemplateData]@{
    ImageId = 'ami-8c1be5f6'
    InstanceType = 'r4.4xlarge'
    NetworkInterfaces = @(
        [Amazon.EC2.Model.LaunchTemplateInstanceNetworkInterfaceSpecificationRequest]@{
            AssociatePublicIpAddress = $true
            DeviceIndex = 0
            Ipv6AddressCount = 1
            SubnetId = 'subnet-7b16de0c'
        }
    )
    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.

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 tag**, and then enter a tag key and value pair.
6. Expand **Source template**, and for **Launch template name** choose a launch template on which to base the new launch template.
7. For **Source template version**, choose the launch template version on which to base the new launch template.
8. Adjust any launch parameters as required, and then choose **Create launch template**.

Create a launch template from an instance

Console

To create a launch template from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and choose **Actions, Create template from instance**.
4. Provide a name, description, and tags, and adjust the launch parameters as required.

Note

When you create a launch template from an instance, the instance's network interface IDs and IP addresses are not included in the template.

5. Choose **Create launch template**.

AWS CLI

You can use the AWS CLI to create a launch template from an existing instance by first getting the launch template data from an instance, and then creating a launch template using the launch template data.

To get launch template data from an instance

- 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

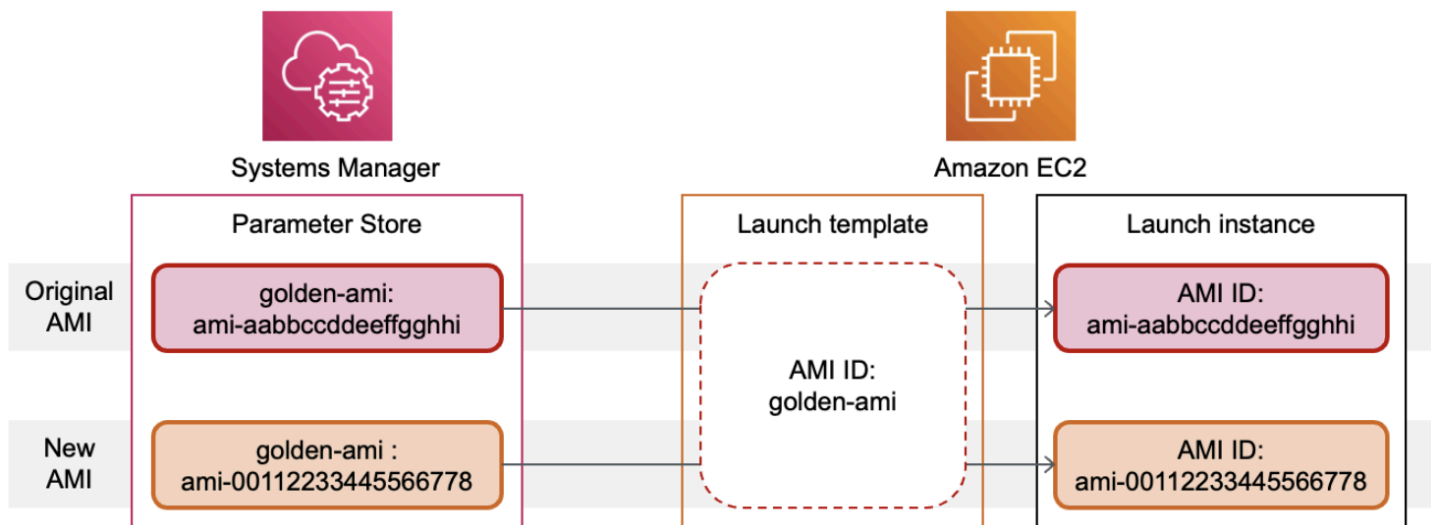
- Use the [create-launch-template](#) command to create a launch template using the output from the previous procedure. For more information about creating a launch template using the AWS CLI, see [Create a launch template from parameters](#).

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 Systems Manager Parameter Store. The Parameter Store provides a central place to store your application configuration values. For more information, see [AWS Systems Manager Parameter Store](#) in the *AWS Systems Manager User Guide*.

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 from 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

- Currently, EC2 Fleets 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 and Spot Fleets, if you specify an AMI in the launch template, 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 that you no longer require.

Tasks

- [Create a launch template version](#)
- [Set the default launch template version](#)
- [Describe a launch template version](#)

- [Delete 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 more information about the launch parameters, see [Create a launch template](#).

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, and 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-c998b6b2"
```

Set the default launch template version

You can set the default version for the launch template. When you launch an instance from a launch template and do not specify a version, the instance is launched using the parameters of the default version.

Console

To set the default launch template version

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 and specify the version that you want to set as the default.

```
aws ec2 modify-launch-template \  
  --launch-template-id lt-0abcd290751193123 \  
  --default-version 2
```

Describe a launch template version

Using the console, you can view all the versions of the selected launch template, or get a list of the launch templates whose latest or default version matches a specific version number. Using the AWS CLI, you can describe all versions, individual versions, or a range of versions of a specified launch template. You can also describe all the latest versions or all the default versions of all the launch templates in your account.

Console

To describe a launch template version

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](#) 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 all the latest and default launch template versions in your account

- Use the [describe-launch-template-versions](#) command and specify `$Latest`, `$Default`, or both. You must omit the launch template ID and name in the call. You cannot specify version numbers.

```
aws ec2 describe-launch-template-versions \  
  --versions "$Latest,$Default"
```

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
```

Delete a launch template

If you no longer require a launch template, you can delete it. Deleting a launch template deletes all of its versions. To delete a specific version of a launch template, see [Delete a launch template version](#).

When you delete a launch template, it doesn't affect any instances that you've launched from the launch template.

Console

To delete a launch template

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Delete template**.
4. Enter **Delete** to confirm deletion, and then choose **Delete**.

AWS CLI

To delete a launch template

- Use the [delete-launch-template](#) (AWS CLI) command and specify the launch template.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

Launch instances from a launch template

Launch templates are supported by several instance launch services. 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.

Topics

- [Launch an instance from a launch template](#)
- [Use launch templates with Amazon EC2 Auto Scaling](#)
- [Use launch templates with EC2 Fleet](#)
- [Use launch templates with Spot Fleet](#)

Launch an instance from a launch template

You can use the parameters contained in a launch template to launch an instance. You have the option to override or add launch parameters before you launch the instance.

Instances that are launched using a launch template are automatically assigned two tags with the keys `aws:ec2launchtemplate:id` and `aws:ec2launchtemplate:version`. You cannot remove or edit these tags.

Console

To launch an instance from a launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Launch instance from template**.
4. For **Source template version**, select the launch template version to use.
5. For **Number of instances**, specify the number of instances to launch.
6. (Optional) You can override or add launch template parameters by changing and adding parameters in the **Instance details** section.
7. Choose **Launch instance from template**.

AWS CLI

To launch an instance from a launch template using the AWS CLI

- Use the [run-instances](#) command and specify the `--launch-template` parameter. Optionally specify the launch template version to use. If you don't specify the version, the default version is used.

```
aws ec2 run-instances \  
  --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- To override a launch template parameter, specify the parameter in the [run-instances](#) command. The following example overrides the instance type that's specified in the launch template (if any).

```
aws ec2 run-instances \  
  --launch-template LaunchTemplateId=lt-0abcd290751193123 \  
  --instance-type t3.micro
```

```
--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 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 instance launch issues](#).

Use launch templates with Amazon EC2 Auto Scaling

You can create an Auto Scaling group and specify a launch template to use for the group. When Amazon EC2 Auto Scaling launches instances in the Auto Scaling group, it uses the launch

parameters defined in the associated launch template. For more information, see [Create a launch template for an Auto Scaling group](#) and [Create a launch template using advanced settings](#) in the *Amazon EC2 Auto Scaling User Guide*.

Before you can create an Auto Scaling group using a launch template, you must create a launch template that includes the parameters required to launch an instance in an Auto Scaling group, such as the ID of the AMI. The console provides guidance to help you create a template that you can use with Amazon EC2 Auto Scaling.

To create a launch template to use with Auto Scaling using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. For **Launch template name**, enter a descriptive name for the launch template.
4. For **Template version description**, provide a brief description of this version of the launch template.
5. Under **Auto Scaling guidance**, select the check box to have Amazon EC2 provide guidance to help create a template to use with Auto Scaling.
6. Modify the launch parameters as required. Because you selected Auto Scaling guidance, some fields are required and some fields are not available. For information about how to configure the launch parameters for Amazon EC2 Auto Scaling, see [Create a launch template for an Auto Scaling group](#) and [Create a launch template using advanced settings](#) in the *Amazon EC2 Auto Scaling User Guide*.
7. Choose **Create launch template**.
8. (Optional) To create an Auto Scaling group using this launch template, on the **Next steps** page, choose **Create Auto Scaling group**.

For examples that show how to use the AWS CLI to create launch templates with various parameter combinations, see [Examples for creating and managing launch templates with the AWS Command Line Interface \(AWS CLI\)](#) 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 about creating or updating an Auto Scaling group using a launch template, see the following topics in the *Amazon EC2 Auto Scaling User Guide*.

- [Create Auto Scaling groups using launch templates](#)
- [Update an Auto Scaling group](#)

Use launch templates with EC2 Fleet

You can create an EC2 Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the EC2 Fleet request, it uses the launch parameters defined in the associated launch template. You can override some of the parameters that are specified in the launch template.

For more information, see [Create an EC2 Fleet](#).

To create an EC2 Fleet with a launch template using the AWS CLI

- Use the [create-fleet](#) command. Use the `--launch-template-configs` parameter to specify the launch template and any overrides for the launch template.

Use launch templates with Spot Fleet

You can create a Spot Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the Spot Fleet request, it uses the launch parameters defined in the associated launch template. You can override some of the parameters that are specified in the launch template.

For more information, see [Create a Spot Fleet request](#).

To create a Spot Fleet request with a launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **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 \(console\)](#).
7. When you're ready to create your Spot Fleet, choose **Launch**.

To create a Spot Fleet request with a launch template using the AWS CLI

- Use the [request-spot-fleet](#) command. Use the `LaunchTemplateConfigs` parameter to specify the launch template and any overrides for the launch template.

Launch an instance using parameters 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

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 instance launch issues](#).

Launch an AWS Marketplace instance

You can subscribe to an AWS Marketplace product and launch an instance from the product's AMI using the Amazon EC2 launch wizard. For more information about paid AMIs, see [Paid AMIs](#). To

cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see [Manage your AWS Marketplace subscriptions](#).

New console


To launch an instance from the AWS Marketplace using the launch wizard

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

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.

- 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 [Launch an instance using defined parameters](#).
 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 Linux instance](#).

 **Important**

Check the vendor's usage instructions carefully, as you might need to use a specific user name 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 instance launch issues](#).

Old console

To launch an instance from the AWS Marketplace using the launch wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 dashboard, choose **Launch instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the **AWS Marketplace** category on the left. Find a suitable AMI by browsing the categories, or using the search functionality. Choose **Select** to choose your product.
4. A dialog displays an overview of the product you've selected. You can view the pricing information, as well as any other information that the vendor has provided. When you're ready, choose **Continue**.

Note

You are not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type, as you will be prompted to select an instance type on the next page of the wizard. Additional taxes may also apply to the product.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. When you're done, choose **Next: Configure Instance Details**.
6. On the next pages of the wizard, you can configure your instance, add storage, and add tags. For more information about the different options you can configure, see [Launch an instance using the old launch instance wizard](#). Choose **Next** until you reach the **Configure Security Group** page.

The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IPv4 addresses (0.0.0.0/0) access on SSH (port 22) on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.

When you are ready, choose **Review and Launch**.

7. On the **Review Instance Launch** page, check the details of the AMI from which you're about to launch the instance, as well as the other configuration details you set up in the wizard. When you're ready, choose **Launch** to select or create a key pair, and launch your instance.

- Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, choose **View Instances** to go to the Instances page.

Note

You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

- When your instance is in the running state, you can connect to it. To do this, select your instance in the list and choose **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see [Connect to your Linux instance](#).

Important

Check the vendor's usage instructions carefully, as you may need to use a specific user name to log in to the instance. For more information about accessing your subscription details, see [Manage your AWS Marketplace subscriptions](#).

- If the instance fails to launch or the state immediately goes to terminated instead of running, see [Troubleshoot instance launch issues](#).

Launch an AWS Marketplace AMI instance using the API and CLI

To launch instances from AWS Marketplace products using the API or command line tools, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance .
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell

Method	Documentation
Query API	Use the RunInstances request.

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.

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 reboot, stop, hibernate, and terminate](#).

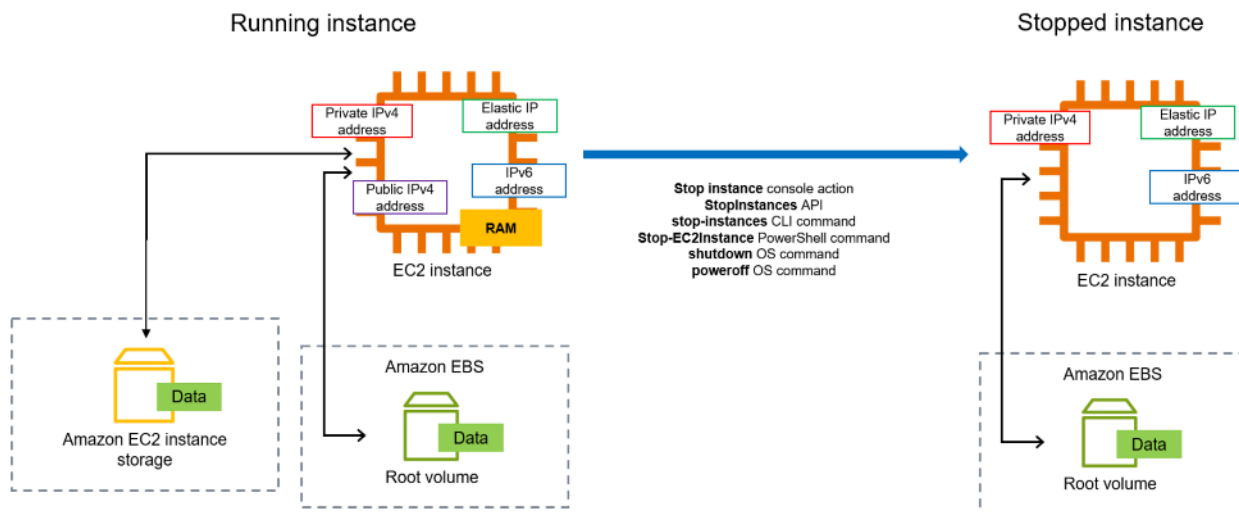
Contents

- [How instance stop and start works](#)
- [Manually stop and start your instances](#)
- [Automatically stop and start your instances](#)
- [Find all running and stopped instances](#)
- [Enable stop protection for your instance](#)

How instance stop and start works

When you stop an instance, changes are registered at the 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.

The following diagram shows what is lost and what persists when an Amazon EC2 instance is stopped. When an instance stops, it loses any attached instance store volumes and the data stored on those volumes, the data stored on the instance RAM, and the assigned public IPv4 address if an Elastic IP address is not associated with the instance. An instance retains assigned private IPv4 addresses, Elastic IP addresses associated with the instance, any IPv6 addresses, and any attached Amazon EBS volumes and the data on those volumes.



What happens when you stop 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 is triggered by the ACPI shutdown button press event from the hypervisor.
- ACPI shutdown is initiated.
- The instance shuts down when the graceful shutdown process exits. There is no configurable OS shutdown time.
- If the instance OS does not cleanly shut down within a few minutes, a hard shutdown is performed.
- The instance stops running.
- The instance status 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 Auto Scaling instances](#) in the *Amazon EC2 Auto Scaling User Guide*.

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 volumes.
- Data stored on the attached Amazon EBS volumes.
- 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](#).

For information about what happens when you stop a Mac instance, see [Stop and terminate your Mac instance](#).

What happens when you start an instance

Changes registered at the OS level

- 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).
- Amazon EC2 assigns a new public IPv4 address to the instance if the instance is configured to receive a public IPv4 address. To retain a public IPv4 address that never changes, you can associate an [Elastic IP address](#) with your instance.

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.

Manually stop and start your instances

You can stop and start your Amazon EBS-backed instances (instances with EBS root devices). You can't stop and start instances with instance store root device.

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.

Console

To stop and start an Amazon EBS-backed instance

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. On the **Storage** tab, verify that **Root device type** is **EBS**. Otherwise, you can't stop the instance.
4. 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.
5. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
6. To start a stopped instance, select the instance, and choose **Instance state**, **Start instance**.
7. It can take a few minutes for the instance to enter the running state.
8. 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 stopping your instance](#).

Command line

Prerequisites

Verify that the root device of the instance is an EBS volume. For example, run the [describe-instances](#) AWS CLI command and verify that `RootDeviceType` is `ebs`, not `instance-store`.

To stop and start an Amazon EBS-backed instance

Use one of the following commands:

- **AWS CLI**—[stop-instances](#) and [start-instances](#).
- **AWS Tools for PowerShell**—[Stop-EC2Instance](#) and [Start-EC2Instance](#).
- **OS commands**—You can initiate a shutdown using the **shutdown** or **poweroff** commands. When you use an OS command, the instance stops by default. You can change this behavior so that it terminates instead. For more information, see [Change the instance initiated shutdown behavior](#).

Note that 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.

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 stop and start Amazon EC2 instances at regular intervals using Lambda?](#)

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 [Amazon EC2 Global View](#).

Enable stop protection for your instance

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 using one of the following methods.

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](#) AWS CLI command to launch the instance, and specify the `disable-api-stop` parameter.

```
aws ec2 run-instances \  
  --image-id ami-a1b2c3d4e5example \  
  --instance-type t3.micro \  
  --key-name MyKeyPair \  
  --disable-api-stop \  
  ...
```

Enable stop protection for a running or stopped instance

You can enable stop protection for an instance while the instance is running or stopped using one of the following methods.

Console

To enable 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. Select the **Enable** check box, and then choose **Save**.

AWS CLI

To enable stop protection for a running or stopped instance

Use the [modify-instance-attribute](#) AWS CLI command and specify the `disable-api-stop` parameter.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --disable-api-stop
```

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** check box, and then choose **Save**.

AWS CLI

To disable stop protection for a running or stopped instance

Use the [modify-instance-attribute](#) AWS CLI command and specify the `no-disable-api-stop` parameter.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --no-disable-api-stop
```

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.

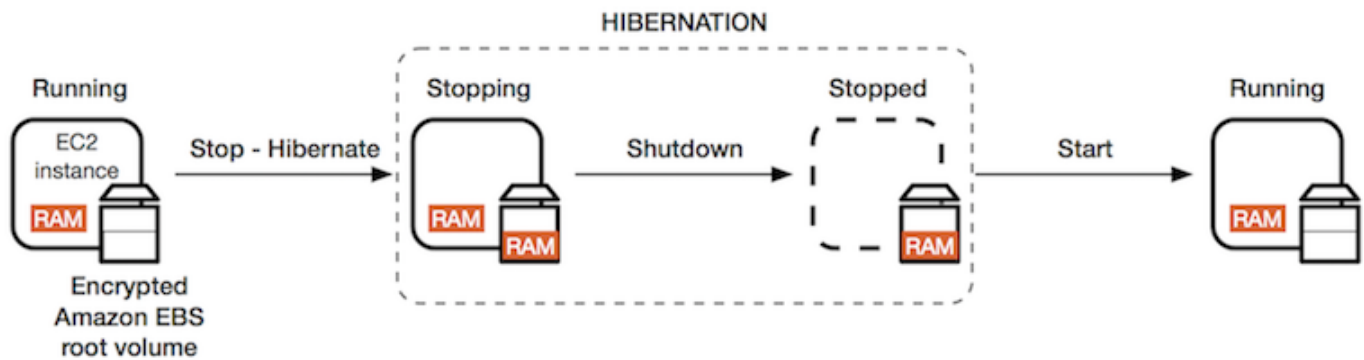
If you no longer need an instance, you can terminate it at any time, including when it is in a stopped (hibernated) state. For more information, see [Terminate Amazon EC2 instances](#).

Contents

- [How Amazon EC2 instance hibernation works](#)
- [Prerequisites for Amazon EC2 instance hibernation](#)
- [Configure a Linux AMI to support hibernation](#)
- [Enable hibernation for an Amazon EC2 instance](#)
- [Disable KASLR on an instance \(Ubuntu only\)](#)
- [Hibernate an Amazon EC2 instance](#)
- [Start a hibernated Amazon EC2 instance](#)
- [Troubleshoot Amazon EC2 instance hibernation](#)

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.
- While your instance is in the **stopped** state, you can modify certain attributes of the instance, including the instance type or size.
- 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 reboot, stop, hibernate, and terminate](#).

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 GB of RAM.
- (Windows instances) You can't hibernate a Windows instance that has more than 16 GB of RAM.
- If you create a snapshot or AMI from an instance that is hibernated or has hibernation enabled, you might not be able to connect to 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 Auto Scaling instances](#) in the *Amazon EC2 Auto Scaling User Guide*.
- You can't hibernate an instance that is configured to boot in UEFI mode 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](#).

- There are several ways in which you can enable hibernation for a Spot Instance. For more information, see [Specify the interruption behavior](#).

Prerequisites for Amazon 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

- AL2023 AMI released 2023.09.20 or later
- 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 ¹ ([Additional configuration](#) is required)
- Fedora version 34 or later AMI ¹ ([Additional configuration](#) is required)
- Red Hat Enterprise Linux (RHEL) 9 AMI ¹ ([Additional configuration](#) is required)
- Red Hat Enterprise Linux (RHEL) 8 AMI ¹ ([Additional configuration](#) is required)
- Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI released with serial number 20230303 or later ²
- Ubuntu 20.04 LTS (Focal Fossa) AMI released with serial number 20210820 or later ²
- Ubuntu 18.04 LTS (Bionic Beaver) AMI released with serial number 20190722.1 or later ^{2 4}
- Ubuntu 16.04 LTS (Xenial Xerus) AMI ^{2 3 4} ([Additional configuration](#) is required)

¹ For CentOS, Fedora, and Red Hat Enterprise Linux, hibernation is supported on Nitro-based instances only.

² 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 \(Ubuntu only\)](#).

³ 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.

⁴ 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.

- General purpose: M3, M4, M5, M5a, M5ad, M5d, M6i, M6id, M7i, M7i-flex, T2, T3, T3a
- Compute optimized: C3, C4, C5, C5d, C6i, C6id, C7a, C7i
- Memory optimized: R3, R4, R5, R5a, R5ad, R5d, R7a, R7i, R7iz
- Storage optimized: I3, I3en

Nitro instances – Bare metal instances are not supported.

To see the available instance types that support hibernation in a specific Region

The available instance types vary by Region. To see the available instance types that support hibernation 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 hibernation and the `--query` parameter to scope the output to the value of InstanceType.

```
aws ec2 describe-instance-types --filters Name=hibernation-supported,Values=true --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

Example output

```
c3.2xlarge  
c3.4xlarge  
c3.8xlarge  
c3.large
```



```
c3.xlarge
c4.2xlarge
c4.4xlarge
c4.8xlarge
...
```

Instance RAM size

Linux instances – Must be less than 150 GB.

Windows instances – Can be up to 16 GB. For hibernating a T3 or T3a Windows instance, we recommend at least 1 GB 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 [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 support hibernation, but to hibernate an instance that was launched with one of these AMIs, additional configuration is required before you can hibernate the instance.

Additional configuration required for:

- [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 more information, see [Update instance software on your Amazon Linux instance](#).

No additional configuration is required for the following AMIs because they're already configured to support hibernation:

- AL2023 AMI released 2023.09.20 or later
- Amazon Linux 2 full AMI released 2019.08.29 or later
- Amazon Linux AMI 2018.03 released 2018.11.16 or later
- Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI released with serial number 20230303 or later
- Ubuntu 20.04 LTS (Focal Fossa) AMI released with serial number 20210820 or later
- Ubuntu 18.04 LTS (Bionic Beaver) AMI released with serial number 20190722.1 or later

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 a Linux AMI from an instance](#).

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 a Linux AMI from an instance](#).

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 --get-selections | 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.

Topics

- [Enable hibernation for On-Demand Instances](#)
- [Enable hibernation for Spot Instances](#)
- [View if an instance is enabled for hibernation](#)

Enable hibernation for On-Demand Instances

Use one the following methods to enable hibernation for your On-Demand Instances.

New 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 Amazon 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 instance using the new launch instance wizard](#).

Old console

To enable hibernation for an On-Demand Instance

1. Follow the [Launch an instance using the old launch instance wizard](#) procedure.
2. On the **Choose an Amazon Machine Image (AMI)** page, select an AMI that supports hibernation. For more information about supported AMIs, see [Prerequisites for Amazon EC2 instance hibernation](#).
3. On the **Choose an Instance Type** page, select a supported instance type, and choose **Next: Configure Instance Details**. For information about supported instance types, see [Prerequisites for Amazon EC2 instance hibernation](#).
4. On the **Configure Instance Details** page, for **Stop - Hibernate Behavior**, select the **Enable hibernation as an additional stop behavior** check box.
5. On the **Add Storage** page, for the root volume, specify the following information:
 - For **Size (GiB)**, enter the EBS root volume size. The volume must be large enough to store the RAM contents and accommodate your expected usage.
 - For **Volume Type**, select a supported EBS volume type, General Purpose SSD (gp2 and gp3) or Provisioned IOPS SSD (io1 and io2).
 - For **Encryption**, select the encryption key for the volume. If you enabled encryption by default in this AWS Region, the default encryption key is selected.

For more information about the prerequisites for the root volume, see [Prerequisites for Amazon EC2 instance hibernation](#).

6. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launch an instance using the old launch instance wizard](#).

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  
    }  
  }  
]
```

Note

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 using the AWS Tools for Windows PowerShell

Use the [New-EC2Instance](#) command to launch an instance. Specify the EBS root volume by first defining the block device mapping, and then adding it to the command using the `-BlockDeviceMappings` parameter. Enable hibernation using the `-HibernationOptions_Configured $true` parameter.

```
PS C:\> $ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
PS C:\> $ebs_encrypt.DeviceName = "/dev/xvda"
PS C:\> $ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
PS C:\> $ebs_encrypt.Ebs.VolumeSize = 30
PS C:\> $ebs_encrypt.Ebs.VolumeType = "gp2"
PS C:\> $ebs_encrypt.Ebs.Encrypted = $true

PS C:\> New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair
```

Note

The value for `DeviceName` must match the root device name associated with the AMI. To find the root device name, use the [Get-EC2Image](#) command.

```
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

Use one of the following methods to enable hibernation for your Spot Instances. For more information about hibernating a Spot Instance on interruption, see [Spot Instance interruptions](#).

Console

You can use the launch instance wizard in the Amazon EC2 console to enable hibernation for a Spot Instance.

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 Amazon 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 instance using the new launch instance wizard](#).

AWS CLI

You can enable hibernation for a Spot Instance using the [run-instances](#) AWS CLI command.

To enable hibernation for a Spot Instance using the `hibernation-options` parameter

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
    }
  }
]
```

Note

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 using the AWS Tools for Windows PowerShell

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.

```
PS C:\> $ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
PS C:\> $ebs_encrypt.DeviceName = "/dev/xvda"
PS C:\> $ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
PS C:\> $ebs_encrypt.Ebs.VolumeSize = 30
PS C:\> $ebs_encrypt.Ebs.VolumeType = "gp2"
```

```
PS C:\> $ebs_encrypt.Ebs.Encrypted = $true

PS C:\> New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair `
    -InstanceMarketOption @(
        MarketType = spot;
        SpotOptions @{
            MaxPrice = 1;
            SpotInstanceType = persistent}
    )
```

Note

The value for DeviceName must match the root device name associated with the AMI. To find the root device name, use the [Get-EC2Image](#) command.

```
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.

There are several ways in which you can enable hibernation for a Spot Instance. For more information, see [Specify the interruption behavior](#).

View if an instance is enabled for hibernation

Use the following instructions to view 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 using the AWS Tools for Windows PowerShell

Use the [Get-EC2Instance](#) command and specify the `-Filter @{ Name="hibernation-options.configured"; Value="true"}` parameter to filter instances that are enabled for hibernation.

```
(Get-EC2Instance -Filter @{Name="hibernation-options.configured";
  Value="true"}).Instances
```

The output lists the EC2 instances that are enabled for hibernation.

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 from Linux or macOS 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.

```
[ec2-user ~]$ sudo update-grub
```

5. Reboot the instance.

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

6. Run the following command to confirm that `nokaslr` has been added.

```
[ec2-user ~]$ 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 Amazon 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 EBS-backed 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 using the AWS Tools for Windows PowerShell

Use the [Stop-EC2Instance](#) command and specify the `-Hibernate $true` parameter.

```
Stop-EC2Instance \  
  -InstanceId i-1234567890abcdef0 \  
  -Hibernate $true
```

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 using the AWS Tools for Windows PowerShell

Use the [Get-EC2Instance](#) command and specify the state-reason-code filter to see the instances on which hibernation was initiated.

```
Get-EC2Instance `
```

```
-Filter @{"Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

The output lists the EC2 instances on which hibernation was initiated.

Start a hibernated Amazon EC2 instance

Start a hibernated instance by starting it in the same way that you would start a stopped instance.

Note

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 using the AWS Tools for Windows PowerShell

Use the [Start-EC2Instance](#) command.

```
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.

If you do not see any logs from these processes, your AMI might not support hibernation. For information about supported AMIs, see [Prerequisites for Amazon 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](#).

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.

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 GB of RAM, try increasing the size of the instance to one that has at least 1 GB of RAM.

Instance "stuck" in the stopping state

If you hibernated your instance and it appears "stuck" in the stopping state, you can forcibly stop it. For more information, see [Troubleshoot stopping your instance](#).

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 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 (with a minimum one-minute charge), unlike [stopping and starting](#) your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see [Scheduled events for your instances](#).

We recommend that you use the Amazon EC2 console, a command line tool, or the Amazon EC2 API to reboot your instance instead of running the operating system reboot command from your instance. If you use the Amazon EC2 console, a command line tool, or the Amazon EC2 API to reboot your instance, we perform a hard reboot if the instance does not cleanly shut down within a few minutes. If you use AWS CloudTrail, then using Amazon EC2 to reboot your instance also creates an API record of when your instance was rebooted.

Console

To reboot an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Instance state**, **Reboot instance**.

Alternatively, select the instance and choose **Actions, Manage instance state**. In the screen that opens, choose **Reboot**, and then **Change state**.

4. Choose **Reboot** when prompted for confirmation.

The instance remains in the `running` state.

Command line

To reboot an instance

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

- [reboot-instances](#) (AWS CLI)
- [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell)

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 additional instances using the same AMI. If you'd rather stop or hibernate an instance, see [Stop and start Amazon EC2 instances](#) or [Hibernate your Amazon EC2 instance](#). For more information, see [Differences between reboot, stop, hibernate, and terminate](#).

Contents

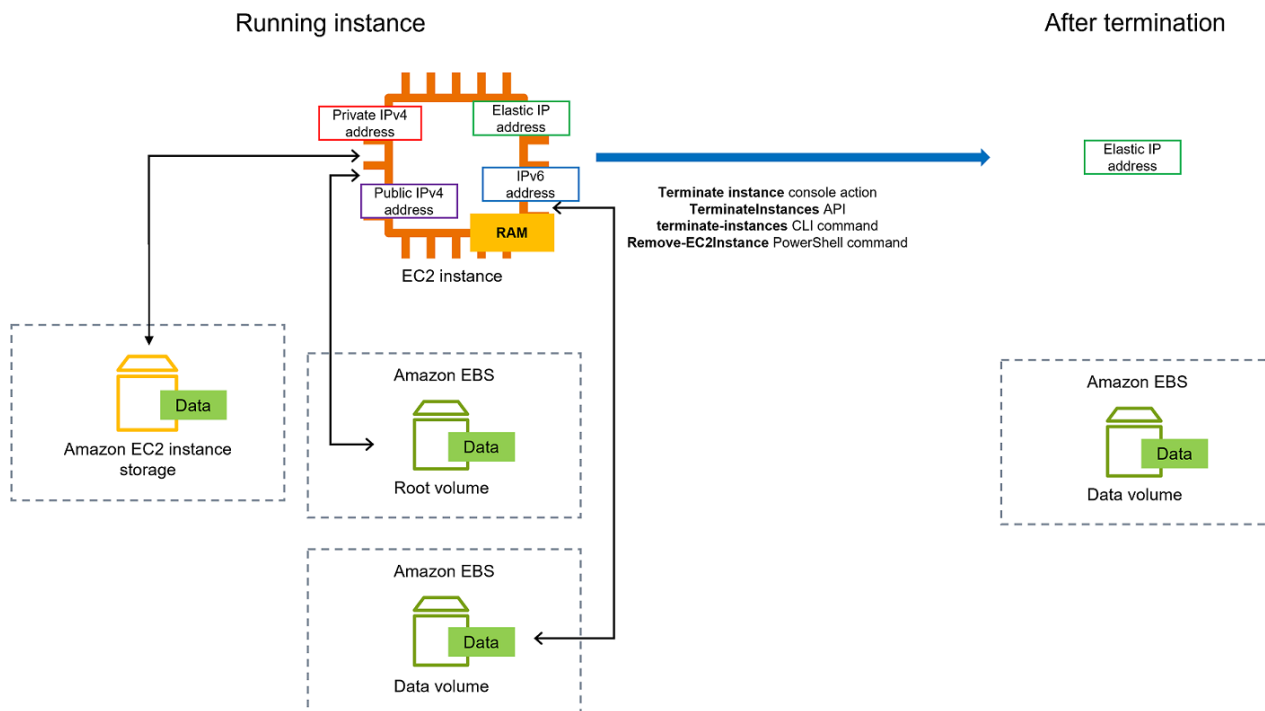
- [How instance termination works](#)
- [Terminate an instance](#)
- [Troubleshoot instance termination](#)
- [Enable termination protection](#)

- [Change the instance initiated shutdown behavior](#)
- [Preserve data when an instance is terminated](#)

How instance termination works

When you terminate an instance, changes are registered at the 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 the instance RAM is erased. Any Elastic IP addresses associated with the instance are detached. For Amazon EBS volumes and the data on those volumes, the outcome depends on the **Delete on termination** setting for the volume. By default, the root volume is deleted and the data volumes are preserved.



Considerations

- When an instance terminates, the data on any instance store volumes associated with that instance is deleted.
- By default, Amazon EBS root device volumes are automatically deleted when the instance terminates. However, any additional EBS volumes that you attach at launch, or any EBS volumes

that you attach to an existing instance persist even after the instance terminates. For more information, see [Preserve data when an instance is terminated](#).

 **Note**

Any volumes that are not deleted upon termination of the instance will continue to incur charges.

- 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](#).
- If you run a script on instance termination, your instance might have an abnormal termination, because we have no way to ensure that shutdown scripts run. Amazon EC2 attempts to shut an instance down cleanly and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.

What happens when you terminate an instance

Changes registered at the OS level

- The API request will send a button press event to the guest.
- Various system services will be stopped as a result of the button press event. In Linux, **systemd** handles a graceful shutdown of the system. Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
- ACPI shutdown will be initiated.
- The instance will shut down when the graceful shutdown process exits. There is no configurable OS shutdown time. The instance remains visible in the console for a short time, then the entry is automatically deleted.

Resources lost

- Data stored on an instance store volume.
- Data stored on Amazon EBS root device volumes if the `DeleteOnTermination` attribute is set to true.

Resources that persist

- Data stored on additional Amazon EBS volumes attached at launch or after launch of an instance.

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](#).

Terminate an instance

You can terminate an instance at any time.

Console

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, and choose **Instance state, Terminate instance**.
4. Choose **Terminate** 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](#).

Command line

To terminate an instance using the command line

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

- [terminate-instances](#) (AWS CLI)
- [Remove-EC2Instance](#) (AWS Tools for Windows PowerShell)

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, it should be cleaned up (terminated) by automated processes within the Amazon EC2 service. For more information, see [Delayed instance termination](#).

Enable termination protection

To prevent your instance from being accidentally terminated, you can enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated using the AWS Management Console, AWS Command Line Interface (AWS CLI), or API. By default, termination protection is disabled for your instance which means that your instance can be terminated using the AWS Management Console, AWS CLI, or API. You can set the value of this attribute when you launch an instance, while the instance is running, or while the instance is stopped (for Amazon EBS backed instances).

The `DisableApiTermination` attribute doesn't prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set. For more information, see [Change the instance initiated shutdown behavior](#).

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.

To enable termination protection for an instance at launch time

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance** and follow the directions in the wizard.
3. On the **Configure Instance Details** page, select the **Enable termination protection** check box.

To enable termination protection for a running or stopped instance

1. Select the instance, and choose **Actions, Instance Settings, Change Termination Protection**.
2. Choose **Yes, Enable**.

To disable termination protection for a running or stopped instance

1. Select the instance, and choose **Actions, Instance Settings, Change Termination Protection**.
2. Choose **Yes, Disable**.

To enable or disable termination protection using the command line

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

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

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 the instance initiated shutdown behavior

By default, when you initiate a shutdown from an Amazon EBS backed instance (using a command such as **shutdown** or **poweroff**), the instance stops. 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.

You can change the `InstanceInitiatedShutdownBehavior` attribute using the Amazon EC2 console or the command line.

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**.

Command line

To change the instance initiated shutdown behavior

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

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Preserve data when an instance is terminated

Depending on your use case, you might want to preserve the data on your instance store volume or Amazon EBS volume when the Amazon EC2 instance is terminated. The data on an instance store volume doesn't persist when an instance is terminated. If you need to preserve the data stored on an instance store volume beyond the lifetime of the instance, you must manually copy that data to more 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](#).

For data on Amazon EBS volumes, Amazon EC2 uses the value of the `DeleteOnTermination` attribute for each attached Amazon EBS volume to determine whether to preserve or delete the volume.

The default value for the `DeleteOnTermination` attribute differs depending on whether the volume is the root volume of the instance or a non-root volume attached to the instance.

Root volume

By default, when you launch an instance the `DeleteOnTermination` attribute for the root volume of an instance is set to `true`. Therefore, the default is to delete the root volume of the instance when the instance terminates.

Non-root volume

By default, when you attach a non-root EBS volume to an instance, its `DeleteOnTermination` attribute is set to `false`. Therefore, the default is to preserve these volumes.

Note

After the instance terminates, you can take a snapshot of the preserved volume or attach it to another instance. You must delete a volume to avoid incurring further charges.

The `DeleteOnTermination` attribute can be set by the creator of an AMI as well as by the person who launches an instance. When the attribute is changed by the creator of an AMI or by the person who launches an instance, the new setting overrides the original AMI default setting. We recommend that you verify the default setting for the `DeleteOnTermination` attribute after you launch an instance with an AMI.

To verify if an Amazon EBS volume will be deleted on instance termination, view the details for volume on the instance's details pane. On the **Storage** tab, under **Block devices**, scroll right to view the **Delete on termination** setting for the volume.

- If **Yes**, the volume will be deleted when the instance is terminated.
- If **No**, the volume will not be deleted when the instance is terminated. Any volumes that are not deleted upon termination of the instance will continue to incur charges.

Change the root volume to persist at launch

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

Use one of the following methods to change the root volume to persist at launch.

Console

To change the root volume of an instance to persist at launch using the console

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. Under **Storage (volumes)**, expand the information under the root volume.
3. For **Delete on termination**, choose **No**
4. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**. For more information, see [Launch an instance using the new launch instance wizard](#).

Command line

To change the root volume of an instance to persist at launch using the command line

When you launch an EBS-backed instance, you can use one of the following commands to change the root device volume to persist. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

In the block device mappings for the volumes that you want to persist, include `--DeleteOnTermination`, and specify `false`.

For example, to persist a volume add the following option to your `run-instances` command:

```
--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
}
]
```

Change the root volume of a running instance to persist

You can use one of the following commands to change the root device volume of a running EBS-backed instance to persist. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

For example, use the following command:

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings file://mapping.json
```

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
    }
  }
]
```

Instance retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware that hosts the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS.

- If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware.
- If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information about the types of instance events, see [Scheduled events for your instances](#).

Contents

- [Identify instances scheduled for retirement](#)
- [Actions to take for EBS-backed instances scheduled for retirement](#)
- [Actions to take for instance-store backed instances scheduled for retirement](#)

Identify instances scheduled for retirement

If your instance is scheduled for retirement, you receive an email prior to the event with the instance ID and retirement date. You can also check for instances that are scheduled for retirement using the Amazon EC2 console or the command line.

Important

If an instance is scheduled for retirement, we recommend that you take action as soon as possible because the instance might be unreachable. (The email notification you receive states the following: "Due to this degradation your instance could already be unreachable.") For more information about the recommended action you should take, see [Check if your instance is reachable](#).

Ways to identify instances scheduled for retirement

- [Email notification](#)
- [Console identification](#)

Email notification

If your instance is scheduled for retirement, you receive an email prior to the event with the instance ID and retirement date.

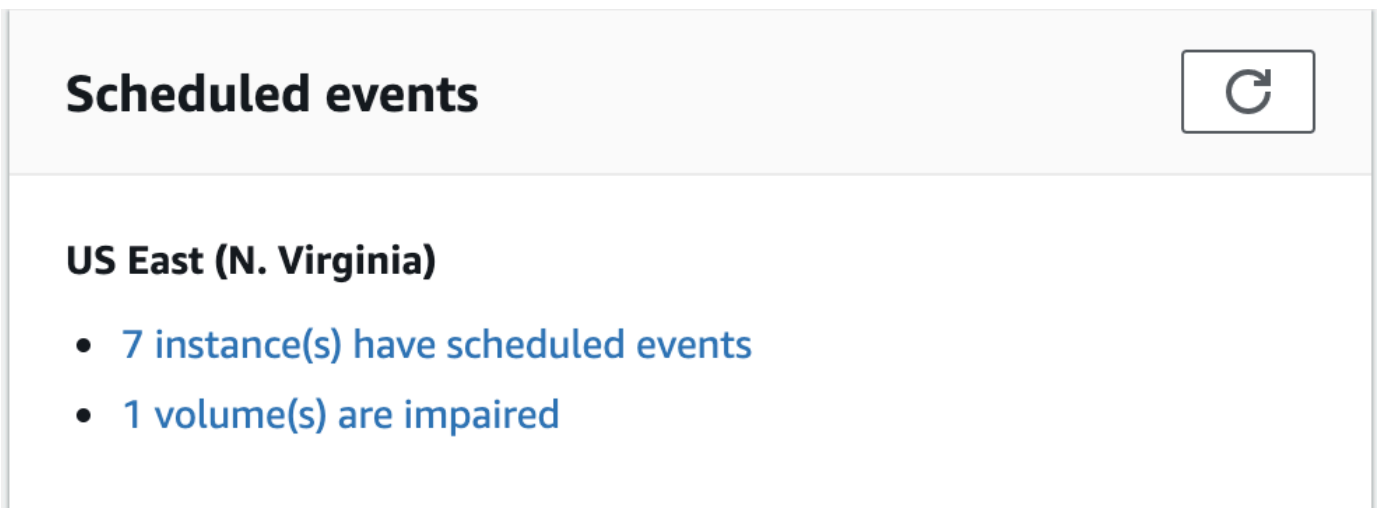
The email is sent to the primary account holder and the operations contact. For more information, see [Adding, changing, or removing alternate contacts](#) in the *AWS Billing User Guide*.

Console identification

If you use an email account that you do not check regularly for instance retirement notifications, you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **EC2 Dashboard**. Under **Scheduled events**, you can see the events that are associated with your Amazon EC2 instances and volumes, organized by Region.



3. If you have an instance with a scheduled event listed, select its link below the Region name to go to the **Events** page.
4. The **Events** page lists all resources that have events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance stop or retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

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

- [describe-instance-status](#) (AWS CLI)
- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)

Actions to take for EBS-backed instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

If you are not sure whether your instance is backed by EBS or instance store, see [Determine the root device type of your 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 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 Linux 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 **Actions, Launch**. Follow the wizard to launch your instance. For more information about each step in the wizard, see [Launch an instance using the new launch instance wizard](#).

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 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 Linux AMI](#). From the Amazon EC2 console, select your new AMI and then choose **Actions, Launch**. Follow the wizard to launch your instance. For more information about each step in the wizard, see [Launch an instance using the new launch instance wizard](#).

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 Amazon EBS-backed AMI](#).

Recover your instance

To automatically recover an instance when a system status check failure occurs, you can use the default configuration of the instance or create an Amazon CloudWatch alarm. If an instance becomes unreachable because of an underlying hardware failure or a problem that requires AWS involvement to repair, the instance is automatically recovered.

A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata. If the impaired instance has a public IPv4 address, the instance retains the public IPv4 address after recovery. If the impaired instance is in a placement group, the recovered instance runs in the placement group. During instance recovery, the instance is migrated as part of an instance reboot, and any data that is in-memory is lost.

Examples of problems that require instance recovery:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

Topics

- [Simplified automatic recovery based on instance configuration](#)
- [Amazon CloudWatch action based recovery](#)
- [Troubleshoot instance recovery failures](#)

Simplified automatic recovery based on instance configuration

Instances that support simplified automatic recovery are configured by default to recover a failed instance. The default configuration applies to new instances that you launch and existing instances that you previously launched. Simplified automatic recovery is initiated in response to system status check failures. Simplified automatic recovery doesn't take place during Service Health Dashboard events, or any other events that impact the underlying hardware. For more information, see [the section called "Troubleshoot instance recovery failures"](#).

When a simplified automatic recovery event succeeds, you are notified by an AWS Health Dashboard event. When a simplified automatic recovery event fails, you are notified by an AWS Health Dashboard event and by email. You can also use Amazon EventBridge rules to monitor for simplified automatic recovery events using the following event codes:

- `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS` — successful events
- `AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE` — failed events

For more information, see [Amazon EventBridge rules](#).

Topics

- [Requirements](#)
- [Limitations](#)
- [Set the recovery behavior](#)

Requirements

Simplified automatic recovery is supported by an instance if the instance has the following characteristics:

- It uses default or dedicated instance tenancy.
- It does not use an Elastic Fabric Adapter.
- It uses one of the following instance types:
 - **General purpose:** A1 | M3 | M4 | M5 | M5a | M5n | M5zn | M6a | M6g | M6i | M6in | M7a | M7g | M7i | M7i-flex | T1 | T2 | T3 | T3a | T4g
 - **Compute optimized:** C3 | C4 | C5 | C5a | C5n | C6a | C6g | C6gn | C6i | C6in | C7a | C7g | C7gn | C7i
 - **Memory optimized:** R3 | R4 | R5 | R5a | R5b | R5n | R6a | R6g | R6i | R6in | R7a | R7g | R7i | R7iz | u-3tb1 | u-6tb1 | u-9tb1 | u-12tb1 | u-18tb1 | u-24tb1 | X1 | X1e | X2iezn
 - **Accelerated computing:** G3 | G3s | G5g | Inf1 | P2 | P3 | VT1
 - **High-performance computing** Hpc6a | Hpc7a | Hpc7g
- It does not have instance store volumes. If a Nitro instance type has instance store volumes, or if a Xen-based instance has mapped instance store volumes in the AMI being used, the instance can't be automatically recovered.

Important

If an instance has instance store volumes attached, stopping and starting the instance will cause any data on the instance store volumes to be lost. You should regularly backup your instance store volume data to more persistent storage, such as Amazon EBS, Amazon S3, or Amazon EFS. In the event of a system status check failure, you can stop and start instances with instance store volumes and then restore the instance store volumes using the backed-up data.

Limitations

- Instances with instance store volumes and metal instance types are not supported by simplified automatic recovery.
- Simplified automatic recovery is not initiated for instances in an Auto Scaling group. If your instance is part of an Auto Scaling group with health checks enabled, then the instance is replaced when it becomes impaired.
- Simplified automatic recovery applies to unplanned events only. It does not apply to scheduled events.
- Terminated or stopped instances can't be recovered.

Set the recovery behavior

You can set the automatic recovery behavior to `disabled` or `default` during or after launching the instance. The default configuration does not enable simplified automatic recovery for an unsupported instance type.

Console

To disable simplified automatic recovery during instance 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**, select **Disabled**.
4. Configure the remaining instance launch settings as needed and then launch the instance.

To disable simplified automatic recovery for a running or stopped 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 then choose **Actions, Instance settings, Change auto-recovery behavior**.
4. Choose **Off**, and then choose **Save**.

To set the automatic recovery behavior to default for a running or stopped 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 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.

```
aws ec2 run-instances \  
--image-id ami-1a2b3c4d \  
--instance-type t2.micro \  
--key-name MyKeyPair \  
--maintenance-options AutoRecovery=Disabled \  
[...]
```

To disable simplified automatic recovery for a running or stopped instance

Use the [modify-instance-maintenance-options](#) command.

```
aws ec2 modify-instance-maintenance-options \  
--instance-id i-0abcdef1234567890 \  
--auto-recovery disabled
```

To set the automatic recovery behavior to default for a running or stopped instance

Use the [modify-instance-maintenance-options](#) command.

```
aws ec2 modify-instance-maintenance-options \  
--instance-id i-0abcdef1234567890 \  
--auto-recovery default
```

Amazon CloudWatch action based recovery

Use Amazon CloudWatch action based recovery if you want to customize when to recover your instance.

When the `StatusCheckFailed_System` alarm is triggered, and the recovery action is initiated, you're notified by the Amazon SNS topic that you selected when you created the alarm and associated the recovery action. When the recovery action is complete, information is published to the Amazon SNS topic you configured for the alarm. Anyone who is subscribed to this Amazon SNS topic receives an email notification that includes the status of the recovery attempt and any further instructions. As a last step in the recovery action, the recovered instance reboots.

You can use Amazon CloudWatch alarms to recover an instance even if simplified automatic recovery is not disabled. For information about creating an Amazon CloudWatch alarm to recover an instance, see [Add recover actions to Amazon CloudWatch alarms](#).

Supported instance types

All of the instance types [supported by simplified automatic recovery](#) are also supported by Amazon CloudWatch action based recovery. Additionally, CloudWatch action based recovery supports bare metal variants of the supported instance types. The following instance families are also supported in addition to those supported by simplified automatic recovery:

- **Memory optimized:** X2idn | X2iedn

Important

For supported instance types that have instance store volumes, any data on these volumes will be lost during a recovery. Stopping and starting the instance will also cause any data on the instance store volume to be lost. You should regularly backup your instance store volume data to more persistent storage, such as Amazon EBS, Amazon S3, or Amazon EFS.

In the event of a system status check failure, you can stop and start instances with instance store volumes and then restore the instance store volumes using the backed-up data.

CloudWatch action based recovery does not support recovery for instances with Dedicated Host tenancy. For Amazon EC2 Dedicated Hosts, you can use [Dedicated Host Auto Recovery](#) to automatically recover unhealthy instances.

You can use the AWS Management Console or the AWS CLI to view the instance types that support CloudWatch action based recovery.

Console

To view the instance types that support Amazon 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, enter **Auto Recovery support: true**. Alternatively, 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 Amazon CloudWatch action based recovery.

AWS CLI

To view the instance types that support Amazon CloudWatch action based recovery

Use the [describe-instance-types](#) command.

```
aws ec2 describe-instance-types --filters Name=auto-recovery-supported,Values=true
--query "InstanceTypes[*].[InstanceType]" --output text | sort
```

Troubleshoot instance recovery failures

The following issues can cause the recovery of your instance to fail:

- During Service Health Dashboard events, simplified automatic recovery might not recover your instance. You might not receive recovery failure notifications for such events. Any ongoing

Service Health Dashboard events might also prevent CloudWatch action based recovery from successfully recovering an instance. For the latest service availability information, see <http://status.aws.amazon.com/>.

- Temporary, insufficient capacity of replacement hardware.
- The instance has reached the maximum daily allowance of three recovery attempts.

The automatic recovery process attempts to recover your instance for up to three separate failures per day. If the instance system status check failure persists, we recommend that you manually stop and start the instance. Data on instance store volumes is lost when the instance is stopped. For more information, see [Stop and start Amazon EC2 instances](#).

Your instance might subsequently be retired if automatic recovery fails and a hardware degradation is determined to be the root cause for the original system status check failure.

Connect

This section of the *Amazon EC2 User Guide for Linux Instances* provides information to help you connect to your Linux instance after you launch it. It also provides information to help you connect your Linux instance to another AWS resource.

For information about how to connect to a **Windows instance**, see [Connect to your Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Topics

- [Connect to your Linux instance](#)
- [Connect to your instances using EC2 Instance Connect Endpoint](#)
- [Connect your EC2 instance to an AWS resource](#)

Connect to your Linux instance

There are many ways to connect to your Linux instance. Some vary depending on the operating system of the local machine that you connect from. Others, like EC2 Instance Connect or AWS Systems Manager Session Manager, don't vary. In this section, you can learn how to connect to your Linux instance and transfer files between your local computer and your instance. For information about how to connect to a **Windows instance**, see [Connect to your Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Before you connect to your Linux instance, complete the following prerequisites.

- [Get information about your instance](#)
- [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.

Options to connect based on your local operating system

- [Connect from a Linux or macOS local machine using SSH](#)
- [Connect from a Windows local machine](#)

Options to connect from any local operating system

- [Connect to your Linux instance with AWS Systems Manager Session Manager](#)
- [Connect to your Linux instance with EC2 Instance Connect.](#)

Note

For instance connection troubleshooting tips, see [Troubleshoot connecting to your instance](#).

To troubleshoot boot, network configuration, and other issues for instances built on the [AWS Nitro System](#), you can use the [EC2 Serial Console for Linux instances](#).

Get information about your instance

To prepare to connect to an instance, get the following information from the Amazon EC2 console or by using the AWS CLI.

The screenshot shows the Amazon EC2 console interface. At the top, there's a notification 'Successfully started i-...' and a 'Launch Instances' button. Below is the 'Instances (1/8)' table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The first instance 'Windows' is in a 'Running' state. Below the table, the 'Instance: i-05' details are shown under the 'Details' tab. The 'Instance summary' section includes fields for Instance ID, IPv6 address, Public IPv4 address, Instance state, Private IP DNS name, Instance type, VPC ID, Subnet ID, and Public IPv4 DNS.

- **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) 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](#).

- **Get the user name 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 users on your Linux instance](#).

- **Get the default username for the AMI that you used to launch your instance:**

AMI used to launch instance	Default username
AL2023	ec2-user
Amazon Linux 2	
Amazon Linux	
CentOS	centos or ec2-user
Debian	admin
Fedora	fedora or ec2-user
RHEL	ec2-user or root
SUSE	ec2-user or root

AMI used to launch instance	Default username
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 connect to your instance. For SSH connections, you must set the permissions so that only you can read the file.

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 [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 Linux instance?](#)

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](#) in the [Connect to your Linux instance from Windows with PuTTY](#) topic in this section.

- **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 [Installing the AWS Command Line Interface](#) 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](#) (AWS CLI) command. If the output is large, [you can pipe it to a text file](#), where it might be easier to read. Note that you must specify an AWS Region when you use the AWS CLI, either explicitly or by setting a default Region. For information about how to set or specify a Region, see [Configuration basics](#) in the *AWS Command Line Interface User Guide*.

```
aws ec2 get-console-output --instance-id instance_id --query Output --output text > temp.txt
```

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 SHA256:14UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY no comment (ECDSA)  
ec2: 256 SHA256:kpEa+rw/Uq3zxaYZN8KT501iBtJ0IdHG52dFi66EEfQ no comment (ED25519)  
ec2: 2048 SHA256:L816pepcA7iqW/jBecQjVZC1UrKY+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 from Linux or macOS using SSH

You can use Secure Shell (SSH) to connect to your Linux instance from a local machine that runs a Linux or macOS operating system, or you can use a platform independent connection tool, such as EC2 Instance Connect or AWS Systems Manager Session Manager. For more information about platform independent tools, see [Connect to your Linux instance](#).

This page explains how to connect to your instance with an SSH client. To connect to your Linux instance from Windows, see [Connect from Windows](#).

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 connecting to your instance](#).

Contents

- [SSH connection prerequisites](#)
- [Connect to your Linux instance using an SSH client](#)
- [Transfer files to Linux instances using an SCP client](#)

SSH connection prerequisites

Before you connect to your Linux instance, complete the following prerequisites.

Check your instance status

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status check** column on the **Instances** page.

Get the public DNS name and user name to connect to your instance

To find the public DNS name or IP address of your instance and the user name that you should use to connect to your instance, see [Get information about your instance](#).

Locate the private key and set the permissions

To locate the private key that is required to connect to your instance, and to set the key permissions, see [Locate the private key and set permissions](#).

Install an SSH client on your local computer as needed

Your local computer might have an SSH client installed by default. You can verify this by typing **ssh** at the command line. If your computer doesn't recognize the command, you can install an SSH client.

- Recent versions of Windows Server 2019 and Windows 10 – OpenSSH is included as an installable component. For more information, see [OpenSSH in Windows](#).
- Earlier versions of Windows – Download and install OpenSSH. For more information, see [Win32-OpenSSH](#).
- Linux and macOS X – Download and install OpenSSH. For more information, see <https://www.openssh.com>.

Connect to your Linux instance using an SSH client

Use the following procedure to connect to your Linux instance using an SSH client. If you receive an error while attempting to connect to your instance, see [Troubleshoot connecting to your instance](#).

Connect to your instance using SSH

1. In a terminal window, use the **ssh** command to connect to the instance. You specify the path and file name of the private key (.pem), the user name for your instance, and the public DNS name or IPv6 address for your instance. For more information about how to find the private key, the user name for your instance, and the DNS name or IPv6 address for an instance, see [Locate the private key and set permissions](#) and [Get information about your instance](#). To connect to your instance, use one of the following commands.

- (Public DNS) To connect using your instance's public DNS name, enter the following command.

```
ssh -i /path/key-pair-name.pem 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 the following command.

```
ssh -i /path/key-pair-name.pem instance-user-name@instance-IPv6-address
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'
can't be established.
ECDSA key fingerprint is 14UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no)?
```

2. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint](#). If these fingerprints don't match, someone might be attempting a man-in-the-middle attack. If they match, continue to the next step.
3. Enter **yes**.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to
the list of known hosts.
```

Transfer files to Linux instances using an SCP client

One way to transfer files between your local computer and a Linux instance is to use the secure copy protocol (SCP). This section describes how to transfer files with SCP. The procedure is similar to the procedure for connecting to an instance with SSH.

Prerequisites

- **Verify the general prerequisites for transferring files to your instance.**

Before you transfer files between your local machine and your instance, take the following actions to ensure that you have all of the information you need.

- [Get information about your instance](#)
- [Locate the private key and set permissions](#)
- [\(Optional\) Get the instance fingerprint](#)

- **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 user name 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 `instance-IPv6-address`.
 - (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@[instance-IPv6-address]: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@[instance-IPv6-address]:path/my-
file.txt path/my-file2.txt
```

Connect to your Linux instance from Windows

You can use the following methods to connect to your Linux instance from a local machine that has a Windows operating system.

- [OpenSSH](#)
- [PuTTY](#)
- [Windows Subsystem for Linux](#)

Connect to your Linux instance from Windows with OpenSSH

The following procedures show you how to connect to your Linux instance from Windows using OpenSSH, an open source connectivity tool for remote login with the SSH protocol. OpenSSH is supported on Windows Server 2019 and later operating systems.

Contents

- [Prerequisites](#)
- [Install OpenSSH for Windows using PowerShell](#)
- [Connect to your Linux instance from Windows using OpenSSH](#)
- [Uninstall OpenSSH from Windows using PowerShell](#)

Prerequisites

Before you connect to your Linux instance from Windows using OpenSSH, complete the following prerequisites.

Verify that the instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status check** column on the **Instances** page.

Verify the general prerequisites for connecting to your instance

To find the public DNS name or IP address of your instance and the username that you should use to connect to your instance, see [Get information about your instance](#).

Verify your Windows version

To connect to your Linux instance from Windows using OpenSSH, the Windows version must be Windows Server 2019 and later.

Verify PowerShell prerequisites

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`.

Install OpenSSH for Windows using PowerShell

To install OpenSSH for Windows using PowerShell, run the following PowerShell command:

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

Expected output:

```
Path          :
Online        : True
RestartNeeded : False
```

Connect to your Linux instance from Windows using OpenSSH

After you install OpenSSH, use the following procedure to connect to your Linux instance from Windows using OpenSSH. If you receive an error while attempting to connect to your instance, see [Troubleshoot connecting to your instance](#).

To connect to your instance using OpenSSH

1. In PowerShell or the Command Prompt, use the `ssh` command to connect to the instance. You specify the path and file name of the private key (`.pem`), the username for your instance, and the public DNS name or IPv6 address for your instance. For more information about how to find the private key, the username for your instance, and the DNS name or IPv6 address for an instance, see [Locate the private key and set permissions](#) and [Get information about your instance](#). To connect to your instance, use one of the following commands.
 - (Public DNS) To connect using your instance's public DNS name, enter the following command.

```
ssh -i /path/key-pair-name.pem 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 the following command.

```
ssh -i /path/key-pair-name.pem instance-user-name@instance-IPv6-address
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'
can't be established.
ECDSA key fingerprint is 14UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

2. (Optional) Verify that the fingerprint in the security alert 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.
3. Enter **yes**.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to
the list of known hosts.
```

Uninstall OpenSSH from Windows using PowerShell

To uninstall OpenSSH from Windows using PowerShell, run the following PowerShell command:

```
Remove-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

Expected output:

```
Path          :
Online        : True
RestartNeeded : True
```

Connect to your Linux instance from Windows with PuTTY

If you're running Windows Server 2019 or later, we recommend using OpenSSH, which is an open source connectivity tool for remote login with the SSH protocol. For steps to connect to a Linux instance from Windows using OpenSSH, see [Connect to your Linux instance from Windows with OpenSSH](#).

The following instructions explain how to connect to your instance using PuTTY, a free SSH client for Windows. If you receive an error while attempting to connect to your instance, see [Troubleshoot connecting to your instance](#).

Contents

- [Prerequisites](#)
 - [Convert your private key using PuTTYgen](#)
- [Connect to your Linux instance](#)
- [Transfer files to your Linux instance using the PuTTY Secure Copy client](#)
- [Transfer files to your Linux instance using WinSCP](#)

Prerequisites

Before you connect to your Linux instance using PuTTY, complete the following prerequisites.

Verify that the instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status check** column on the **Instances** page.

Verify the general prerequisites for connecting to your instance

To find the public DNS name or IP address of your instance and the username that you should use to connect to your instance, see [Get information about your instance](#).

Install PuTTY on your local computer

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 .pem key to .ppk using PuTTYgen

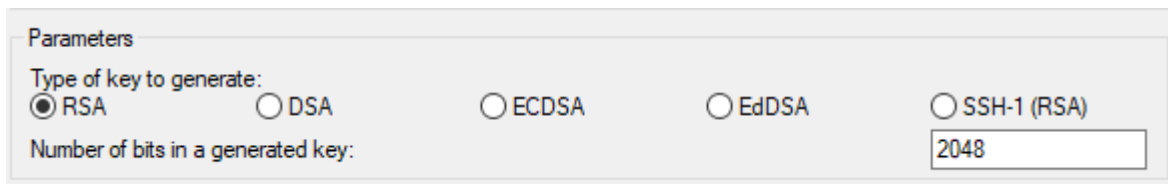
For the key pair that you specified when you launched the instance, if you chose to create the private key in the .pem format, you must convert it to a .ppk file for use with PuTTY. Locate the private .pem file, and then follow the steps in the next section.

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. You must convert your private key (.pem file) into this format (.ppk file) as follows in order to connect to your instance using PuTTY.

To convert your private .pem key to .ppk

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 [Convert your private key using PuTTYgen](#) in the preceding section. If you receive an error while attempting to connect to your instance, see [Troubleshoot connecting to your instance](#).

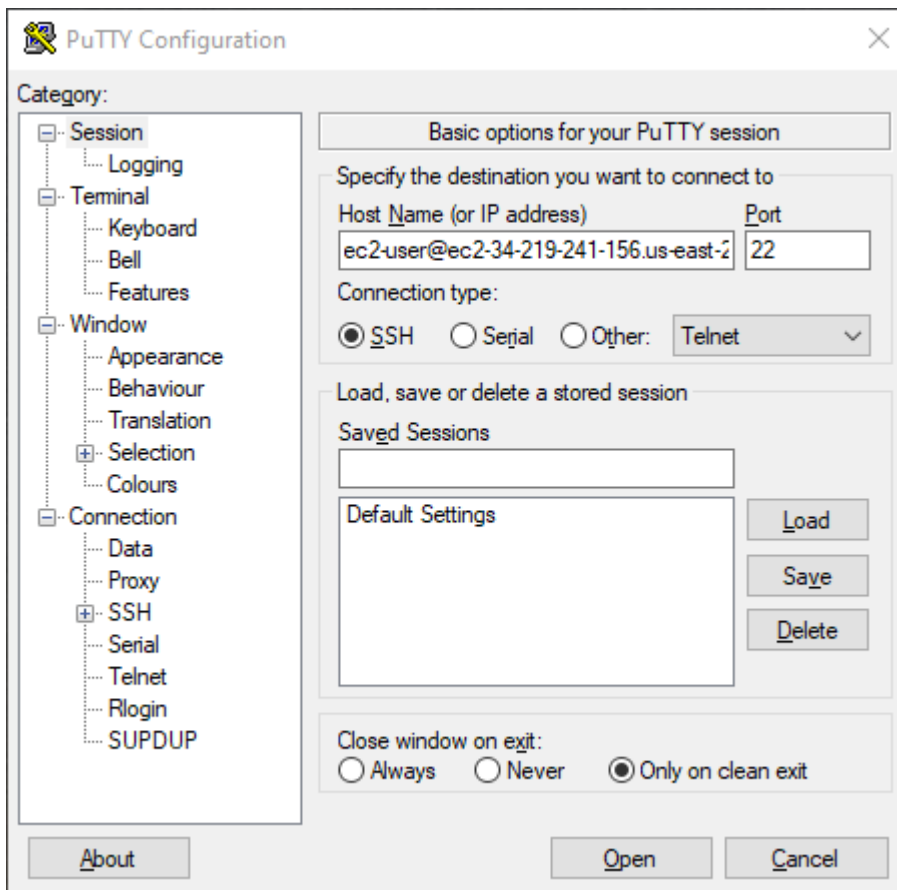
Last tested version of PuTTY: .78

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@instance-IPv6-address*.

For information about how to get the username for your instance, and the public DNS name or IPv6 address of your instance, see [Get information about your instance](#).

- 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 connecting to your instance](#).

Transfer files to your Linux instance using the PuTTY Secure Copy client

The PuTTY Secure Copy client (PSCP) is a command line tool that you can use to transfer files between your Windows computer and your Linux instance. If you prefer a graphical user interface (GUI), you can use an open source GUI tool named WinSCP. For more information, see [Transfer files to your Linux instance using WinSCP](#).

To use PSCP, you need the private key you generated in [Convert your private key using PuTTYgen](#). You also need the public DNS name of your Linux instance, or the IPv6 address if your instance has one.

The following example transfers the file `Sample_file.txt` from the `C:\` drive on a Windows computer to the `instance-user-name` home directory on an Amazon Linux instance. To transfer a file, use one of the following commands.

- (Public DNS) To transfer a file using your instance's public DNS name, enter the following command.

```
pscp -i C:\path\my-key-pair.ppk C:\path\Sample_file.txt instance-user-name@instance-public-dns-name:/home/instance-user-name/Sample_file.txt
```

- (IPv6) Alternatively, if your instance has an IPv6 address, to transfer a file using your instance's IPv6 address, enter the following command. The IPv6 address must be enclosed in square brackets ([]).

```
pscp -i C:\path\my-key-pair.ppk C:\path\Sample_file.txt instance-user-name@[instance-IPv6-address]:/home/instance-user-name/Sample_file.txt
```

Transfer files to your Linux instance using WinSCP

WinSCP is a GUI-based file manager for Windows that you can use to upload and transfer files to a remote computer using the SFTP, SCP, FTP, and FTPS protocols. With WinSCP, you can drag and drop files from your Windows computer to your Linux instance or synchronize entire directory structures between the two systems.


Requirements

- You must have the private key that you generated in [Convert your private key using PuTTYgen](#).
- You must have the public DNS name of your Linux instance.
- Your Linux instance must have scp installed. For some operating systems, you install the openssh-clients package. For others, such as the Amazon ECS-optimized AMI, you install the scp package. Check the documentation for your Linux distribution.

To connect to your instance using WinSCP

1. Download and install WinSCP from <http://winscp.net/eng/download.php>. For most users, the default installation options are OK.
2. Start WinSCP.
3. At the **WinSCP login** screen, for **Host name**, enter one of the following:
 - (Public DNS or IPv4 address) To log in using your instance's public DNS name or public IPv4 address, enter the public DNS name or public IPv4 address for your instance.
 - (IPv6) Alternatively, if your instance has an IPv6 address, to log in using your instance's IPv6 address, enter the IPv6 address for your instance.
4. For **User name**, enter the default username for your AMI.
 - For AL2023, Amazon Linux 2, or the Amazon Linux AMI, the user name is ec2-user.
 - For a CentOS AMI, the user name is centos or ec2-user.
 - For a Debian AMI, the user name is admin.
 - For a Fedora AMI, the user name is fedora or ec2-user.

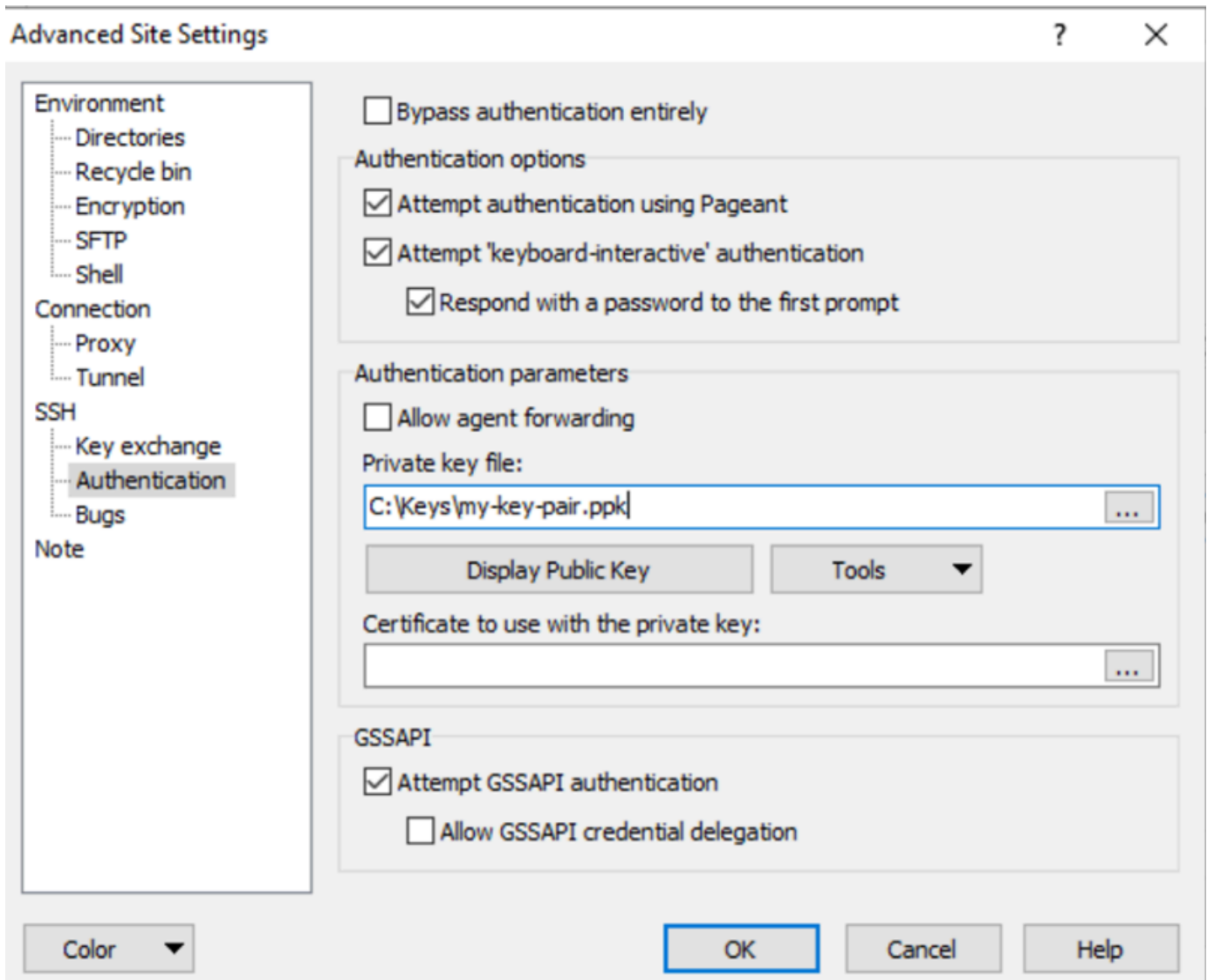
- For a RHEL AMI, the user name is `ec2-user` or `root`.
- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- For an Oracle AMI, the user name is `ec2-user`.
- For a Bitnami AMI, the user name is `bitnami`.

 **Note**

To find the default user name for other Linux distributions, check with the AMI provider.

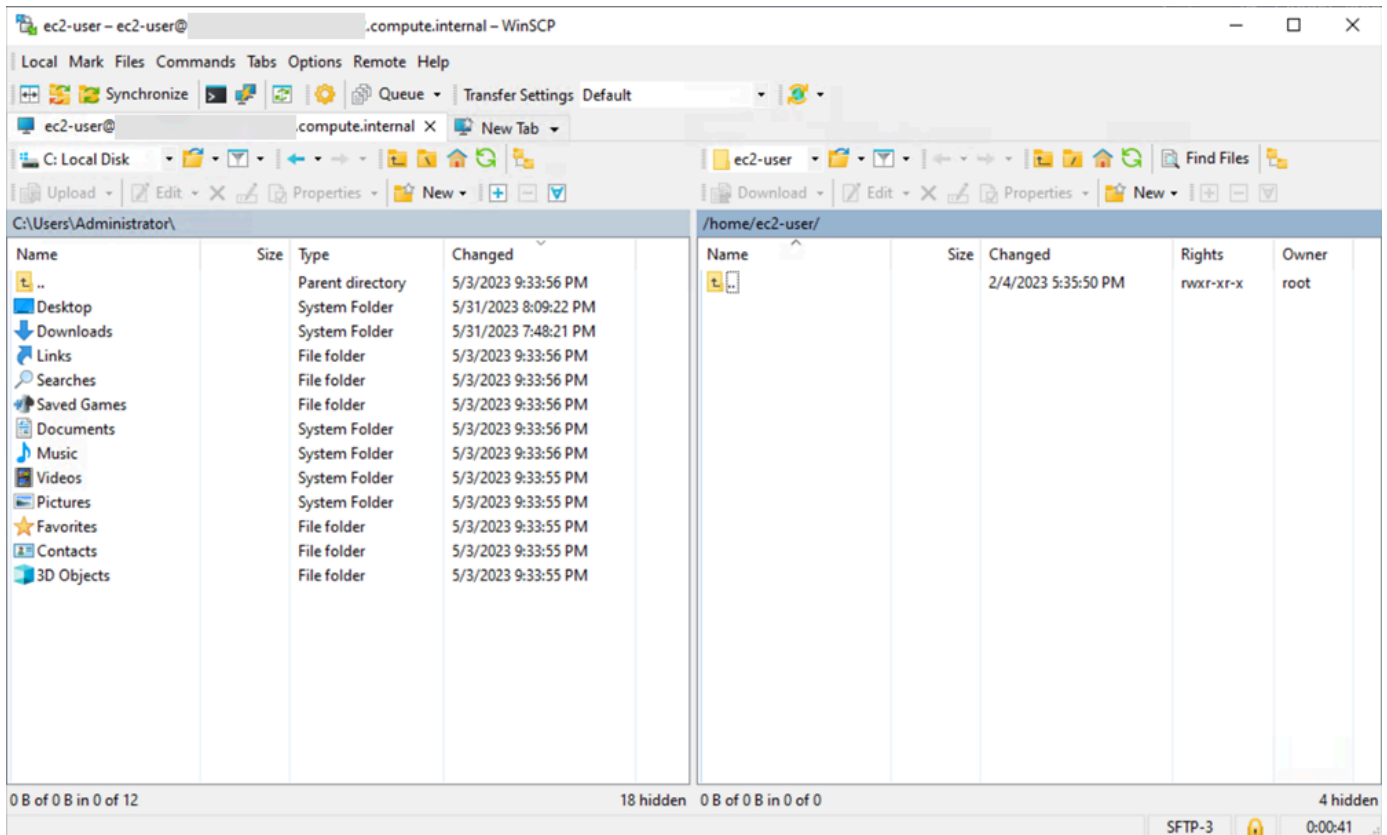
5. Specify the private key file for your instance.
 - a. Choose the **Advanced...** button.
 - b. Under SSH, choose **Authentication**.
 - c. Specify the path for your private key file, or choose the **...** button to browse to the key pair file.
 - d. Choose **OK**.

Here is a screenshot from WinSCP version 6.1:



WinSCP requires a PuTTY private key file (.ppk). You can convert a .pem security key file to the .ppk format using PuTTYgen. For more information, see [Convert your private key using PuTTYgen](#).

6. (Optional) In the left panel, choose **Directories**. For **Remote directory**, enter the path for the directory to which to add files. To open the advanced site settings for newer versions of WinSCP, choose **Advanced**. To find the **Remote directory** setting, under **Environment**, choose **Directories**.
7. Choose **Login**. To add the host fingerprint to the host cache, choose **Yes**.



8. After the connection is established, in the connection window your Linux instance is on the right and your local machine is on the left. You can drag and drop files between the remote file system and your local machine. For more information on WinSCP, see the project documentation at <http://winscp.net/eng/docs/start>.

If you receive an error that you cannot run SCP to start the transfer, verify that you installed **scp** on the Linux instance.

Connect to your Linux instance from Windows with Windows Subsystem for Linux (WSL)

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

The following instructions explain how to connect to your instance with a Linux distribution on the Windows Subsystem for Linux (WSL). WSL is a free download and enables you to run native Linux command line tools directly on Windows, alongside your traditional Windows desktop, without the overhead of a virtual machine.

By installing WSL, you can use a native Linux environment to connect to your Linux EC2 instances instead of using PuTTY or PuTTYgen. The Linux environment makes it easier to connect to your

Linux instances because it comes with a native SSH client that you can use to connect to your Linux instances and change the permissions of the .pem key file. The Amazon EC2 console provides the SSH command for connecting to the Linux instance, and you can get verbose output from the SSH command for troubleshooting. For more information, see the [Windows Subsystem for Linux Documentation](#).

Note

After you've installed the WSL, all the prerequisites and steps are the same as those described in [Connect to your Linux instance from Linux or macOS using SSH](#), and the experience is just like using native Linux.

If you receive an error while attempting to connect to your instance, see [Troubleshoot connecting to your instance](#).

Contents

- [Prerequisites](#)
- [Connect to your Linux instance using WSL](#)
- [Transfer files to Linux instances from Linux using SCP](#)
- [Uninstall WSL](#)

Prerequisites

Before you connect to your Linux instance, complete the following prerequisites.

Verify that the instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status check** column on the **Instances** page.

Verify the general prerequisites for connecting to your instance

To find the public DNS name or IP address of your instance and the user name that you should use to connect to your instance, see [Get information about your instance](#).

Install the Windows Subsystem for Linux (WSL) and a Linux distribution on your local computer

Install the WSL and a Linux distribution using the instructions in the [Windows 10 Installation Guide](#). The example in the instructions installs the Ubuntu distribution of Linux, but you can install any distribution. You are prompted to restart your computer for the changes to take effect.

Copy the private key from Windows to WSL

In a WSL terminal window, copy the `.pem` file (for the key pair that you specified when you launched the instance) from Windows to WSL. Note the fully-qualified path to the `.pem` file on WSL to use when connecting to your instance. For information about how to specify the path to your Windows hard drive, see [How do I access my C drive?](#). For more information about key pairs and Windows instances, see [Amazon EC2 key pairs and Windows instances](#).

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

Connect to your Linux instance using WSL

Use the following procedure to connect to your Linux instance using the Windows Subsystem for Linux (WSL). If you receive an error while attempting to connect to your instance, see [Troubleshoot connecting to your instance](#).

To connect to your instance using SSH

1. In a terminal window, use the `ssh` command to connect to the instance. You specify the path and file name of the private key (`.pem`), the user name for your instance, and the public DNS name or IPv6 address for your instance. For more information about how to find the private key, the user name for your instance, and the DNS name or IPv6 address for an instance, see [Locate the private key and set permissions](#) and [Get information about your instance](#). To connect to your instance, use one of the following commands.

- (Public DNS) To connect using your instance's public DNS name, enter the following command.

```
ssh -i /path/key-pair-name.pem instance-user-name@my-instance-public-dns-name
```

- (IPv6) Alternatively, if your instance has an IPv6 address, you can connect to the instance using its IPv6 address. Specify the `ssh` command with the path to the private key (`.pem`) file, the appropriate user name, and the IPv6 address.


```
ssh -i /path/key-pair-name.pem instance-user-name@my-instance-IPv6-address
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'
can't be established.
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

2. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint](#). If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
3. Enter yes.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
```

Transfer files to Linux instances from Linux using SCP

One way to transfer files between your local computer and a Linux instance is to use the secure copy protocol (SCP). This section describes how to transfer files with SCP. The procedure is similar to the procedure for connecting to an instance with SSH.

Prerequisites

- **Verify the general prerequisites for transferring files to your instance.**

Before you transfer files between your local machine and your instance, take the following actions to ensure that you have all of the information you need.

- [Get information about your instance](#)
- [Locate the private key and set permissions](#)
- [\(Optional\) Get the instance fingerprint](#)

- **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. If you've already connected to the instance with SSH and have verified its fingerprints, you can start with the step that contains the SCP command (step 4).

To use SCP to transfer a file

1. Transfer a file to your instance using the instance's public DNS name. For example, if the name of the private key file is `key-pair-name`, the file to transfer is `SampleFile.txt`, the user name is `instance-user-name`, and the public DNS name of the instance is `my-instance-public-dns-name` or the IPv6 address is `my-instance-IPv6-address`, use one of the following commands to copy the file to the `instance-user-name` home directory.

- (Public DNS) To transfer a file using your instance's public DNS name, enter the following command.

```
scp -i /path/key-pair-name.pem /path/SampleFile.txt instance-user-name@my-  
instance-public-dns-name:~
```

- (IPv6) Alternatively, if your instance has an IPv6 address, you can transfer a file using the instance's IPv6 address. The IPv6 address must be enclosed in square brackets ([]), which must be escaped (\).

```
scp -i /path/key-pair-name.pem /path/SampleFile.txt instance-user-name@[my-  
instance-IPv6-address]:~
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'  
can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

2. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint](#). If these fingerprints don't

match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.

3. Enter **yes**.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
Sending file modes: C0644 20 SampleFile.txt
Sink: C0644 20 SampleFile.txt
SampleFile.txt                               100%  20    0.0KB/s   00:00
```

If you receive a "bash: scp: command not found" error, you must first install **scp** on your Linux instance. For some operating systems, this is located in the `openssh-clients` package. For Amazon Linux variants, such as the Amazon ECS-optimized AMI, use the following command to install **scp**:

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

4. To transfer files in the other direction (from your Amazon EC2 instance to your local computer), reverse the order of the host parameters. For example, to transfer the `SampleFile.txt` file from your EC2 instance back to the home directory on your local computer as `SampleFile2.txt`, use one of the following commands on your local computer.

- (Public DNS) To transfer a file using your instance's public DNS name, enter the following command.

```
scp -i /path/key-pair-name.pem instance-user-
name@ec2-198-51-100-1.compute-1.amazonaws.com:~/SampleFile.txt ~/
SampleFile2.txt
```

- (IPv6) Alternatively, if your instance has an IPv6 address, to transfer files in the other direction using the instance's IPv6 address, enter the following command.

```
scp -i /path/key-pair-name.pem instance-user-name@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~/SampleFile.txt ~/SampleFile2.txt
```

Uninstall WSL

For information about uninstalling Windows Subsystem for Linux, see [How do I uninstall a WSL Distribution?](#)

Connect to your Linux instance with EC2 Instance Connect

Amazon EC2 Instance Connect provides a simple and secure way to connect to your Linux instances with 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 an SSH client of your choice.

When you connect to an instance using EC2 Instance Connect, the 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 Instance Connect is installed, to look up the public key from the instance metadata for authentication, and connects you to the instance.

You can use EC2 Instance Connect to connect to instances that have public or private IP addresses. For more information, see [Connect using EC2 Instance Connect](#).

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](#).

Tip

EC2 Instance Connect is one of the options to connect to your Linux instance. For other options, see [Connect to your Linux instance](#). To connect to a Windows instance, see [Connect to your Windows instance](#).

Contents

- [Tutorial: Complete the configuration required to connect to your instance using EC2 Instance Connect](#)

- [Prerequisites](#)
- [Grant IAM permissions for EC2 Instance Connect](#)
- [Install EC2 Instance Connect on your EC2 instances](#)
- [Connect using EC2 Instance Connect](#)
- [Uninstall 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: Create and attach an IAM policy to allow you 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: Create a security group to 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: Create and attach an IAM policy to allow you 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

In this task, 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.

Once 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:

```
{
  "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](#).

Steps to create and attach the IAM policy

Use the following steps to create and attach the IAM policy. To view an animation of the steps, see [View an animation: Create an IAM policy](#) and [View an animation: Attach an IAM policy](#).

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 check box 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

The screenshot displays the AWS Management Console Home page. At the top, there is a navigation bar with a hamburger menu icon on the left, the text "Console Home" with an "Info" link, a "Reset to default layout" button, and an "Add widgets" button. On the right side of the console, there are three utility icons: a help icon, a refresh icon, and a warning icon.

The main content area is divided into several widgets:

- Recently visited**: A grid of service tiles including IAM, EC2, CloudWatch, Amazon Bedrock, VPC, Resource Access Manager, RDS, Systems Manager, AWS FIS, AWS Outposts, and AWS Marketplace. A "View all services" link is at the bottom.
- Welcome to AWS**: A central widget with three sections: "Getting started with AWS" (with a rocket icon), "Training and certification" (with a document icon), and "What's new with AWS?".
- AWS Health**: A widget showing "Open Issues" (0), "Scheduled changes" (2), and "Other notifications" (0), all for the "Past 7 days". A "Go to AWS Health" link is at the bottom.
- Cost and usage**: A widget showing "Current month costs" as "\$5,588.24" and "Cost (\$)" as "15K".
- Build a solution**: A widget with two tiles: "Launch a virtual machine" (With EC2 (2 mins)) and "Start migrating to AWS" (With AWS MGN (2 mins)).

View an animation: Attach an IAM policy

Task 2: Create a security group to 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.

The IP address ranges for the AWS services are available at <https://ip-ranges.amazonaws.com/ip-ranges.json>. The EC2 Instance Connect IP address ranges are identified by "service": "EC2_INSTANCE_CONNECT".

Task objective

In this task, first you'll find the IP address range for EC2_INSTANCE_CONNECT in the AWS Region in which your instance is located. Then you'll create a security group that allows inbound SSH traffic on port 22 from that IP address range.

Steps to create the security group

Use the following steps to create the security group. To view an animation of the steps, see [View an animation: Get the IP address range for EC2 Instance Connect for a specific Region](#) and [View an animation: Configure a security group](#).

To create a security group that allows inbound traffic from the EC2 Instance Connect service to your instance

1. First get the IP address range for the EC2 Instance Connect service

- a. Open the AWS IP address ranges JSON file at <https://ip-ranges.amazonaws.com/ip-ranges.json>.
- b. Choose **Raw Data**.
- c. Find the IP address range for EC2_INSTANCE_CONNECT for the AWS Region in which your instance is located. You can use the browser search field to search for the service EC2_INSTANCE_CONNECT, and keep searching until you find the Region in which your instance is located.

For example, if your instance is located in the US East (N. Virginia) (us-east-1) Region, the IP address range for EC2_INSTANCE_CONNECT in that Region is 18.206.107.24/29.

Note

The IP address ranges are different for each AWS Region.

- d. Copy the IP address range that appears next to `ip_prefix`. You will use this IP address range later in this procedure.

For more information about downloading the AWS IP address ranges JSON file and filtering by service, see [AWS IP address ranges](#) in the *Amazon VPC User Guide*.

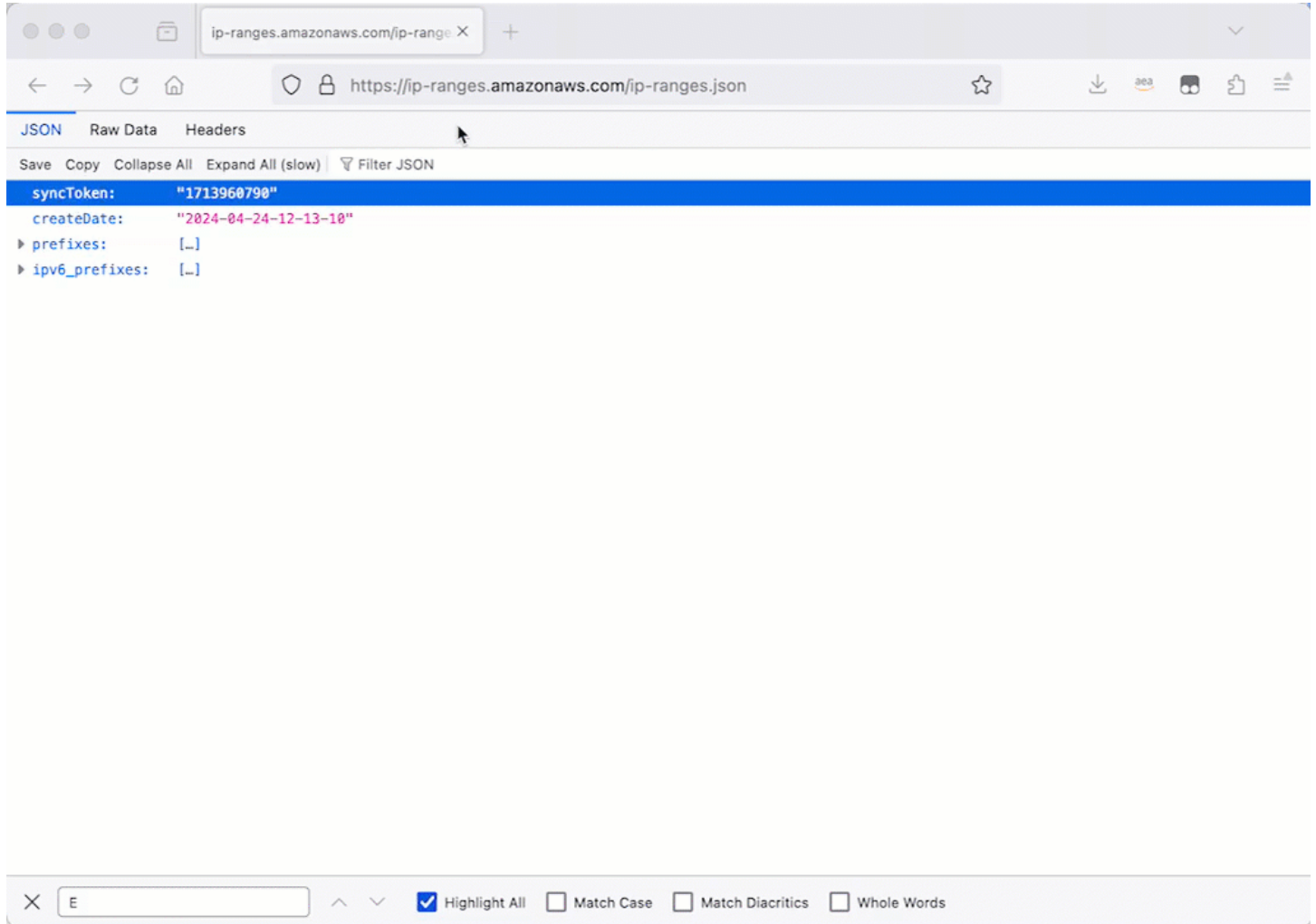
2. Then create the security group with an inbound rule to allow traffic from the copied IP address range

- a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- b. In the navigation pane, choose **Security Groups**.
- c. Choose **Create security group**.

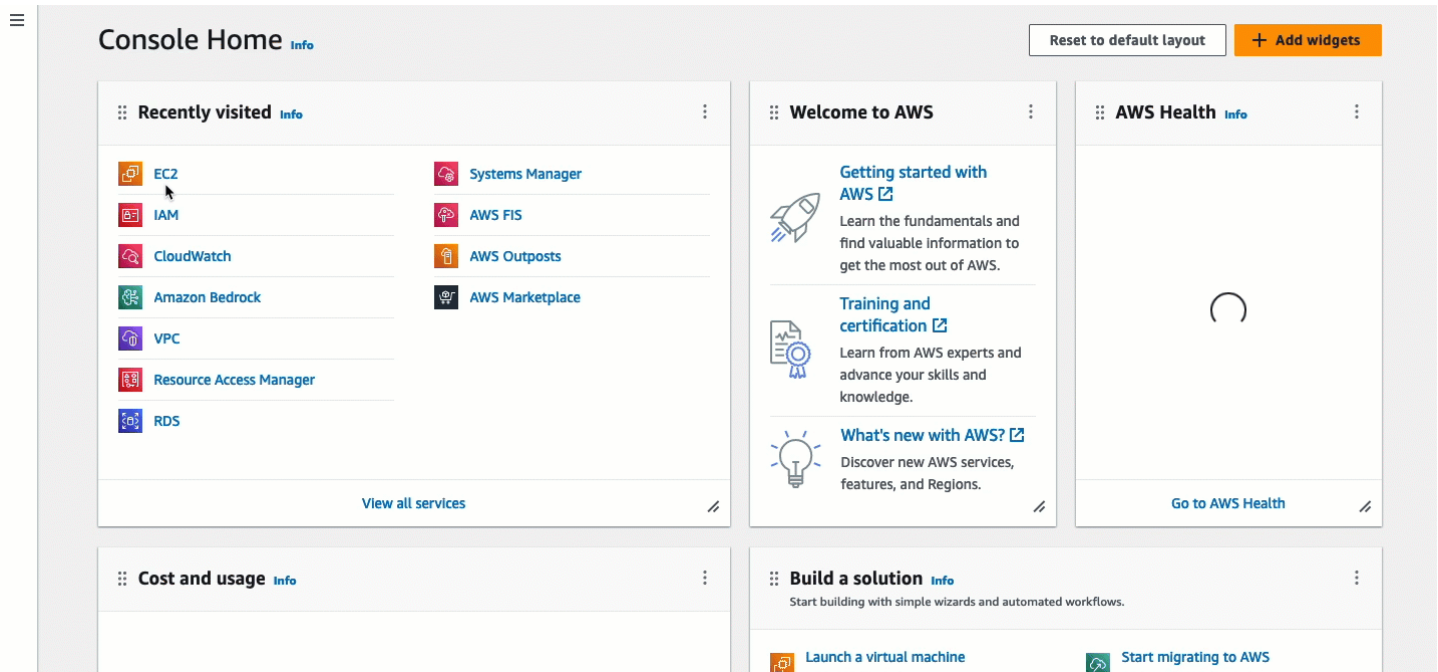
- d. Under **Basic details**, do the following:
 - i. For **Security group name**, enter a meaningful name for your security group.
 - ii. For **Description**, enter a meaningful description for your security group.
- e. Under **Inbound rules**, do the following:
 - i. Choose **Add rule**.
 - ii. For **Type**, choose **SSH**.
 - iii. For **Source**, leave **Custom**.
 - iv. In the field next to **Source**, paste the IP address range for the EC2 Instance Connect service that you copied earlier in this procedure.

For example, if your instance is located in the US East (N. Virginia) (us-east-1) Region, paste the following IP address range into the field: 18.206.107.24/29
- f. Choose **Create security group**.

View an animation: Get the IP address range for EC2 Instance Connect for a specific Region



View an animation: Configure a 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

In this task, 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. But you must ensure that your instance has a public IPv4 address because using EC2 Instance Connect in the Amazon EC2 console supports connecting only to instances with public IPv4 addresses.

Steps to launch your instance

Use the following steps to launch your instance. To view an animation of the steps, see [View an animation: Launch your instance](#).

To launch an instance that can use EC2 Instance Connect in the Amazon EC2 console for connection

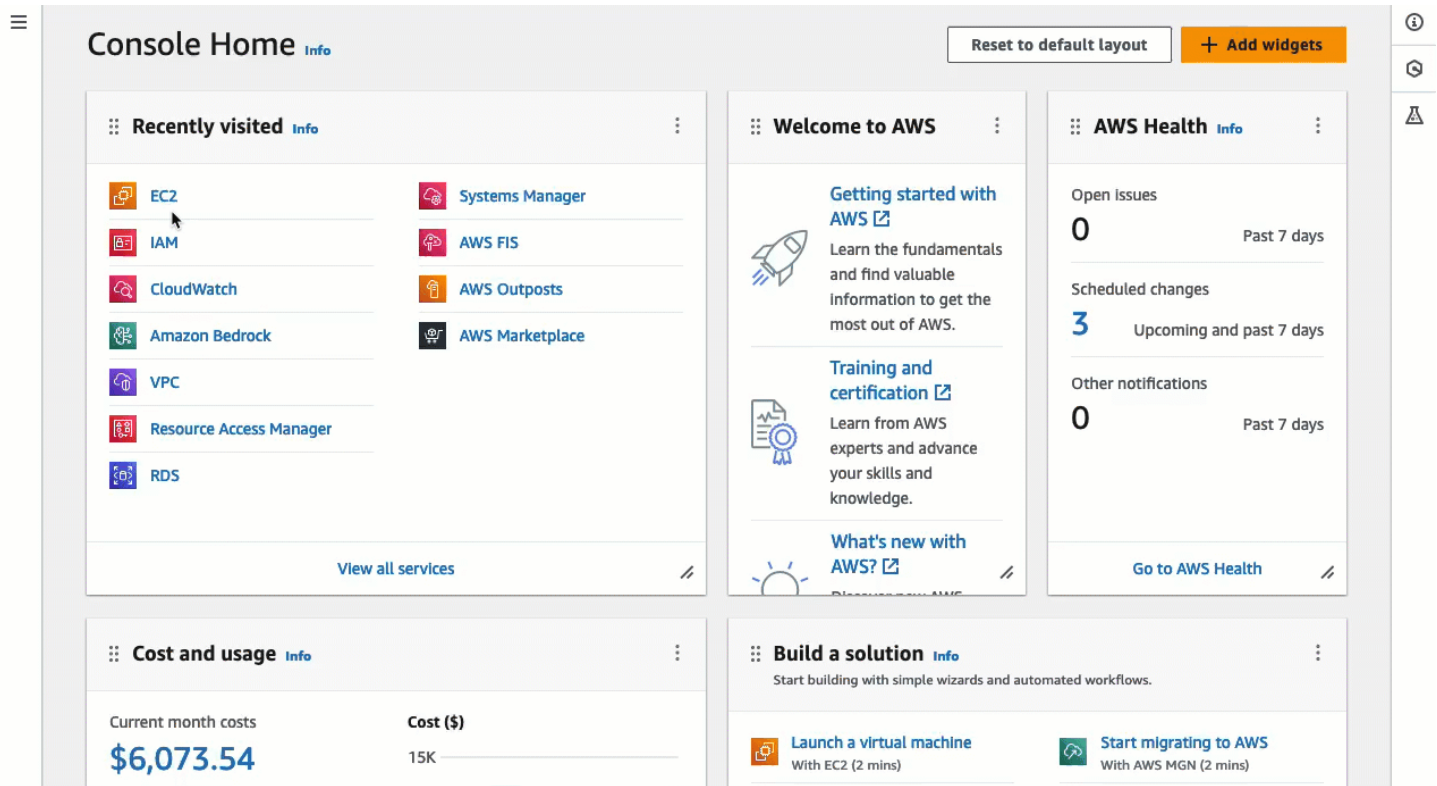
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current AWS Region is displayed (for example, **Ireland**). Select a Region in which to launch your instance. This choice is important because you created a security group that 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 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](#).

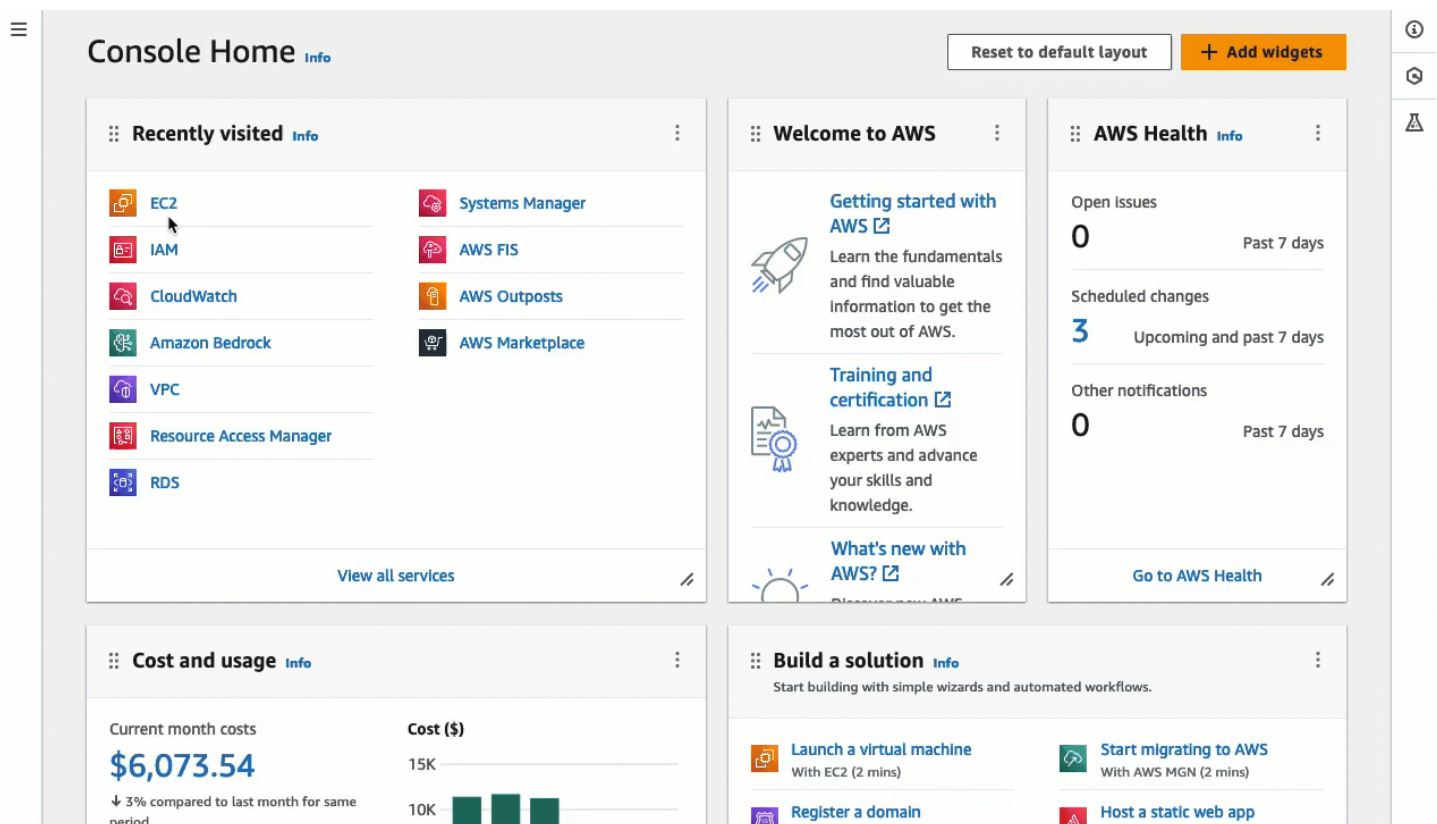
To connect an instance using EC2 Instance Connect in the Amazon EC2 console

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. For **Connection type**, choose **Connect using EC2 Instance Connect**.
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

The following are the prerequisites for installing EC2 Instance Connect and for using EC2 Instance Connect to connect to an instance:

- [AWS Regions](#)
- [Local Zones](#)

- [AMIs](#)
- [Install EC2 Instance Connect](#)
- [IPv4 address](#)
- [Network access](#)
- [Security group rule](#)
- [Grant permissions](#)
- [Local computer setup](#)
- [Username](#)

AWS Regions

Supported in all AWS Regions except Canada West (Calgary).

Local Zones

Not supported.

AMIs

EC2 Instance Connect comes pre-installed on the following AMIs:

- AL2023
- 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

EC2 Instance Connect is not pre-installed on the following AMIs, but you can install it on instances that are launched using the following AMIs:

- Amazon Linux 2 prior to version 2.0.20190618
- 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 or 18.04

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 the supported AMIs, see the preceding section. For the installation instructions, see [Install EC2 Instance Connect on your EC2 instances](#).

IPv4 address

Your instance must have an IPv4 address (either private or public). EC2 Instance Connect does not support connecting using an IPv6 address.

Network access

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 IP address and be in a public subnet. For more information, see [Enable internet access](#) in the *Amazon VPC User Guide*.
- If your users will connect to your instance through the instance's private IP 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 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. EC2 Instance Connect Endpoint allows you to connect to an instance via SSH or RDP without requiring the instance to have a public IPv4 address. For more information, see [Connect to your Linux instance using the Amazon EC2 console](#).

Security group rule

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](#).

EC2 Instance Connect uses specific IP address ranges for browser-based SSH connections to your instance (when users use the Amazon EC2 console to connect to an instance). If your users will use the Amazon EC2 console to connect to an instance, ensure that the security group associated with your instance allows inbound SSH traffic from the IP address range for EC2_INSTANCE_CONNECT. To identify the address range, download the JSON file provided by AWS and filter for the subset for EC2 Instance Connect, using EC2_INSTANCE_CONNECT as the service value. These IP address ranges differ between AWS Regions. For more information about downloading the JSON file and filtering by service, see [AWS IP address ranges](#) in the *Amazon VPC User Guide*.

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](#).

Local computer setup

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 only use the Amazon EC2 console to connect to an instance.

Username

When using EC2 Instance Connect to connect to an instance, the username must meet the following prerequisites:

- 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.

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.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-instance-connect:SendSSHPublicKey",
    "Resource": [
      "arn:aws:ec2:region:account-id:instance/i-1234567890abcdef0",
      "arn:aws:ec2:region:account-id:instance/i-0598c7d356eba48d7"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:osuser": "ami-username"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DescribeInstances",
    "Resource": "*"
  }
]
```

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 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.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-instance-connect:SendSSHPublicKey",
    "Resource": "arn:aws:ec2:region:account-id:instance/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/tag-key": "tag-value"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DescribeInstances",
    "Resource": "*"
  }
]
```

Install EC2 Instance Connect on your EC2 instances

To connect to an instance using EC2 Instance Connect, the instance must have EC2 Instance Connect installed.

The following AMIs come pre-installed with EC2 Instance Connect:

- AL2023
- 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

If your instance was launched with one of the AMIs in the preceding list, you can skip this procedure.

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 cannot use EC2 Instance Connect.

Prerequisites for installing EC2 Instance Connect

- **Launch the instance with one of the following supported AMIs:**

Amazon Linux 2 prior to version 2.0.20190618

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

If your instance was launched with a later version of Amazon Linux 2, macOS Sonoma, Ventura, or Monterey, or Ubuntu, it comes pre-installed with EC2 Instance Connect and you can skip this procedure.

- **Verify the general prerequisites for EC2 Instance Connect.**

For more information, see [Prerequisites](#).

- **Verify the prerequisites for connecting to your instance using an SSH client on your local machine.**

If your local machine is Linux or macOS, see [Connect to your Linux instance from Linux or macOS using SSH](#). If your local machine is Windows, see [Prerequisites](#).

For more information, see [SSH connection prerequisites](#).

- **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.

- **Install an SSH client on your local computer.**

Your local computer most likely has an SSH client installed by default. You can check for an SSH client by typing `ssh` at the command line. If your local computer doesn't recognize the command, you can install an SSH client. For information about installing an SSH client on Linux or macOS X, see <http://www.openssh.com>. For information about installing an SSH client on Windows 10, see [OpenSSH in Windows](#).

- **(Ubuntu) Install the AWS CLI on your instance.**

To install EC2 Instance Connect on an Ubuntu instance, you must use the AWS CLI on the instance. For more information about installing the AWS CLI, see [Installing the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Install EC2 Instance Connect

Installing EC2 Instance Connect configures the SSH daemon on the instance.

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 from Linux or macOS using SSH](#).

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 from Linux or macOS using SSH](#).

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.rhel8.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.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.rhel8.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.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.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.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.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.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/ssh.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 from Linux or macOS using SSH](#).

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 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 from Linux or macOS using SSH](#).

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.rhel8.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.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.rhel8.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.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.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.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.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.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/ssh.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 from Linux or macOS using SSH](#).

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 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.

For more information about the EC2 Instance Connect package, see [aws/aws-ec2-instance-connect-config](https://github.com/aws/aws-ec2-instance-connect-config) on the GitHub website.

Connect using EC2 Instance Connect

The following instructions explain how to connect to your Linux instance using EC2 Instance Connect.

Decide which connection option to use. The connection option to use depends on whether your instance has a public IPv4 address:

- Amazon EC2 console – To connect using the Amazon EC2 console, the instance must have a public IPv4 address.
- SSH client – If the instance does not have a public IP address, you can connect to the instance over a private network using an SSH client. For example, you can connect from within the same VPC or through a VPN connection, transit gateway, or AWS Direct Connect.

EC2 Instance Connect does not support connecting using an IPv6 address.

Tip

EC2 Instance Connect is one of the options to connect to your Linux instance. For other options, see [Connect to your Linux instance](#). To connect to a Windows instance, see [Connect to your Windows instance](#).

Connection options for EC2 Instance Connect

- [Connect using the Amazon EC2 console](#)
- [Connect using your own key and SSH client](#)

- [Connect using the AWS CLI](#)
- [Troubleshoot](#)

Connect using the Amazon EC2 console

You can connect to an instance using the Amazon EC2 console by selecting the instance from the console and choosing to connect using EC2 Instance Connect. Instance Connect handles the permissions and provides a successful connection.

To connect using the Amazon EC2 console, the instance must have a public IPv4 address. Before connecting, be sure to review all the [prerequisites](#).

To connect to your instance using the browser-based client from the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Connect**.
4. Choose the **EC2 Instance Connect** tab.
5. For **Connection type**, choose **Connect using EC2 Instance Connect**.
6. For **Username**, verify the username.
7. Choose **Connect** to open a terminal window.

Connect using your own key and SSH client

You can use your own SSH key and connect to your instance from the SSH client of your choice while using the EC2 Instance Connect API. This enables you to benefit from the Instance Connect capability to push a public key to the instance. 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 only has 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](#).

Before connecting, be sure to review all the [prerequisites](#).

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
```

Connect using the AWS CLI

If you know your instance ID, you can use the [ec2-instance-connect](#) AWS CLI command to connect to your instance using an SSH client. If you don't specify a connection type, EC2 Instance Connect automatically tries to connect to your instance's public IPv4 address. If your instance doesn't have a public IPv4 address, EC2 Instance Connect then tries to connect to your instance's private IPv4 address through an [EC2 Instance Connect Endpoint](#). If your instance doesn't have a private IPv4 address or your VPC doesn't have an EC2 Instance Connect Endpoint, then EC2 Instance Connect tries to connect to your instance's IPv6 address.

Important

Before you connect with this method, ensure that you have configured the AWS CLI, including the credentials that it uses, and that you're using the *latest* version of the AWS CLI. For more information, see [Installing or updating the latest version of the AWS CLI](#) and [Configuring the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Connection types

auto (default)

The CLI tries to connect using the instance's IP addresses in the following order and with the corresponding connection type:

- Public IPv4: `direct`
- Private IPv4: `eice`
- IPv6: `direct`

direct

The CLI tries to connect using the instance's IP addresses in the following order (it does not connect over an EC2 Instance Connect Endpoint):

- Public IPv4
- IPv6
- Private IPv4

eice

The CLI always uses the instance's private IPv4 address.

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`.

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.

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](#) CLI command and specify the `ssh` parameter and the instance ID.

```
aws ec2-instance-connect ssh --instance-id i-1234567890example
```

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*.

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
```

Troubleshoot

If you receive an error while attempting to connect to your instance, see the following:

- [Troubleshoot connecting to your 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 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 with Amazon Linux 2

1. Connect to your instance using SSH. Specify the SSH key pair you used for your instance when you launched it and the default 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 with an Ubuntu AMI

1. Connect to your instance using SSH. Specify the SSH key pair you used for your instance when you launched it and the default 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 Linux instance with AWS Systems Manager Session Manager

Session Manager is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances through an interactive one-click browser-based shell or through the AWS CLI. You can use Session Manager to start a session with an instance in your account. After the session is started, you can run bash commands as you would through any other connection type. For more information about Session Manager, see [AWS Systems Manager Session Manager](#) in the *AWS Systems Manager User Guide*.

Prerequisites

Before attempting to connect to an instance using Session Manager, ensure that the necessary setup steps have been completed. For more information and instructions, see [Setting up Session Manager](#).

To connect to a Linux instance using Session Manager with 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 **Session Manager**.
5. Choose **Connect**.

Troubleshooting

If you receive an error that you're not authorized to perform one or more Systems Manager actions (ssm: *command-name*), then you must update your policies to allow you to start sessions from the Amazon EC2 console. For more information, see [Sample IAM policies for Session Manager](#) in the *AWS Systems Manager User Guide*.

Connect to your instances using 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](#)
- [Considerations](#)
- [Grant permissions to use EC2 Instance Connect Endpoint](#)
- [Security groups for EC2 Instance Connect Endpoint](#)
- [Create an EC2 Instance Connect Endpoint](#)
- [Connect to a Linux instance using EC2 Instance Connect Endpoint](#)
- [Log connections established over EC2 Instance Connect Endpoint](#)
- [Delete an EC2 Instance Connect Endpoint](#)
- [Service-linked role for EC2 Instance Connect Endpoint](#)
- [Quotas for EC2 Instance Connect Endpoint](#)

How it works

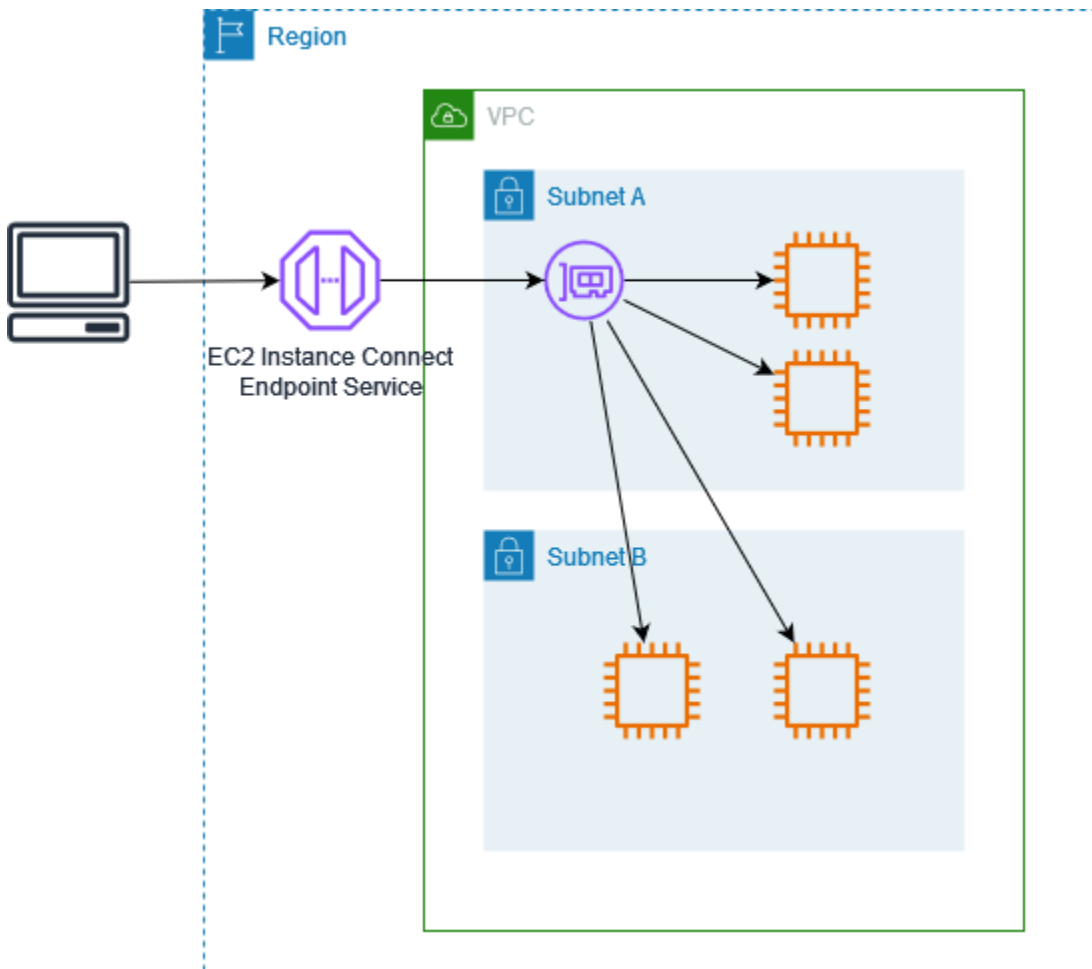
EC2 Instance Connect Endpoint is an identity-aware TCP proxy. The EC2 Instance Connect Endpoint Service establishes a private tunnel from your computer to the endpoint using the credentials for your IAM entity. Traffic is authenticated and authorized before it reaches your VPC.

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](#).
- 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](#).
- EC2 Instance Connect Endpoint is not supported in Canada West (Calgary).

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, 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, and delete EC2 Instance Connect Endpoints

To create an EC2 Instance Connect Endpoint, users require permissions for the following actions:

- `ec2:CreateInstanceConnectEndpoint`
- `ec2:CreateNetworkInterface`
- `ec2:CreateTags`
- `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, 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 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.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "GrantAllActionsInAllSubnets",
    "Action": [
      "ec2:CreateInstanceConnectEndpoint",
      "ec2>DeleteInstanceConnectEndpoint",
      "ec2:CreateNetworkInterface",
      "ec2:CreateTags",
      "iam:CreateServiceLinkedRole"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:region:account-id:subnet/*"
  },
  {
    "Action": [
      "ec2:CreateNetworkInterface"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:::security-group/*"
  },
  {
    "Sid": "DescribeInstanceConnectEndpoints",
    "Action": [
      "ec2:DescribeInstanceConnectEndpoints"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
  ]
}

```

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 a Linux 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*.

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 policy allows an IAM principal to connect to an instance using only the specified EC2 Instance Connect Endpoint, identified by the specified endpoint ID `oice-123456789abcdef`. The connection is successfully established only if all the conditions are satisfied, for example, if the SSH connection is established on port 22 of the instance, 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).

The `ec2:Describe*` API actions do not support resource-level permissions. Therefore, the `*` wildcard is necessary in the `Resource` element.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "EC2InstanceConnect",
    "Action": "ec2-instance-connect:OpenTunnel",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:region:account-id: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": "*"
  }
]
}

```

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*.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-instance-connect:OpenTunnel",
    "Resource": "arn:aws:ec2:region:account-id: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	<i>ID of instance security group</i>	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	<i>VPC IPv4 CIDR</i>	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	<i>ID of endpoint security group</i>	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	<i>VPC IPv4 CIDR</i>	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	<i>Public IPv4 address range</i>	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.

You can't modify an EC2 Instance Connect Endpoint after you've created it. Instead, you must delete the EC2 Instance Connect Endpoint and create a new one with the settings that you need.

Prerequisites

You must have the required IAM permissions to create an EC2 Instance Connect Endpoint. For more information, see [Permissions to create, describe, and delete EC2 Instance Connect Endpoints](#).

Create the endpoint using the console

Use the following procedure to create an EC2 Instance Connect Endpoint.

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 **Service category**, choose **EC2 Instance Connect Endpoint**.
 - c. For **VPC**, select the VPC that has the target instances.
 - d. (Optional) To preserve client IP addresses, expand **Additional settings** and select the 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](#).

Create the endpoint using the AWS CLI

Use the [create-instance-connect-endpoint](#) AWS CLI command.

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 create the endpoint

Use the following command to create an endpoint network interface for your EC2 Instance Connect Endpoint in the specified subnet.

```
aws ec2 create-instance-connect-endpoint --subnet-id subnet-0123456789example
```

The following is example output.

```
{
  "OwnerId": "111111111111",
  "InstanceConnectEndpointId": "oice-0123456789example",
  "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/oice-0123456789example",
  "State": "create-complete",
  "StateMessage": "",
  "DnsName": "oice-0123456789example.0123abcd.ec2-instance-connect-endpoint.us-east-1.amazonaws.com",
  "FipsDnsName": "oice-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](#) AWS CLI 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 oice-0123456789example --query InstanceConnectEndpoints[*].State --output text
```

The following is example output.

```
create-complete
```

Connect to a Linux instance using EC2 Instance Connect Endpoint

You can use EC2 Instance Connect Endpoint to connect to a Linux instance that supports SSH.

For information about how to connect to a **Windows instance**, see [Connect to a Windows instance using EC2 Instance Connect Endpoint](#) in the *Amazon EC2 User Guide for Windows Instances*.

Contents

- [Prerequisites](#)
- [Connect to your Linux instance using the Amazon EC2 console](#)
- [Connect to your Linux instance using SSH](#)
- [Connect to your Linux instance using the AWS CLI](#)
- [Troubleshoot](#)

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](#).
- To use the 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](#).

Connect to your Linux instance using the Amazon EC2 console

You can connect to an instance using the Amazon EC2 console as follows.

To connect to your instance using the browser-based client

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 **Connect**.
4. Choose the **EC2 Instance Connect** tab.
5. For **Connection type**, choose **Connect using EC2 Instance Connect Endpoint**.
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 user name other than `ec2-user`, enter the correct user name.
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-0123456789example \  
    -o ProxyCommand='aws ec2-instance-connect open-tunnel --instance-  
id i-0123456789example'
```

For:

- `-i` – Specify the key pair that was used to launch the instance.
- *ec2-user@i-0123456789example* – 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-0123456789example \  
    --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](#) AWS CLI command to connect to your instance using an SSH client. For more information about using the [ec2-instance-connect](#) command, 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](#) 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 --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*.

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, see [Troubleshoot connecting to your instance](#).

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 and Amazon EBS API calls with 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": "ABCDEFGONGNOM00CB6XYTQEXAMPLE",
    "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGUKZHNAW40SN2AEXAMPLE",
    "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"
}
```

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, 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-endpoints](#) AWS CLI command and specify the ID of the EC2 Instance Connect Endpoint to delete.

```
aws ec2 delete-instance-connect-endpoint --instance-connect-endpoint-id oice-03f5e49b83924bbc7
```

Example output

```
{
  "InstanceConnectEndpoint": {
    "OwnerId": "111111111111",
    "InstanceConnectEndpointId": "oice-0123456789example",
    "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/oice-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"
  }
}
```

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 required so that Amazon EC2 can call other AWS services on your behalf. For more information, see [Using 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 services to assume the role:

- `ec2-instance-connect.amazonaws.com`

The **AWSServiceRoleForEC2InstanceConnect** service-linked role uses the managed policy **Ec2InstanceConnectEndpoint**. To view the permissions for this policy, see [Ec2InstanceConnectEndpoint](#) in the *AWS Managed Policy Reference*.

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 EC2 Instance Connect Endpoint

You don't need to manually create the 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 [Deleting a service-linked role](#) 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.

Description	Quota
Maximum number of EC2 Instance Connect Endpoints per AWS account per AWS Region	5
Maximum number of EC2 Instance Connect Endpoints per VPC	1
Maximum number of EC2 Instance Connect Endpoints per subnet	1
Maximum number of concurrent connections per EC2 Instance Connect Endpoint	20

Connect your EC2 instance to an AWS resource

After you launch an instance, you can connect it to one or more AWS resources.

This section describes how to automatically connect an Amazon EC2 instance to an Amazon RDS database.

Automatically connect an EC2 instance to an RDS database

You can use the automatic connection functionality in the Amazon EC2 console to quickly connect one or more EC2 instances to an RDS database to allow traffic between them.

For more information, see [How the connection is automatically configured](#). For a detailed walkthrough, which includes other ways to connect an EC2 instance and an RDS database, see [Tutorial: Connect an Amazon EC2 instance to an Amazon RDS database](#).

Topics

- [Costs](#)
- [Prerequisites](#)
- [Automatically connect an instance and a database](#)
- [How the connection is automatically configured](#)

Costs

While there is no charge to automatically connect your EC2 instance to an RDS database, you are charged for the underlying services. Data transfer fees will apply if your EC2 instance and RDS database are in different Availability Zones. For information about data transfer fees, see [Data Transfer](#) on the Amazon EC2 On-Demand Pricing page.

Prerequisites

Before you can automatically connect an EC2 instance to an RDS database, check the following:

- The EC2 instances must be in the **Running** state. You can't connect an EC2 instance if it's in another state.
- The EC2 instances and the RDS database must be in the same virtual private cloud (VPC). The automatic connection feature is not supported if an EC2 instance and RDS database are in different VPCs.

Automatically connect an instance and a database

You can automatically connect an EC2 instance to an RDS database immediately after you've launched your instance, or later.

Automatically connect immediately after launch

Use the following steps to automatically connect an EC2 instance to an RDS database immediately after you've launched the EC2 instance.

To view an animation of these steps, see [View an animation: Automatically connect a newly-launched EC2 instance to an RDS database](#).

To automatically connect a newly-launched EC2 instance to an RDS database using the EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch instances**, and then follow the steps to [launch an instance](#).
3. On the instance launch confirmation page, choose **Connect an RDS database**.
4. In the **Connect RDS Database** dialog box, do the following:

- a. For **Database role**, choose either **Cluster** or **Instance**.
- b. For **RDS database**, select a database to connect to.

Note

The EC2 instances and the RDS database must be in the same VPC in order to connect to each other.

- c. Choose **Connect**.

View an animation: Automatically connect a newly-launched EC2 instance to an RDS database

The screenshot shows the AWS Management Console interface for EC2 resources in the Europe (Stockholm) Region. The left sidebar contains navigation options like 'EC2 Dashboard', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main content area is divided into several sections:

- Resources:** A summary table showing the following counts:

Instances (running)	1	Dedicated Hosts	0	Elastic IPs	0
Instances	1	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	9	Snapshots	1
Volumes	2				
- Launch instance:** A section with a 'Launch instance' button and a 'Migrate a server' link. Below it, a note states: 'Note: Your instances will launch in the Europe (Stockholm) Region'.
- Scheduled events:** A section showing 'Europe (Stockholm)' with 'No scheduled events'.
- Migrate a server:** A section with the text: 'Use AWS Application Migration Service to simplify and expedite migration'.
- Service health:** A section showing 'Region: Europe (Stockholm)' and 'Status: This service is operating normally'.
- Zones:** A table listing available zones:

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2
eu-north-1c	eun1-az3

Automatically connect an existing instance

Use the following steps to automatically connect an existing EC2 instance to an RDS database.

To view an animation of these steps, see [View an animation: Automatically connect an existing EC2 instance to an RDS database](#).

To automatically connect an existing EC2 instance to an RDS database using the EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select one or more EC2 instances to connect to an RDS database, and then choose **Actions**, **Networking**, **Connect RDS database**.

If **Connect RDS database** is not available, check that the EC2 instances are in the **Running** state and that they are in the same VPC.

4. In the **Connect RDS Database** dialog box, do the following:
 - a. For **Database role**, choose either **Cluster** or **Instance**.
 - b. For **RDS database**, select a database to connect to.

Note

The EC2 instances and the RDS database must be in the same VPC in order to connect to each other.

- c. Choose **Connect**.

View an animation: Automatically connect an existing EC2 instance to an RDS database

The screenshot displays the Amazon EC2 console interface. On the left is a navigation pane with categories like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area is divided into several panels:

- Resources:** A table showing EC2 resources in the Europe (Stockholm) Region.

Resource	Count	Resource	Count	Resource	Count
Instances (running)	2	Dedicated Hosts	0	Elastic IPs	0
Instances	2	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	10	Snapshots	1
Volumes	3				
- Launch instance:** A section with a "Launch instance" button and a "Migrate a server" button. Below it, a note states: "Note: Your instances will launch in the Europe (Stockholm) Region".
- Scheduled events:** A section showing "Europe (Stockholm)" with "No scheduled events".
- Migrate a server:** A section with the text: "Use AWS Application Migration Service to simplify and expedite migration".
- Service health:** A section showing "Region: Europe (Stockholm)" and "Status: This service is operating normally". Below it is a table of zones:

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2
eu-north-1c	eun1-az3
- Account attributes:** A section on the right showing "Supported platforms" (VPC, Default VPC, Settings, EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments).
- Explore AWS:** A section on the right with promotional cards for "Amazon GuardDuty Malware Protection", "Enable Best Price-Performance with AWS Graviton2", and "Get Up to 40% Better Price Performance".

For information about how to use the Amazon RDS console to automatically connect an EC2 instance to an RDS database, see [Configure automatic network connectivity with an EC2 instance](#) in the *Amazon RDS User Guide*.

How the connection is automatically configured

When you use the EC2 console to automatically configure the connection between an EC2 instance and an RDS database to allow traffic between them, the connection is configured by [security groups](#).

The security groups are automatically created and added to the EC2 instance and RDS database, as follows:

- Amazon EC2 creates a security group called **ec2-rds-x** and adds it to the EC2 instance. It has one outbound rule that allows traffic to the database by specifying **rds-ec2-x** (the database security group) as its destination.
- Amazon RDS creates a security group called **rds-ec2-x** and adds it to the database. It has one inbound rule that allows traffic from the EC2 instance by specifying **ec2-rds-x** (the EC2 instance security group) as its source.

The security groups reference each other as the destination and source, and only allow traffic on the database port. You can reuse these security groups so that any database with the **rds-ec2-x** security group can talk to any EC2 instance with the **ec2-rds-x** security group.

The security group names follow a pattern. For the security groups created by Amazon EC2, the pattern is **ec2-rds-x**, and for the security groups created by Amazon RDS, the pattern is **rds-ec2-x**. **x** is a number, which increases by 1 each time a new security group is automatically created.

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 your EC2 instance to your 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 your EC2 instance to your 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 your EC2 instance to your RDS database by mimicking the automatic connection feature](#)

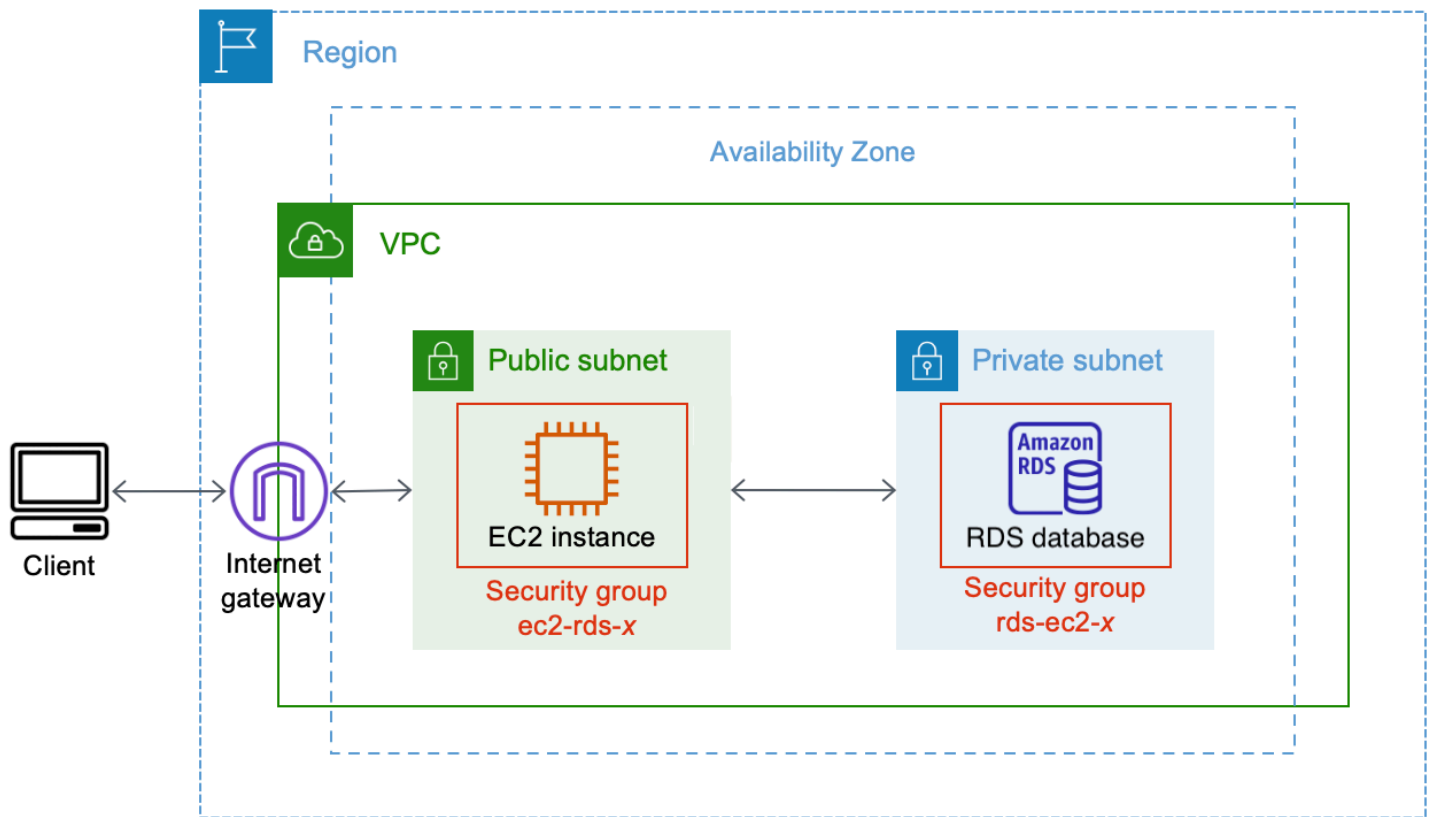
Configure the connection between your EC2 instance to your RDS database by manually configuring and assigning the security groups to reproduce the configuration that is automatically created by the automatic connection feature in Option 1 and Option 2.

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 is less than 12 months old and you configure your resources according to the free tier requirements.

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 [Clean up](#).

Option 1: Automatically connect your EC2 instance to your RDS database using the EC2 console

Objective

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.

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`

Tasks to complete Option 1

- [Task 1: Create an RDS database – optional](#)
- [Task 2: Launch an EC2 instance – optional](#)
- [Task 3: Automatically connect your EC2 instance to your RDS database](#)
- [Task 4: Verify the connection configuration](#)

Task 1: Create an RDS database – *optional*

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.

Task objective

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. If you have an RDS database that you can use, you can skip this task.

Important

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.

Steps to create an RDS database

Use the following steps to create an RDS database.

To view an animation of these steps, see [View an animation: Create an RDS database](#).

RDS database configuration

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 a database at no cost. However, note that the free tier is only available if your account is less than 12 months old. Other restrictions apply. 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 is less than 12 months, you can use this database class for free. Other restrictions apply. 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

The screenshot shows the Amazon RDS console interface. On the left is a navigation sidebar with the following items: **Amazon RDS** (with a close icon), **Dashboard** (highlighted in orange), Databases, Performance insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, and Certificate update.

The main content area features a light blue banner at the top with an information icon and the text: "Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL. For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional commit latencies instances by deploying the Multi-AZ DB cluster. [Learn more](#)". Below this is an orange "Create database" button with a mouse cursor hovering over it, and the text "Or, Restore Multi-AZ DB Cluster from Snapshot".

Below the banner is a "Resources" section titled "Resources" with the text: "You are using the following Amazon RDS resources in the EU (Stockholm) region (used/quota)". It lists various resource categories and their usage:

- DB Instances (3/40)**: Allocated storage (0.3 TB/100 TB), [Increase DB instances limit](#) (with an external link icon).
- DB Clusters (1/40)**
- Reserved instances (0/40)**
- Snapshots (1)**:
 - Manual**: DB Cluster (0/100), DB Instance (0/100)
 - Automated**: DB Cluster (1), DB Instance (0)
- Recent events (5)**
- Event subscriptions (0/20)**
- Parameter groups (2)**: Default (2), Custom (0/100)
- Option groups (1)**: Default (1), Custom (0/20)
- Subnet groups (1/50)**
- Supported platforms VPC**
- Default network vpc-78678c**

At the bottom of the main content area is a "Create database" section with the text: "Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database I".

You're now ready for [Task 2: Launch an EC2 instance – optional](#).

Task 2: Launch an EC2 instance – *optional*

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.

Task objective

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. If you have an EC2 instance that you can use, you can skip this task.

Important

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.

Steps to launch an EC2 instance

Use the following steps to launch an EC2 instance for this tutorial.

To view an animation of these steps, see [View an animation: Launch an EC2 instance](#).

EC2 instance configuration

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`.

ℹ Note

You can use Amazon EC2 under the [Free tier](#) provided your AWS account is less than 12 months old and you choose a `t2.micro` instance type (or `t3.micro` in Regions where `t2.micro` is not available).

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** check box, choose **My IP**.

- c. To allow traffic from the internet to your instance, select the following check boxes:
- **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 instance launch issues](#).

For more information about launching an instance, see [Launch an instance using the new launch instance wizard](#).

View an animation: Launch an EC2 instance

Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	2	Dedicated Hosts	0	Elastic IPs	0
Instances	2	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	10	Snapshots	1
Volumes	3				

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#) [Migrate a server](#)

Note: Your instances will launch in the Europe (Stockholm) Region

Scheduled events

Europe (Stockholm)
No scheduled events

Service health

Region: Europe (Stockholm)
Status: ✔ This service is operating normally

Zones

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2
eu-north-1c	eun1-az3

You're now ready for [Task 3: Automatically connect your EC2 instance to your RDS database](#).

Task 3: Automatically connect your EC2 instance to your RDS database

Task objective

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.

Steps to connect your EC2 instance and RDS database

Use the following steps to connect your EC2 instance and RDS database using the automatic feature in the EC2 console.

To view an animation of these steps, see [View an animation: Automatically connect a newly-launched EC2 instance to an 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.

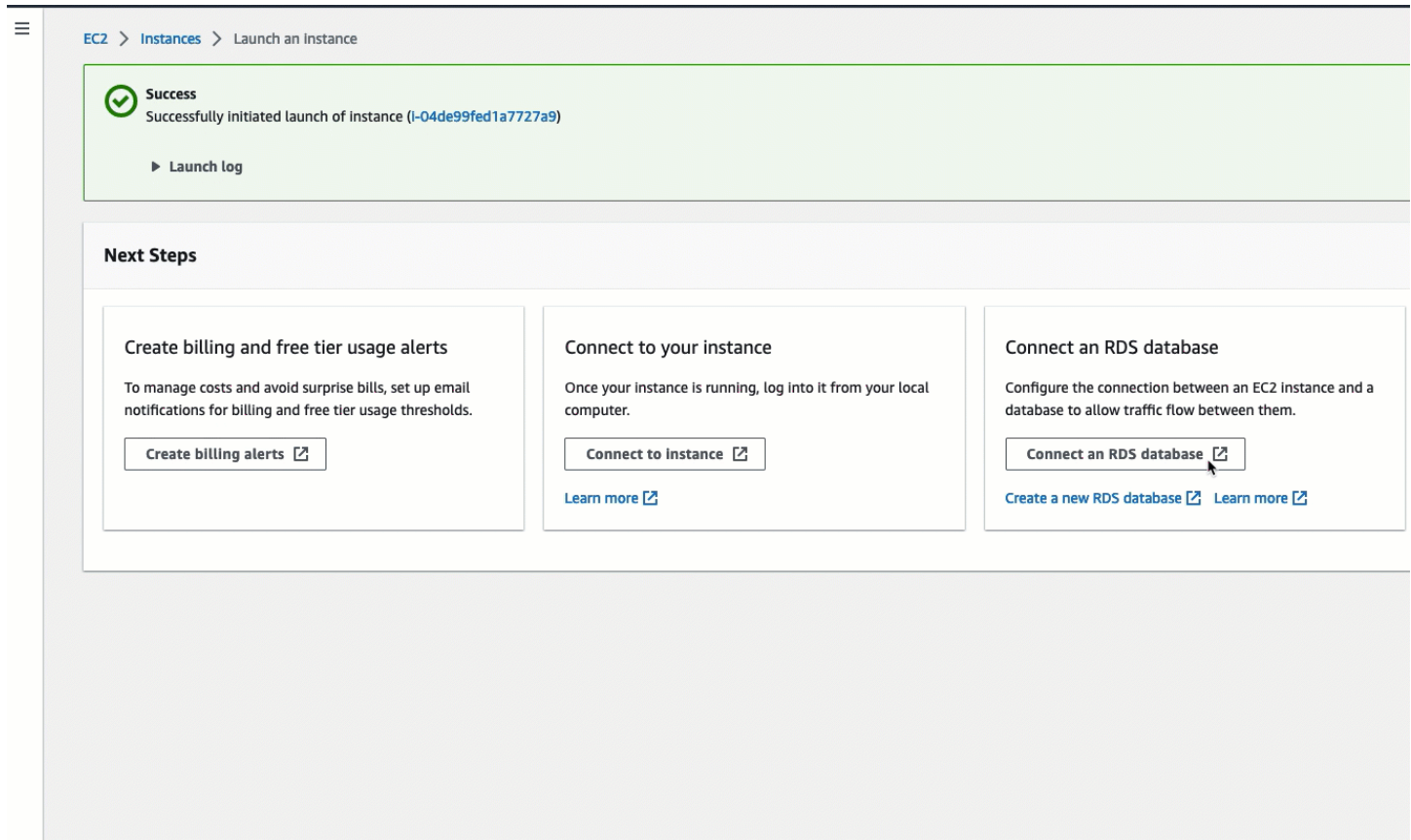
2. For **Database role**, choose **Instance**. *Instance* in this case refers to the database instance.
3. For **RDS database**, choose the RDS database that you created in Task 1.

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



EC2 > Instances > Launch an Instance

Success
Successfully initiated launch of instance (i-04de99fed1a7727a9)

▶ Launch log

Next Steps

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Create billing alerts [↗](#)

Connect to your instance

Once your instance is running, log into it from your local computer.

Connect to instance [↗](#)

[Learn more](#) [↗](#)

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Connect an RDS database [↗](#)

[Create a new RDS database](#) [↗](#) [Learn more](#) [↗](#)

You're now ready for [Task 4: Verify the connection configuration](#).

Task 4: Verify the connection configuration

Task objective

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 EC2 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.

Steps to verify the connection configuration

Use the following steps to verify the connection configuration.

To view an animation of these steps, see [View an animation: Verify the connection configuration](#).

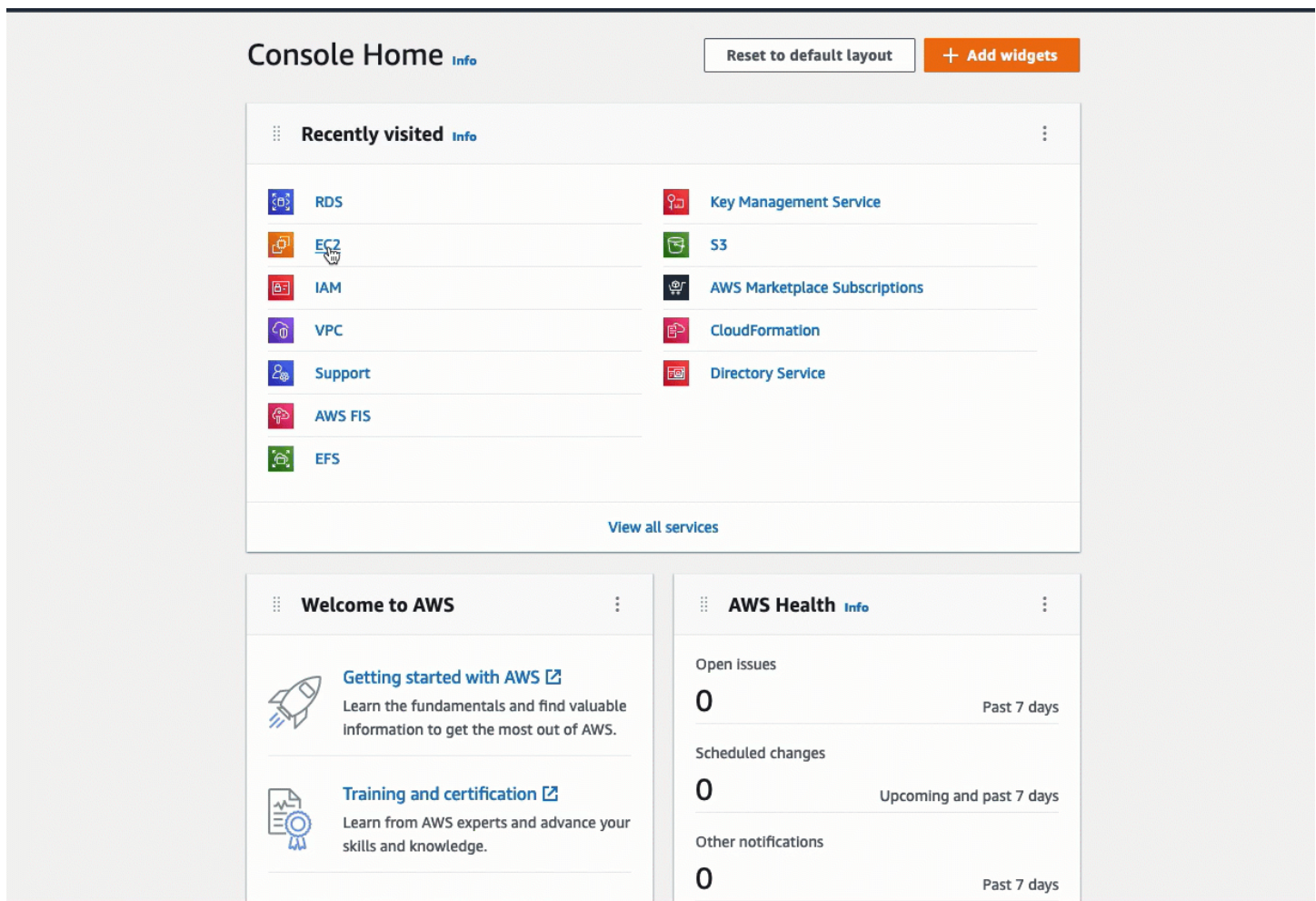
To verify the connection configuration using the console

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



The screenshot displays the AWS Management Console Home page. At the top, it shows "Console Home" with an "Info" link, a "Reset to default layout" button, and a "+ Add widgets" button. Below this is a "Recently visited" section with a list of service tiles: RDS, EC2, IAM, VPC, Support, AWS FIS, EFS, Key Management Service, S3, AWS Marketplace Subscriptions, CloudFormation, and Directory Service. A "View all services" link is at the bottom of this list. Below the "Recently visited" section are two main panels: "Welcome to AWS" and "AWS Health". The "Welcome to AWS" panel contains two links: "Getting started with AWS" and "Training and certification". The "AWS Health" panel shows three metrics: "Open Issues" (0), "Scheduled changes" (0), and "Other notifications" (0), each with a time range of "Past 7 days".

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.

Option 2: Automatically connect your EC2 instance to your RDS database using the RDS console

Objective

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.

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`

Tasks to complete Option 2

- [Task 1: Launch an EC2 instance – optional](#)
- [Task 2: Create an RDS database and automatically connect it to your EC2 instance](#)
- [Task 3: Verify the connection configuration](#)

Task 1: Launch an EC2 instance – optional

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.

Task objective

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. If you have an EC2 instance that you can use, you can skip this task.

Steps to launch an EC2 instance

Use the following steps to launch an EC2 instance for this tutorial.

To view an animation of these steps, see [View an animation: Launch an EC2 instance](#).

EC2 instance configuration

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`.

ℹ Note

You can use Amazon EC2 under the [Free tier](#) provided your AWS account is less than 12 months old and you choose a `t2.micro` instance type (or `t3.micro` in Regions where `t2.micro` is not available).

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** check box, choose **My IP**.

- c. To allow traffic from the internet to your instance, select the following check boxes:
 - **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 instance launch issues](#).

For more information about launching an instance, see [Launch an instance using the new launch instance wizard](#).

View an animation: Launch an EC2 instance

Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	2	Dedicated Hosts	0	Elastic IPs	0
Instances	2	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	10	Snapshots	1
Volumes	3				

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#) [Migrate a server](#)

Note: Your instances will launch in the Europe (Stockholm) Region

Scheduled events

Europe (Stockholm)
No scheduled events

Service health

Region: Europe (Stockholm)
Status: ✔ This service is operating normally

Zones

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2
eu-north-1c	eun1-az3

You're now ready for [Task 2: Create an RDS database and automatically connect it to your EC2 instance](#).

Task 2: Create an RDS database and automatically connect it to your EC2 instance

Task objective

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.

Steps to create an RDS database

Use the following steps to create an RDS database and connect it to your EC2 instance using the automatic feature in the RDS console.

To view an animation of these steps, see [View an animation: Create an RDS database and automatically connect it to an EC2 instance](#).

DB instance configuration

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 is less than 12 months old. Other restrictions apply. 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 is less than 12 months, you can use this instance for free. Other restrictions apply. 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 configured 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

The screenshot shows the Amazon RDS console interface. On the left is a navigation sidebar with the following items: **Amazon RDS** (with a close icon), **Dashboard** (highlighted in orange), Databases, Performance Insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, and Certificate update.

The main content area features a light blue informational banner at the top with an information icon and the text: "Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL. For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional instances by deploying the Multi-AZ DB cluster. [Learn more](#)". Below this is a prominent orange button labeled "Create database" with a mouse cursor hovering over it. Underneath the button, it says "Or, Restore Multi-AZ DB Cluster from Snapshot".

Below the banner is a "Resources" section with the heading "Resources". The text reads: "You are using the following Amazon RDS resources in the EU (Stockholm) region (used/qu...". It lists several resource categories with their respective counts and links to configuration options:

- DB Instances (5/40)**: Includes "Allocated storage (0.34 TB/100 TB)" and a link to "Increase DB instances limit".
- DB Clusters (1/40)**: Includes a link to "Option groups".
- Reserved instances (0/40)**: Includes a link to "Default".
- Snapshots (2)**: Includes a link to "Custom".
- Manual**: Includes "DB Cluster (0/100)" and "DB Instance (0/100)", both with links to "Subnet groups" and "Supported".
- Automated**: Includes "DB Cluster (1)" and "DB Instance (1)", both with links to "Default ne".
- Recent events (10)**
- Event subscriptions (0/20)**

At the bottom of the main content area is a section titled "Create database" with the introductory text: "Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a rel".

You're now ready for [Task 3: Verify the connection configuration](#).

Task 3: Verify the connection configuration

Task objective

The objective of this task is to verify that the two security groups were created and assigned to the instance and the database.

When you use the automatic connection feature in the RDS 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.

Steps to verify the connection configuration

Use the following steps to verify the connection configuration.

To view an animation of these steps, see [View an animation: Verify the connection configuration](#).

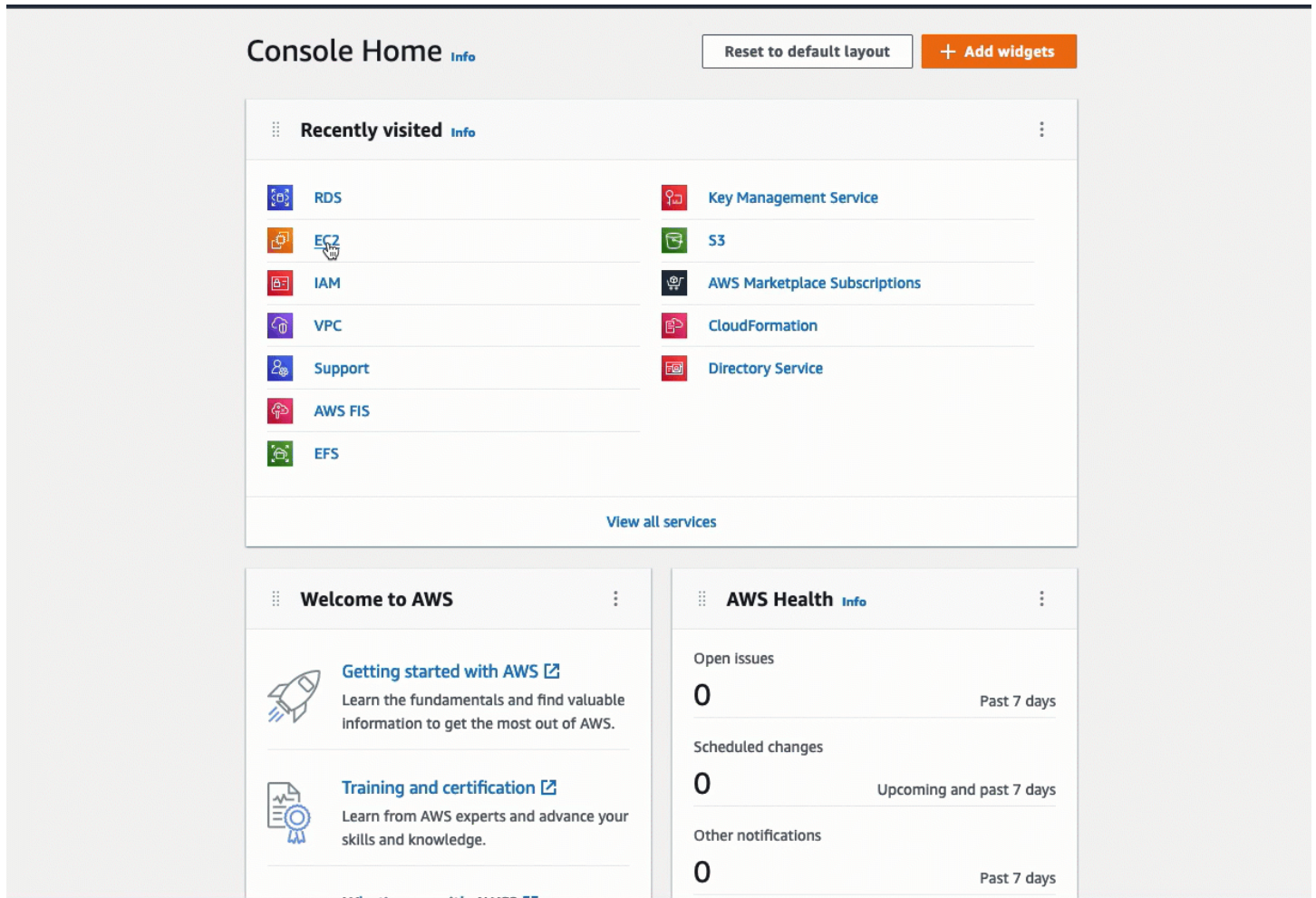
To verify the connection configuration using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose the EC2 instance that you selected to connect to the RDS database in the previous task, and choose the **Security** tab.
4. Under **Security details**, **Security groups**, verify that a security group called **ec2-rds-x** is in the list. **x** is a number.
5. Choose the **ec2-rds-x** security group to open it.
6. Choose the **Outbound rules** tab.
7. 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**
8. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
9. In the navigation page, choose **Databases**.

10. Choose the RDS database that you created for this tutorial.
11. On the **Connectivity & security** tab, under **Security, VPC security groups**, verify that a security group called **rds-ec2-x** is displayed.
12. Choose the **rds-ec2-x** security group. The **Security Groups** screen in EC2 console opens.
13. Choose the **rds-ec2-x** security group open it.
14. Choose the **Inbound rules** tab.
15. 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**

By verifying that these security groups and security group rules exist and that they are assigned to the EC2 instance and RDS database 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.

Option 3: Manually connect your EC2 instance to your RDS database by mimicking the automatic connection feature

Objective

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.

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. If you have completed Option 1 of this tutorial, you already have these permissions.
 - `ec2:AssociateRouteTable`
 - `ec2:AuthorizeSecurityGroupEgress`
 - `ec2:CreateRouteTable`
 - `ec2:CreateSecurityGroup`
 - `ec2:CreateSubnet`
 - `ec2:DescribeInstances`
 - `ec2:DescribeNetworkInterfaces`
 - `ec2:DescribeRouteTables`
 - `ec2:DescribeSecurityGroups`
 - `ec2:DescribeSubnets`
 - `ec2:ModifyNetworkInterfaceAttribute`
 - `ec2:RevokeSecurityGroupEgress`

Tasks to complete Option 3

- [Task 1: Launch an EC2 instance – optional](#)
- [Task 2: Create an RDS database – optional](#)
- [Task 3: Manually connect your EC2 instance to your RDS database by creating security groups and assigning them to the instances](#)

Task 1: Launch an EC2 instance – optional

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.

Task objective

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.

Steps to launch an EC2 instance

Use the following steps to launch an EC2 instance for this tutorial.

To view an animation of these steps, see [View an animation: Launch an EC2 instance](#).

EC2 instance configuration

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**

You can use Amazon EC2 under the [Free tier](#) provided your AWS account is less than 12 months old and you choose a `t2.micro` instance type (or `t3.micro` in Regions where `t2.micro` is not available).

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** check box, choose **My IP**.

- c. To allow traffic from the internet to your instance, select the following check boxes:

- **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 instance launch issues](#).

For more information about launching an instance, see [Launch an instance using the new launch instance wizard](#).

View an animation: Launch an EC2 instance

Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	2	Dedicated Hosts	0	Elastic IPs	0
Instances	2	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	10	Snapshots	1
Volumes	3				

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#) [Migrate a server](#)

Note: Your instances will launch in the Europe (Stockholm) Region

Scheduled events

Europe (Stockholm)
No scheduled events

Service health

Region: Europe (Stockholm)
Status: ✔ This service is operating normally

Zones

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2
eu-north-1c	eun1-az3

You are now ready for [Task 2: Create an RDS database – optional](#).

Task 2: Create an RDS database – *optional*

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.

Task objective

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.

Steps to create an RDS database

Use the following steps to create an RDS database for Option 3 of this tutorial.

To view an animation of these steps, see [View an animation: Create a DB instance](#).

RDS database configuration

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.

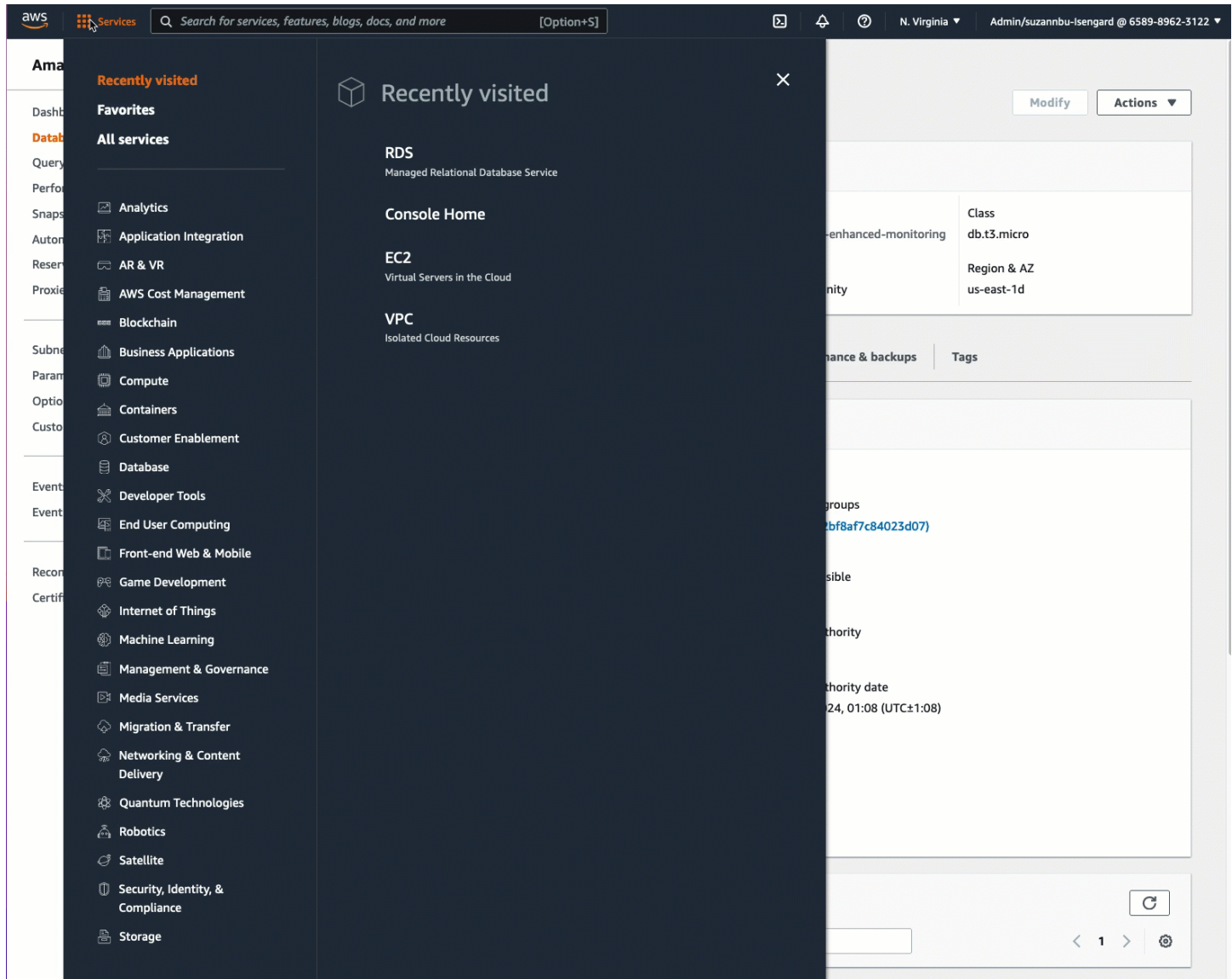
To create a MySQL DB instance

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



You are now ready for [Task 3: Manually connect your EC2 instance to your RDS database by creating security groups and assigning them to the instances.](#)

Task 3: Manually connect your EC2 instance to your RDS database by creating security groups and assigning them to the instances

Task objective

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.

Steps to create new security groups and add them to the instances

Use the following steps to connect an EC2 instance to your RDS database by creating two new security groups. You 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 **Source**, 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 [Clean up](#).

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.

Topics

- [Terminate your EC2 instance](#)
- [Delete your RDS database](#)

Terminate your EC2 instance

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.

Delete your RDS database

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**.

Configure your Amazon Linux instance

After you have successfully launched and logged into your Amazon Linux instance, you can make changes to it. There are many different ways you can configure an instance to meet the needs of a specific application. The following are some common tasks to help get you started.

Contents

- [Common configuration scenarios](#)
- [Manage software on your Amazon Linux instance](#)
- [Manage users on your Linux instance](#)
- [Processor state control for your EC2 instance](#)
- [I/O scheduler](#)
- [Set the time for your Linux instance](#)
- [Configure your GPU instances](#)
- [Optimize CPU options](#)
- [CPU features](#)
- [Change the hostname of your Amazon Linux instance](#)
- [Set up dynamic DNS on your Amazon Linux instance](#)
- [Run commands on your Linux instance at launch](#)
- [Instance metadata and user data](#)

Common configuration scenarios

The base distribution of Amazon Linux contains the software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code. For more information on installing and building software from these locations, see [Manage software on your Amazon Linux instance](#).

Amazon Linux instances come pre-configured with an `ec2-user`, but you may want to add other users that do not have super-user privileges. For more information on adding and removing users, see [Manage users on your Linux instance](#).

The default time configuration for Amazon Linux instances uses Amazon Time Sync Service to set the system time on an instance. The default time zone is UTC. For more information on setting the time zone for an instance or using your own time server, see [Set the time for your Linux instance](#).

If you have your own network with a domain name registered to it, you can change the hostname of an instance to identify itself as part of that domain. You can also change the system prompt to show a more meaningful name without changing the hostname settings. For more information, see [Change the hostname of your Amazon Linux instance](#). You can configure an instance to use a dynamic DNS service provider. For more information, see [Set up dynamic DNS on your Amazon Linux instance](#).

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: cloud-init directives and shell scripts. For more information, see [Run commands on your Linux instance at launch](#).

Manage software on your Amazon Linux instance

The base distribution of Amazon Linux contains the software packages and utilities that are required for basic server operations.

This information applies to Amazon Linux 2 and Amazon Linux. For information about AL2023, see [Managing packages and operating system updates](#) in the *AL2023 User Guide*.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

Contents

- [Update instance software on your Amazon Linux instance](#)
- [Add repositories on an Amazon Linux instance](#)
- [Find and install software packages on an Amazon Linux instance](#)
- [Prepare to compile software on an Amazon Linux instance](#)

It is important to keep software up to date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. For more information, see [Update instance software on your Amazon Linux instance](#).

By default, Amazon Linux instances launch with the following repositories enabled:

- Amazon Linux 2: `amzn2-core` and `amzn2extra-docker`
- Amazon Linux AMI: `amzn-main` and `amzn-updates`

While there are many packages available in these repositories that are updated by Amazon Web Services, there might be a package that you want to install that is contained in another repository. For more information, see [Add repositories on an Amazon Linux instance](#). For help finding and installing packages in enabled repositories, see [Find and install software packages on an Amazon Linux instance](#).

Not all software is available in software packages stored in repositories; some software must be compiled on an instance from its source code. For more information, see [Prepare to compile software on an Amazon Linux instance](#).

Amazon Linux instances manage their software using the `yum` package manager. The `yum` package manager can install, remove, and update software, as well as manage all of the dependencies for each package. Debian-based Linux distributions, like Ubuntu, use the `apt-get` command and `dpkg` package manager, so the `yum` examples in the following sections do not work for those distributions.

Update instance software on your Amazon Linux instance

It is important to keep software up to date. Packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. When you first launch and connect to an Amazon Linux instance, you might see a message asking you to update software packages for security purposes. This section shows how to update an entire system, or just a single package.

This information applies to Amazon Linux 2 and Amazon Linux. For information about AL2023, see [Managing packages and operating system updates](#) in the *AL2023 User Guide*.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about

the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

For information about changes and updates to Amazon Linux 2, see [Amazon Linux 2 release notes](#).

For information about changes and updates to AL2023, see [AL2023 release notes](#).

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

Important

If you launched an EC2 instance that uses an Amazon Linux 2 AMI into an IPv6-only subnet, you must connect to the instance and run `sudo amazon-linux-https disable`. This lets your AL2 instance connect to the yum repository in S3 over IPv6 using the http patch service.

To update all packages on an Amazon Linux instance

1. (Optional) Start a **screen** session in your shell window. Sometimes you might experience a network interruption that can disconnect the SSH connection to your instance. If this happens during a long software update, it can leave the instance in a recoverable, although confused state. A **screen** session allows you to continue running the update even if your connection is interrupted, and you can reconnect to the session later without problems.
 - a. Execute the **screen** command to begin the session.

```
[ec2-user ~]$ screen
```

- b. If your session is disconnected, log back into your instance and list the available screens.

```
[ec2-user ~]$ screen -ls  
There is a screen on:
```

```
17793.pts-0.ip-12-34-56-78 (Detached)
1 Socket in /var/run/screen/S-ec2-user.
```

- c. Reconnect to the screen using the **screen -r** command and the process ID from the previous command.

```
[ec2-user ~]$ screen -r 17793
```

- d. When you are finished using **screen**, use the **exit** command to close the session.

```
[ec2-user ~]$ exit
[screen is terminating]
```

2. Run the **yum update** command. Optionally, you can add the **--security** flag to apply only security updates.

```
[ec2-user ~]$ sudo yum update
```

3. Review the packages listed, enter **y**, and press Enter to accept the updates. Updating all of the packages on a system can take several minutes. The **yum** output shows the status of the update while it is running.
4. (Optional) [Reboot your instance](#) to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any **glibc** libraries should also be followed by a reboot. For updates to packages that control services, it might be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

To update a single package on an Amazon Linux instance

Use this procedure to update a single package (and its dependencies) and not the entire system.

1. Run the **yum update** command with the name of the package to update.

```
[ec2-user ~]$ sudo yum update openssl
```

2. Review the package information listed, enter **y**, and press Enter to accept the update or updates. Sometimes there will be more than one package listed if there are package dependencies that must be resolved. The **yum** output shows the status of the update while it is running.

3. (Optional) [Reboot your instance](#) to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any `glibc` libraries should also be followed by a reboot. For updates to packages that control services, it might be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

Add repositories on an Amazon Linux instance

This information applies to Amazon Linux 2 and Amazon Linux. For information about AL2023, see [Using Deterministic upgrades through versioned repository](#) in the *AL2023 User Guide*.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

By default, Amazon Linux instances launch with the following repositories enabled:

- Amazon Linux 2: `amzn2-core` and `amzn2extra-docker`
- Amazon Linux AMI: `amzn-main` and `amzn-updates`

While there are many packages available in these repositories that are updated by Amazon Web Services, there might be a package that you want to install that is contained in another repository.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To install a package from a different repository with `yum`, you need to add the repository information to the `/etc/yum.conf` file or to its own `repository.repo` file in the `/etc/`

`yum.repos.d` directory. You can do this manually, but most yum repositories provide their own *repository*.repo file at their repository URL.

To determine what yum repositories are already installed

- List the installed yum repositories with the following command:

```
[ec2-user ~]$ yum repolist all
```

The resulting output lists the installed repositories and reports the status of each. Enabled repositories display the number of packages they contain.

To add a yum repository to `/etc/yum.repos.d`

- Find the location of the .repo file. This will vary depending on the repository you are adding. In this example, the .repo file is at `https://www.example.com/repository.repo`.
- Add the repository with the **yum-config-manager** command.

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://
www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo | 4.0 kB 00:00
repo saved to /etc/yum.repos.d/repository.repo
```

After you install a repository, you must enable it as described in the next procedure.

To enable a yum repository in `/etc/yum.repos.d`

- Use the **yum-config-manager** command with the `--enable repository` flag. The following command enables the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project. By default, this repository is present in `/etc/yum.repos.d` on Amazon Linux AMI instances, but it is not enabled.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

Note

To enable the EPEL repository on Amazon Linux 2, use the following command:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

For information on enabling the EPEL repository on other distributions, such as Red Hat and CentOS, see the EPEL documentation at <https://fedoraproject.org/wiki/EPEL>.

Find and install software packages on an Amazon Linux instance

You can use a package management tool to find and install software packages. In Amazon Linux 2 and Amazon Linux, the default software package management tool is YUM. In AL2023, the default software package management tool is DNF. For more information about AL2023, see [Package management tool](#) in the *AL2023 User Guide*.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

Find software packages on an Amazon Linux instance

You can use the **yum search** command to search the descriptions of packages that are available in your configured repositories. This is especially helpful if you don't know the exact name of the package you want to install. Simply append the keyword search to the command; for multiple word searches, wrap the search query with quotation marks.

⚠ Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

```
[ec2-user ~]$ yum search "find"
```

The following is example output for Amazon Linux 2.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
===== N/S matched: find =====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
ocaml-findlib-devel.x86_64 : Development files for ocaml-findlib
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
  File::Find
robotfindskitten.x86_64 : A game/zen simulation. You are robot. Your job is to find
  kitten.
mlocate.x86_64 : An utility for finding files by name
ocaml-findlib.x86_64 : Objective CAML package manager and build helper
perl-Devel-Cycle.noarch : Find memory cycles in objects
perl-Devel-EnforceEncapsulation.noarch : Find access violations to blessed objects
perl-File-Find-Rule-Perl.noarch : Common rules for searching for Perl things
perl-File-HomeDir.noarch : Find your home and other directories on any platform
perl-IPC-Cmd.noarch : Finding and running system commands made easy
perl-Perl-MinimumVersion.noarch : Find a minimum required version of perl for Perl code
texlive-xesearch.noarch : A string finder for XeTeX
valgrind.x86_64 : Tool for finding memory management bugs in programs
valgrind.i686 : Tool for finding memory management bugs in programs
```

The following is example output for Amazon Linux.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
===== N/S Matched: find =====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
  File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png, jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png, jpg)
```

```
mlocate.x86_64 : An utility for finding files by name
```

Multiple word search queries in quotation marks only return results that match the exact query. If you don't see the expected package, simplify your search to one keyword and then scan the results. You can also try keyword synonyms to broaden your search.

For more information about packages for Amazon Linux 2 and Amazon Linux, see the following:

- [Extras library \(Amazon Linux 2\)](#)
- [Package repository](#)

Install software packages on an Amazon Linux instance

In Amazon Linux 2 and Amazon Linux, the yum package management tool searches all of your enabled repositories for different software packages and handles any dependencies in the software installation process. For information about installing software packages in AL2023, see [Managing packages and operating system updates](#) in the *AL2023 User Guide*.

To install a package from a repository

Use the **yum install *package*** command, replacing *package* with the name of the software to install. For example, to install the **links** text-based web browser, enter the following command.

```
[ec2-user ~]$ sudo yum install links
```

To install RPM package files that you have downloaded

You can also use **yum install** to install RPM package files that you have downloaded from the internet. To do this, append the path name of an RPM file to the installation command instead of a repository package name.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

To list installed packages

To view a list of installed packages on your instance, use the following command.

```
[ec2-user ~]$ yum list installed
```

Prepare to compile software on an Amazon Linux instance

Open-source software is available on the internet that has not been pre-compiled and made available for download from a package repository. You might eventually discover a software package that you need to compile yourself, from its source code. For your system to be able to compile software in Amazon Linux 2 and Amazon Linux, you need to install several development tools, such as **make**, **gcc**, and **autoconf**.

Note

The Amazon Linux AMI reached its end-of-life on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information about the Amazon Linux AMI end-of-life and maintenance support, see the blog post [Update on Amazon Linux AMI end-of-life](#). We recommend that you upgrade applications to AL2023, which includes long-term support until 2028.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

Because software compilation is not a task that every Amazon EC2 instance requires, these tools are not installed by default, but they are available in a package group called "Development Tools" that is easily added to an instance with the **yum groupinstall** command.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

Software source code packages are often available for download (from websites such as <https://github.com/> and <http://sourceforge.net/>) as a compressed archive file, called a tarball. These tarballs will usually have the `.tar.gz` file extension. You can decompress these archives with the **tar** command.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

After you have decompressed and unarchived the source code package, you should look for a README or INSTALL file in the source code directory that can provide you with further instructions for compiling and installing the source code.

To retrieve source code for Amazon Linux packages

Amazon Web Services provides the source code for maintained packages. You can download the source code for any installed packages with the **yumdownloader --source** command.

- Run the **yumdownloader --source *package*** command to download the source code for *package*. For example, to download the source code for the htop package, enter the following command.

```
[ec2-user ~]$ yumdownloader --source htop

Loaded plugins: priorities, update-motd, upgrade-helper
Enabling amzn-updates-source repository
Enabling amzn-main-source repository
amzn-main-source
| 1.9 kB 00:00:00
amzn-updates-source
| 1.9 kB 00:00:00
(1/2): amzn-updates-source/latest/primary_db
| 52 kB 00:00:00
(2/2): amzn-main-source/latest/primary_db
| 734 kB 00:00:00
htop-1.0.1-2.3.amzn1.src.rpm
```

The location of the source RPM is in the directory from which you ran the command.

Manage users on your 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 user name](#) 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](#).

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 user names](#)
- [Considerations](#)
- [Create a user](#)
- [Remove a user](#)

Default user names

The default user name for your EC2 instance is determined by the AMI that was specified when you launched the instance.

The default user names are:

- For AL2023, Amazon Linux 2, or the Amazon Linux AMI, the user name is `ec2-user`.
- For a CentOS AMI, the user name is `centos` or `ec2-user`.
- For a Debian AMI, the user name is `admin`.
- For a Fedora AMI, the user name is `fedora` or `ec2-user`.
- For a RHEL AMI, the user name is `ec2-user` or `root`.
- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.

- For an Oracle AMI, the user name is `ec2-user`.
- For a Bitnami AMI, the user name is `bitnami`.

Note

To find the default user name 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 with 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.

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  
AAAAB3NzaC1yc2EAAAADAQABAAQAClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ItxCih  
+PnDSUaw+WNQn/mZphTk/a/gU8jEzo0WbkM4yxyb/wB96xbiFveSFJu0p/
```



```
d6RJhJ0I0iBXr1sLnBItnckij7FbtXJMXLvVwJryDUi1BMTjYtwB+QhYXUM0zce5Pjz5/  
i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco  
+CY/5WriUBkriHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi  
+z7wB3RbBQoQzd8v7yeb70z1PnW0yN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

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*.

- Amazon Linux and Amazon Linux 2

With Amazon Linux 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](#).

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
```

Processor state control for your EC2 instance

C-states control the sleep levels that a core can enter when it is idle. C-states are numbered starting with C0 (the shallowest state where the core is totally awake and executing instructions) and go to C6 (the deepest idle state where a core is powered off).

P-states control the desired performance (in CPU frequency) from a core. P-states are numbered starting from P0 (the highest performance setting where the core is allowed to use Intel Turbo Boost Technology to increase frequency if possible), and they go from P1 (the P-state that requests the maximum baseline frequency) to P15 (the lowest possible frequency).

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 | m5.metal | m5d.metal | m5n.metal | m5zn.metal | m6i.metal | m6id.metal | m7i.metal-24x1 | m7i.metal-48x1
- Compute optimized: c4.8xlarge | c5.metal | c5an.metal | c5adn.metal | c5n.metal | c6i.metal | c6id.metal | c7i.metal-24x1 | c7i.metal-48x1
- Memory optimized: r4.8xlarge | r4.16xlarge | r5.metal | r5b.metal | r5d.metal | r6i.metal | r7i.metal-24x1 | r7i.metal-48x1 | r7iz.metal-16x1 | r7iz.metal-32x1 | u-6tb1.metal | u-9tb1.metal | u-12tb1.metal | u-18tb1.metal | u-24tb1.metal | x1.16xlarge | x1.32xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge | z1d.metal
- Storage optimized: d2.8xlarge | d3.metal | d3en.metal | i3.8xlarge | i3.16xlarge | i3.metal | i3en.metal | h1.8xlarge | h1.16xlarge
- Accelerated computing: f1.16xlarge | g3.16xlarge | g4dn.metal | p2.16xlarge | p3.16xlarge

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 | m6a.24xlarge | m6a.48xlarge | m6ad.metal | m6i.16xlarge | m6i.32xlarge | m7i.large | m7i.xlarge | m7i.2xlarge | m7i.4xlarge | m7i.8xlarge | m7i.12xlarge | m7i.16xlarge | m7i.24xlarge | m7i.48xlarge
- 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 | c7i.large | c7i.xlarge | c7i.2xlarge | c7i.4xlarge | c7i.8xlarge | c7i.12xlarge | c7i.16xlarge | c7i.24xlarge | c7i.48xlarge
- Memory optimized: r5.12xlarge | r5.24xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge | r5dn.24xlarge | r6a.24xlarge | r6a.48xlarge | r6i.16xlarge | r6i.32xlarge | r6id.32xlarge | r6in.32xlarge | 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 | u-6tb1.56xlarge | u-6tb1.112xlarge | u-9tb1.112xlarge | u-12tb1.112xlarge | u-18tb1.112xlarge | u-24tb1.112xlarge | z1d.6xlarge | z1d.12xlarge
- Storage optimized: d3en.12xlarge | dl1.24xlarge | i3en.12xlarge | i3en.24xlarge | i4i.metal | r5b.12xlarge | r5b.24xlarge | i4i.16xlarge
- Accelerated computing: dl1.24xlarge | g5.24xlarge | g5.48xlarge | g6.24xlarge | g6.48xlarge | inf1.24xlarge | p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | vt1.24xlarge

AWS Graviton processors have built-in power saving modes and operate at a fixed frequency. Therefore, they do not provide the ability for the operating system to control C-states and P-states.

You might want to change the C-state or P-state settings to increase processor performance consistency, reduce latency, or tune your instance for a specific workload. The default C-state and

P-state settings provide maximum performance, which is optimal for most workloads. However, if your application would benefit from reduced latency at the cost of higher single- or dual-core frequencies, or from consistent performance at lower frequencies as opposed to bursty Turbo Boost frequencies, consider experimenting with the C-state or P-state settings that are available to these instances.

The following sections describe the different processor state configurations and how to monitor the effects of your configuration. These procedures were written for, and apply to Amazon Linux; however, they may also work for other Linux distributions with a Linux kernel version of 3.9 or newer. For more information about other Linux distributions and processor state control, see your system-specific documentation.

Note

The examples on this page use the following:

- The **turbostat** utility to display processor frequency and C-state information. The **turbostat** utility is available on Amazon Linux by default.
- The **stress** command to simulate a workload. To install **stress**, first enable the EPEL repository by running **sudo amazon-linux-extras install epel**, and then run **sudo yum install -y stress**.

If the output does not display the C-state information, include the **--debug** option in the command (**sudo turbostat --debug stress <options>**).

Contents

- [Highest performance with maximum Turbo Boost frequency](#)
- [High performance and low latency by limiting deeper C-states](#)
- [Baseline performance with the lowest variability](#)

Highest performance with maximum Turbo Boost frequency

This is the default processor state control configuration for the Amazon Linux AMI, and it is recommended for most workloads. This configuration provides the highest performance with lower variability. Allowing inactive cores to enter deeper sleep states provides the thermal headroom required for single or dual core processes to reach their maximum Turbo Boost potential.

The following example shows a `c4.8xlarge` instance with two cores actively performing work reaching their maximum processor Turbo Boost frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU   %c0 GHz TSC SMI   %c1   %c3   %c6   %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
           5.54 3.44 2.90  0  9.18  0.00  85.28  0.00  0.00  0.00  0.00  0.00
 94.04 32.70 54.18  0.00
0  0  0  0.12 3.26 2.90  0  3.61  0.00  96.27  0.00  0.00  0.00  0.00
48.12 18.88 26.02  0.00
0  0  18  0.12 3.26 2.90  0  3.61
0  1  1  0.12 3.26 2.90  0  4.11  0.00  95.77  0.00
0  1  19  0.13 3.27 2.90  0  4.11
0  2  2  0.13 3.28 2.90  0  4.45  0.00  95.42  0.00
0  2  20  0.11 3.27 2.90  0  4.47
0  3  3  0.05 3.42 2.90  0  99.91  0.00  0.05  0.00
0  3  21  97.84 3.45 2.90  0  2.11
...
1  1  10  0.06 3.33 2.90  0  99.88  0.01  0.06  0.00
1  1  28  97.61 3.44 2.90  0  2.32
...
10.002556 sec
```

In this example, vCPUs 21 and 28 are running at their maximum Turbo Boost frequency because the other cores have entered the C6 sleep state to save power and provide both power and thermal headroom for the working cores. vCPUs 3 and 10 (each sharing a processor core with vCPUs 21 and 28) are in the C1 state, waiting for instruction.

In the following example, all 18 cores are actively performing work, so there is no headroom for maximum Turbo Boost, but they are all running at the "all core Turbo Boost" speed of 3.2 GHz.

```
[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU   %c0 GHz TSC SMI   %c1   %c3   %c6   %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
           99.27 3.20 2.90  0  0.26  0.00  0.47  0.00  0.00  0.00  0.00  0.00
228.59 31.33 199.26  0.00
0  0  0  99.08 3.20 2.90  0  0.27  0.01  0.64  0.00  0.00  0.00  0.00
114.69 18.55 99.32  0.00
```

```

0  0  18  98.74  3.20  2.90  0  0.62
0  1   1  99.14  3.20  2.90  0  0.09  0.00  0.76  0.00
0  1  19  98.75  3.20  2.90  0  0.49
0  2   2  99.07  3.20  2.90  0  0.10  0.02  0.81  0.00
0  2  20  98.73  3.20  2.90  0  0.44
0  3   3  99.02  3.20  2.90  0  0.24  0.00  0.74  0.00
0  3  21  99.13  3.20  2.90  0  0.13
0  4   4  99.26  3.20  2.90  0  0.09  0.00  0.65  0.00
0  4  22  98.68  3.20  2.90  0  0.67
0  5   5  99.19  3.20  2.90  0  0.08  0.00  0.73  0.00
0  5  23  98.58  3.20  2.90  0  0.69
0  6   6  99.01  3.20  2.90  0  0.11  0.00  0.89  0.00
0  6  24  98.72  3.20  2.90  0  0.39
...

```

High performance and low latency by limiting deeper C-states

C-states control the sleep levels that a core may enter when it is inactive. You may want to control C-states to tune your system for latency versus performance. Putting cores to sleep takes time, and although a sleeping core allows more headroom for another core to boost to a higher frequency, it takes time for that sleeping core to wake back up and perform work. For example, if a core that is assigned to handle network packet interrupts is asleep, there may be a delay in servicing that interrupt. You can configure the system to not use deeper C-states, which reduces the processor reaction latency, but that in turn also reduces the headroom available to other cores for Turbo Boost.

A common scenario for disabling deeper sleep states is a Redis database application, which stores the database in system memory for the fastest possible query response time.

To limit deeper sleep states on Amazon Linux 2

1. Open the `/etc/default/grub` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /etc/default/grub
```

2. Edit the `GRUB_CMDLINE_LINUX_DEFAULT` line and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1
processor.max_cstate=1"
```

```
GRUB_TIMEOUT=0
```

The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.
4. Run the following command to rebuild the boot configuration.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

To limit deeper sleep states on Amazon Linux AMI

1. Open the `/boot/grub/grub.conf` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. Edit the kernel line of the first entry and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
intel_idle.max_cstate=1 processor.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-

based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.
4. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

The following example shows a `c4.8xlarge` instance with two cores actively performing work at the "all core Turbo Boost" core frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5322] successful run completed in 10s
pk cor CPU   %c0 GHz  TSC SMI   %c1   %c3   %c6   %c7   %pc2  %pc3  %pc6  %pc7
  Pkg_W RAM_W PKG_% RAM_%
      5.56 3.20 2.90   0 94.44  0.00  0.00  0.00  0.00  0.00  0.00  0.00
131.90 31.11 199.47 0.00
  0  0  0  0.03 2.08 2.90   0 99.97  0.00  0.00  0.00  0.00  0.00  0.00
 67.23 17.11 99.76  0.00
  0  0 18  0.01 1.93 2.90   0 99.99
  0  1  1  0.02 1.96 2.90   0 99.98  0.00  0.00  0.00
  0  1 19 99.70 3.20 2.90   0  0.30
...
  1  1 10  0.02 1.97 2.90   0 99.98  0.00  0.00  0.00
  1  1 28 99.67 3.20 2.90   0  0.33
  1  2 11  0.04 2.63 2.90   0 99.96  0.00  0.00  0.00
  1  2 29  0.02 2.11 2.90   0 99.98
...
```

In this example, the cores for vCPUs 19 and 28 are running at 3.2 GHz, and the other cores are in the C1 C-state, awaiting instruction. Although the working cores are not reaching their maximum Turbo Boost frequency, the inactive cores will be much faster to respond to new requests than they would be in the deeper C6 C-state.

Baseline performance with the lowest variability

You can reduce the variability of processor frequency with P-states. P-states control the desired performance (in CPU frequency) from a core. Most workloads perform better in P0, which requests

Turbo Boost. But you may want to tune your system for consistent performance rather than bursty performance that can happen when Turbo Boost frequencies are enabled.

Intel Advanced Vector Extensions (AVX or AVX2) workloads can perform well at lower frequencies, and AVX instructions can use more power. Running the processor at a lower frequency, by disabling Turbo Boost, can reduce the amount of power used and keep the speed more consistent. For more information about optimizing your instance configuration and workload for AVX, see the [Intel website](#).

CPU idle drivers control P-state. Newer CPU generations require updated CPU idle drivers that correspond to the kernel level as follows:

- Linux kernel versions 5.6 and higher (for example, m6i) – Supports Intel Icelake.
- Linux kernel versions 5.10 and higher (for example, m6a) – Supports AMD Milan.

To detect if a running system's kernel recognizes the CPU, run the following command.

```
if [ -d /sys/devices/system/cpu/cpu0/cpuidle ]; then echo "C-state control enabled";  
else echo "Kernel cpuidle driver does not recognize this CPU generation"; fi
```

If the output of this command indicates a lack of support, we recommend that you upgrade the kernel.

This section describes how to limit deeper sleep states and disable Turbo Boost (by requesting the P1 P-state) to provide low-latency and the lowest processor speed variability for these types of workloads.

To limit deeper sleep states and disable Turbo Boost on Amazon Linux 2

1. Open the `/etc/default/grub` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /etc/default/grub
```

2. Edit the `GRUB_CMDLINE_LINUX_DEFAULT` line and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0  
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1  
processor.max_cstate=1"
```

```
GRUB_TIMEOUT=0
```

The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.
4. Run the following command to rebuild the boot configuration.

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

6. When you need the low processor speed variability that the P1 P-state provides, run the following command to disable Turbo Boost.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

7. When your workload is finished, you can re-enable Turbo Boost with the following command.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

To limit deeper sleep states and disable Turbo Boost on Amazon Linux AMI

1. Open the `/boot/grub/grub.conf` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. Edit the `kernel` line of the first entry and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
```

```
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
intel_idle.max_cstate=1 processor.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.
4. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

5. When you need the low processor speed variability that the P1 P-state provides, run the following command to disable Turbo Boost.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

6. When your workload is finished, you can re-enable Turbo Boost with the following command.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

The following example shows a `c4.8xlarge` instance with two vCPUs actively performing work at the baseline core frequency, with no Turbo Boost.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU   %c0  GHz  TSC SMI   %c1   %c3   %c6   %c7   %pc2  %pc3  %pc6  %pc7
 Pkg_W RAM_W PKG_% RAM_%
      5.59 2.90 2.90   0 94.41  0.00  0.00  0.00  0.00  0.00  0.00  0.00
128.48 33.54 200.00  0.00
 0   0   0   0.04 2.90 2.90   0 99.96  0.00  0.00  0.00  0.00  0.00  0.00
 65.33 19.02 100.00  0.00
 0   0  18   0.04 2.90 2.90   0 99.96
 0   1   1   0.05 2.90 2.90   0 99.95  0.00  0.00  0.00
 0   1  19   0.04 2.90 2.90   0 99.96
 0   2   2   0.04 2.90 2.90   0 99.96  0.00  0.00  0.00
```

```

0  2  20  0.04 2.90 2.90  0  99.96
0  3   3  0.05 2.90 2.90  0  99.95  0.00  0.00  0.00
0  3  21  99.95 2.90 2.90  0  0.05
...
1  1  28  99.92 2.90 2.90  0  0.08
1  2  11  0.06 2.90 2.90  0  99.94  0.00  0.00  0.00
1  2  29  0.05 2.90 2.90  0  99.95

```

The cores for vCPUs 21 and 28 are actively performing work at the baseline processor speed of 2.9 GHz, and all inactive cores are also running at the baseline speed in the C1 C-state, ready to accept instructions.

I/O scheduler

The I/O scheduler is a part of the Linux operating system that sorts and merges I/O requests and determines the order in which they are processed.

I/O schedulers are particularly beneficial for devices such as magnetic hard drives, where seek time can be expensive and where it is optimal to merge co-located requests. I/O schedulers have less of an effect with solid state devices and virtualized environments. This is because for solid state devices, sequential and random access don't differ, and for virtualized environments, the host provides its own layer of scheduling.

This topic discusses the Amazon Linux I/O scheduler. For more information about the I/O scheduler used by other Linux distributions, refer to their respective documentation.

Topics

- [Supported schedulers](#)
- [Default scheduler](#)
- [Change the scheduler](#)

Supported schedulers

Amazon Linux supports the following I/O schedulers:

- `deadline` — The *Deadline* I/O scheduler sorts I/O requests and handles them in the most efficient order. It guarantees a start time for each I/O request. It also gives I/O requests that have been pending for too long a higher priority.

- `cfq` — The *Completely Fair Queueing* (CFQ) I/O scheduler attempts to fairly allocate I/O resources between processes. It sorts and inserts I/O requests into per-process queues.
- `noop` — The *No Operation* (noop) I/O scheduler inserts all I/O requests into a FIFO queue and then merges them into a single request. This scheduler does not do any request sorting.

Default scheduler

No Operation (noop) is the default I/O scheduler for Amazon Linux. This scheduler is used for the following reasons:

- Many instance types use virtualized devices where the underlying host performs scheduling for the instance.
- Solid state devices are used in many instance types where the benefits of an I/O scheduler have less effect.
- It is the least invasive I/O scheduler, and it can be customized if needed.

Change the scheduler

Changing the I/O scheduler can increase or decrease performance based on whether the scheduler results in more or fewer I/O requests being completed in a given time. This is largely dependent on your workload, the generation of the instance type that's being used, and the type of device being accessed. If you change the I/O scheduler being used, we recommend that you use a tool, such as **iotop**, to measure I/O performance and to determine whether the change is beneficial for your use case.

You can view the I/O scheduler for a device using the following command, which uses `nvme0n1` as an example. Replace `nvme0n1` in the following command with the device listed in `/sys/block` on your instance.

```
$ cat /sys/block/nvme0n1/queue/scheduler
```

To set the I/O scheduler for the device, use the following command.

```
$ echo cfq|deadline|noop > /sys/block/nvme0n1/queue/scheduler
```

For example, to set the I/O scheduler for an `xvda` device from `noop` to `cfq`, use the following command.

```
$ echo cfq > /sys/block/xvda/queue/scheduler
```

Set the time for your Linux instance

A consistent and accurate time reference on your Linux 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.

The 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 NTP or a direct PTP connection. 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](#).

For a backup to the local Amazon Time Sync Service on your instance, and 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.

For Windows instances, see [Set the time for your Windows instance](#).

Topics

- [Set your instance to use the local Amazon Time Sync Service](#)
- [Set your instance or any internet-connected device to use the public Amazon Time Sync Service](#)
- [Compare timestamps](#)
- [Change the time zone on your instance](#)

- [Leap seconds](#)
- [Related resources](#)

Set your instance to use the local Amazon Time Sync Service

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 by [instances built on the AWS Nitro System](#).)
- Through a direct PTP connection to connect to a local PTP hardware clock:
 - PHC0

NTP and PTP connections do not require any VPC configuration changes, and your instance does not require access to the internet.

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 to the PTP hardware clock are the same as the regular Amazon Time Sync Service connection over IPv4 or IPv6. If your software is configured to the NTP endpoint and is running on an instance with a PTP hardware clock, it will be connected to the PTP hardware clock automatically over NTP.

AL2023 and the latest versions of Amazon Linux 2 and Amazon Linux AMIs are configured to use the Amazon Time Sync Service IPv4 endpoint by default. 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. For the IPv6 configuration instructions, see [Connect to the IPv6 endpoint of the Amazon Time Sync Service](#). For the PTP hardware clock configuration instructions, see [Connect to the PTP hardware clock](#).

If you're using an AMI that doesn't have the Amazon Time Sync Service configured by default, use one of the following procedures to configure the Amazon Time Sync Service on your instance using the chrony client.

Contents

- [Connect to the IPv4 endpoint of the Amazon Time Sync Service](#)

- [Connect to the IPv6 endpoint of the Amazon Time Sync Service](#)
- [Connect to the PTP hardware clock](#)

Connect to the IPv4 endpoint of the Amazon Time Sync Service

This section describes how to configure your instance to use the local Amazon Time Sync Service through the IPv4 endpoint. It requires adding a server entry for the Amazon Time Sync Service to the `chrony` configuration file. On the latest versions of these AMIs, `chrony` is already installed and configured to use the Amazon Time Sync Service IPv4 endpoint.

Use the instructions for your instance's operating system.

Amazon Linux

Note

On AL2023 and the latest versions of Amazon Linux 2 and Amazon Linux AMI, `chrony` is already installed and configured to use the Amazon Time Sync Service IPv4 endpoint. If you're using one of these latest AMIs, you can skip this procedure.

To connect to the IPv4 endpoint of the Amazon Time Sync Service on Amazon Linux using `chrony`

1. Connect to your instance and uninstall the NTP service.

```
[ec2-user ~]$ sudo yum erase 'ntp*'
```

2. Install the `chrony` package.

```
[ec2-user ~]$ sudo yum install chrony
```

3. Open the `/etc/chrony.conf` file using a text editor (such as **vim** or **nano**). Verify that the file includes the following line:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

If the line is present, then the Amazon Time Sync Service is already configured to use the IPv4 endpoint of the Amazon Time Sync Service and you can go to the next step. If not, add

the line after any other `server` or `pool` statements that are already present in the file, and save your changes.

- Restart the `chronyd` 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`.

- To configure `chronyd` to start at each system boot, use the `chkconfig` command.

```
[ec2-user ~]$ sudo chkconfig chronyd on
```

- Verify that `chrony` is using the `169.254.169.123` IPv4 endpoint to synchronize the time.

```
[ec2-user ~]$ chronyc sources -v
```

```
210 Number of sources = 7

    .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
   /  .- Source state '*' = current synced, '+' = combined , '-' = not
combined,
  | /   '?' = unreachable, 'x' = time may be in error, '~' = time too
variable.
  ||                                     .- xxxx [ yyyy ] +/-
zzzz                                     |
  ||      Reachability register (octal) -.   |  xxxx = adjusted
offset,                                  |
  ||      Log2(Polling interval) --.      |  |  yyyy = measured
offset,                                  |  |
  ||                                     \   |  |  zzzz = estimated
error.                                  |  |
  ||                                     |  |  \
MS Name/IP address                     Stratum Poll Reach LastRx Last sample
```

```

=====
^* 169.254.169.123          3   6   17   43   -30us[ -226us] +/-
287us
^- ec2-12-34-231-12.eu-west>  2   6   17   43   -388us[ -388us] +/-
11ms
^- tshirt.heanet.ie        1   6   17   44   +178us[ +25us] +/-
1959us
^? tbag.heanet.ie          0   6   0    -    +0ns[ +0ns] +/-
0ns
^? bray.walcz.net          0   6   0    -    +0ns[ +0ns] +/-
0ns
^? 2a05:d018:c43:e312:ce77:>  0   6   0    -    +0ns[ +0ns] +/-
0ns
^? 2a05:d018:dab:2701:b70:b>  0   6   0    -    +0ns[ +0ns] +/-
0ns

```

In the output that's returned, `^*` indicates the preferred time source.

7. Verify the time synchronization metrics that are reported by `chrony`.

```
[ec2-user ~]$ chronyc tracking
```

```

Reference ID      : A9FEA97B (169.254.169.123)
Stratum           : 4
Ref time (UTC)   : Wed Nov 22 13:18:34 2017
System time      : 0.000000626 seconds slow of NTP time
Last offset      : +0.002852759 seconds
RMS offset       : 0.002852759 seconds
Frequency        : 1.187 ppm fast
Residual freq    : +0.020 ppm
Skew             : 24.388 ppm
Root delay       : 0.000504752 seconds
Root dispersion  : 0.001112565 seconds
Update interval  : 64.4 seconds
Leap status      : Normal

```

Ubuntu

To connect to the IPv4 endpoint of the Amazon Time Sync Service on Ubuntu using chrony

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
```

```
210 Number of sources = 7

    .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
    /  .- Source state '*' = current synced, '+' = combined , '-' = not
combined,
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too
variable.
||                                     .- xxxx [ yyyy ]
+/- zzzz
||      Reachability register (octal) -.    |  xxxx =
adjusted offset,
```

```

||      Log2(Polling interval) --.      |      |      yyyy =
measured offset,
||                                     \      |      |      zzzz =
estimated error.
||                                     |      |      \
MS Name/IP address      Stratum Poll Reach LastRx Last sample

=====
^* 169.254.169.123      3  6  17  12  +15us[ +57us]
+/- 320us
^- tbag.heanet.ie      1  6  17  13  -3488us[-3446us]
+/- 1779us
^- ec2-12-34-231-12.eu-west- 2  6  17  13  +893us[ +935us]
+/- 7710us
^? 2a05:d018:c43:e312:ce77:6 0  6  0  10y  +0ns[ +0ns]
+/- 0ns
^? 2a05:d018:d34:9000:d8c6:5 0  6  0  10y  +0ns[ +0ns]
+/- 0ns
^? tshirt.heanet.ie    0  6  0  10y  +0ns[ +0ns]
+/- 0ns
^? bray.walcz.net      0  6  0  10y  +0ns[ +0ns]
+/- 0ns

```

In the output that's returned, on the line starting with `^*` indicates the preferred time source.

5. Verify the time synchronization metrics that are reported by `chrony`.

```
ubuntu:~$ chronyc tracking
```

```

Reference ID      : 169.254.169.123 (169.254.169.123)
Stratum          : 4
Ref time (UTC)   : Wed Nov 29 07:41:57 2017
System time      : 0.000000011 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 connect to IPv4 endpoint of the Amazon Time Sync Service on SUSE Linux using `chrony`

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.

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 [instances built on the AWS Nitro System](#).

Note

We don't recommend using both the IPv4 and IPv6 endpoint entries together in your `chrony.conf` file. 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.

Depending on the Linux distribution you are 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.

Connect to the PTP hardware clock

This section describes how to configure your 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 to the `chrony` configuration file.

If your instance has a PTP hardware clock and you configured the NTP connection (to either the IPv4 or the IPv6 endpoint), your instance time is automatically sourced from the PTP hardware clock. The steps below configure the *direct* PTP connection, which will give you more accurate time than the NTP connection.

Requirements

The PTP hardware clock is available on an instance when the following requirements are met:

- Supported AWS Regions: US East (N. Virginia) and Asia Pacific (Tokyo)
- Supported instance families: C7a | C7i | M7a | M7g | M7i | R7a | R7g | R7i
- 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*.

To connect to the PTP hardware clock

1. Connect to your instance and install the Linux kernel driver for Elastic Network Adapter (ENA) version 2.10.0 or later. For the installation instructions, see [Linux kernel driver for Elastic Network Adapter \(ENA\) family](#) on *GitHub*.
2. Verify that the `/dev/ptp0` device shows up on your instance.

```
[ec2-user ~]$ ls /dev/ptp0
```

The following is the expected output. If `/dev/ptp0` is not in the output, the ENA driver was not correctly installed. Review step 1 in this procedure for installing the driver.

```
/dev/ptp0
```

3. Edit `/etc/chrony.conf` using a text editor and add the following line anywhere in the file.

```
refclock PHC /dev/ptp0 poll 0 delay 0.000010 prefer
```

4. Restart `chrony` using the following command.

```
[ec2-user ~]$ sudo systemctl restart chronyd
```

5. 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 your 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.

Depending on the operating system of your instance or device, use one of the following procedures to set your instance or device to use the public Amazon Time Sync Service.

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 check box 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

If you're using the Amazon Time Sync Service, you can compare the timestamps on your Amazon EC2 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*.

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 on your instance

Amazon Linux instances are set to the UTC (Coordinated Universal Time) time zone by default. You can change the time on an instance to the local time zone or to another time zone in your network.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To change the time zone on an AL2023 or Amazon Linux 2 instance

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
```

To change the time zone on an Amazon Linux instance

1. Identify the time zone to use on the instance. The `/usr/share/zoneinfo` directory contains a hierarchy of time zone data files. Browse the directory structure at that location to find a file for your time zone.

```
[ec2-user ~]$ ls /usr/share/zoneinfo
Africa      Chile      GB         Indian     Mideast    posixrules US
America     CST6CDT   GB-Eire    Iran       MST        PRC        UTC
Antarctica  Cuba      GMT        iso3166.tab MST7MDT    PST8PDT    WET
Arctic      EET       GMT0       Israel     Navajo     right      W-SU
...
```

Some of the entries at this location are directories (such as `America`), and these directories contain time zone files for specific cities. Find your city (or a city in your time zone) to use for the instance.

2. Update the `/etc/sysconfig/clock` file with the new time zone. In this example, we use the time zone data file for Los Angeles, `/usr/share/zoneinfo/America/Los_Angeles`.
 - a. Open the `/etc/sysconfig/clock` file with a text editor (such as **vim** or **nano**). You need to use **sudo** with your editor command because `/etc/sysconfig/clock` is owned by `root`.

```
[ec2-user ~]$ sudo nano /etc/sysconfig/clock
```

- b. Locate the `ZONE` entry, and change it to the time zone file (omitting the `/usr/share/zoneinfo` section of the path). For example, to change to the Los Angeles time zone, change the `ZONE` entry to the following:

```
ZONE="America/Los_Angeles"
```

Note

Do not change the `UTC=true` entry to another value. This entry is for the hardware clock, and does not need to be adjusted when you're setting a different time zone on your instance.

- c. Save the file and exit the text editor.

3. Create a symbolic link between `/etc/localtime` and the time zone file so that the instance finds the time zone file when it references local time information.

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

4. Reboot the system to pick up the new time zone information in all services and applications.

```
[ec2-user ~]$ sudo reboot
```

5. (Optional) Confirm that the current time zone is updated to the new time zone by using the `date` command. The current time zone appears in the output. In the following example, the current time zone is PDT, which refers to the Los Angeles time zone.

```
[ec2-user ~]$ date
Sun Aug 16 05:45:16 PDT 2020
```

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-smearred 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-smearred time sources in your time client configuration during a leap second event.

Related resources

- <https://chrony-project.org/>

Configure your 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.

To get started with a GPU-based instance, you must install the appropriate drivers.

Contents

- [Install NVIDIA drivers on Linux instances](#)
- [Install AMD drivers on Linux instances](#)
- [Activate NVIDIA GRID Virtual Applications on your Amazon EC2 GPU-based instances](#)
- [Set up Dual 4K displays on G4ad instances](#)
- [Get started with P5 instances](#)
- [Optimize GPU settings on Amazon EC2 instances](#)

Install NVIDIA drivers on Linux instances

An instance with an attached NVIDIA GPU, such as a P3 or G4dn instance, must have the appropriate NVIDIA driver installed. Depending on the instance type, you can either download a public NVIDIA driver, download a driver from Amazon S3 that is available only to AWS customers, or use an AMI with the driver pre-installed.

To install AMD drivers on a Linux instance with an attached AMD GPU, such as a G4ad instance, see [Install AMD drivers](#). To install NVIDIA drivers on a Windows instance, see [Install NVIDIA drivers on Windows instances](#). To install NVIDIA drivers on a Linux instance, see [Install NVIDIA drivers on a Linux instance](#).

Contents

- [Types of NVIDIA drivers](#)
- [Available drivers by instance type](#)
- [Installation options](#)
 - [Option 1: AMIs with the NVIDIA drivers installed](#)
 - [Option 2: Public NVIDIA drivers](#)

- [Option 3: GRID drivers \(G6, Gr6, G5, G4dn, and G3 instances\)](#)
- [Option 4: NVIDIA gaming drivers \(G5 and G4dn instances\)](#)
- [Install an additional version of CUDA](#)

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
G6	Yes	Yes	No
Gr6	Yes	Yes	No
P2	Yes	No	No
P3	Yes	No	No
P4d	Yes	No	No
P4de	Yes	No	No

¹ This Tesla driver also supports optimized graphics applications specific to the ARM64 platform

² Using Marketplace AMIs only

Installation options

Use one of the following options to get the NVIDIA drivers required for your GPU instance.

Options

- [Option 1: AMIs with the NVIDIA drivers installed](#)

- [Option 2: Public NVIDIA drivers](#)
- [Option 3: GRID drivers \(G6, Gr6, G5, G4dn, and G3 instances\)](#)
- [Option 4: NVIDIA gaming drivers \(G5 and G4dn instances\)](#)

Option 1: AMIs with the NVIDIA drivers installed

AWS and NVIDIA offer different Amazon Machine Images (AMI) that come with the NVIDIA drivers installed.

- [Marketplace offerings with the Tesla driver](#)
- [Marketplace offerings with the GRID driver](#)
- [Marketplace offerings with the Gaming driver](#)

To 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 [using Sysprep](#) to ensure that the GRID driver works.

Option 2: Public NVIDIA drivers

The options offered by AWS come with the necessary license for the driver. Alternatively, you can install the public drivers and bring your own license. To install a public driver, download it from the NVIDIA site as described [here](#).

Alternatively, you can use the options offered by AWS instead of the public drivers. To use a GRID driver on a P3 instance, use the AWS Marketplace AMIs as described in [Option 1](#). To use a GRID driver on a G6, Gr6, G5, G4dn, or G3 instance, use the AWS Marketplace AMIs as described in [Option 1](#) or install the NVIDIA drivers provided by AWS as described in [Option 3: GRID drivers \(G6, Gr6, G5, G4dn, and G3 instances\)](#).

To download a public NVIDIA driver

Log on to your Linux instance and download the 64-bit NVIDIA driver appropriate for the instance type from <http://www.nvidia.com/Download/Find.aspx>. For **Product Type**, **Product Series**, and **Product**, use the options in the following table.

Instance	Product Type	Product Series	Product
G3	Tesla	M-Class	M60
G4dn	Tesla	T-Series	T4
G5 ¹	Tesla	A-Series	A10
G5g ²	Tesla	T-Series	NVIDIA T4G
G6 ³	Tesla	L-Series	L4
Gr6 ³	Tesla	L-Series	L4
P2	Tesla	K-Series	K80
P3	Tesla	V-Series	V100
P4d	Tesla	A-Series	A100
P4de	Tesla	A-Series	A100
P5 ⁴	Tesla	H-Series	H100

¹ G5 instances require driver version 470.00 or later

² G5g instances require driver version 470.82.01 or later. The operating systems is Linux aarch64

³ G6 and Gr6 instances require driver version 525.0 or later.

⁴ P5 instances require driver version 530 or later.

To install the NVIDIA driver on Linux

For more information about installing and configuring the driver, see the [NVIDIA Driver Installation Quickstart Guide](#).

Option 3: GRID drivers (G6, Gr6, 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 A10G, NVIDIA Tesla T4, or NVIDIA Tesla M60 hardware. Upon installation of the software, you are bound by the terms of the [NVIDIA GRID Cloud End User License Agreement](#). For information about the version of the NVIDIA GRID driver for your operating system, see the [NVIDIA® Virtual GPU \(vGPU\) Software Documentation](#) on the *NVIDIA website*.

Considerations

- G6 and Gr6 instances require GRID 17 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 [using Sysprep](#) to ensure that the GRID driver works.
- NVIDIA GRID driver releases 17.0 and later do not support Windows Server 2019.
- NVIDIA GRID driver releases 14.2 and later do not support Windows Server 2016.

Amazon Linux and Amazon Linux 2

To install the NVIDIA GRID driver on your 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 **gcc** and **make**, if they are not already installed.

```
[ec2-user ~]$ sudo yum install gcc make
```

4. Update your package cache and get the package updates for your instance.

```
[ec2-user ~]$ sudo yum update -y
```

5. Reboot your instance to load the latest kernel version.

```
[ec2-user ~]$ sudo reboot
```

6. Reconnect to your instance after it has rebooted.
7. Install the **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

8. 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/
```

9. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

Note

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).

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.

```
[ec2-user ~]$ 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, on why this is required visit [NVIDIA's 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) Depending on your use case, you might complete the following optional steps. If you do not require this functionality, do not complete these steps.

- a. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol [NICE DCV](#).

- b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications 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 Linux instance. Install **gcc** and **make**, if they are not already installed.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y gcc 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, on why this is required visit [NVIDIA's 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 [NICE DCV](#).
- b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications 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 Linux instance. Install **gcc** and **make**, if they are not already installed.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo dnf install -y make gcc 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, on why this is required visit [NVIDIA's 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 [NICE DCV](#).
 - b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications 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 Linux instance. Install **gcc** and **make**, if they are not already installed.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo dnf install -y make gcc 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, on why this is required visit [NVIDIA's 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 [NICE DCV](#).
- b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications on your Amazon EC2 GPU-based instances](#).

Ubuntu and Debian

To install the NVIDIA GRID driver on your instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.
2. Update your package cache and get the package updates for your instance.

```
$ sudo apt-get update -y
```

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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
$ sudo apt-get install -y gcc make 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 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, on why this is required visit [NVIDIA's 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 [NICE DCV](#).
 - b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications 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. 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 Windows 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. 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 Documentation](#) 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).

4. 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.

5. (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
```

6. (Optional) Depending on your use case, you might complete the following optional steps. If you do not require this functionality, do not complete these steps.
 - a. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol, [NICE DCV](#).
 - b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications on your Amazon EC2 GPU-based instances](#).

Option 4: NVIDIA gaming drivers (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 A10G, and NVIDIA Tesla T4

hardware. 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.

Amazon Linux and Amazon Linux 2

To install the NVIDIA gaming driver on your 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 **gcc** and **make**, if they are not already installed.

```
[ec2-user ~]$ sudo yum install gcc make
```

4. Update your package cache and get the package updates for your instance.

```
[ec2-user ~]$ sudo yum update -y
```

5. Reboot your instance to load the latest kernel version.

```
[ec2-user ~]$ sudo reboot
```

6. Reconnect to your instance after it has rebooted.
7. Install the **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

8. Download the gaming driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

9. Extract the gaming driver installation utility from the downloaded .zip archive.

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

10. Add permissions to run the driver installation utility using the following command:

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

11. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

Note

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).

12. Use the following command to create the required configuration file.

```
[ec2-user ~]$ 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:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2023_9_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"
```

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, on why this is required visit [NVIDIA's 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
```

15. Reboot the instance.

```
[ec2-user ~]$ sudo reboot
```

16. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol [NICE 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.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y unzip gcc 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 vGPUSW-*vGaming-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/Linux/NVIDIA-Linux-x86_64*-grid.run
```

10. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./nvidia-drivers/Linux/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_2023_9_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, on why this is required visit [NVIDIA's 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 [NICE 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.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo yum install -y unzip gcc 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 vGPUSW-*vGaming-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/Linux/NVIDIA-Linux-x86_64*-grid.run
```

9. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./nvidia-drivers/Linux/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_2023_9_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, on why this is required visit [NVIDIA's 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 [NICE 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.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
[ec2-user ~]$ sudo dnf install -y unzip gcc make 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 vGPUSW-*vGaming-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/Linux/NVIDIA-Linux-x86_64*-grid.run
```

9. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./nvidia-drivers/Linux/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_2023_9_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, on why this is required visit [NVIDIA's 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 [NICE 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.
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 **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

```
$ sudo apt-get install -y unzip gcc make 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 vGPUSW-*vGaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

10. Add permissions to run the driver installation utility using the following command:

```
$ chmod +x nvidia-drivers/Linux/NVIDIA-Linux-x86_64*-grid.run
```

11. Run the installer using the following command:

```
$ sudo ./nvidia-drivers/Linux/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_2023_9_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, on why this is required visit [NVIDIA's 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 [NICE 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 [using Sysprep](#) to ensure that the gaming driver works.
- 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 Windows PowerShell User Guide*.
- Your users 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*.

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 461.40 or later:


```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertWindows_2023_9_22.cert" -OutFile "$Env:PUBLIC\Documents\nvidia-cert.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\nvidia-cert.txt"
```

- For earlier versions:

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Windows_2019_09.cert" -OutFile "$Env:PUBLIC\Documents\nvidia-cert.txt"
```

6. Reboot your instance.
7. Verify the NVIDIA Gaming license using the following command.

```
C:\Windows\System32\DriverStore\FileRepository\nv_dispswi.inf_*\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 [NICE DCV](#). If you do not require this functionality, do not complete this step.

Install an additional version of CUDA

After you install an NVIDIA graphics driver on your instance, you can install a version of CUDA other than the version that is bundled with the graphics driver. The following procedure demonstrates how to configure multiple versions of CUDA on the instance.

To install the CUDA toolkit

1. Connect to your 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 AMD drivers on Linux instances

An instance with an attached AMD GPU, such as a G4ad instance, must have the appropriate AMD driver installed. Depending on your requirements, you can either use an AMI with the driver preinstalled or download a driver from Amazon S3.

To install NVIDIA drivers on an instance with an attached NVIDIA GPU, such as a G4dn instance, see [Install NVIDIA drivers](#) instead. To install AMD drivers on a Windows instance, see [Install AMD drivers on Windows instances](#).

Contents

- [AMD Radeon Pro Software for Enterprise Driver](#)
- [AMIs with the AMD driver installed](#)
- [AMD driver download](#)
- [Set up an interactive desktop](#)

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

AMIs with the AMD driver installed

AWS offers different Amazon Machine Images (AMI) that come with the AMD drivers installed. Open [Marketplace offerings with the AMD driver](#).

AMD driver download

If you aren't using an AMI with the AMD driver installed, you can download the AMD driver and install it on your instance. The AMD driver is only supported on Amazon Linux 2 with kernel version 4.14.

Note

AMD driver version **amdgpu-pro-20.20-1184451** and newer driver releases require kernel version 5.15 or higher.

These downloads are available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the AMD Radeon Pro V520 hardware. Upon installation of the software, you are bound by the terms of the [AMD Software End User License Agreement](#).

To install the AMD driver on your 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 **gcc** and **make**, if they are not already installed.

```
$ sudo yum install gcc make
```

4. Update your package cache and get the package updates for your instance.

- For Amazon Linux 2:

```
$ sudo amazon-linux-extras install epel -y
$ sudo yum update -y
```

- For Ubuntu 22.04:

```
$ wget https://repo.radeon.com/.preview/a0e4ef1dffbc95b4abb54e891f265e61/amdgpu-
install/5.5.02.05.2/ubuntu/jammy/amdgpu-install_5.5.02.05.50502-1_all.deb
$ sudo apt install ./amdgpu-install_5.5.02.05.50502-1_all.deb
$ sudo sed -i 's#repo.radeon.com#&/.preview/a0e4ef1dffbc95b4abb54e891f265e61#' /
etc/apt/sources.list.d/{amdgpu.list,rocm.list,amdgpu-proprietary.list}
```

- For other Ubuntu versions:

```
$ sudo dpkg --add-architecture i386
$ sudo apt-get update -y && sudo apt upgrade -y
```

- For CentOS:

```
$ sudo yum install epel-release -y
$ sudo yum update -y
```

5. Reboot the instance.

```
$ sudo reboot
```

6. Reconnect to the instance after it reboots.
7. Download the latest AMD driver.

Note

Skip this step for Ubuntu 22.04.

```
$ aws s3 cp --recursive s3://ec2-amd-linux-drivers/latest/ .
```

8. Extract the file.

- For Amazon Linux 2 and CentOS:

```
$ tar -xf amdgpu-pro-*rhel*.tar.xz
```

- For Ubuntu:

Note

Skip this step for Ubuntu 22.04.

```
$ tar -xf amdgpu-pro*ubuntu*.xz
```

9. Change to the folder for the extracted driver.
10. Add the missing modules for the driver installation.

- For Amazon Linux 2 and CentOS:

Skip this step.

- For Ubuntu:

Note

Skip this step for Ubuntu 22.04.

```
$ sudo apt install linux-modules-extra-$(uname -r) -y
```

11. Run the self install script to install the full graphics stack.

- For Ubuntu 22.04:

```
$ sudo amdgpu-install --usecase=workstation --vulkan=pro --opengl=rocr,legacy -y
```

- For Amazon Linux 2 and CentOS and other Ubuntu versions:

```
$ ./amdgpu-pro-install -y --opengl=pa1,legacy
```

12. Reboot the instance.

```
$ sudo reboot
```

13. Confirm that the driver is functional.

```
$ dmesg | grep amdgpu
```

The response should look like the following:

```
Initialized amdgpu
```

Set up an interactive desktop

After you confirm that your 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.

```
$ sudo amazon-linux-extras install epel -y
```

2. Install the MATE desktop.

```
$ sudo amazon-linux-extras install mate-desktop1.x -y  
$ 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 NICE DCV server](#) to use NICE DCV as a high-performance display protocol, and then [connect to a NICE 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 NICE DCV server](#) to use NICE DCV as a high-performance display protocol, and then [connect to a NICE 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 NICE DCV server](#) to use NICE DCV as a high-performance display protocol, and then [connect to a NICE DCV session](#) using your preferred client.

Activate NVIDIA GRID Virtual Applications on your Amazon EC2 GPU-based instances

To activate the GRID Virtual Applications on G3, G4dn, and G5 instances (NVIDIA GRID Virtual Workstation is enabled by default), you must define the product type for the driver in the `/etc/nvidia/gridd.conf` file.

To activate GRID Virtual Applications on Linux instances

1. Create the `/etc/nvidia/gridd.conf` file from the provided template file.

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. Open the `/etc/nvidia/gridd.conf` file in your favorite text editor.
3. Find the `FeatureType` line, and set it equal to `0`. Then add a line with `IgnoreSP=TRUE`.

```
FeatureType=0
IgnoreSP=TRUE
```

4. Save the file and exit.
5. Reboot the instance to pick up the new configuration.

```
[ec2-user ~]$ sudo reboot
```

Set up Dual 4K displays on G4ad instances

Launch a G4ad instance

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. Follow the instructions [here](#) to install the AMD drivers on Linux. 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

```

- When you connect into DCV, change the resolution to 2x4K, confirming the dual monitor support is registered by DCV.



Get started with P5 instances

P5 instances provide 8 NVIDIA H100 GPUs with 640 GB of high-bandwidth GPU memory. They feature 3rd generation AMD EPYC processors and provide 2 TB of system memory, 30 TB of local NVMe instance storage, 3,200 Gbps aggregated network bandwidth, and GPUDirect RDMA support. P5 instances also support Amazon EC2 UltraCluster technology, which provides lower latency and improved network performance using EFA.

The following table provides a summary of the p5.48xlarge specifications.

vCPUs	System memory	GPUs	GPU memory	Network bandwidth	GPUDirect RDMA	GPU peer to peer	Instance storage
192	2 TiB	8 NVIDIA H100 GPUs	640 GB HBM3	3200 Gbps with EFAv2	Supported	900 GB/s NVSwi	8 x 3,800 GB NVMe SSD volumes

Software configuration

The easiest way to get started with P5 instances is to launch an instance using an AWS Deep Learning AMI that is preconfigured with all of the required software. For the latest AWS Deep Learning AMI for use with P5 instances, see the [AWS Deep Learning Base GPU AMI \(Ubuntu 20.04\)](#).

If you need to build a custom AMI for use with P5 instances, we recommend installing the following minimum software versions:

- NVIDIA driver 535.54.03 or later
- CUDA 12.1 or later
- NVIDIA GDRCopy 2.3 or later
- EFA installer 1.24.1 or later
- NCCL 2.18.3 or later
- aws-ofi-ncl plugin 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](#). The latest AWS Deep Learning Base GPU AMI is preconfigured to not use deeper C-states.

Ubuntu 20.04 specific recommendations

The following recommendations for Ubuntu 20.04 help prevent unpredictable interface naming on boot:

- Ensure you are running `systemd 245.4-4ubuntu3.19` or later with the following command:

```
systemd --version
```

- Ensure you have configured GRUB:
 - Open the `/etc/default/grub` configuration file in a text editor.
 - Edit the `GRUB_CMDLINE_LINUX_DEFAULT` entry to include `net.naming-scheme=v247`.
 - Reboot your instance by running `sudo update-grub`.

Networking and EFA configuration

P5 instances deliver 3200 Gbps of networking bandwidth by using multiple EFA interfaces. P5 instances support 32 network cards. We recommend that you define a single EFA network interface per network card. To configure these interfaces at launch we recommend the following settings:

- For network interface 0, specify device index 0
- For network interfaces 1 through 31, specify device index 1

For more information about how to configure your P5 instances for EFA see [Get started with P5 instances and EFA](#).

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 autoboot feature, which varies the GPU clock speeds. By disabling autoboot and setting the GPU clock speeds to their maximum frequency, you can consistently achieve the maximum performance with your GPU instances. The following procedure helps you to configure the GPU settings to be persistent, disable the autoboot feature if needed, and set the GPU clock speeds to their maximum frequency.

The following steps are for optimizing GPU settings on a Linux instance. For Windows instances, see [Optimize GPU settings](#) in the *Amazon EC2 User Guide for Windows Instances*.

To optimize GPU settings

1. Configure the GPU settings to be persistent. This command can take several minutes to run.

```
[ec2-user ~]$ sudo nvidia-persistenced
```


2. [G3, and P2 instances only] Disable the autoboot 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
```

- 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 CPU options

Many Amazon EC2 instances support simultaneous multithreading, which enables multiple threads to run concurrently on a single CPU core. Each thread is represented as a virtual CPU (vCPU) on the instance. An instance has a default number of CPU cores, which varies according to instance type. For example, an `m5.xlarge` instance type has two CPU cores and two threads per core by default—four vCPUs in total.

Note

Each vCPU is a thread of a CPU core, except for T2 instances, M7a instances, Apple silicon Mac instances, and 64-bit ARM platforms such as instances powered by AWS Graviton processors.

In most cases, there is an Amazon EC2 instance type that has a combination of memory and number of vCPUs to suit your workloads. However, you can specify the following CPU options to optimize your instance for specific workloads or business needs:

- **Number of CPU cores:** You can customize the number of CPU cores for the instance. You might do this to potentially optimize the licensing costs of your software with an instance that has sufficient amounts of RAM for memory-intensive workloads but fewer CPU cores.
- **Threads per core:** You can disable multithreading by specifying a single thread per CPU core. You might do this for certain workloads, such as high performance computing (HPC) workloads.

You can specify these CPU options during instance launch. There is no additional or reduced charge for specifying CPU options. You're charged the same as instances that are launched with default CPU options.

Contents

- [Rules for specifying CPU options](#)
- [CPU cores and threads per CPU core per instance type](#)
- [Specify CPU options for your instance](#)

- [View the CPU options for your instance](#)

Rules for specifying CPU options

To specify the CPU options for your instance, be aware of the following rules:

- You can't specify CPU options for bare metal instances.
- CPU options can only be specified during instance launch and cannot be modified after launch.
- When you launch an instance, you must specify both the number of CPU cores and threads per core in the request. For example requests, see [Specify CPU options for your instance](#).
- The number of vCPUs for the instance is the number of CPU cores multiplied by the threads per core. To specify a custom number of vCPUs, you must specify a valid number of CPU cores and threads per core for the instance type. You cannot exceed the default number of vCPUs for the instance. For more information, see [CPU cores and threads per CPU core per instance type](#).
- To disable multithreading, specify one thread per core.
- When you [change the instance type](#) of an existing instance, the CPU options automatically change to the default CPU options for the new instance type.
- The specified CPU options persist after you stop, start, or reboot an instance.

CPU cores and threads per CPU core per instance type

The following tables list the instance types that support specifying CPU options.

Contents

- [General purpose instances](#)
- [Compute optimized instances](#)
- [Memory optimized instances](#)
- [Storage optimized instances](#)
- [Accelerated computing instances](#)
- [High-performance computing instances](#)

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
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,	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
				24, 26, 28, 30, 32	
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5dn.large	2	1	2	1	1, 2
m5dn.xlarge	4	2	2	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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m5n.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6a.large	2	1	2	1	1, 2
m6a.xlarge	4	2	2	1, 2	1, 2
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6g.xlarge	4	4	1	1, 2, 3, 4	1
m6g.2xlarge	8	8	1	1, 2, 3, 4, 5, 6, 7, 8	1
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
t3.nano	2	1	2	1	1, 2
t3.micro	2	1	2	1	1, 2
t3.small	2	1	2	1	1, 2
t3.medium	2	1	2	1	1, 2
t3.large	2	1	2	1	1, 2
t3.xlarge	4	2	2	2	1, 2
t3.2xlarge	8	4	2	2, 4	1, 2
t3a.nano	2	1	2	1	1, 2
t3a.micro	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c5a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5a.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5a.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5a.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5a.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
c5a.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2
c5ad.large	2	1	2	1	1, 2
c5ad.xlarge	4	2	2	1, 2	1, 2
c5ad.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5ad.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r5.large	2	1	2	1	1, 2
r5.xlarge	4	2	2	2	1, 2
r5.2xlarge	8	4	2	2, 4	1, 2
r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5.16xlarge	64	32	2	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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
r5d.xlarge	4	2	2	2	1, 2
r5d.2xlarge	8	4	2	2, 4	1, 2
r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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
u-3tb1.56xlarge	224	112	2	4, 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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
u-6tb1.56 xlarge	224	224	1	8, 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
u-6tb1.11 2xlarge	448	224	2	8, 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-9tb1.11 2xlarge	448	224	2	8, 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-12tb1.1 12xlarge	448	224	2	8, 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-18tb1.1 12xlarge	448	224	2	8, 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-24tb1.1 12xlarge	448	224	2	8, 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
x1.16xlarge	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
x1.32xlarge	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.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x2iedn.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
x2iedn.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
x2iezn.2xlarge	8	4	2	2, 4	1, 2
x2iezn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
x2iezn.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
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
x1e.xlarge	4	2	2	1, 2	1, 2
x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
d3en.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
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
i3.xlarge	4	2	2	1, 2	1, 2
i3.2xlarge	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
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
i3en.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
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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
d11.24xlarge	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
d12q.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
g3.4xlarge	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
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
g4dn.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
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
gr6.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
gr6.8xlarge	32	16	2	1, 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
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
inf2.48xlarge	192	96	2	4, 8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
p2.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
p2.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
p2.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
p3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
p3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
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
p5.48xlarge	192	96	2	12, 24, 36, 48, 60, 72, 84, 96	1, 2
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
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 your instance

You can specify CPU options during instance launch.

The following examples describe how to specify the CPU options when using the launch instance wizard in the EC2 console and the [run-instances](#) AWS CLI command, and the create launch template page in the EC2 console and the [create-launch-template](#) AWS CLI command. For EC2 Fleet or Spot Fleet, you must specify the CPU options in a launch template.

The following examples are for an `r5.4xlarge` instance type, which has the following [default values](#):

- Default CPU cores: 8
- Default threads per core: 2
- Default vCPUs: 16 (8 * 2)
- Valid number of CPU cores: 2, 4, 6, 8
- Valid number of threads per core: 1, 2

Disable multithreading

To disable multithreading, specify 1 thread per core.

New console

To disable multithreading during instance launch

1. Follow the [Quickly launch an instance](#) procedure and configure your instance as needed.
2. Expand **Advanced details**, and select the **Specify CPU options** check box.
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 multithreading, 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 instance using the new launch instance wizard](#).

Old console

To disable multithreading during instance launch

1. Follow the [Launch an instance using the old launch instance wizard](#) procedure.
2. On the **Configure Instance Details** page, for **CPU options**, choose **Specify CPU options**.
3. For **Core count**, choose the number of required CPU cores. In this example, to specify the default CPU core count for an `r5.4xlarge` instance, choose 8.
4. To disable multithreading, for **Threads per core**, choose **1**.
5. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launch an instance using the old launch instance wizard](#).

AWS CLI

To disable multithreading 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 `r5.4xlarge` instance, specify a value of 8.

```
aws ec2 run-instances \  
  --image-id ami-1a2b3c4d \  
  --instance-type r5.4xlarge \  
  --cpu-options "CoreCount=8,ThreadsPerCore=1" \  
  --key-name MyKeyPair
```

Specify a custom number of vCPUs at launch

You can customize the number of CPU cores and threads per core for the instance.

The following example launches an `r5.4xlarge` instance with 4 vCPUs.

New console

To specify a custom number of vCPUs during instance launch

1. Follow the [Quickly launch an instance](#) procedure and configure your instance as needed.
2. Expand **Advanced details**, and select the **Specify CPU options** check box.
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 instance using the new launch instance wizard](#).

Old console

To specify a custom number of vCPUs during instance launch

1. Follow the [Launch an instance using the old launch instance wizard](#) procedure.
2. On the **Configure Instance Details** page, for **CPU options**, choose **Specify CPU options**.
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. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launch an instance using the old launch instance wizard](#).

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-1a2b3c4d \  
  --instance-type r5.4xlarge \  
  --cpu-options "CoreCount=2,ThreadsPerCore=2" \  
  --key-name MyKeyPair
```

Alternatively, specify 4 CPU cores and 1 thread per core (disable multithreading) to get 4 vCPUs:

```
aws ec2 run-instances \  
  --image-id ami-1a2b3c4d \  
  --instance-type r5.4xlarge \  
  --cpu-options "CoreCount=4,ThreadsPerCore=1" \  
  --key-name MyKeyPair
```

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 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 from parameters](#) procedure and configure your launch template as needed.

2. Expand **Advanced details**, and select the **Specify CPU options** check box.
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 [Launch an instance from a launch template](#).

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-7b16de0c"  
  }],  
  "ImageId": "ami-8c1be5f6",  
  "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 multithreading) to get 4 vCPUs:

```
{
  "NetworkInterfaces": [{
    "AssociatePublicIpAddress": true,
    "DeviceIndex": 0,
    "Ipv6AddressCount": 1,
    "SubnetId": "subnet-7b16de0c"
  }],
  "ImageId": "ami-8c1be5f6",
  "InstanceType": "r5.4xlarge",
  "TagSpecifications": [{
    "ResourceType": "instance",
    "Tags": [{
      "Key": "Name",
      "Value": "webserver"
    }]
  }],
  "CpuOptions": {
    "CoreCount":4,
    "ThreadsPerCore":1
  }
}
```

View the CPU options for your instance

You can view the CPU options for an existing instance in the Amazon EC2 console or by describing the instance using the AWS CLI.

Console

To view the CPU options for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances** and select the instance.

3. On the **Details** tab, under **Host and placement group**, find **Number of vCPUs**.

AWS CLI

To view the CPU options for an instance (AWS CLI)

Use the [describe-instances](#) command.

```
aws ec2 describe-instances --instance-ids i-123456789abcde123
```

```
...
  "Instances": [
    {
      "Monitoring": {
        "State": "disabled"
      },
      "PublicDnsName": "ec2-198-51-100-5.eu-central-1.compute.amazonaws.com",
      "State": {
        "Code": 16,
        "Name": "running"
      },
      "EbsOptimized": false,
      "LaunchTime": "2018-05-08T13:40:33.000Z",
      "PublicIpAddress": "198.51.100.5",
      "PrivateIpAddress": "172.31.2.206",
      "ProductCodes": [],
      "VpcId": "vpc-1a2b3c4d",
      "CpuOptions": {
        "CoreCount": 34,
        "ThreadsPerCore": 1
      },
      "StateTransitionReason": "",
      ...
    }
  ]
...
```

In the output that's returned, the `CoreCount` field indicates the number of cores for the instance. The `ThreadsPerCore` field indicates the number of threads per core.

Alternatively, connect to your instance and use a tool such as **lscpu** to view the CPU information for your instance.

You can use AWS Config to record, assess, audit, and evaluate configuration changes for instances, including terminated instances. For more information, see [Getting Started with AWS Config](#) in the *AWS Config Developer Guide*.

CPU features

Amazon EC2 instances support the following CPU features. Support varies by instance type. Refer to the documentation for each CPU feature for more information about the supported instance types.

Topics

- [AMD SEV-SNP](#)

AMD SEV-SNP

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 [Attestation 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.

Pricing

When you launch an Amazon EC2 instance with AMD SEV-SNP turned on, 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](#) turned on, 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.

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 turn on AMD SEV-SNP when you launch an instance. When AMD SEV-SNP is turned on for your instance launch, the following rules apply.

- AMD SEV-SNP can't be turned off. It remains on 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.

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](#).

Working with AMD SEV-SNP

Topics

- [Find supported instance types](#)
- [Turn on AMD SEV-SNP at launch](#)
- [Check AMD SEV-SNP status](#)

Find supported instance types

You can use the AWS CLI to find instance types that support AMD SEV-SNP.

To find the instance types that support AMD SEV-SNP using the AWS CLI, 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'
```

Example output.

```
[  
  "r6a.2xlarge",  
  "m6a.large",  
  "m6a.2xlarge",  
  "r6a.xlarge",  
  "c6a.16xlarge",  
  "c6a.8xlarge",  
  "m6a.4xlarge",  
  "c6a.12xlarge",  
  "r6a.4xlarge",  
  "c6a.xlarge",  
  "c6a.4xlarge",  
  "c6a.2xlarge",  
  "m6a.xlarge",  
  "c6a.large",  
  "r6a.large",  
  "m6a.8xlarge"  
]
```

Turn on AMD SEV-SNP at launch

You can use the AWS CLI to launch an instance with AMD SEV-SNP turned on.

To launch an instance with AMD SEV-SNP turned on using the AWS CLI, use the [run-instances](#) command and include the `--cpu-options AmdSevSnp=enabled` option. For `--image-id`, specify an AMI with the `uefi` or `uefi-preferred` boot mode and an operating system that supports AMD SEV-SNP. For `--instance-type`, specify a supported instance type.

```
$ aws ec2 run-instances \  
--image-id supported_ami_id \  
--instance-type supported_instance_type \  
--key-name key_pair_name \  
--subnet-id subnet_id \  
--cpu-options AmdSevSnp=enabled
```

Check AMD SEV-SNP status

You can use one of the following methods to find instance types that support AMD SEV-SNP.

AWS CLI

To check whether AMD SEV-SNP is turned on for an instance using the AWS CLI, use the [describe-instances](#) command. For `--instance-ids`, specify the ID of the instance to check.

```
$ aws ec2 describe-instances --instance-ids instance_id
```

In the command output, the value for `AmdSevSnp` in `CpuOptions` indicates whether AMD SEV-SNP is turned on or off.

AWS CloudTrail

In the AWS CloudTrail event for the instance launch request, a value of `"cpuOptions": {"AmdSevSnp": enabled}` indicates that AMD SEV-SNP is turned on for the instance.

Attestation with AMD SEV-SNP

Attestation is a process that allows your instance to prove its state and identity. When you turn on 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.

To perform attestation with AMD SEV-SNP, complete the following steps.

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. 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
```


2. 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
```

3. 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
```

Expected output:

```
certs/vcek.pem: OK
```

3. Use the `snpguest` utility to validate that the attestation report is signed by the VLEK certificate.

```
$ ./snpguest verify attestation ./ report.bin
```

Expected 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!
```

Change the hostname of your Amazon Linux instance

When you launch an instance into a private VPC, Amazon EC2 assigns a guest OS hostname. The type of hostname that Amazon EC2 assigns depends on your subnet settings. For more information about EC2 hostnames, see [Amazon EC2 instance hostname types](#).

A typical Amazon EC2 private DNS name for an EC2 instance configured to use IP-based naming with an IPv4 address looks something like this: `ip-12-34-56-78.us-west-2.compute.internal`, where the name consists of the internal domain, the service (in this case, `compute`), the region, and a form of the private IPv4 address. Part of this hostname is displayed at the shell prompt when you log into your instance (for example, `ip-12-34-56-78`). Each time you stop and restart your Amazon EC2 instance (unless you are using an Elastic IP address), the public IPv4 address changes, and so does your public DNS name, system hostname, and shell prompt.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

Change the system hostname

If you have a public DNS name registered for the IP address of your instance (such as `webserver.mydomain.com`), you can set the system hostname so your instance identifies itself as a part of that domain. This also changes the shell prompt so that it displays the first portion of this name instead of the hostname supplied by AWS (for example, `ip-12-34-56-78`). If you do not

have a public DNS name registered, you can still change the hostname, but the process is a little different.

In order for your hostname update to persist, you must verify that the `preserve_hostname` cloud-init setting is set to `true`. You can run the following command to edit or add this setting:

```
sudo vi /etc/cloud/cloud.cfg
```

If the `preserve_hostname` setting is not listed, add the following line of text to the end of the file:

```
preserve_hostname: true
```

To change the system hostname to a public DNS name

Follow this procedure if you already have a public DNS name registered.

1. • For Amazon Linux 2: Use the `hostnamectl` command to set your hostname to reflect the fully qualified domain name (such as `webserver.mydomain.com`).

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.mydomain.com
```

- For Amazon Linux AMI: On your instance, open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the fully qualified domain name (such as `webserver.mydomain.com`).

```
HOSTNAME=webserver.mydomain.com
```

2. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

Alternatively, you can reboot using the Amazon EC2 console (on the **Instances** page, select the instance and choose **Instance state**, **Reboot instance**).

3. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the `hostname` command should show the fully-qualified domain name.

```
[ec2-user@webserver ~]$ hostname
```

```
webserver.mydomain.com
```

To change the system hostname without a public DNS name

1. • For Amazon Linux 2: Use the `hostnamectl` command to set your hostname to reflect the desired system hostname (such as `webserver`).

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.localdomain
```

- For Amazon Linux AMI: On your instance, open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the desired system hostname (such as `webserver`).

```
HOSTNAME=webserver.localdomain
```

2. Open the `/etc/hosts` file in your favorite text editor and change the entry beginning with `127.0.0.1` to match the example below, substituting your own hostname.

```
127.0.0.1 webserver.localdomain webserver localhost4 localhost4.localdomain4
```

3. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

Alternatively, you can reboot using the Amazon EC2 console (on the **Instances** page, select the instance and choose **Instance state, Reboot instance**).

4. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the `hostname` command should show the fully-qualified domain name.

```
[ec2-user@webserver ~]$ hostname  
webserver.localdomain
```

You can also implement more programmatic solutions, such as specifying user data to configure your instance. If your instance is part of an Auto Scaling group, you can use lifecycle hooks to define user data. For more information, see [Run commands on your Linux instance at launch](#) and [Lifecycle hook for instance launch](#) in the *AWS CloudFormation User Guide*.

Change the shell prompt without affecting the hostname

If you do not want to modify the hostname for your instance, but you would like to have a more useful system name (such as **webserver**) displayed than the private name supplied by AWS (for example, `ip-12-34-56-78`), you can edit the shell prompt configuration files to display your system nickname instead of the hostname.

To change the shell prompt to a host nickname

1. Create a file in `/etc/profile.d` that sets the environment variable called `NICKNAME` to the value you want in the shell prompt. For example, to set the system nickname to **webserver**, run the following command.

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/  
prompt.sh'
```

2. Open the `/etc/bashrc` (Red Hat) or `/etc/bash.bashrc` (Debian/Ubuntu) file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/bashrc` and `/etc/bash.bashrc` are owned by `root`.
3. Edit the file and change the shell prompt variable (`PS1`) to display your nickname instead of the hostname. Find the following line that sets the shell prompt in `/etc/bashrc` or `/etc/bash.bashrc` (several surrounding lines are shown below for context; look for the line that starts with `["$PS1"`):

```
# Turn on checkwinsize  
shopt -s checkwinsize  
[ "$PS1" = "\\s-\\v\\\\"$ " ] && PS1="[\\u@\\h \\W]\\\\"$ "  
# You might want to have e.g. tty in prompt (e.g. more virtual machines)  
# and console windows
```

Change the `\\h` (the symbol for hostname) in that line to the value of the `NICKNAME` variable.

```
# Turn on checkwinsize  
shopt -s checkwinsize  
[ "$PS1" = "\\s-\\v\\\\"$ " ] && PS1="[\\u@$NICKNAME \\W]\\\\"$ "  
# You might want to have e.g. tty in prompt (e.g. more virtual machines)  
# and console windows
```

4. (Optional) To set the title on shell windows to the new nickname, complete the following steps.

- a. Create a file named `/etc/sysconfig/bash-prompt-xterm`.

```
[ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
```

- b. Make the file executable using the following command.

```
[ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
```

- c. Open the `/etc/sysconfig/bash-prompt-xterm` file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/sysconfig/bash-prompt-xterm` is owned by `root`.
- d. Add the following line to the file.

```
echo -ne "\033]0;${USER}@${NICKNAME}:${PWD/#$HOME/~}\007"
```

5. Log out and then log back in to pick up the new nickname value.

Change the hostname on other Linux distributions

The procedures on this page are intended for use with Amazon Linux only. For more information about other Linux distributions, see their specific documentation and the following articles:

- [How do I assign a static hostname to a private Amazon EC2 instance running RHEL 7 or Centos 7?](#)

Set up dynamic DNS on your Amazon Linux instance

When you launch an EC2 instance, it is assigned a public IP address and a public Domain Name System (DNS) name that you can use to reach it from the internet. Because there are so many hosts in the Amazon Web Services domain, these public names must be quite long for each name to remain unique. A typical Amazon EC2 public DNS name looks something like this: `ec2-12-34-56-78.us-west-2.compute.amazonaws.com`, where the name consists of the Amazon Web Services domain, the service (in this case, `compute`), the AWS Region, and a form of the public IP address.

Dynamic DNS services provide custom DNS host names within their domain area that can be easy to remember and that can also be more relevant to your host's use case. Some of these services are also free of charge. You can use a dynamic DNS provider with Amazon EC2 and configure the instance to update the IP address associated with a public DNS name each time the instance starts.

There are many different providers to choose from, and the specific details of choosing a provider and registering a name with them are outside the scope of this guide.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To use dynamic DNS with Amazon EC2

1. Sign up with a dynamic DNS service provider and register a public DNS name with their service. This procedure uses the free service from noip.com/free as an example.
2. Configure the dynamic DNS update client. After you have a dynamic DNS service provider and a public DNS name registered with their service, point the DNS name to the IP address for your instance. Many providers (including noip.com) allow you to do this manually from your account page on their website, but many also support software update clients. If an update client is running on your EC2 instance, your dynamic DNS record is updated each time the IP address changes, as happens after a shutdown and restart. In this example, you install the `noip2` client, which works with the service provided by noip.com.
 - a. Enable the Extra Packages for Enterprise Linux (EPEL) repository to gain access to the `noip2` client.

Note

Amazon Linux instances have the GPG keys and repository information for the EPEL repository installed by default; however, Red Hat and CentOS instances must first install the `epel-release` package before you can enable the EPEL repository. For more information and to download the latest version of this package, see <https://fedoraproject.org/wiki/EPEL>.

- For Amazon Linux 2

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- For Amazon Linux AMI

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

- b. Install the noip package.

```
[ec2-user ~]$ sudo yum install -y noip
```

- c. Create the configuration file. Enter the login and password information when prompted and answer the subsequent questions to configure the client.

```
[ec2-user ~]$ sudo noip2 -C
```

3. Enable the noip service.

- For Amazon Linux 2

```
[ec2-user ~]$ sudo systemctl enable noip.service
```

- For Amazon Linux AMI

```
[ec2-user ~]$ sudo chkconfig noip on
```

4. Start the noip service.

- For Amazon Linux 2

```
[ec2-user ~]$ sudo systemctl start noip.service
```

- For Amazon Linux AMI

```
[ec2-user ~]$ sudo service noip start
```

This command starts the client, which reads the configuration file (`/etc/no-ip2.conf`) that you created earlier and updates the IP address for the public DNS name that you chose.

5. Verify that the update client has set the correct IP address for your dynamic DNS name. Allow a few minutes for the DNS records to update, and then try to connect to your instance using SSH with the public DNS name that you configured in this procedure.

Run commands on your Linux instance at launch

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch instance wizard as plain text, as a file (this is useful for launching instances using the command line tools), or as base64-encoded text (for API calls).

If you're interested in more complex automation scenarios, you might consider AWS CloudFormation or AWS OpsWorks. For more information, see the following:

- [Deploying applications on Amazon EC2 with AWS CloudFormation](#) in the *AWS CloudFormation User Guide*.
- [AWS OpsWorks User Guide](#).

For information about running commands on your Windows instance at launch, see [Run commands on your Windows instance at launch](#) and [Manage Windows instance configuration](#) in the *Amazon EC2 User Guide for Windows Instances*.

In the following examples, the commands from the [Install a LAMP Web Server on Amazon Linux 2](#) are converted to a shell script and a set of cloud-init directives that run when the instance launches. In each example, the following tasks are performed by the user data:

- The distribution software packages are updated.
- The necessary web server, php, and mariadb packages are installed.
- The httpd service is started and turned on via **systemctl**.
- The `ec2-user` is added to the apache group.
- The appropriate ownership and file permissions are set for the web directory and the files contained within it.
- A simple web page is created to test the web server and PHP engine.

Contents

- [Prerequisites](#)
- [User data and shell scripts](#)
- [User data and the console](#)

- [User data and cloud-init directives](#)
- [User data and the AWS CLI](#)
- [Combine shell scripts and cloud-init directives](#)

Prerequisites

The examples in this topic assume the following:

- Your instance has a public DNS name that is reachable from the internet. For more information, see **Auto-assign Public IP** in the [Network settings](#) section and [Create a security group](#).
- 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. For more information, see [Create a security group](#).
- Your instance is launched with an Amazon Linux 2 AMI. These instructions are intended for use with Amazon Linux 2, and the commands and directives may not work for other Linux distributions. For more information about other distributions, such as their support for cloud-init, see their specific documentation.

User data and shell scripts

If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch. Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.

Important

By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance. For more information, see [How can I 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 [Using instance profiles](#) in the IAM User Guide. The permissions you assign to the IAM role depend on which services you are calling with the API. For more information, see [IAM roles for Amazon EC2](#).

The cloud-init output log file 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.

User data and the console

You can specify instance user data when you launch the instance. If the root volume of the instance is an EBS volume, you can also stop the instance and update its user data.

Specify instance user data at launch

Follow the procedure for [launching an instance](#). The **User data** field is located in the [Advanced details](#) section of the launch instance wizard. Enter your shell script in the **User data** field, and then complete the instance launch procedure.

In the example script below, the script creates and configures our web server.

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
```

```
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Allow enough time for the instance to launch and run the commands in your script, and then check to see that your script has completed the tasks that you intended.

For our example, in a web browser, enter the URL of the PHP test file the script created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page. If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For more information, see [Add rules to a security group](#).

(Optional) If your script did not accomplish the tasks you were expecting it to, or if you just want to verify that your script completed without errors, [connect to the instance](#), examine the cloud-init output log file (`/var/log/cloud-init-output.log`), and look for error messages in the output.

For additional debugging information, you can create a Mime multipart archive that includes a cloud-init data section with the following directive:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends command output from your script to `/var/log/cloud-init-output.log`. For more information about cloud-init data formats and creating Mime multi part archive, see [cloud-init Formats](#).

View and update the instance user data

To 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 [cloud-init](#).

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 <http://cloudinit.readthedocs.org/en/latest/index.html>.

⚠ Important

By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance. For more

information, see [How can I 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 instance with user data

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 example below, the directives create and configure a web server on Amazon Linux 2. The `#cloud-config` line at the top is required in order to identify the commands as cloud-init directives.

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
- systemctl start httpd
- sudo systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. Allow enough time for the instance to launch and 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 [Add rules to a security group](#).

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 **runcommand** output to `/var/log/cloud-init-output.log`.

User data and the AWS CLI

You can use the AWS CLI to specify, modify, and view the user data for your instance. For information about viewing user data from your instance using instance metadata, see [Retrieve instance user data from your 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](#) in the *Amazon EC2 User Guide for Windows Instances*.

Example: Specify user data at launch

To specify user data when you launch your instance, use the [run-instances](#) command with the `--user-data` parameter. With **run-instances**, the AWS CLI performs base64 encoding of the user data for you.

The following example shows how to specify a script as a string on the command line:

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
```

```
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \  
--user-data echo user data
```

The following example shows how to specify a script using a text file. Be sure to use the `file://` prefix to specify the file.

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \  
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \  
--user-data file://my_script.txt
```

The following is an example text file with a shell script.

```
#!/bin/bash  
yum update -y  
service httpd start  
chkconfig httpd on
```

Example: Modify the user data of a stopped instance

You can modify the user data of a stopped instance using the [modify-instance-attribute](#) command. With **modify-instance-attribute**, the AWS CLI does not perform base64 encoding of the user data for you.

- On a **Linux** computer, use the `base64` command to encode the user data.

```
base64 my_script.txt >my_script_base64.txt
```

- On a **Windows** computer, use the `certutil` command to encode the user data. Before you can use this file with the AWS CLI, you must remove the first (BEGIN CERTIFICATE) and last (END CERTIFICATE) lines.

```
certutil -encode my_script.txt my_script_base64.txt  
notepad my_script_base64.txt
```

Use the `--attribute` and `--value` parameters to use the encoded text file to specify the user data. Be sure to use the `file://` prefix to specify the file.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute  
userData --value file://my_script_base64.txt
```


Example: Clear the user data of a stopped instance

To delete the existing user data, use the [modify-instance-attribute](#) command as follows:

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --user-data Value=
```

Example: View user data

To retrieve the user data for an instance, use the [describe-instance-attribute](#) command. With **describe-instance-attribute**, the AWS CLI does not perform base64 decoding of the user data for you.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute  
userData
```

The following is example output with the user data base64 encoded.

```
{  
  "UserData": {  
    "Value":  
    "IyEvYm1uL2Jhc2gKeXVtIHVwZGF0ZSAteQpzZXJ2aWNlIGh0dHBkIHNoYXJ0cmNoa2NvbWZpZyBodHRwZCBvbG=="  
  },  
  "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
--//--
```

Instance metadata and user data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into [categories](#), for example, host name, events, and security groups.

You can also use instance metadata to access *user data* that you specified when launching your instance. For example, you can specify parameters for configuring your instance, or include a simple script. You can build generic AMIs and use user data to modify the configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same generic AMI and retrieve their content from the Amazon S3 bucket that you specify in the user data at launch. To add a new customer at any time, create a bucket for the customer, add their content, and launch your AMI with the unique bucket name provided to your

code in the user data. If you launch 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 [Example: AMI launch index value](#).

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched. For more information, see [Dynamic data categories](#).

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by authentication or cryptographic methods. Anyone who has direct access to the instance, and potentially any software running on the instance, can view its metadata. Therefore, you should not store sensitive data, such as passwords or long-lived encryption keys, as user data.

Note

The examples in this topic 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 [instances built on the AWS Nitro System](#).

Contents

- [Use IMDSv2](#)
- [Configure the instance metadata options](#)
- [Retrieve instance metadata](#)
- [Work with instance user data](#)
- [Retrieve dynamic data](#)
- [Instance metadata categories](#)
- [Example: AMI launch index value](#)

- [Instance identity documents](#)
- [Instance identity roles](#)

Use IMDSv2

You can access instance metadata from a running instance using one of the following methods:

- Instance Metadata Service Version 1 (IMDSv1) – a request/response method
- Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method

By default, you can use either IMDSv1 or IMDSv2, or both.

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 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](#).

To retrieve instance metadata, see [Retrieve instance metadata](#).

Topics

- [How Instance Metadata Service Version 2 works](#)
- [Transition to using Instance Metadata Service Version 2](#)
- [Use a supported AWS SDK](#)

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 [instances built on the AWS Nitro System](#).

The following example uses a Linux shell script and IMDSv2 to retrieve the top-level instance metadata items. The example:

- Creates a session token lasting six hours (21,600 seconds) using the PUT request
- Stores the session token header in a variable named TOKEN
- Requests the top-level metadata items using the token

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
```

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](#).

Transition to using Instance Metadata Service Version 2

When migrating to 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, updating, and uninstalling 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 [IMDS Packet Analyzer](#) 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 information about updating the CLI, see [Upgrading to the latest version of the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Then, change your software that directly accesses instance metadata (in other words, that does not use an SDK) using the IMDSv2 requests. 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](#) CLI 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](#) CLI 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.

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 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 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

Configure the instance metadata 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 [instances built on the AWS Nitro System](#).

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, we recommend setting the hop limit to 2. For more information, see [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 [Work with instance tags in instance metadata](#).

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
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

Launch parameter	Account level default	AMI default	Resulting instance configuration
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.

Configure instance metadata options for new instances

You can configure the following instance metadata options for new instances.

Options

- [Require the use of IMDSv2](#)
- [Configure IPv4 and IPv6 endpoints](#)
- [Turn off access to instance metadata](#)

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 requires IMDSv2; 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 requires IMDSv2; otherwise it defaults to `1`.

```
aws ec2 modify-instance-metadata-defaults \  
  --region us-east-1 \  
  --http-tokens required \  
  --http-put-response-hop-limit 2
```

Expected 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
```

Example output

```
{  
  "AccountLevel": {  
    "HttpTokens": "required",  
    "HttpPutResponseHopLimit": 2  
  }  
}
```

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).

New 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)**.

For more information, see [Advanced details](#).

Old console

To require the use of IMDSv2 on a new instance

- When launching a new instance in the Amazon EC2 console, select the following options on the **Configure Instance Details** page:
 - Under **Advanced Details**, for **Metadata accessible**, select **Enabled**.
 - For **Metadata version**, select **V2 (token required)**.

For more information, see [Step 3: Configure Instance Details](#).

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.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type c6i.large \  
  ...  
  --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

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.

```
Import-Module AWS.Tools.EC2 # Required for Amazon.EC2.Model object creation.  
Register-EC2Image `br/>  -Name 'my-image' `br/>  -RootDeviceName /dev/xvda `br/>  -BlockDeviceMapping (  
New-Object `br/>  -TypeName Amazon.EC2.Model.BlockDeviceMapping `br/>  -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-0123456789example \
  --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](#).

Configure IPv4 and IPv6 endpoints

By default, the IPv6 endpoint is disabled. This is true even if you are launching an instance into an IPv6-only subnet. You can choose to enable the IPv6 endpoint when you launch an instance.

The IPv6 endpoint for the IMDS is only accessible on [instances built on the AWS Nitro System](#).

Use one of the following methods to launch an instance with the IPv6 endpoint enabled for IMDS.

New console

To enable the IMDS IPv6 endpoint at launch

- [Launch the instance](#) in the Amazon EC2 console with the following specified under **Advanced details**:
 - For **Metadata transport**, choose **Enabled**.

For more information, see [Advanced details](#).

AWS CLI

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

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.

New 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](#).

Old console

To turn off access to instance metadata at launch

- Launch the instance in the Amazon EC2 console with the following option selected on the **Configure Instance Details** page:
 - Under **Advanced Details**, for **Metadata accessible**, select **Disabled**.

For more information, see [Step 3: Configure Instance Details](#).

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*.

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](#).

Query instance metadata options for existing instances

You can query the instance metadata options for your existing instances by using one of the following methods.

Console

To query the instance metadata options for an existing instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance.
4. Choose **Actions, Instance settings, Modify instance metadata options**.
5. Review the current instance metadata options in the **Modify instance metadata options** dialog box.

AWS CLI

To query the instance metadata options for an existing instance using the AWS CLI

Use the [describe-instances](#) CLI 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
```

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-1234567898abcdef0 \  
  --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-1234567898abcdef0 \  
  -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-1234567898abcdef0 \  
  --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-1234567898abcdef0 \  
  -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-1234567898abcdef0 \  
  --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-1234567898abcdef0 \  
  -HttpPutResponseHopLimit 3 \  
  -HttpEndpoint enabled).InstanceMetadataOptions
```

Enable the IPv6 endpoint for your instance

By default, the IPv6 endpoint is disabled. This is true even if you have launched an instance into an IPv6-only subnet. The IPv6 endpoint for the IMDS is only accessible on [instances built on the AWS Nitro System](#).

Currently only the AWS CLI and AWS SDKs support enabling the IPv6 endpoint for your instance.

AWS CLI

To enable the 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-1234567898abcdef0 \  
  --http-protocol-ipv6 enabled \  
  --http-endpoint enabled
```

```
--http-endpoint enabled
```

PowerShell

To enable the 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-1234567898abcdef0 `
  -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-1234567898abcdef0 \  
  --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-1234567898abcdef0 \  
  -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-1234567898abcdef0 \  
  --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-1234567898abcdef0 \  
  -HttpEndpoint disabled).InstanceMetadataOptions
```

Retrieve instance metadata

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

Instance metadata is divided into categories. For a description of each instance metadata category, see [Instance metadata categories](#).

To view all categories of instance metadata from within a running instance, use the following IPv4 or IPv6 URIs.

IPv4

```
http://169.254.169.254/latest/meta-data/
```

IPv6

```
http://[fd00:ec2::254]/latest/meta-data/
```

The IP addresses are link-local addresses and are valid only from the instance. For more information, see [Link-local addresses](#) in this user guide and [Link-local address](#) on Wikipedia.

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 [instances built on the AWS Nitro System](#).

The command format is different, depending on whether you use IMDSv1 or IMDSv2. By default, you can use both versions of the IMDS. To require the use of IMDSv2, see [Use IMDSv2](#).

You can use a tool such as cURL, as shown in the following example.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

For the command to retrieve instance metadata from a Windows instance, see [Retrieve instance metadata](#) in the *Amazon EC2 User Guide for Windows Instances*.

Costs

You are not billed for HTTP requests used to retrieve instance metadata and user data.

Considerations

To avoid problems with instance metadata retrieval, consider the following:

- **In a container environment, we recommend setting the hop limit to 2.**

The AWS SDKs use IMDSv2 calls by default. If the IMDSv2 call receives no response, the SDK retries the call and, if still unsuccessful, uses IMDSv1. This can result in a delay, especially in a container environment. In a container environment, if the hop limit is 1, the IMDSv2 response does not return because going to the container is considered an additional network hop. To avoid the process of falling back to IMDSv1 and the resultant delay, in a container environment we recommend that you set the hop limit to 2. For more information, see [Configure the instance metadata options](#).

- **For IMDSv2, you must use `/latest/api/token` when retrieving the token.**

Issuing PUT requests to any version-specific path, for example `/2021-03-23/api/token`, will result in the metadata service returning 403 Forbidden errors. This behavior is intended.

- **If IMDSv2 is required, IMDSv1 does not work.**

You can check if IMDSv2 is required for an instance, as follows: Select the instance to view its details, and check the value for **IMDSv2**. The value is either **Required** (only IMDSv2 can be used) or **Optional** (IMDSv2 and IMDSv1 can be used).

Responses and error messages

All instance metadata is returned as text (HTTP content type `text/plain`).

A request for a specific metadata resource returns the appropriate value, or a `404 - Not Found` HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a `/`) returns a list of available resources, or a `404 - Not Found` HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

For requests made using Instance Metadata Service Version 2, the following HTTP error codes can be returned:

- `400 - Missing or Invalid Parameters` – The PUT request is not valid.
- `401 - Unauthorized` – The GET request uses an invalid token. The recommended action is to generate a new token.
- `403 - Forbidden` – The request is not allowed or the IMDS is turned off.

Examples of retrieving instance metadata

The following examples provide commands that you can use on a Linux instance. For the commands to retrieve instance metadata from a Windows instance, see [Retrieve instance metadata](#) in the *Amazon EC2 User Guide for Windows Instances*.

Examples

- [Get the available versions of the instance metadata](#)
- [Get the top-level 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.

Note

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. For example, use `latest` as follows:

```
curl http://169.254.169.254/latest/meta-data/ami-id
```

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/  
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
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

Get the top-level metadata items

This example gets the top-level metadata items. For more information, see [Instance metadata 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/meta-
data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
```

```
events/  
hostname  
iam/  
instance-action  
instance-id  
instance-type  
local-hostname  
local-ipv4  
mac  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
services/
```

The following examples get the values of some of the top-level metadata items that were obtained in the preceding example. The IMDSv2 requests use the stored token that was created in the preceding example command, assuming it has not expired.

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/  
latest/meta-data/ami-id  
ami-0abcdef1234567890
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id  
ami-0abcdef1234567890
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/  
latest/meta-data/reservation-id  
r-0efghijk987654321
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id  
r-0efghijk987654321
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/  
latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/  
latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

Get the list of available public keys

This example gets the list of available public keys.

IMDSv2

```
[ec2-user ~]$ 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
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/  
0=my-public-key
```

Show the formats in which public key 0 is available

This example shows the formats in which public key 0 is available.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-  
aws-ec2-metadata-token-ttl-seconds: 21600" ` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-  
data/public-keys/0/  
openssh-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/  
openssh-key
```

Get public key 0 (in the OpenSSH key format)

This example gets public key 0 (in the OpenSSH key format).

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key
ssh-rsa MIICiTCcAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCcAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

Get the subnet ID for an instance

This example gets the subnet ID for an instance.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

Get the instance tags for an instance

In the following examples, the sample instance has [tags on instance metadata enabled](#) and the instance tags Name=MyInstance and Environment=Dev.

This example gets all the instance tag keys for an instance.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance
Name
Environment
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/tags/instance
Name
Environment
```

The following example gets the value of the Name key that was obtained in the preceding example. The IMDSv2 request uses the stored token that was created in the preceding example command, assuming it has not expired.

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/  
latest/meta-data/tags/instance/Name  
MyInstance
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/tags/instance/Name  
MyInstance
```

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.

Limit IMDS access

You can consider using local firewall rules to disable access from some or all processes to the IMDS.

Note

For [instances built on the AWS Nitro System](#), 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 IMDS [fd00:ec2::254].

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.

Work with instance user data

You can use *instance user data* to customize your instances. When you launch an instance, you can store parameters or scripts as user data. Any scripts in user data are run when you launch the instance. You can view user data as an instance attribute. You can also view user data from your instance through the Instance Metadata Service (IMDS).

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 you or accept base64-encoded input.
- User data is limited to 16 KB, in raw form, before it is base64-encoded. The size of a string of length n after base64-encoding is $\text{ceil}(n/3)*4$.
- User data must be base64-decoded when you retrieve it. If you retrieve the data using instance metadata or the console, it's decoded for you automatically.
- If you stop an instance, modify its user data, and start the instance, the updated user data is not run when you 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.

Specify instance user data at launch

You can specify user data when you launch an instance. For console directions, see [Specify instance user data at launch](#). For an example that uses the AWS CLI, see [the section called "User data and the AWS CLI"](#).

Modify instance user data

You can modify user data for instances with an EBS root volume. The instance must be in the stopped state. For console directions, see [View and update the instance user data](#). For an example that uses the AWS CLI, see [modify-instance-attribute](#).

Retrieve instance user data from your instance

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 [instances built on the AWS Nitro System](#).

To retrieve user data from an instance, use the following URI.

```
http://169.254.169.254/latest/user-data
```

A request for user data returns the data as it is (content type application/octet-stream). If the instance does not have any user data, the request returns 404 - Not Found.

This example returns user data that was provided as comma-separated text.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-
data
1234, john, reboot, true | 4512, richard, | 173, , ,
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
1234, john, reboot, true | 4512, richard, | 173, , ,
```

This example returns user data that was provided as a script.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-
data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

Retrieve instance user data from your computer

You can retrieve user data for an instance from your own computer. For console directions, see [View and update the instance user data](#). For an example that uses the AWS CLI, see [User data and the AWS CLI](#).

Retrieve dynamic data

To retrieve dynamic data from within a running instance, use the following URI.

```
http://169.254.169.254/latest/dynamic/
```

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 [instances built on the AWS Nitro System](#).

This example shows how to retrieve the high-level instance identity categories.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/  
instance-identity/  
rsa2048  
pkcs7  
document  
signature  
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/  
rsa2048  
pkcs7  
document  
signature  
dsa2048
```

For more information about dynamic data and examples of how to retrieve it, see [Instance identity documents](#).

Instance metadata categories

Instance metadata is divided into categories. To retrieve instance metadata, 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 instances built on the [Nitro system](#), 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
<code>ami-id</code>	The AMI ID used to launch the instance.	1.0
<code>ami-launch-index</code>	If you launch multiple instances using the same <code>RunInstances</code> 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

Category	Description	Version when category was released
<code>ami-manifest-path</code>	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 <code>unknown</code> .	1.0
<code>ancestor-ami-ids</code>	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the AMI manifest file contained an <code>ancestor-amis</code> key.	2007-10-10
<code>autoscaling/target-lifecycle-state</code>	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: <code>Detached</code> <code>InService</code> <code>Standby</code> <code>Terminated</code> <code>Warmed:Hi</code> <code>Warmed:Ru</code> <code>Warmed:Stopped</code> <code>Warmed:Terminated</code> . See Retrieve the target lifecycle state through instance metadata in the <i>Amazon EC2 Auto Scaling User Guide</i> .	2021-07-15
<code>block-device-mapping/ami</code>	The virtual device that contains the root/boot file system.	2007-12-15

Category	Description	Version when category was released
block-device-mapping/ ebs <i>N</i>	The virtual devices associated with any Amazon EBS volumes. Amazon EBS volumes are only available in metadata if they were present at launch time or when the instance was last started. The <i>N</i> indicates the index of the Amazon EBS volume (such as ebs1 or ebs2).	2007-12-15
block-device-mapping/ ephemeral <i>N</i>	The virtual devices for any non-NVMe instance store volumes. The <i>N</i> indicates the index of each volume. The number of instance store volumes in the block device mapping might not match the actual number of instance store volumes for the instance. The instance type determines the number of instance store volumes that are available to an instance. If the number of instance store volumes in a block device mapping exceeds the number available to an instance, the additional instance store volumes are ignored.	2007-12-15
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

Category	Description	Version when category was released
block-device-mapping/ swap	The virtual devices associated with swap. Not always present.	2007-12-15
elastic-gpus/associations/ <i>elastic-gpu-id</i>	If there is an Elastic GPU attached to the instance, contains a JSON string with information about the Elastic GPU, including its ID and connection information.	2016-11-30
elastic-inference/associations/ <i>eia-id</i>	If there is an Elastic Inference accelerator attached to the instance, contains a JSON string with information about the Elastic Inference accelerator, including its ID and type.	2018-11-29
events/maintenance/ history	If there are completed or canceled maintenance events for the instance, contains a JSON string with information about the events. For more information, see To view event history about completed or canceled events.	2018-08-17
events/maintenance/ scheduled	If there are active maintenance events for the instance, contains a JSON string with information about the events. For more information, see View scheduled events.	2018-08-17

Category	Description	Version when category was released
events/recommendations/rebalance	<p>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:</p> <pre>{"noticeTime": "2020-11-05T08:22:00Z"}</pre> <p>This category is available only after the notification is emitted. For more information, see EC2 instance rebalance recommendations.</p>	2020-10-27
hostname	<p>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 Amazon EC2 instance hostname types.</p>	1.0

Category	Description	Version when category was released
iam/info	If there is an IAM role associated with the instance, contains information about the last time the instance profile was updated, including the instance's LastUpdated date, InstanceProfileArn, and InstanceProfileId. Otherwise, not present.	2012-01-12
iam/security-credentials/role-name	If there is an IAM role associated with the instance, <i>role-name</i> is the name of the role, and <i>role-name</i> contains the temporary security credentials associated with the role (for more information, see Retrieve security credentials from instance metadata). Otherwise, not present.	2012-01-12
identity-credentials/ec2/info	Information about the credentials in identity-credentials/ec2/security-credentials/ec2-instance .	2018-05-23

Category	Description	Version when category was released
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 .	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 Instance 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 Amazon EC2 instance hostname types .	2007-01-19

Category	Description	Version when category was released
<code>local-ipv4</code>	The private IPv4 address of the instance. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the <code>device-number</code> is 0). If this is an IPv6-only instance, this item is not set and results in an HTTP 404 response.	1.0
<code>mac</code>	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the <code>eth0</code> device (the device for which the <code>device-number</code> is 0).	2011-01-01
<code>metrics/vhostmd</code>	No longer available.	2011-05-01
<code>network/interfaces/macs/mac/device-number</code>	The unique device number associated with that interface. The device number corresponds to the device name; for example, a <code>device-number</code> of 2 is for the <code>eth2</code> device. This category corresponds to the <code>DeviceIndex</code> and <code>device-index</code> fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01
<code>network/interfaces/macs/mac/interface-id</code>	The ID of the network interface.	2011-01-01

Category	Description	Version when category was released
network/interfaces/ macs/mac/ipv4-as sociations/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-h ostname	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 Amazon EC2 instance hostname types .	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/ <i>mac</i> /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/mac/mac/owner-id	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
network/interfaces/mac/mac/public-hostname	The interface's public DNS (IPv4). This category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see DNS attributes for your VPC in the <i>Amazon VPC User Guide</i> . 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/mac/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/mac/mac/security-groups	Security groups to which the network interface belongs.	2011-01-01
network/interfaces/mac/mac/security-group-ids	The IDs of the security groups to which the network interface belongs.	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. The Availability Zone ID is consistent 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 <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see DNS attributes for your VPC in the <i>Amazon VPC User Guide</i> . 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	<p>The names of the security groups applied to the instance.</p> <p>After launch, you can change the security groups of the instances. Such changes are reflected here and in <code>network/interfaces/macs/<i>mac</i>/security-groups</code>.</p>	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 <code>aws</code> . If you have resources in other partitions, the partition is <code>aws-<i>partition name</i></code> . For example, the partition for resources in the China (Beijing) Region is <code>aws-cn</code> .	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 Allow 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 .	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance identity documents .	2009-04-04
instance-identity/signature	Data that can be used by other parties to verify its origin and authenticity. See Instance identity documents .	2009-04-04

Example: AMI launch index value

This example demonstrates how you can use both user data and instance metadata to configure your 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 [instances built on the AWS Nitro System](#).

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 certain unique metadata.

Instance 1

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

Instance 2

Metadata	Value
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

Instance 3

Metadata	Value
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com

Metadata	Value
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

Instance 4

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice can use the `ami-launch-index` value to determine which portion of the user data is applicable to a particular instance.

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
```

Instance identity documents

Each instance that you launch has an instance identity document that provides information about the instance itself. You can use the instance identity document to validate the attributes of the instance.

The instance identity document is generated when the instance is stopped and started, restarted, or launched. The instance identity document is exposed (in plaintext JSON format) through the Instance Metadata Service (IMDS). The IPv4 address `169.254.169.254` is a link-local address and is valid only from the instance. For more information, see [Link-local address](#) on Wikipedia. The IPv6 address `[fd00:ec2::254]` is a unique local address and is valid only from the instance. For more information, see [Unique local address](#) on Wikipedia.

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 [Instances built on the AWS Nitro System](#).

You can retrieve the instance identity document from a running instance at any time. The instance identity document includes the following information:

Data	Description
<code>accountId</code>	The ID of the AWS account that launched the instance.
<code>architecture</code>	The architecture of the AMI used to launch the instance (i386 x86_64 arm64).
<code>availabilityZone</code>	The Availability Zone in which the instance is running.
<code>billingProducts</code>	The billing products of the instance.
<code>devpayProductCodes</code>	Deprecated.

Data	Description
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 plaintext instance identity document

To retrieve the plaintext instance identity document

Connect to the instance and run one of the following commands depending on the version of the IMDS 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
```

IMDSv1

```
$ curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

The following is example output.

```
{
  "devpayProductCodes" : null,
  "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
  "availabilityZone" : "us-west-2b",
  "privateIp" : "10.158.112.84",
  "version" : "2017-09-30",
  "instanceId" : "i-1234567890abcdef0",
  "billingProducts" : null,
  "instanceType" : "t2.micro",
  "accountId" : "123456789012",
  "imageId" : "ami-5fb8c835",
  "pendingTime" : "2016-11-19T16:32:11Z",
  "architecture" : "x86_64",
  "kernelId" : null,
  "ramdiskId" : null,
  "region" : "us-west-2"
}
```

Verify the instance identity document

If you intend to use the contents of the instance identity document for an important purpose, you should verify its contents and authenticity before using it.

The plaintext instance identity document is accompanied by three hashed and encrypted signatures. You can use these signatures to verify the origin and authenticity of the instance identity document and the information that it includes. The following signatures are provided:

- Base64-encoded signature—This is a base64-encoded SHA256 hash of the instance identity document that is encrypted using an RSA key pair.
- PKCS7 signature—This is a SHA1 hash of the instance identity document that is encrypted using a DSA key pair.
- RSA-2048 signature—This is a SHA256 hash of the instance identity document that is encrypted using an RSA-2048 key pair.

Each signature is available at a different endpoint in the instance metadata. You can use any one of these signatures depending on your hashing and encryption requirements. To verify the signatures, you must use the corresponding AWS public certificate.

The following topics provide detailed steps for validating the instance identity document using each signature.

- [Use the PKCS7 signature to verify the instance identity document](#)
- [Use the base64-encoded signature to verify the instance identity document](#)
- [Use the RSA-2048 signature to verify the instance identity document](#)

Use the PKCS7 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate.

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](#) 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 AWS Support.

Use the base64-encoded signature to verify the instance identity document

This topic explains how to verify the instance identity document using the base64-encoded signature and the AWS RSA public certificate.

To validate the instance identity document using the base64-encoded signature and the AWS RSA public certificate

1. Connect to the instance.
2. Retrieve the base64-encoded signature from the instance metadata, convert it to 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](#) 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 AWS Support.

Use the RSA-2048 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate.

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](#) 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 AWS Support.

AWS public certificates

The following AWS public certificates can be used to verify the contents of an instance's instance identity document as described in the following topics:

- [Verify using the PKCS7 signature](#)
- [Verify using the base64-encoded signature](#)
- [Verify using the RSA-2048 signature](#)

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 base6-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 (Ohio) — `us-east-2`

DSA

```
-----BEGIN CERTIFICATE-----  
MIIC7TCCAqQCCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw  
FwYDVoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYD  
VQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z  
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVoQIEExBXYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYDVoQKEXdBbWF6b24gV2ViIFN1  
cnZpY2VzIEExMQzCCAbcwggEsBgqhkJ00AQBMIIbHwKBgQCjkvcS2bb1VQ4yt/5e
```

```
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLcLnd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNMp9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUUVJTC+h0U+8Gk3J1qsX438Dk5c58wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWVlU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTE3MTE0V0xDTI5MDQyODE3MTE0VowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWVlU2Vydm1jZXMgTEExDMIGFMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCChvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RwqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXk3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcwWdUUIzvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcwWdUUIzvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDQVQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQVQHEwdT
ZWf0dGx1MSAwHgYDQVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUUVJTC+h0U
+8Gk3J1qsX438Dk5c58wEgYDVR0TAQH/BAGwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQAyWJQaVNWJqW0R0T0xV0SoN1GLk9x9kKEuN67RN9CLin4dA97qa7M1r5W4P
FZ6vnh5Cj0hQBRXV9xJUeYSdqVItNAUfK/fEzDdjf1nUfP1Q30J49u6CV01NoJ9m
usvY9kWcV46dqn2bk2MyfTTgvmepP8fiMRPxxnVRkSz11dP5Fg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM07oeX4xevdMA0GCSqGSIb3DQEBwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDQVQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQVQHEwdTZWf0
dGx1MSAwHgYDQVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUUVJTC+h0U
MjU4MThaGA8yMTk1MTEExNDEyNTgxOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVlU2Vydm1jZXMgTEExDMIGFMA0GCSqGSIb3DQEBAQUAA4OCQAQ8AMIIB
```

```

CgKCAQEA6v6kGMnRmFDLxBEqXzP4nnpL65000kmQ7w8YXQygSdmNIOscGSU5wfh9
mZdcvCxcdxgALFsFqPvH8fqIE9ttI0fEfuZvH0s8wUsIdKr0Zz0MjSx3cik4tKET
ch0EKfMnzK0gDBavraCDeX1rUDU0Rg7HFqNA0ry3uqDmnqtk00XC9GenS3z/7ebJ
fIBEPAam5oYMFVpX6M6St77WdNE8wEU8SuerQughiMVx9kMB07imeVHBiELbMQ0N
lwSWRL/61fA02keGSTfSp/0m3u+lesf2VwVFhqIJs+JbsEscPx0kIRlzy8mGd/JV
ONb/DQpTedzUKLgXbw7Kt03HTG9iXQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAD
BgNVHQ4EFgQU2CTGYE5fTjx7gQXzdZSGPEWAJY4wgY4GA1UdIwSBhjCBg4AU2CTG
YE5fTjx7gQXzdZSGPEWAJY6hYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEyBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFNlcnZpY2VzIEExMQ4IJAM07oeX4xevdMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANDqkIpVypr2PveqUsAKke1wKCOsuw1UmH9k
xX1/VRoHbrI/UznrXtPQ0PmHA2LKSTedwsJuorUn3cFH6qNs8ixBDrl8pZwfk0Y
IBJcTFBbI1xBEFkZo03wczzo5+8vPQ60RVqAaYb+iCa1HFJpccC30vajfa4GRdNb
n6FYnluIcDbmpcQePoVQwX7W3o0YLB1QLN7fE6H1j4TBIsFd030uKzmaifQlWLYt
DVxVCNDabp0r6Uozd5ASm4ihPPoEoK07I1p0f0T6fZ41U2xWA4+HF/89UoygZS07
K+cQ90xGxJ+gmlybLFR5rbJ0LfjrgDAb2ogbFy8LzHo2ZtSe60M=
-----END CERTIFICATE-----

```

US East (Virginia) — us-east-1

DSA

```

-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEyBXN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFNlcnZpY2VzIEExMQ4IJAM07oeX4xevdMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANDqkIpVypr2PveqUsAKke1wKCOsuw1UmH9k
xX1/VRoHbrI/UznrXtPQ0PmHA2LKSTedwsJuorUn3cFH6qNs8ixBDrl8pZwfk0Y
IBJcTFBbI1xBEFkZo03wczzo5+8vPQ60RVqAaYb+iCa1HFJpccC30vajfa4GRdNb
n6FYnluIcDbmpcQePoVQwX7W3o0YLB1QLN7fE6H1j4TBIsFd030uKzmaifQlWLYt
DVxVCNDabp0r6Uozd5ASm4ihPPoEoK07I1p0f0T6fZ41U2xWA4+HF/89UoygZS07
K+cQ90xGxJ+gmlybLFR5rbJ0LfjrgDAb2ogbFy8LzHo2ZtSe60M=
-----END CERTIFICATE-----

```



```

UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUUIzvvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQQIEyBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUK2zmY9PU
STR7rc1k20wPYu4+g7wwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQA1Ng4QmN4n7iPh5CnadS0c0ZfM7by0dBePwZJyGv0Hdaw6P6E/vEk76KsC
Q8p+akuzVzVPkU4kBK/TRqLp19wEwVvhhTaxHjQ1tTRHqXIVlRkw4JrtFbeNM21
G1kSLonuzmNZdivn9WuQYeGe7nUD4w3q9GgiF3CPorJe+UxtbA==
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIIIEjCCAvqgAwIBAgIJANNPkIpcyEtIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEyBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUK2zmY9PU
OTAzMDdaGA8yMTk1MDQwMzA5MMDMwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2Vydm1jZXMgTEExIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIB
CgKCAQEApHQgVhVq3SVCzDrC7575BW7GWLzCj8CLqYcL3YY7Jffupz70jCft057Z
4fo5Pj0CaS8DtPzh8+8vdwUSMbiJ6cDd3ooio3MnCc6DwzmsY+pY7CiI3UVG7KcH
4TriDqr1Iii7nB5MiPJ8wTeAqX89T3SYaf6Vo+4Gcb3LCDGvnkZ9TrGcz2CHKJsj
AIGwgopFpwhIjVYm7obmuIxSIUv+oNH0wXgDL029Zd98SnIYQd/njiqkzE+lvXgk
4h4Tu17xZIKBgFcTtWPky+POGu81DYFqiWVEyR2JKKm2/iR1dL1YsT39kbNg47xY
aR129sS4nB5Vw3TRQA2jL0ToTIxzhQIDAQABo4HUMIHRMASGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQUgepyi0Ns8j+q67dmcWu+mKKDa+gwgY4GA1UdIwSBhjCBg4AUgepy
i0Ns8j+q67dmcWu+mKKDa+ihYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEyBx
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IANNPkIpcyEtIMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGLFwYutf1u0xcAc+kmnMPqtc/Q6b79VIX0E
tNoKMI2KR81cV8ZELXDb0NC6v8UeLpe1WBKjaWQtEjL1ifKg9hdY9RjJ4RXIDSK7
33qCQ8juF4vep2U5TTBd6hfWxt1Izi88xudjixmbpUU4YKr8UPbmixldYR+BEx0u
B1KJi9l11xvuc/Igy/xeh0AZEjAXzVvH8Bne33VvWmiMxWECZCiJxE4I7+Y6fjQJ
pLLSFFJKbNaFyX1DiJ3kXyePEZSc1xiWeyRB2ZbTi5eu7vMG4i3AYWuFVLthaBgu
lPfhafJpj/JDcqt2vKUKfur5edQ6j1CGdxqqjawn0TEqcN8m7us=
-----END CERTIFICATE-----

```

US West (Oregon) — us-west-2

DSA

```

-----BEGIN CERTIFICATE-----
MIIC7TCCAqQCCQCWukjZ5V4aZzAJBgqhkiG9w0BAQDMFwxCzAJBgNVBAYTA1VTMRkw

```



```

BAYTA1VTMRkwFwYDVQQIEExBXyXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTA4MTQw
OTAxMzJaGA8yMTk1MDExNzA5MDEzMlowXDELMakGA1UEBhMCMVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWVgU2Vydm1jZXMgTEExDMIIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAE02Y59qtAA0a6uZo7nEQcnJ260KF+LRPwZfixBH+EbEN/Fx0gYy1jppCP
s5+VRNg6/WbfqAsV6X2VSjUKN59ZMnMY9ALA/Ipz0n00Huxj38EBZmX/NdNqKm7C
qWu1q5kmIvYjKGIadfb0U8wLwLcHo8ywwfgI6FiGGsE09VMC56E/hL6Cohko11LW
dizyvRcvG/IidazVkJQCN/4zC9PU0VYKdhw33jXy8BTg/QH927QuNk+ZzD7HH//y
tIYxDhR6TIZsSnRjz3b0cEHxt1nsidc65mY0ejQty4hy7ioSiapw316mdbtE+RTN
fcH9FPIFKQNBpiqfAW5Ebp3La13/+wIDAQABo4HUMIHRMASGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQU7coQx8Qnd75qA9XotSWT3IhvJmowgY4GA1UdIwSBhjCBg4AU7coQ
x8Qnd75qA9XotSWT3IhvJmqhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IjALZL31rQCSTMMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFZ1e2MnzRaXCaLwEC1pW/f0oRG8nHr1PZ9W
OYZEWbh+QanRgaikBNDtVTwARQcZm3z+HWSkaIx3cyb6vM0DSkZuiwzm1LJ9rDpc
aBm03SEt5v8mcc7sXWvgFjCnUpz0smky6JheCD401Cf8k0oLZ93FQnTrbg620K0h
83mGCDeVKU3hLH97FYuQ+3N/IliWFDhviBAYYKFJydZLhIdlCiiB99AM6Sg53rm
oukS3csyUxZyTU2hQfdjyo1nqW9yhvFAKjnnnggiwxNKTPZzstKW8+cnYwiiTwJN
QpVoZdt0SfbuNnmwRUMi+QbuccXweav29QeQ3ADqjgB0CZdSRKk=
-----END CERTIFICATE-----

```

Africa (Cape Town) — af-south-1

DSA

```

-----BEGIN CERTIFICATE-----
MIIC7DCCAqwCCQCncbCtQbjuyzAJBgqhkiG9w0BAQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXyXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xOTA2MDQxMjQ4MDVaFw00
NTA2MDQxMjQ4MDVaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXyXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbYwggErBgqhkiG9w0BAQBMIIIBHgKBgQC12Nr1gMrHcFSZ7S/A
pQBSCMHWmn2qeoQTMVWqe50fnTd0zGFxDdIjKxUK58/8zjWG5uR4TXRzmZpGpmXB
bSufAR6BGqud2LnT/HIWGJAsnX2u0tSyNfCoJigqwhea5w+CqZ6I7iBDdnB4TtTw
q06T1nExHFVj8LMky1ZgiaE1CQIVAIhdobse4K0QnbAhCL6R2euQz1oXAoGAV/21
WUuMz/79Ga0JvQcz1FNy1sT0pU9rU4TenqLQIt5iccn/7EIfNtvV05TZKu1IKq7J
gXZr0x/KIT8zsNweetL0aGehPIYRMPX0vunMMR7hN7qA7W17WZv/76adywIsnDKq
ekfe15jinaX8MsKUDyDK7Y+ifCG4PVhoM4+W2XwDgYQAAGAIxOKbVgwLxnb6Pi2
6hB0ihFv16jKxAQI0hHzXJLV0Vyv9QwnqjJJRf0Cy3dB0zicLXiIxeIdYfvqJr+u
h1N8rGxEZYYJjEUKMGvsc0DW85jonXz0bnfcp0aaKH01KKVjL+0Zi5n2kn9wgd05

```



```
F3CVnM18BUra8A1Tr2yrrE6TVZ4wCQYHKOzIzjgEAwMvADAsAhQfa7MCJZ+/TEY5
AUr0J4wm8VzjoAIUSYZVu2NdRJ/ERPmDfhW5EsjH1CA=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAKumfZiRrNvHMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbW50b24gV2ViIFN1cnZpY2VzIEExMQZAgFw0x0TEwMjcw
NzE0MDVaGA8yMTk5MDUwMjA3MTQwNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACjB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVzIGU2VydmLjZXMgTEwDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFd571nUzVtke3rPyRkYfvs3jh0C0EMzzG72boyUNjnfW1+m0TeFraTLKb9T6F
7TuB/ZEN+vm1Yqr2+5Va8U8qLbPF0bRH+FdaKjhgWZdYXxGzQzU3ioy5W5ZM1VyB
7iUsxEALxSybC3ziPYaHI42UiTkQNaHmoroNeqVyHNnBpQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAAJLy1WyeLEgOpW4B1XPYRVD4pAds8Guw2+krqkY0HxLCdjosuH
RytGDGN+q75aAoXzW5a7SGpxLxk6Hfv0xp3RjDHsoeP0i1d8MD3hAC5ezxS4oukK
s5gbP0nokhKTMpXbTdRn5ZifCbWlx+bYN/mTYKvxho7b5SVg2o1La9aK
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJAIIFI+05A6/ZIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbW50b24gV2ViIFN1cnZpY2VzIEExMQZAgFw0x0TA2MDQx
MjQ4MDRaGA8yMTk4MTEwNzEyNDgwNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACjB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVzIGU2VydmLjZXMgTEwDMiIIBjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAY7/WHBBH0rk+20aumT07g8rxrSM0UXgki3eYgKauPCG4Xx//vwQbuZwI
oeVmR9nqnfhij2w0cQdbLandh0EGtbxerete3IoXzd1KXJb11Pvmzrzyu5SPBPuP
iCeV4qdjjkXo2YWM6t9YQ911hcG96YSp89TBXFYU3KLxfqAdTVhuC0NRGhXpyii
j/czo9njofHhqhTr7UEyPun8NVS2QWctLQ86N5zWR3Q0GRoVqqMrJs0cowHTrVw2
9Qr7QBjjB0VbyYmtYxm/DtiKprYV/e6bCAVok015X1sZDd3oC0QNoG1v5XbHJe2o
JFD8GRRy2rkW0/1NwVFDcweC6zC3QwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCE
goqzjpCpmMgCpszFHwvRaSMbspKtK7wNImUjrSB0fBjsfFu1yg1Zgn2nDCK7kQhx
jMjMNIvXbps3yMqQ2cHUKKcKf5t+WldfeT4VklRz6HSA8sd0kgVcIesIaoy2aaXU
VEB/oQziRGyKdN1d4TGYVZXG44CkrzSDvlbmfiTq5tL+kAieznVF3bzHgPZW6hKP
EXC3G/IXrXicFEe6YyE1Rakl62VncYSXiGe/i2XvsiNH3Qlmmx5XS7W0SCN0oAxW
EH9twibauv82DVg1W0kQu8EwFw8hfde9X0Rkiu0qVcuU81JgFEvPWMDFU5sGB6ZM
gkEKTzMvLZpPbBhg99Jl
-----END CERTIFICATE-----
```

Asia Pacific (Hong Kong) — ap-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQC07MJe5Y3VLjAJBgqhkhj00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXYYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEEMQzAeFw0xOTAyMDMwMjIxMjFaFw00
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEEMQzCCAbgwgEsBgcqhkhj00AQBMIIBHwKBgQDvQ9RzVvf4MAwGbbqfX
b1CvCoVb99570kLGN/04CowHXJ+vTBR7eyIa6AoX1tsQXB0mrJswToFKKxT4gbuw
jK7s9QXX4CmTRWcEg02RXtZSVj0hsUQMh+yf7Ht40VL97LWnNfGsX2cwjcRWHYgI
71vnuBNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGkd9FAoGBA0CG
eSNmXpW4QFu4pI1Aykm6EntZKKHT87gdXkAkfoC5fAf0xxhnE2HezZHp9Ap2tMV5
8bWNVPHvoKCCqwfM+OUB1AxC/3vqoVkkL2mG1KgUH9+hrtPMtkw03RREnKe7I50
x9qDimJp0ihrl4I0dYvy9xU0oz+DzFAW8+y1WVYpA4GFAAKBgQDbnBAKSxWt9QH
Y6Dt+EFdGz61AZLedeBKpaP53Z1DT034J0C55YbJTWBTFGqPtOLxnUVD1GiD6GbmC
80f3jvogPR1mSmGsydbNbZnbUEVWtRhe+y5zJ3g9qs/DWmDW0deEFvkhWVnLJkFJ
9pd0u/ibRPH11E2nz6pK7Gb0QtLyHTAJBgqhkhj00AQDAzAAMC0CFQCoJlWgtJQC
cLoM4p/jtVF0j26xbgIUUS4pDKyHaG/eaygLTtFpFJqzWHc=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICszCCAbQCCQDtQvkVxRvK9TANBgqhkhkiG9w0BAQsFADBqMQswCQYDVQQGEwJV
UzETMBEGA1UECBMKV2FzaGluZ3Rvb3JlEQMA4GA1UEBxMHU2VhdHRsZTEYMBYGA1UE
ChMPQW1hem9uLmNvbSBjb250bW9uMR0wGAYDVQQDEExFLYzIuYW1hem9uYXZzLmNvbTAe
Fw0xOTAyMDMwMzAwMDZaFw0yOTAyMDIwMzAwMDZaMGoxCzAJBgNVBAYTA1VTMRMw
EQYDVQQIEWpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGx1MRgwFgYDVQQKEw9B
bWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3MuY29tMIGfMA0G
CSqGSIb3DQEBQUAA4GNADCBiQKBgQC1kkHXyTfc7gY5Q55JJhjTieHAgacaQkiR
Pity9QPDE3b+NXDh4UdP1xdIw73JcIIG3sG9RhWiXVCHh6KkuCTqJfPUknIKk8vs
M3RXflUpBe8Pf+P92pxqPMCz1Fr2NehS3JhhpkCZVGxxwLC5gaG0Lr4rFORubjYY
Rh84dK98VwIDAQABMA0GCSqGSIb3DQEBGwUAA4GBAA6xV9f0HMqXjPHuGILDyaNN
dKcVp1NFwDTydvG32MNUbAGnecoEBtUPTxBsLoVYXC0b+b5/ZMDubPF9tU/vSXuo
TpYM5Bq57gJzDRaB0ntQbX9bgHiUxw6XZwaTS/6xjRJDT5p3S1E0mPI31P/eJv4o
Ezk5zb3eIf10/sqt4756
-----END CERTIFICATE-----
```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJAMoxixvs3YssMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWw6b24gV2ViIFN1cnZpY2VzIExMQzAgFw0xODA3MjAw
ODQ0NDRaGA8yMTk3MTIyMzA4NDQ0NFowXDELMAkGA1UEBhMCMVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBACTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2Vydm1jZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIB
CgKCAQEA4T1PNs0g0FDrG1WePoHe0Sm0JTA3HCry5LSbYD33GFU2eBr0IxoU/+SM
rInKu3GghAMfH7WxPW3etIAZiyTDDU5RLcUq2Qwdr/ZpXAWpYocNc/CEmBFtfbxF
z4uwBIN3/diM0RSbe/wP9EcgmNUGQMMZWeAji8sMtwp0b1NWAP9BniUG0Flcz6Dp
uPovwDTLdAYT3TyhzlohKL3f6048TR5yTaV+3Ran2SGRhyJjfh3FRpP4VC+z5LnT
WPQHn74Kdq35UgrUxNhJraMGCzzno1UuoR/tFMwR93401GsM9fVA7SW3jjCGF81z
PSzjy+ArKyQqIpLW1YGWDFk3sf08FQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDK
2/+C3nPMgty0FX/I3Cyk+Pui44Ig0wCsIdNGwuJysdq5VIfnjegEu2zIMWJSKG0
1MzoQXjffkVZ97J7RNDW06oB7kj3WVE8a7U4WE0fn0/CbMUF/x99CckNDwpjgW+
K8V8SzAsQDvYZs2KaE+18GFfLVF1TGUYK2rPSZMHyX+v/TI1c/qUceBycrIQ/kke
jDFsihUMLqgm0V2hXKUpIsmiWMGrFQV4AeV0iXP8L/ZhcepLft5SbsGdUA3AUy1
3If8s81uTheiQjwY5t9nM0SY/1Th/tL3+RaEI79VNEVfG1FQ8mgqCK0ar4m0zJ1
tmmEJM7xeURdpBBx36Di
-----END CERTIFICATE-----

```

Asia Pacific (Hyderabad) — ap-south-2

DSA

```

-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXjrQ4+XMAKGBYqGSM44BAMwXDELMAkGA1UEBhMCMVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s50f2EbdSP09EAMMeP4C2USZpRV1AI1H7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdirmVClpJ+f6AR7ECLCT7up1/63xhv401fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzriith1yrv8iIDGZ3RSAHHAhUA12BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v60uqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFh03zwyjMim4TwWeotUfI0o4K0uHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZ16Ae1U1ZAFM0/7PSSoDgYUAAoGBAJCKGGBBoxIUxqBk94JHhwZZbgvbP0DA0oHENQWxp/981I7/
Y0fYJ0VMJS22aCnHDurofmo5rvNIkgXi7Rztbhu
+1ko9rK6DgmpUwBU0WZtf34aZ2IWNBwHaVhHvWAQf9/46u18dMa2YucK1Wi+Vc+M
+KldrvGxmhym6ErNlzhJyMAKGBYqGSM44BAMDlwAwLAIUaaPKxa0HoYvwz709xXpsQueIq+UCFFa/
GpzoD0Sok11057NU/2hnsiW4
-----END CERTIFICATE-----

```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAzygAwIBAgIGAXjwLj9CMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIDBBXYXNoaW50
+sFcobrijvcAYm0PNRD8f4R1jAzvoLt2+qGe0TAY01Httj6cmsYN3AP1hN5iYuppFiYs12eNPa/
CD0Vg0BAfDF1V5rzjpA0j7TJabVh4kj7JvtD+xYMi6wEQA4x6SP0NY40eZ2+8o/
HS8nucpWDVdPR06ciWU1MhjmDmwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAAy6sgTdRkTqELHBeWj69q60xHyUmsWqHAQ
TGGbYP0yP2qfM10cCIImzRI5W0gn8gogderVfeT7nH5ih0TWEy/QDwfKQ601L4erm4yh4YQq8vcqAPSkf04N
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAIvWfPw/X82fMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQZAgFw0yMjA3MDQx
NDMwMjhaGA8yMjAxMTIwODE0MzAyOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgt
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBZXWV2VydmljZXMgTEExMjA3MDQxNDMwMjhaGA8yMjAxMTIwODE0MzAyOFow
CgKCAQEAg29QEFriG+qFEjYw/v62nN701MJY/Hevx5TtmU/VIYBPQa3HUGTBabbI
2Tmy8UMpa8kZeaYeI3RAfiQWt0Ws7wUrBu02Pdp518WDPaJUH7RWEuu1BDDkyZRw
NAMNPCn3ph70d243IFcLGku7HVeke15poqRpSfojrMasjlf+CvixUeAJbmFoxUHK
kh5unzG2sZy04wHXcJPQRf5a8zSTPe9YZP1kXPPEv4p/jTSggaYPxXyS6QVaT1V
zLeLFZ0fesLPMeil3KYQtV7IKLQiEA2F6dxWnxNWQlyMHtdq6PucfEmVx17i/Xza
yNBRo0azY8WUNVkeXrRhp/pU8Nh3GQIDAQABO4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU9A01aZk9RLXk2ZvRVoUxYvQy9uwwgY4GA1UdIwSBhjCBg4AU9A01
aZk9RLXk2ZvRVoUxYvQy9uyhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExB
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAIVWfPw/X82fMBlGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADEXluMRQRftqViahCnauEWGdMvLCBr8A+Yr
6hJq0guoxEk/1ahxR137DnfMPuSbi1Rx5QKo7oBrWfG/zsgQUnF2IwHTzwD+i/2m
XCane6FiS5RpK31GdILq8ZmlhQk+6iI8yoZLr0LCfTh+CLgIKH0knfR51FzgzAiF
SI8/Q9mm+uvYtSTZECI6Zh57QZPoETAG/y1+9ji0y21Aelqa/k1i+Qo8gMf0c+Pm
dwY7o6fV+oucgrlSdey6VM45LeyILQqv0RXtVzjuowanzmCCFMjgqi09oZAWu40h
+F3unijELo01vZJs8s2N3KG1o3/jtUFTX6RTKShZ1APLwBi5GMI=
-----END CERTIFICATE-----
```

Asia Pacific (Jakarta) — ap-southeast-3

DSA

```
-----BEGIN CERTIFICATE-----
```

```

MIIC8DCCArCgAwIBAgIGAXbVDEikMAkGByqGSM44BAMwXDELMakGA1UEBhMCMVVMxGTAXBgNVBAGMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s50f2EbdSP09EAMMeP4C2USZpRV1AI1H7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdirmVClpJ+f6AR7ECLCT7up1/63xhv401fnxqimFQ8E
+4P208UewwI1VBNAFpEy9nXzriith1yrv8iIDGZ3RSAHHAhUAL2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtCnrhXuXmUr7v60uqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFh03zwyjMim4TwWeotUfI0o4K0uHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZ16Ae1U1LZAFM0/7PSSoDgYUAAoGBAPjuiEx05N3JQ6cVwntJie67D80uNo4jGRn
+crEtL7Y00jSVB9zGE1ga
+UgRPIaYETL293S8rTJTVgXAqdpBwfaHC6NUzre8U8iJ8FMNn1P9Gw1oUIlgQBj0RyynVJexoB31TDZM
+/52g90/bpq1QqNyKbeIgyBB1c1dAtr1QLnsMAkGByqGSM44BAMDlwAwLAIUK8E6RDIRtwK+9qnaTOBhv0/
njuQCFfocyt10xK+UDR888oNsdgtif2Sf
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIICmzCCAZygAwIBAgIGAXbVDG2yMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIDBBXYXNoaW50
Vbt0gQ1ebWcur2hS07PnJifE40PxQ7RgSA1c4/spJp1sDP+ZrS0L01ZJfKhXf1R9S3AUwLncs7b
+IuVXdY5LK9RKqu64nyXP5dx170zOL81oEycSuRR2fs+04i2QsWBVP+KFNA7P5L1EHRjkGT08kjNKviwRV
+0kP9ab5wIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAI4WUy6+DKh0JDSzQEZNYBgN1SoSuC2owtMxCwGB6nBfzzfcekWvs
+87w/g91NwUnUt0ZHYyh2tuBG6hVJuUEwDJ/z3wDd6wQviL0TF3MITawt9P8siR1hXqLJNxpjRQFZrgHqi
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMtdyRcH51j9MA0GCSqGSIb3DQEBQwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXXYXNoaW50ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0yMjA0MDgX
MjM5MTZaGA8yMjAxMDkxMjE5MzcxNlowXDELMakGA1UEBhMCMVVMxGTAXBgNVBAGT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWlGyU2Vydm1jZXMgTEExDMIIIBjANBgkqhkiG9w0BAQEFAAOCQA8AMIIB
CgKCAQEAvUsKCxoH6KXRYJLeYTWAQfaBQeCwhJaR56mfUeFHJE4g8aFjWkiN4uc1
Tv0yYNNIZKTHWmzmulmdinWNbwP0GiR0Hb/i7ro0HhvnptyycGt8ag8affiIbx5X
7ohdwSN2KJ6G0IKf1Ix7f2NEI0oAMM/9k+T1eVF+MVWzpZoiDp8frLNkqp8+RAgz
ScZsbRfww3u/if5xJAVdg2nckIWDMSHEVPoz01Jo7v0ZuDtwWsL1LHnL5ozvsKEk
+ZJyEi23r+U1hIT1NTBdp4yoigNQexedtwCSr7q36o0dDwvZpqY1kLi3uxZ4ta+a
01pz0STwMLgQZSbKWQrPmvsIAPrxoQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAD
BgNVHQ4EFgQU1GgnGdNpbnL31LF30Jomg7Ji9hYwgY4GA1UdIwSBhjCBg4AU1Ggn
GdNpbnL31LF30Jomg7Ji9hahYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExB
XYXNoaW50ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAMtdyRcH51j9MBIGA1UdEwEB/wQIMAYBAf8C

```

```
AQAwdQYJKoZIhvcNAQELBQADggEBACV100qQ1atBKVeiWMrhpczsJroxDx1ZT0ba
6wTMzk7c3akb6XM0SZFbGaiFkeBPzqTHEhDlrClM2j9AIlYcCx6YCrTf4cuhn2mD
gcJN33143e0WSaeRY3ee4j+V9ne98y3k02wLz95VrRgc1PFR8po2iWGzGhwUi+FG
q8dXeCH3N0DZgQsSgQWwmdNQXZZej6RHLU/8In5trHKLY0ppnLBjn/UZQbeTyW5q
RJB3GaveXjfgFUWj2q0cDuRGaikdS+dYaLsi5z9cA3FolHzWxx9M0s8io8vKqQzV
XUrLTNwWuhZy88c0lqGPxnoRbw7TmifwPw/cunNrsjUU0gs6ZTk=
-----END CERTIFICATE-----
```

Asia Pacific (Melbourne) — ap-southeast-4

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjWF7P2MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s50f2EbdSP09EAMMeP4C2USZpRV1AI1H7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYldmVClpJ+f6AR7ECLCT7up1/63xhv401fnxqimFQ8E
+4P208UewwI1VBNAFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUA12BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v60uqC+VdMCz0HgmdRWVe0utRZT
+ZxBxCBGLRjFEnEj6EwoFh03zwyjMim4TwWeotUfI0o4K0uHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZ16Ae1U1ZAFM0/7PSSoDgYQAAoGAPRXSsQP9E3dw8QXK1rgBgEVCprLHdK/bbrMas0XMU1Eh0D
+q
+0PcTr8+iwbtoX1Y5MCeatWIp1GrXQjVqsF8vQqx1EuRuYKbR3nq4mWwaeG1x9AG5EjQHRa3GQ44wWH0dof0M3NRI1MF
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICmzCCAZygAwIBAgIGAXjSh40SMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIDDBBYXNoaW50
+qWTGAbGsPeMX4hBMjAJUKys2NIRcRZaLM/BCew2FIPVjNtlaj6Gwn9ipU4M1z3zIwAMWi1AvGMSreppt
+wV6MRtf0jh0Dvj/veJe88aEZJMoZNgkJFRS
+WFwSckQeL56tf6kY6QT1No8V/0CsQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAF7vpPghH0FRo5gu49EAirRNPrIvW1egM
wGkqIwwuXYj+1rh1L+/
iMpQWjdVGEqIZSeXn5fLmdx50eegFCwND837r9e8XYTiQS143Sxt9+Yi6BZ7U7YD8kK9NBWoJxFqUeHdpRCs007C0jT3
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAN4GTQ64zVs8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBYXNoaW50dG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
-----END CERTIFICATE-----
```

```
dGx1MSAwHgYDVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0yMjA3MTMx
MzMzMDBaGA8yMjAxMTIxNzEzMzMwMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVudmUyVydmljZXMgTExDMiIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEa2BYgeCr+Rk/jIAED0HS7wJq162vc83QEwjuzk0q0FEReIZz1N1fBRNXK
g0T178Kd3gLYcE59wEFbTe/X5y0A1Lo95x1anSAo7R+Cisf9C2HQUJp+gVb+zx71
lniPF7gHzIGpm0M8DdAU/IW+wkZwGbP4z7Hq9+bJ0P21tvPJ5yxSgkFuDsI9VBHa
CLOprHsChh2VdP8KcMgQQMmHe1NmBpyTk0uL/aLmQkCQEX6ZIRG0eq228fwlh/t+
Ho+jv87duihvKic6MrL32S1D+maX0LSDUydwda0LLTGkh7oV7+bFuH6msrXUu+Ur
ZEP1r/MidCWMhfgrFzeTBz0HA97qxQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQUcHMD1cHqzmsQ5hpUK3EMLhHdsi4wgY4GA1UdIwSBhjCBg4AUcHMD
1cHqzmsQ5hpUK3EMLhHdsi6hYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQKQIEExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQKHEwdTZWF0dGx1MSAwHgYDVQKQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAN4GTQ64zVs8MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAl4PFyVN+7EGS0bioiPnv0LL0f70SSzUZJ8p
X090d4rWea7jIbgZ2AKb+ErynkU9xVg7XQQ5k6KDWgp/4jYFL2dqnt/YAY4PS0un
RSrYE1awxLT0BcLn4rcSDC79vQe1xGC5//wDdV6b399COAHRAK6axWYy5w32u9PL
uw0cIp3Ch8JoNwgcTHKRRGzePmBeR4PNqhHTArG4/dJk6/aU040pX0WzI6L67CGY
6Nex3dau+gkLCK93dTEkrXtyXHu4wB0J9zd1w+iQ0SEa9eKc78/NjEsF/FZdGrWC
t571IM00XJhQ1kRgSwNeZdQWV1dRakv06sfcvVYkfj1wAvZvvAw=
-----END CERTIFICATE-----
```

Asia Pacific (Mumbai) — ap-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkiG9w0BAQ0DMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQKQIEExBXNoaW5ndG9uIFN0YXR1MRAwDgYDVQKHEwdTZWF0dGx1MSAwHgYD
VQKQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODAxMDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQKQIEExBXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQKHEwdTZWF0dGx1MSAwHgYDVQKQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkiG9w0BAQ0BMiIBHwKBGCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAl1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCySFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNMp9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKml1qx41lHW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
```

```
7HX32MxXYruse9ACFBNGmdX2ZB1rVNG1rN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUdLA+x6tTAP3LRT1r0z6n0xfsozdMwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAGTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBAClB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWVjZjZlZmVjZmVjZmVj
MB4XDTE0MDQyOTE0MTMwMVowXDTI0MDQyODE0MTMwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAGTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWVjZjZlZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVj
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJO+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnf0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHGDAdBgNVHQ4EFgQUJdbMCBXXKtvCcWdwUizvtUF2
UTgwgZkGA1UdIwSBkTCBjjoAUJdbMCBXXKtvCcWdwUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQIQIExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUDLA+x6tT
AP3LRT1r0z6n0xfsozdMwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAZ7rYKoAwwiiH1M5GJbrT/BEk3002VrEPw8ZxgppQ/EK1zM10s/0Cyrmp7
UYyUgYfQe5nq37Z94r0USeMgv/WRxaMwrL1LqD78cuF9DSkXaZIX/kECtVaUnjk8
BZx0QhoIH0pQocJUS1m/dLeMuE0+0A3HNR6JVktGsUdv9u1mKw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAi0gAwIBAgIJAPRYyD8TtmC0MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQIQIExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUDLA+x6tTAP3L
MDQ1MDFaGA8yMTk1MDgxMTEwNDUwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAGT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVjZjZlZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVjZmVj
CgKCAQEA0LSS5I/eCT2PM0+qusorBx67QL26BIWQHd/yF6ARtHBb/1DdFLRqE5Dj
07Xw7eENC+T79m0x0AbeWg91Ka0D0zw6i9I/2/HpK0+NDEdD6sPKDA1d45jRra+v
CqAjI+nV9Vw91wv7HjMk3RcjWGziM8/hw+3YNIutt7aQzZRwIW1Bpcqx3/AFd8Eu
2UsRMSHgkGUW6UzUF+h/U8218XfrauKNGmNKDYUhtmyBrHT+k6J0hQ4pN7fe6h+Z
w9RVHm24BGh1LxLHlms0IxvbrF277uX9Dxu1HfKfu5D2kimTY7xSZDNLR2dt+kNY
/+iWdIeEFpPT0PLSILt52wP6stF+3QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBIE
6w+WWC2gCfoJ06c9HMyGLMFEpqZmz1n5IcQt1h9iy07Vkm1wkJiZsMhXpk73zXf
TPxuXEacTX3S0Ea070IMCFwkus05f61e0yFTynHCzBgZ3U0UkRVZA3WcpcbNB6Dwy
h7ysV1qyT9WZd7E0Ym5j5oue2G2xdei+6etgn5UjyWm6liZGrc0F6WPTdmzqa6WG
```



```
ApEqanpkQd/HM+hUYex/ZS6zEhd4CCDLgYkIjlrFbFb3pJ10VLztIfSN5J40o1pu
JVCfIq5u1NkpzL7ys/Ub8eYipbzI6P+yxXiUSuF0v9b98ymczMYjrSQXIf1e8In3
OP2Cc1Choz8XDQcvvKAh
-----END CERTIFICATE-----
```

Asia Pacific (Osaka) — ap-northeast-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXyXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEEMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXyXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEEMQzCCAbcwggEsBgcqhkJ00AQDMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmveve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAKGBYqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUHTRhxHhBZF0GvTFKxHoy9+f5H18wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWlGU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTE2NTQwN1oXDTE1MDQyODE2NTQwN1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWlGU2Vydm1jZXMgTEExDMIGFMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXk3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUUizvtUF2UtiHYKReMFwxCzAJ
-----END CERTIFICATE-----
```

```
BgNVBAYTA1VTMRkwFwYDVQIQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUHTRhxHhB
ZFOGvTFKxHoy9+f5H18wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQAUZx7DcYbhWNTD4BNghr5beruT20UoGHH9J73UKxwdqeb9bH1LIWhIZ00X
/1mjn3bWBgCwfoS8gjZwsVB6fZbNBRy8urdBZJ87xF/4JPBjt7S9oGx/zthDUYrC
yK0Y0v4G0PgiS81CvYLG09LpmYhLSJbXENlkC04v5yxdKxZxyg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAi0gAwIBAgIJAMn1yPk22ditMA0GCSqGSIb3DQEBCwUAMFwxZzA3MTkx
BAYTA1VTMRkwFwYDVQIQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNzA3MTkx
MTEyNTA4yMTk2MTIyMjExMTI1FowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBACjTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVzIGU2VydmIjZXMgTEExMTI1FowXDELMAkGA1UEBhMCVVMxGTAXBgNV
CgKCAQEAznEYef8IjhrJoazI0QGZkmlmHm/4rEbyQbMNifxjsDE8YwTHNwaM91z
zmyK6Sk/tKlWxcn13g31iq305ziyFPEewe5Qbwf1iz2cMsvfNBcTh/E6u+mBPH3J
gvGanqUJt6c4IbipdEouIjjnyVWd4D6erLl/ENijeR10xVpaqSW5SBK7jms49E
pw3wtbchEl3qsE42Ip4IYmWxqjgaxB7vps91n4kfyZAjUmklcqTfMfPckzmJCRgp
Vh1C79vRQhmriVKD6BXwfZ8tG3a7mijeDn7kTsQzg007Z2SAE63PI048JK8Hc0bh
tXORUQ/XF1jzi/SIaUJZT7kq3kw18wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBj
Tht09dLvU2QmKuXAhXjsIdlQgGG3ZGh/Vke4If1ymgLx95v2Vj9Moxk+gJuUSRL
BzFte3TT6b3jPolbECgmAorjj8NxcjC17N8QAAI1d0S0gI8kqkG7V8iRyPIFekv+M
pcai1+cIv5IV5qAz8Q0MGYfGdYkcoBjsgiyvMJU/2N2UbZJNGWvcEGkdjGJUYY00
NaspCAFm+6HA/K7BD9zXB1IKsprLgqhiIUgEaW3UFEbThJT+z8UfHG9fQjzzfN/J
nT6vuY/0RRu1xAZPyh2gr5okN/s6rnmh2zmBHU1n8cbCc64MVfXe2g3EZ9G1q/9n
izPrI09hMypJDP04ugQc
-----END CERTIFICATE-----
```

Asia Pacific (Seoul) — ap-northeast-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzA3BgcqhkiG9w0AQDMFwxZzA3BAYTA1VTMRkw
FwYDVQIQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODAxMDUxMjU2MTJhMFwxZzA3BAYTA1VTMRkwFwYDVQIQIEExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkiG9w0AQBMIIIBHwKBGQCjKvcS2bb1VQ4yt/5e
```

```
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLcLnd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUbBSn2UI06vYk4iNwV0RPxJJtH1gwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWVlU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTEzZmZg0Nl0XDTI5MDQyODEzZmZg0Nl0wXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWVlU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCChvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RwqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXk3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUuizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUuizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDQVQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQVQHEwdT
ZWf0dGx1MSAwHgYDQVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUbBSn2UI0
6vYk4iNwV0RPxJJtH1gwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAmjTja1G8MGLqWTC2uYqEM8nzI3px1eo0ArvFRsyqQ3fgmWcQpxExqUqRy
l3+2134Kv8dFab04Gut5wlfrtc20wPKKicmv/IXGN+9bKFnQFjTqif08NIzrDZch
aFT/uvxrIiM+oN2YsHq66GUh02+xVRXDxvM/V0bFgPERbJpyA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAi0gAwIBAgIJANuCGcCht0JhMA0GCSqGSIb3DQEBwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDQVQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQVQHEwdTZWf0
dGx1MSAwHgYDQVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTA5MTQx
NTU3NDRaGA8yMTk1MDIxNzE1NTc0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVlU2Vydm1jZXMgTEExDMIIIBjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
```

```
CgKCAQEA66iNv6pJPmGM20W8HbVYJS1KcAg2vUGx8xeAbzZIQdpGfKabVcUHGB6m
Gy59VXDMD1rJckDDk6dxU0hmcX9z785TtVZURq1fua9QosdbTzX4kAgHGdp4xQEs
m06QZqg5qKjBP6xr3+PshfQ1rB8Bmwg0gXEm22CC7o77+7N7Mu2sWzWbiUR7vi14
9FjWS8XmMNwFT1Shp4l1TDTevDWW/uYmC30RThM9S4QPvTZ0rAS18hHVam8BCTxa
LHaVCH/Yy52rsz0hM/F1ghnSnK105ZKj+b+KIp3adBL80MCjgc/Pxi0+j3HQLdYE
32+FaXWU84D2iP2gDT28evnstzuYTQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC1
mA4q+12pxy7By6g3nBk1s34PmWikNRJBw0qhF8ucGRv8aiNhRRye9lokcxomwo8r
KHbbqvTK8510xUZp/Cx4sm4aTgcMvfJP29jGLclDzeqADIVkWEJ4+xncxSYV1S9x
+78TvF/+8h9U2LnS164PXaKdxHy2IsHIVRN4GtoaP2Xhpa1S0M328Jykq/571nfn
1WRD1c/fQf1edgzRjhQ4whcAhv7WRRF+qTbfQJ/vDxy8lki0svU9XzUaZ0fZSfXX
wXzZamQb0NvFcXVHY/0PSiM8nQoUmkkBQuK1eDwRWvkoJKYKyir3jvXK7HIWtMr04
jmXe0aMy3thyK6g5sJVg
-----END CERTIFICATE-----
```

Asia Pacific (Singapore) — ap-southeast-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqQCCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYD
VQKQExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVoQIEExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYDVoQKQExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkJ00AQBMIIbHwKBgQCjkvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJ1/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKml1qx41lHW
MXrs3IgiB6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAKGBYqGSM44BAMDLwAwLAIUWXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUSqP6ih+++5KF07NXngirWf26mhSUwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
```

```
BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWlGU2VydmJjZXMgTExD
MB4XDTI0MDQyOFE0MzAxNFoXDTI5MDQyOFE0MzAxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAGTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWlGU2VydmJjZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJO+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnf0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUuizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUuizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQREExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQHEwdT
ZWF0dGx1MSAwHgYDVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUSqP6ih++
+5KF07NXngRwF26mhSUwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQAw13Bxw11U/JL58j//Fmk7qqtrZTqXmaz1qm2WlIpJpW750M0cP4ux1uPy
eM0RdVZ4jHSMv5gtLAv/PjExBfw9n6vNck+5GZG4Xec5DoapBZHxmfMo93sjxBFP
4x9rWn0GuwAV09ukjYPevq2Rerilrq5VvppHtbATVNY2qecXDA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAjVMGw5SHkcvMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQREExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQHEwdTZWF0
dGx1MSAwHgYDVQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTEwMjkw
ODU3MTlaGA8yMTk1MDQwMzA4NTcxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAGTE
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWlGU2VydmJjZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAAQ0CAQ8AMIIB
CgKCAQEAlaSSLfB170gmikjLReHuNhVuvM20dCsVzptUyRbut+KmIEEc24wd/xVy
2RMIrydGedk4tUjkUy0yfET50AyT43jTzDPHZTkRSVkyjBdcYbe9o/0Q4P7IVS3
X1vwruUu0qo9nSID0mxMn0oF118KAqnn10tQ0W+1NSTkasW7QVzcb+3okPEVhPA0q
MnLY3vkMQGI8zX4i0KbEcSVIzf6wuIffXMGHVC/JjwihJ2USQ8fq6oy686g54P4w
R0g415kLYcodjqThmGJPNUApAZ7M0c5Z4pymFuCHgNAZNvjhZDA8420jecqm62zcm
Tzh/pNMNeGCRYq2EQX0aQtY0Ij7b0QIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU6SSB+3qALorPMVNjToM1Bj3oJMswgY4GA1UdIwSBhjCBg4AU6SSB
+3qALorPMVNjToM1Bj3oJMuhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQREExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQHEwdTZWF0dGx1MSAwHgYDVQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAjVMGw5SHkcvMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAF/0dWqkIEZK5rca8oP0VS+to1JJE/FRZO
atH0eaQbWzyac6NEwjYeeV2kY63skJ+QPuYbSuIBLM8p/uTRIVYM4LZYImLGuvo0
IdtJ8mAzq8CZ3ipdMs1hrqF5GRp81g4w2QpX+PfhW47iIOBiqSAUkIr3Y3BDaDn
EjeXF6qS4iPIvBaQQ0cvdddNh/pE33/ceghbkZNTYkrwMyBkQ1RTTVKXFN7pCRUV
+L9FuQ9y8mP0BYZa5e1sdkwebyDU+eqVzsil98ntkhpjvRkaJ5+Drs8TjGaJWlRw
5Wu0r8unKj7YxdL1bv7//RtVYVVi2961doRUyv4ScvJF11z00dQ=
-----END CERTIFICATE-----
```



```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL2b0gb+dq9rMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEyBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTEwMjkx
OTAwNTdaGA8yMTk1MDQwMzA5MDA1N1owXDELMAKGA1UEBhMCVVMxGTAXBgNVBAGT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBAClB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBZXWV2VydmljZXMgTEExDMIIIBjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAmRcyLWraysQS8yDC1b5Abs3TUaJabjqWu7d5gHik5Icd6dK18EYpQSeS
vz6pLhkg04xBbCRGlge8LS/0ijcZ5HwdrxBiKbicR1YvIPaIyEQQvF5sX6UWkGYw
Ma5IRGj4YbRmJkBybw+AAV9Icb5LJNOMWpi340WM+2tMh+8L234v/JA6ogpdPuDr
sM6YFHMZ0NWo58MQ0FnEj2D7H58Ti//vFP10TaaPwaAIRF85zBiJtKcFJ6vPidqK
f2/SDuAvZmyHC8ZBHg1moX9bR5FsU3QazfbW+c+JzAQWHj2AaQrGSCITxCM1S9sJ
l51DeoZBjnx8cnRe+HCaC4YoRbiqIQIDAQABo4HUMIHRMASGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQU/wHIo+r5U31VIsPoWoRVsNXGxowwgY4GA1UdIwSBhjCBg4AU/wHI
o+r5U31VIsPoWoRVsNXGxoyhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEyBXY
XNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAL2b0gb+dq9rMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAQobLv8IxlQyORTz/9q7/VJL509/p4HAeve
92riHp6+Moi0/dSEYPeFTgdWB9W3YCnc34Ss9TJq2D7t/zLGG1bI4wYXU6VJjL0S
hCjWeIyBXUZ0ZKfCb0DSJeUElsTRSXSfUvRz9EAwjLvHni3BaC9Ve34iP71ifr75
8Tpk6PEj0+JwiiJFH8E4GhcV5chB0/iooU6ioQqJrMwFYnwo1cVZJD5v6D0mu9bS
TMIJLJKv4QQQpPsNdjib7G9bfbk6trP8fUVYLHLsV1Iy5lGx+tgwFEYkG1N8I00/
2LCawwaWm8FYAFd3IZl04RImNs/IMG7VmH1bf4swH0BHgCN1uYo=
-----END CERTIFICATE-----
```

Asia Pacific (Tokyo) — ap-northeast-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkiG9w0AQMDFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEyBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODAxMDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEyBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgqhkiG9w0AQMIIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
-----END CERTIFICATE-----
```

```

hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buycU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNMp9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULgwDh7TiDrPPBJwscqDwiBHkEFQwDQYJKoZIhvcNAQEL
BQAwXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWlgaU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTYyMjM0MDQyOTYyMjM0MDQyOTYyMjM0MDQyOTYyMjM0MDQyOTYy
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWlgaU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCChvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHGDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDQ0EIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQ0QEwdT
ZWf0dGx1MSAwHgYDQ0KExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IULgwDh7Ti
DrPPBJwscqDwiBHkEFQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQBtjAg1Bde1t4F9EHCZ0j4qnY6Gigy070u54i+1R77MhbpzE8V28Li9l+YT
QMIn6SszJqU3/fIycIro10VY11HmaKYgPGSEZxBenSBHfzwdLRmC9oRp4QMe0BjOC
gepj11UoiN70A6PtA+ycN1sP0oJvdBjhvayLiuM3tUfLTrgHbw==
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL9KIB7Fgvg/MA0GCSqGSIb3DQEBwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDQ0EIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQ0QEwdTZWf0
dGx1MSAwHgYDQ0KExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTA4MTQw
OTAwMjVhGA8yMTk1MDExNzA5MDAyNVowXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWlgaU2Vydm1jZXMgTEExDMIIIBjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAz0djWUcmRW85C5CiCKPFiTiVj6y20uopFxE5d3Wtab10bm06vnXVKXu
tz3AndG+Dg0zIL0gM1U+QmrSR0PH2Pfv9iejfLak9iwdm1WbwRrCEAj5VxPe0Q+I

```



```
Kezn0txzqQ5Wo5NLE9bA61sziUAFNVsTFUzphEwRohcekYyd3bBC4v/RuAjCXHVx
40z6AIksnA0GN2VABM1TeMnVPItKOCIErL111SqXX1gbtL1gxSW40JWdF3WPB68E
e+/1U3F70Er7XqmNOD0L6yh92QqZ8fHjG+af0L9Y2Hc4g+P1nk4w4iohQ0PABqzb
MPjK7B2Rze0f90Ec51GBQu13kxkWWQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU5DS5IFdU/QwYbikgtWvkU3fDwRgwY4GA1UdIwSBhjCBg4AU5DS5
IFdU/QwYbikgtWvkU3fDwRihYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFNlcnZpY2VzIEExMQ4IJAL9KIB7Fgvg/MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAG/N7ua8IE9IMyno0n5T57erBvLT0Q79fIJN
Mf+mKRM7qRRsdg/eumFft0rL0Ko54pJ+Kim2cngCWNhkcZctRHBV567AJNt4+ZDG5
hDgV0Ixw01+eaLE4qzqWP/9Vr0+p3reuumgFZLVpvVpwXBBEBFUf2drUR14awfI2
L/6VGINXYS7uP8v/2VBS7r6XZRnPBuY/R4hv5efYXnjwA9gq8+a3stC2ur8m5yS1
faKSWE4H320yAyaZWH4gpwUdbU1YgPHtm/ohRtiWPrN7KEG5Wq/REzMIjZCnx0fS
6KR6PNj1hxBsImQhmBvz6j5PLQx0xBZIpDoiK278e/1Wqm9LrBc=
-----END CERTIFICATE-----
```

Canada (Central) — ca-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXyXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKEXdBbWF6b24gV2ViIFNlcnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXyXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFNl
cnZpY2VzIEExMQzCCAbcwggEsBgqhkJ00AQBMIIbHwKBgQCjkvcS2bb1VQ4yt/5e
ih5006kK/n1Lz11r7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJ1/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
1Ra2v1ntMX3carVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKml1qx411HW
MXrs3Igb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziziqQYMAKGBYqGSM44BAMDlwAwLAIUwXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
```

```

MIIDITCCAoqgAwIBAgIUIrLgixJJB5C4G8z6pZ5rB0JU2aQwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWVlU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTE1MzU0M1oXDTE1MDQyODE1MzU0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWVlU2Vydm1jZXMgTEExDMIGFMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCChvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJO+eIB
UqPfQG09kZlwpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHGDAdBgNVHQ4EFgQUJdbMCBXXtvCcWduUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWduUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWV0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUIrLgixJJ
B5C4G8z6pZ5rB0JU2aQwEgYDVR0TAAQH/BAGwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQBHIQJmzyFAaSYs8SpiRijIDZW2RIo7qBk/pI3rqK6y0WD1PuMr6yNI81D
IrKGGftg4Z+2KETYU4x76HSf0s//vfH3QA57qFaAwdhKyy4BhteFQ1/Wex3xTLX
LiwI07kwJvJy3mS6UfQ4HcvZy219tY+0iy0Wrz/jVxwq7T0kCw==
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJAJNKhJhaJ0uMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWV0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNjA3MjIx
MTM3MTdaGA8yMTk2MDEwMjExMzU0M1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVlU2Vydm1jZXMgTEExDMIIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAhDUh6j1ACSt057nSxAcwMaGr8Ez87VA2RW2HyY819XoHndnxmP50Cqld
+26AJtltlqHpI1YdtnZ60rVgVhXcVtbvte0lZ3ldEzC3PMvmISBhHs6A3SWHA9ln
InHbToLX/SWqBHL0X78HkPRaG2k0C0HpRy+fg9gvz8HCiQaXCbWNFDHZev90ToNI
xhXBVzIa3AgUnGma1CYZuh5AFVRCEeALG60kxMMC8IoAN7+HG+pMdqAhJxGUCm00
LBvmTGGewhi04MUZwf0kwn9JjQZuyLg6B10D4Y6s0LB2P1MovmSJKGY4JcF8Qu3z
xxUb17Bh9pvzFR5gJN1pjM2n3gJEPwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAJ
UNKM+gIIHNk0G0tzv6vZBT+o/vt+tp81EoZwaPqh1121iw/I7ZvhMLAigx7eyvf
IxUt9/nf8pxWaeGzi98RbSmbap+uxYRynqe1p5rifTam0sguuPrhVp1120gRWLcT
rjg/K60UMXRsmg2w/cxV45pUBcyVb5h60p5uEVAVq+CVns13ExiQL6kk3guG4+Yq
LvP1p4DZfeC33a2Rfre2IHLsJH5D4SdWcYqBsfTpf3FQThH010KoacGrXtsedsxs
9aRd70zuSEJ+mBxmzxSjSwM840oh78DjkdpQgv967p3d+8NiSLt3/n7MgnUy6WwB
KtDujDnB+ttEHwRRngX7
-----END CERTIFICATE-----

```



```
UpsAsBs7phaoN+X/5hIERfbp5L fVnqq54pNG5Knu4KynfW9+kA/WS4cJ6FTTN5t+
y0P1HvcL+BL2RuDy6T2bB21xw5WqtQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQURTVu/Dd4zDnmS5G5CfVlnmUBN0swgY4GA1UdIwSBhjCBg4AURTVu
/Dd4zDnmS5G5CfVlnmUBN0uhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEsBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEsdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJALyTn5IHrIZjMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFt523A3Aug6/F8xxyITgA8gkU0btFh1XNSP
U4U20Q9n0tWI9WqnKNWH3KBxwY5EPitU6b3LM4xc9lDwPz7h2Pto+WhxP9LVKe6f
r8r7teTLCVZ7cfYZHzHg+f1ZjVpAgzE5BVfrR1j3QKpv0hYT3J1wMtI++Vorq5NF
aPjzedehJLhmZVALwnfqfLrgv6/gmraP9Vmoa8U4D6AljNiQGYaLwyoPoRm3bUs2
v1Mh9GkEQ1b9+1pFXcqqzJJTGRuiPCyPbECI79FAnx5JM/CkGJV8H10mjIW1qkK1
Y2qT7wzErrKLJyB53Pw15BdIM1onbDAQreZb0yZQLdoEl/tx7Uk=
-----END CERTIFICATE-----
```

Europe (Frankfurt) — eu-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEsBXIXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKEsdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEsBXIXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEsdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkJ00AQBMIIBHwKBQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJ1/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKml1qx41lHW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFD5GsmkxRuecttwsCG763m3u63UwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhZGUxEDA0
```

```
BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWlgaU2Vydm1jZXMgTExD
MB4XDTI0MDQyOTE1NTUyOVowXDTI5MDQyODE1NTUyOVowXDELMakGA1UEBhMCMVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWlgaU2Vydm1jZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnf0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUuizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUuizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUFD5Gsmkx
RuecttwsCG763m3u63UwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBBh0WaX1BsW56Hqk588MmJxs0rvcKfDjF57RgEDgnGnQaJcStCVWD09UYO
JX2tdsPw+E7AjDqjsuxYaotLn3Mr3mK0sN0Xq9BljBnWD4pARg89KZnZI8FN35HQ
0/LY0VHCknuPL123VmVRNs51qQA9hkPjvw21UzpdLxaUxt9Z/w==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAKD+v6LeR/WrMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUFD5Gsmkx
OTA4MTlaGA8yMTk1MDExNzA5MDg0VowXDELMakGA1UEBhMCMVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2Vydm1jZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAAQACAQ8AMIIB
CgKCAQEAKa8FLhxs1cSJKG+Q+q/vTf8zVnDAPZ3U6oqpp0W/cupCtpwMAQcky8DY
Yb62GF7+C6usniaq/9W6xPn/3o//wti0cNt6MLsiUeHqN15H/4U/Q/fr+GA8pJ+L
npqZDG2tFi1WmVvGhGgIbScrjR4V03TuKy+rZXYvMRk1RXZ9gPhk6evFnviwHsE
jv5AEjxLz3duD+u/SjPp1vloxe2KuWnyC+EKInnka909s14ZAUh+qIYfZK85DAjm
GJP4W036E9wTJQF2hZJrzsib1MGyC1WI9veRISd30izzZL6VVXLXUtHwVHnVASrS
zZDVpzj+3yD5hRXsvFigGhY0FCVFnwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUxC216pvJaRflgu3MudN6zTuP6YcwgY4GA1UdIwSBhjCBg4AUxC21
6pvJaRflgu3MudN6zTuP6YehYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExB
XYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAkD+v6LeR/WrMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAIK+DtbUPppJXFqQMv1f2Gky5/82ZwgbbfXa
HBeGSii55b3tsyC3ZW5Z1MJ7Dtnr3vUkiWbV1EUaZG0UIndUFtXUMABCb/coDndw
CAr53XTv7UwGVNe/AF0/6pQDdPxXn3xBhF0mTKPrOGdvYmjZUtQMSVb91bMwCFfs
w+SwDLnm5NF4yZchIcTs2fdpoyZp0HDXy0xgx01gWhKTnYbaZ0xkJvEvckcxVAwJ
obF8NyJ1a0/pwdjh1HafEXEN81yxyTTY0a0BGTuY0BD2cTYynauVKY4fqHUKr3v
Z6fboaHEd4RFamShM8uvSu6eEFD+qRmvq1codbpsS0huGNLzh0Q=
-----END CERTIFICATE-----
```

Europe (Ireland) — eu-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqQCCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEiExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFNlcnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODAxMDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEiExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIEExMQzCCAbcwggESBgqhkJ00AQBMIIIBHwKBQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAOqgAwIBAgIUakDaQ1Zqy87Hy9ESXA1pFC116HkwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBhZG9uY2VzIEExMQzAeFw0z
ODAxMDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEiExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIEExMQzCCAbcwggESBgqhkJ00AQBMIIIBHwKBQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJA0rmqHuaUt0vMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXlYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTEwMjkw
OTA2MTlaGA8yMTk1MDQwMzA5MDYxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBAClTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWlgaU2Vydm1jZXMgTEExDMIIIBjANBgkqhkiG9w0BAQEFAAOCQAQ8AMIIB
CgKCAQEAEjE7nVu+aHLtzp9FYV25Qs1mvJ1JXD7J0iQ1Gs/RirW9a5ZECctc4ssnf
zQHq2JRVr0GRchvDrbm1HaP/avtFQR/Thvf1twu9AR0VT22dU0TvERdkNzveoFCy
hf52Rqf0DMrLXG8ZmQPPXPDFAv+sVMWCDftcChxRYZ6mP90+TpgYNT1krD5PdvJU
7HcXrkNHDYqbsg8A+Mu2hzl0QkvUET83Csg1ibeK54HP9w+FSD6F5W+6ZSHGJ881
FI+qYKs7xsjJQYgXWfEt6bbckWs1kZiAI0yMzYdPF6ClYzEec/UhIe/uJyUUNfpT
VIsI50ltBbcPF4c7Y20j0IwwI2Sg0QIDAQABo4HUMIHRMAsGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQUF2DgPUZivKQR/Zl8mB/MxIkjZDUwgY4GA1UdIwSBhjCBg4AUF2Dg
PUZivKQR/Zl8mB/MxIkjZDWhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
lYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJA0rmqHuaUt0vMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAgm6+57W5brzJ3+T8/XsIdLTuiBSe5ALgSqI
qn05usUKAeQsa+kZIJPyEri5i8LEodh46DAF1R1XTMYgXXx10YggX88XPmPtok17
l4hib/D9/lu4IaFIyLzYNSzsETyWkWoGve7ZFz60MTRTwY2u8YgJ5dec7gQgPSGj
avB0vTIgoW41G58sfw5b+wjXCsh0nR0on79RcQFFhGnvup0MZ+JbljyhZUYFzCli
31jPZiKzqWa87xh2DbAyyvj2KZrZtTe2LQ48Z4G8wWytJzxEeZdREe4NoETf+Mu5G
4CqoaPR05KwkdNudGNwXewydb3+agdCgfTs+uAjeXKNdSpbhMYg=
-----END CERTIFICATE-----
```

Europe (London) — eu-west-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkiG9w0BAQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXlYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODAxMDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXlYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgqhkiG9w0BAQMIIBHwKBQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
-----END CERTIFICATE-----
```

```

hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buycU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNMp9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUCgCV/DPxYNND/swDgEKGiC5I+EwwDQYJKoZIhvcNAQEL
BQAwXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWlgaU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTE2MjJkxNFoXDTI0MDQyODE2MjJkxNFoXDELMAKGA1UEBhMVCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWlgaU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCChvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnf0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHGDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUuizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUuizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDQoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQoQHEwdT
ZWf0dGx1MSAwHgYDQoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUCgCV/DPx
YNND/swDgEKGiC5I+EwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQATPu/sOE2esNa4+XPEGK1EJSgqzyBSQLQc+VWo6FAJhGG9fp7D97jhHeLC
5vwfmtTAFnGBxadfa0T3ASKxn0ZhXtnRna460LtnNHm7ArCVgXKJo7uBn6ViXtFh
uEEw4y6p9YaLQna+VC8Xtgw6WKq2JXuKzuhuNKSFAgGw9vRcHg==
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJANBx0E2b0CEPMA0GCSqGSIb3DQEBcWUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDQoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDQoQHEwdTZWf0
dGx1MSAwHgYDQoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNjA4MTEEx
NDU2NDJaGA8yMTk2MDExNTE0NTY0M1owXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCgKCAQEArYS3mJLGA1UdIwYDQoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQz
d8QvW306Yleec45M4F2RA3J4hWHtShzsM10JVRt+Yu1GeTf90CP126QmIFfs5nD4

```



```
fgsJQEry2MBSGA9Fxfq3Cw6qkWcr0PsCR+bH0U0XykdK10MnIbpBf0kTfciAupQEA
dEHnM2J1L2iI0NTLbgKxy5PXLH9weX20BFauNmHH9/J070pwL20SN5f8TxcM9+pj
Lbk8h1V4KdIwVQpdWkbDL9BCGLYjyadQJxSxz1J343NzrnDM0M4h4HtVaKOS7bQo
Bqt2ruopLRCYgcuFHck/1348iAmbRQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBGB
wujwU10tpi3iBgmhJMClgZyMMn0aQIxMigoFNqXMUNx1Mq/e/Tx+SNa0EAu0n2FF
aiYjvY0/hX0x75ewzZvM7/zJWIdLdsgewpUq0BH4DXFhbSk2TxggSPb0WRqTBxq5
Ed7F7+7GRIeBbRzdLqmISDnfqey8ufW0ks51XcQNomDIRG5s9XZ5KHviDCar8FgL
HngBCdFI04CMagM+pwT09XN1Ivt+NzUj208ca3oP1IwEAd5KhIhPLcihBQA5/Lpi
h1s3170z1JQ1HZbDrH1pgp+8hSI0DwwDVb3IIH8kPR/J0Qn+hv012H0paUg2Ly0E
pt1RCZe+W7/dF4zsbqWk
-----END CERTIFICATE-----
```

Europe (Milan) — eu-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqwCCQCME1HPdwG37jAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xOTA0MjkyMDM1MjJaFw00
NTA0MjkyMDM1MjJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbYwggErBgcqhkJ00AQBMIIIBHgKBgQDAkoL4YfdMI/MrQ0oL
NPfeEk94eiCQA5xN0nU7+2eVQtEqjFbDADFENh1p3sh9Q90oheLFH8qpSfNDWn/0
ktCS909ApTY6Esx1ExjGSeQq/U+SC2JSuuTT4WFMKJ63a/czMtFkEPPnVIjJJmT
HJSKSsVUgpdDIRvJXuyB0zdB+wIVALQ30LaVGd1PMNfS1nD/Yyn+32wnAoGAPBQ3
7XHg5NLOS4326eFRUT+4oInQFjJjP6dp3p0BEzpImNmZTtkCNNUKE4Go9hv5T41h
R0p0DvWv0CBupMAZVBP90bp1XPCyEIZtuDqVa7ukPOUpQNqQhLLAqkigTyXV0Smt
ECBj9tu5WNP/x3iTZTHJ+g0rhIqpgh012UwJpKADgYQAAoGAV10EQPYQUg5/M3xf
6vE7jKTxxyFWEyjkfJK7PZCz0IGrE/swgACy4PYQW+AwcUweS1K/Hx20aZVUKzWo
wDUbeu65DcRdw2rSwCBTU342sitFo/iGCV/Gjf+BaiAJtxniZze7J1ob8v0BeLv
uaMQmg0YeZ5e0f104GtqP1+lhcQwCQYHkoZIZjgEAwMwADAtAhQdoeWLRkm0K49+
AeBK+j6m2h9SKQIVAIBNhS2a8cQVABDCQXVXrc0t0m08
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICnjCCAZ+gAwIBAgIJA0Z3GEIaDcugMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xOTEwMjQx
NTE5MD1aGA8yMTk5MMDy0TE1MTkw0VowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgt
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2VydmVjZXMgTEExDMIGFMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCjiiPgW3vsXRj4JoA16WQDyoPc/eh3QBARaApJEc4nPIGoUolpAXcjFhWp1o20+
ivgfcCsc4AU90pYdApha3spLey/bhHPri1JZHRNqScKP0hzsCNmKhfnZTIEQCFvsp
DRp4zr91/W506/f1JFBYJ6JHhp0KwM81XQG591V6kkow7QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAGLLrY3P+HH6C57dYgtJkuGZGT2+rMkk2n81/abzTJvsqRqGRrWv
XRKRX1KdM/dfiuYGokDGxiC0Mg6TYy6wvsR2qRhtXW10tZkiHwcQCn0ttz+8vpew
wx8JGMvowtuKB1iMsbwYRqZkFYLCvH+0pfb/Aayi20/ChQLdI6M2R5VU
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAi0gAwIBAgIJA0/+DgYF78KwMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVoQHEwdTZWF0
dGx1MSAwHgYDVoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xOTA0Mjky
MDM1MjJaGA8yMTk4MTAwMjIwMzUyM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAGT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2VydmVjZXMgTEExDMIIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAv1ZLV+Z/P6INq+R1qLkzETBg7sFGKPiwHekbpuB61rRxKHhj8V9vaReM
lInv1Ur5LAPpMPYDsUJ4WoUbPYAqVqyMAo7ikJHCCM1cXgZJefgN6z9bpS+uA3YVh
V/0ipHh/X2hc2S9wvxKWiSHu6Aq9GVpqL035tJQD+NJuqFd+nXrtcw4yGtmvA6w1
5Bjn8WdsP3x0TKjrByYY1BhXpP/f1ohU9jE9dstsRXLa+XTgTPWcWdCS2oRTWPGR
c5Aeh47nnDsyQfP9gLxHeYeQItV/BD9kU/2Hn6mnRg/B9/TYH8qz1RTzLapXp4/5
iNwusrTNexG18BgvAPrfhjDpdgYuTwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQB7
5ya11K/hKgvaRTvZwVv8G1VZt0CGPtNv0i4AR/UN6Tmm51BzUB5nurB4z0R2MoY0
Uts9sLgVsfALJ4otoB77hyNpH3drttU1CVVwal/yK/RQLSon/IoUkaGEbqalu+mH
nYad5IG4tEbmeP456XXc058MKmnczNbPyw3FRzUZQtI/sf94qBwJ1Xo6XbzPKMy
xjL57LHIZCsd+XPifXay690FlsCIgLim11HgPkRIHE0XLSf3dsW9r+4CjoZqB/Z
jj/P4TLCxbYCLkvg1wamjgEWF40img0fhx7yT2X92MiSrs3oncv/IqfdVTiN80Xq
jgnq1bf+EZEZKvb6UCQV
-----END CERTIFICATE-----
```

Europe (Paris) — eu-west-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkiG00AQMDFwxCzAJBgNVBAYTA1VTMRkw
FwYDVoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYD
VoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVoQIEExBXYXNoaW5ndG9u
```

```

IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkJ00AQBMIIIBHwKBgQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLcLnd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buycU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCySfYDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAKGBYqGSM44BAMDlwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUaC9fX57UDr6u1vBvsCsECKBZQyIwDQYJKoZIhvcNAQEL
BQAwXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBZXWlgaU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTE2MzczOFoXDTI0MDQyOTE2MzczOFowXDELMAKGA1UEBhMVCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBZXWlgaU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnf0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHGdAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUizvtUF2
UTgwgZkGA1UdIwSBKTCBjoAUJdbMCBXXtvCcWdwUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUaC9fX57U
Dr6u1vBvsCsECKBZQyIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQCARv1bQEDaMEzYI0nPlu8GHcMXgmgA94HyrXhMMcaI1QwocGBs6VILGVhM
TXP2r3JfFaPEpmXSQNQHvGA13c1KwAZbni8wtzv6qXb4L4muF34iQRHF0nYrEDoK7
mMPR8+oXKKuPO/mv/XKo6XAV5DDERdSYHX5kkA2R9wtvyZjPnQ==
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJALWSfgHuT/ARMA0GCSqGSIb3DQEBwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNzA1MzEx
MTE4MTZaGA8yMTk2MTEwMzExMTg5N1owXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgT

```

```

EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWlGyU2VydmLjZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAY5V7KDqnEvF3DrSProFcgU/oL+QYD62b1U+Naq8aPulJJe127Sm9WnWA
EBd0SASk0aQ9fzjCPoG5SGgWkxYoZjsevHpmzjVv9+Ci+F57bSuMbjgUbvRIFUB
bxQojVoXQPHgK5v4330DxkQ4sjRyUbf4YV1AFdfU7zabC698YgPV0ExGhXP1Tvco
8mlc631ubw2g52j0lzaozUkHPSbknTomhQIv06kUfX0e0TDMH4jLDG2ZIrUB1L4r
0WKG4KetduFrRZyDHF6ILZu+s6ywiMicUd+2U11DFC6oas+a8D11hm0/rpWU/ieV
jj4rWAFrsebnp+Nhgy96iiVUGS2LuQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDE
iYv6FQ6knXCg+sv1caQG9q59xUC5z8HvJZ1+SxzPKKC4PKQdKvIIfe8GxVXq1ZG1
c15WKTDFDMapnzb9RV/DTaVzWx3cMYT77vm1H11XGjhx611CGcENH1egI310TILsa
+KfopuJEEQ9TDMAIkGjha+KieU/U5Ctv9fdej6d0GC60EuwKkTNzPWue6UMq8d4H
2xqJboWsE1t4nybEosvZfQJcZ8jyIYcYBnsG13vCLM+ixjuU5MVVQNMV/gBJzqJB
V+U0QiGiuT5cYgY/QihxdHt99zwGaE0ZBC7213NKr1NuLSrghDI2NLU8NsExq0Fy
0mY0v/xVmQUQ126jJXaM
-----END CERTIFICATE-----

```

Europe (Spain) — eu-south-2

DSA

```

-----BEGIN CERTIFICATE-----
MIIC8DCCAq
+gAwIBAgIGAXjwLk46MAkGBYqGSM44BAMwXDELMakGA1UEBhMCMVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWlGyU2VydmLjZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAY5V7KDqnEvF3DrSProFcgU/oL+QYD62b1U+Naq8aPulJJe127Sm9WnWAEbD0SASk0aQ9fzjCPoG5SGgWkxYoZjsevHpmzjVv9+Ci+F57bSuMbjgUbvRIFUBbxQojVoXQPHgK5v4330DxkQ4sjRyUbf4YV1AFdfU7zabC698YgPV0ExGhXP1Tvco8mlc631ubw2g52j0lzaozUkHPSbknTomhQIv06kUfX0e0TDMH4jLDG2ZIrUB1L4r0WKG4KetduFrRZyDHF6ILZu+s6ywiMicUd+2U11DFC6oas+a8D11hm0/rpWU/ieVjj4rWAFrsebnp+Nhgy96iiVUGS2LuQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDEiYv6FQ6knXCg+sv1caQG9q59xUC5z8HvJZ1+SxzPKKC4PKQdKvIIfe8GxVXq1ZG1c15WKTDFDMapnzb9RV/DTaVzWx3cMYT77vm1H11XGjhx611CGcENH1egI310TILsa+KfopuJEEQ9TDMAIkGjha+KieU/U5Ctv9fdej6d0GC60EuwKkTNzPWue6UMq8d4H2xqJboWsE1t4nybEosvZfQJcZ8jyIYcYBnsG13vCLM+ixjuU5MVVQNMV/gBJzqJBV+U0QiGiuT5cYgY/QihxdHt99zwGaE0ZBC7213NKr1NuLSrghDI2NLU8NsExq0Fy0mY0v/xVmQUQ126jJXaM
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIICmzCCAzygAwIBAgIGAXjwLkiaMA0GCSqGSIb3DQEBCwUAMFwxZAJBgNVBAYTA1VTMRkwFwYDVQQIDDBBXNoaW5nVvR1+45Aey5zn3vPk6xBm5o9grSDL6D2iAuprQnfVXn8CIbSDbWFhA3fi5ippjKkh3s18VyCvCOUXKd0aNrYBrPRkrdH+3m/
rxIUZ2IK1fDlC6sWAjddf6sBrV2w2a78H0H8EuwWiSgttURBjwJ7KPPJCqaqrQIDAQABMA0GCSqGSIb3DQEBCwUAA4GB+FzqQDzun/
-----END CERTIFICATE-----

```

```
iMMzcFucmLM15BxEblrFX0z7IIu0eiGkndmrqUeDCykztLku45s7hxdNy41tTuVAaE5aNBdw5J8U1mRvsKvHLy2ThH6h
+hBgiphYp84DUbWVYeP8YqLEJSqscKscWC
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIIEjCCAvqgAwIBAgIJALWsm06DvSpQMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQLExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0yMjA3MTgx
MzU4NDNaGA8yMjAxMTIyMjEzNTg0M1owXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWVzIGU2VydmIjZXMgTExDMIIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIB
CgKCAQEAAhuSpsHC00/fD2zN1BDpNLRndi9qbHsNeuz3WqN7Samj2aSrM2hS+i
hUxx0BspZj0tZC0sbpPZ+i74N0EQtFeqQoEGvKhB1nJiF4y5I81HDhs5qHvoIivm
7rbbik3zgm1PqS/DmDjVQaXPcD31Rd9ILwBmWEwJqHigyNV1xYtCzTQcrlBrvNZM
dnNgCDAdX/HBEFxx9012xeu0bSt0s+PJWZ1RTbYrNe7LIH6ntUqHxP/ziQ5trXEZ
uqy7aWk1L8uK4jmyNph0lbaqBa3Y6pYmU1nC27UE4i3fnPB0LSiAr+SrWvVx1g4z
ilo8kr+tbIF+JmcgYLBv08Jwp+EUqQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQUwvGzKJL9A5LReJ4Fxo5K6I20xcowgY4GA1UdIwSBhjCBg4AUwvGz
KJL9A5LReJ4Fxo5K6I20xcqhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQLExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJALWsm06DvSpQMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAJAZd31jyoTGLawAD2+v/vQsaB9vZIx5EImi
G8YGkd61uFwEhAmtrwyE/i6FDSIphDrMHBkw/D3BsQK+Ev/JOK/VYuaYDx/8fp
H4cwp9jC57CXzdIDREWNf6M9PsHFg2WA9XNNtC10ZL5WJiJwel8eDSg+sqJUxE01
MW+QChq/20F6niyaRK4bXrZq14as7h+F9u3A9xHE0VP7Zk9C2ehrBXzCMLSDt3GV
fEuMea2RxBmhozw34Hkdb6j18qoCfygubulovRNQjKw/cEmgPR16KfZPP5caILVt
9qkYPvePmbiVswZDee73cDymJYxLqILp0ZwyXvUH8StiH42FHZQ=
-----END CERTIFICATE-----
```

Europe (Stockholm) — eu-north-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqQCCQCWukjZ5V4aZzAJBgqhkiG9w0BAQDAQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQLExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODAxMDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQLExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgqhkiG9w0BAQMIIBHwKBGQCjKvcS2bb1VQ4yt/5e
```

```
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLcLnd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUN1c9U6U/xiVDFgJcYKZB4NkH1QEwDQYJKoZIhvcNAQEL
BQAwXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBZXWlU2Vydm1jZXMgTEExD
MB4XDTI0MDQyOTE2MDYwM1oXDTI1MDQyODE2MDYwM1owXDELMAKGA1UEBhMVCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBZXWlU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCChvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUuizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUuizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWZ0dGx1MSAwHgYDVQQKEXdBbWZ6b24gV2ViIFN1cnZpY2VzIEExMQ4IUN1c9U6U/
xiVDFgJcYKZB4NkH1QEwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBTIQdoFSDRHkppNPUBz9WXR205v/9bpmHojMYZb3Hw46wsaRso7STiGGX/
tRqjIkPUIXsdhZ3+7S/RmhFznmZc8e0bjU4n5vi9CJtQSt+1u4E17+V2bF+D3h/7
wcfE013414Q8JaTDtEf/aF3F0uyBvr4MDM7mFvAMmDmBPS1A==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJALc/uRxxg++EnMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWZ0
dGx1MSAwHgYDVQQKEXdBbWZ6b24gV2ViIFN1cnZpY2VzIEExMQ4IUN1c9U6U/
NDAwMTFaGA8yMTk3MDkxMzE0MDAxMVowXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACeTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBZXWlU2Vydm1jZXMgTEExDMIGfMA0GCSqGSIb3DQEBAQUAA4OCQAQ8AMIIB
```

```
CgKCAQEazwCGJEJIxqtr2PD2a1mA6LhRzKhTba1AZsg3eYfpETXIV1rpojMfvVoN
qHvGshWLgrGTT6os/3gsaADheSaJKavxwX3X6tJA8fvEGqr3a1C1MffH9hBwbQqC
LbfUTAbkwis4GdTUw0wPjT1Cm3u9R/VzilCNwkj7iQ65AFai8Enmsw3UGldEsop4
yChkB3KW3WI0FTh0+gD0YtjrqqYJxpG0YBpJp5vwd3fZ4t1vidmDms7liv4f9Bx
p0oSmUobU4GUlFhBchK1DukICVQdn0VzdMonYm7s+HtpFbVHR8yf6QoixBKGdSa1
mBf7+y0ixjCn0pnC0VLVooGo4mi17QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
40NZiixgk2sjJctwbyD5WKLTH6+mxYcDw+3y/F0fwz561YORhP2FNnPOmEkf0S1/
Jqk4svzJbCbQeMzRoyaya/46d7UioXMRZam5IaGBh0dQbi97R4VsQjwQj0RmQsq
yDueDyuKTwwLK9KvI+ZA6e6bRkdNGf1K4N8GGKQ+fBhPwVELkbT9f160Jkezeen
S+F/gDADGJgmPXfjogICb4Kvshq0H5Lm/xZ1DULF2g/cYhyNY6E0I/eS5m1I7R8p
D/m6WoyZdpInxJfxW6160MkxQMRVsruLTNGtby3u1g6ScjmpFtvAMhYeJBSdzKG4
FEyxIdEjoe01jhTsck3R
-----END CERTIFICATE-----
```

Europe (Zurich) — eu-central-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXiKJnMAKGBYqGSM44BAMwXDELMaKGA1UEBhMVCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s50f2EbdSP09EAMMeP4C2USZpRV1AI1H7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdirmVClpJ+f6AR7ECLCT7up1/63xhv401fnxqimFQ8E
+4P208UewwI1VBNAfPEy9nXzriith1yrv8iIDGZ3RSAHHAhUA12BQjxUjC8yykrmCouEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v60uqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFh03zwyjMim4TwWeotUfI0o4K0uHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZ16Ae1ULZAFM0/7PSSoDgYQAAoGAYNjaCNg/
cfGQ011BUj5C1UulqwZ9Q+SfDzPZ9D2C0VbiRANiZoxrV8RdgmzzC5T7VcriVwjwvta2Ch//
b+sZ86E5h0XWwR+BeEjD9cu3eDj12XB5sWEbNHNx49p5Tmtu5r2LDt1L8X/
Rpfalu2Z20JgjFJWGF7hRwx456n
+lowCQYHkoZIZjgEAWMvADAsAhRChsLcj4U5CVb2cp5M0RE1XbXmhAIUeGSnH+aiUQIWmPEFja+itWDufIk=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICmzCCAzygAwIBAgIGAXjSGFGiMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIDBBXYXNoaW50
opKZAUusJx2hpgU3pUhh1p9Ath/VeVD582jTd9IY
+8t5MDa6Z3fGliByEiXz0LEHdi8MBacLREu1TwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAILlpoE3k9o7KdALAXsFJNi
+g3RMzdbiFM+7MA63Nv5fsf+0xgcjSNBELvPCDKFvTJ14Q0hToy0561105GvdS9RK
+H8xrP2mrqngApoKTApv93vHBixgFSn5KrczR00YSm30jkqbydU7DFlmkXXR7GYE+5jbHvQHYiT1J5sMu
-----END CERTIFICATE-----
```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIIIEejCCAvqgAwIBAgIJALvT012pxTxNMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXlYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWw6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0yMjA3MTgx
NTEyMDdaGA8yMjAxMTIyMjE1MTIwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBACTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBxZWlgaU2Vydm1jZXMgTEExMIIBIjANBgkqhkiG9w0BAQEFAAOCQAQ8AMIIB
CgKCAQEAYn+Lsnq1ykrfYlZkk6aAAYNREnd9Iw8AUwCBkg0r2eBiBBepYxHwU85N
++moQ+j0EV2VaahBeTLShGZZS1HsyK8+cYT2QzpgHoamcYhrPXyIxlWiRQ1aqSg
0FiE9bsqL3rCF5Vz+t0iTe5W/7ojf0Fls6++g7ZpobwJlpMbuJepqyeHMPyjv05A
age811Jewc4bxo2ntaW0HCqNksqfYB78j6X6kn3PFpX7FaYAwZA+Xx6C7UCY7rNi
UdQzfa08htfJi4chz7frpUdQ9k13I0QrsLshBB5fFUjl09NiFipCGBwi+8ZMeSn1
5qwBI01BWXPFg7WX60wyjhmh6JtE1wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHGDAD
BgNVHQ4EFgQU8HN4vvJrsZgPQeksMBgJb9xR1yYwgY4GA1UdIwSBhjCBg4AU8HN4
vvJrsZgPQeksMBgJb9xR1yahYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExB
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWw6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJALvT012pxTxNMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAG1HYDtC HpfBvdHx9HeQE8HgNugJUPdEqxun
t9U33p8VFrs+uLPtr0d9HDJEGvvs5h84EUie/oGJxRt7V1Vlid1PvHf6cRmpjgqY
YdggAVkZtY/PnFvmzf2bMV1SQPrqCl7U0zaw2Kvnj4zgX0rZyCetgrRZSUSxotyp
978W9y9ccXwVSeYG/YAr5rJpS6ZH7eRQvUY0IzwFNea0Pg0TEVpcjW1V6+MQEvsEx
W85q+s6AVr49eppEx8SLJs10C23yB+L+t32tAveQImRWtJMpZ5cxh/sYgDVeOC0
85H1NK/7H9fAzT1cPu1oHSnB0xYzzHG0AmXmusMfwUk8fL1RQkE=
-----END CERTIFICATE-----

```

Israel (Tel Aviv) — il-central-1

DSA

```

-----BEGIN CERTIFICATE-----
MIIC7zCCAq+gAwIBAgIGAX0QPj
+9MAkGBByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBACMB1NlYX
U4EddRipUt9KnC7s50f2EbdSP09EAMMeP4C2USZpRV1AI1H7WT2NWPq/
xfw6MPbLm1Vs14E7gB00b/JmYldrmVC1pJ+f6AR7ECLCT7up1/63xhv401fnxqimFQ8E
+4P208UewwI1VBNAfPey9nXzrith1yrv8iIDGZ3RSAHHAhUA12BQjxUjC8yykrmCouEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v60uqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBGLRJFnEj6EwoFh03zwykjMim4TwWeotUfI0o4K0uHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZ16Ae1U1LZAFM0/7PSSoDgYQAAoGAbazCL5XXyPmcw3+oMYQUF5/9YogW6D0FZbYuyPgj0oUwWdl6fj1zWca

```



```

pq+11ezuK2DF0zNTEyPEwwCQYHKoZIzjgEAwMvADAsAhRt1jKpXsvrS
+xTo2M9h2s2uLAhEQIU0Z2FcNtSrshF2EIdixZZwtNv66Q=
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIICmzCCAZygAwIBAgIGAX0QQGVLMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIDBBXYXNoaW50
+S8v0y5hpLoRe4Rk0rY0cM3bN07GdEMlin5mU0y1t8y3ct4YewvmkgT42kTyMM
+t1K4S0xsqjXxxS716uGYh7eWtkxrCihj8AbXN/6pa095h
+7TZy12n83keiNUzM2KoqQVMwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBADwA6VVEIIZD2YL00F12po40xDLzIc9XvqFPS
FmU7H8s62/jD6c0R1A1cClIyZUe1yT1ZbPySCs43J+Thr8i8FSRxxzDBSZZi5foW
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJA0Vp1h2I9wW7MA0GCSqGSIb3DQEBwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKEXdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0yMjA3MTUx
MjQ0MTJaGA8yMjAxMTIx0TEyNDQxMjE0YXQwMDEwXDE1MDEwMDEwMDEwMDEwMDEw
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBACTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWJgU2VydmVjZXMgTEwMIIBIjANBgkqhkiG9w0BAQEFAAOCQAQ8AMIIB
CgKCAQEAl3PkyWv161iV/SYf01UF076UpDfPm2SF/Rz/o33cm699X++EYPxTnoEc
vmWeS0I7eDXc40CUiToG0sEx0k1E0CX1Z1tK6qJ+zgWQLZ9SZEC9H0NsSA6LhrHu
Nq0dzeK3LjhdFcX46/4GqdiptpdTuM4m/h0Q5yx4JMQ/n1sdpv4M5VLRwWw9Lem
ufb79Id709SispxgRsz1KXIjp7N9S4BY7itSXz97uSyzTqEjWZ6mDUhTu3t21GKC
6f1ALGTTTrG2yghEhz53rkvLsvwzjPSS1T6LI f0mrRPzHaf+EdaKoasELE1SHh+ZH
9mI81HywPE+HZ+W+5hBCvjYp90Y1fwIDAQABo4HUMIHRMASGA1UdDwQEAwIHGDAd
BgNVHQ4EFgQU58tN2J0+yEGq5JbIXxGi4vRVPyIwgY4GA1UdIwSBhjCBg4AU58tN
2J0+yEGq5JbIXxGi4vRVPyKhYKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJA0Vp1h2I9wW7MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANBN0e1EqNy4+IU2yQzMJ+Wy5ZIOtTP6GSBR
7muVY1bDeAwtNTE0pwgrZV1C7/xq5Q0LC1y0Z70hHXEF8au7qStaAoUtxzvHTAZI
NC01woFU56UFw4N0vZII17iqEfoqRC4PpI30xqEJHFy0VL1vAzJoKB4QLLqDAYVA
LXCi0LoVT+y9tRysxw5My00Bi6fxQIIAD12bE9xkunTN1Jkkwqo3LxNy/ryz4QWR
8K7jHUItifv4h/hxBKpHEquN8CkdvM9oeG17I8PFrSFEpGr1euDXy0euZzzYiDBV
m6GpTJgzpVsEuIX52dPcPemwQncoIfZyhWDW85MJUnby2WTEcFo=
-----END CERTIFICATE-----

```

Middle East (Bahrain) — me-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCVWIGSmP8RhTAJBgcqhkj00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbgwgEsBgcqhkj00AQBMIIbHwKBgQDcwojQfgWdV1Q1i00B
8n6cLZ38VE7ZmrjZ90QV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q
PH1P1WGL8IZ34BUgRTtG4TVo1vp0smjkmVyrU5hIdKtzjV93Ccx15gVgyk+o1IEG
fZ2Kbw/Dd8JfoPS7KaSCmJKxXQIVAIzIaDFRga2qcMkW2HWASyND17bAoGBANTz
IdhfMq+12I5iofY2oj3HI21Kj3LtZrWEg3W+/4rVhL31Tm0Nne1r19yGujrjQwy5
Zp9V4A/w9w2010Lx4K6hj34Eefy/aQnZwNdnhv/FQP7Az0fju+Y16L1300HQrL0z
Q+9cF7zEosekEnBQx3v6psNknKgD3Shgx+G0/LpCA4GFAAKBgQCVS7m77nuNA1Z8
wvUqcooxXMPkxJF154NxAsAu19KP9KN4svm003Zrb7t2F0tXRM8zU3TqMpryq1o5
mpMPsZDg6RXo9BF7Hn0DoZ6PJTamkFA6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr
12AztQ8bFwSrTgTzPE3p6U5ckcgV1TAJBgcqhkj00AQDAy8AMCwCFB2NZGwm5ED1
86ayV3c1PEDukgQIAhQow38rQkN/VwHvEsw9DqEshXHjuQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDPDCCAqWgAwIBAgIJAM16uIV/zqJFMA0GCSqGSIb3DQEBCwUAMHIXCzAJBgNV
BAYTA1VTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0dGx1MSAw
HgYDVQQKDBdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzEaMBGGA1UEAwRZWMyLmFt
YXpvbmF3cy5jb20wIBcNMTEwNDI2MTQzMjQ3WhgPMjE5ODA5MjE5NDMyNDdaMHIX
CzAJBgNVBAYTA1VTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0
dGx1MSAwHgYDVQQKDBdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzEaMBGGA1UEAwR
ZWMyLmFtYXpvbmF3cy5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALVN
CDTZEEnIeoX1SEYqq6k1BV0Z1pY5y3Kno0reCAE589TwS4MX5+8Fzd6AmACmugeBP
Qk7Hm6b2+g/d4tWycyXLaQ1cq81DB1GmXehRkZRgGeRge1ePwD1TUA0I8P/QBT7S
gUePm/kANSFU+P7s7u1NN1+vynyi0wUUr7/wIZTAgMBAAGjgdwgdQwHQYDVR00
BBYEFILtMd+T4YgH1cgc+hVsV0V+480FMIGkBgNVHSMGZwwgZmAFILtMd+T4YgH
1cgc+hVsV0V+480FoXakdDBYMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGlu
Z3Rvb3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3RvZ3Rv
aWNLcyBMTEMxGjAYBgNVBAMMEWVjMi5hbWF6b24gV2VzIEExMQzEaMBGGA1UEAwR
DAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0BAQsFAA0BgQBhKNTBIFgWfd+ZhC/LhRUY
40jEiykmbEp6hlzQ79T0Tfbn5A4NYDI2icBP0+hmf6qSnIhwJF6typyd1yPK5Fqt
NTpxxcXmUKquX+pHmIkK1LKD08rNE84jqxrxRsfdi6by82fjVYf2pgjJW8R1FAw+
```



```
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
```

```
MIICMzCCAZygAwIBAgIGAXjRrnDjMA0GCSqGSIb3DQEEBQUAMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIDBBXYXNoaW50
KyA6zyruJQrYy00a6wqLA7eeUzk3bMiTkLsTeDQfrkaZMfBAjGaa0ymRo1C3qzE4rIenmahvUp1u9ZmLwL1idWXMxR2R
+d2SeoK0KQWoc2U0FZMHYxDue7zkyk1CIRaBukTeY13/
RIr1c6X61zJ5BBtZX1HwayjQIDAQABMA0GCSqGSIb3DQEEBQUAA4GBABTqTy3R6RXKPW45FA+cgo7YZEj/
Cnz5YaoUivRRdX2A83BHuBTvJE2+WX00FTEj4hRVjameE1nEno08Z7fUV1oAFD1Do69fhkJesvn51D1WRrPnoWGgEfr1
B+Wqm3kVEz/QNcz6npmA6
```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
```

```
MIIEEjCCAvqgAwIBAgIJAM4h7b1CVhqqMA0GCSqGSIb3DQEBQwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0yMjA0MTEy
MDE1MDNaGA8yMjAxMDkxNTEwMTUwM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhZGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBhZG9uU2Vudm1jZXMgTEExMDE1MjA0MTEyMDE1MDNaGA8yMjAxMDkxNTEw
M1UwM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
CgKCAQEApYbTWFm0hSoMpqPo72eqAmnn1dXGZM+G8EoZXzWwT/+IHEXNB4q5N6k
tudYLre1bJxuzEw+iProSHjmb9bB9YscRTofjVhB1t35Fc+i8BaMeH94SR/eE8Q0
m1l8gnLNW3d62lyuhzuyv1e5wV1RqzYw+X2zRH4/wRD0C0pzjKoHIgyPKsMgsw5
aTZhNMsGxZN9dbkf0iCGeQLDytwU/JTh/HqvSr3VfU0apTJJiyAxCtZWgp1/7wC
Rv0CSMRJobpUqxZgl/VsttWnkikSFz1wGkcYeSQvk+odbnYQckA8tdddoVI56eD4
qtREQvfpmAX5v7fcqLex15d5vH8uZQIDAQABo4HUMIHRMASGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU0adrTs+0hzwoAgUJ7RqQNdwufkwyY4GA1UdIwSBhjCBg4AU0adr
bTs+0hzwoAgUJ7RqQNdwufmhyKReMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQ4IJAM4h7b1CVhqqMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAICTdA0GE0nII8HaGCpCB8us/hGFaLptJaAf
D5SJAyVy66/mdfjGzE1BKkKxnbxemEVUIzbrid0nyilB+pKwN3edAjTZtWdpVA0V
R/G/qQPmcV1jtycBz4VC6Su0UYf1GzLH1GZ6GJWbuDtFzw8r7HGdRN1wrEPe3UF2
sMpuVezqnRUdvVRoVQP4jFgNsE7kNvtN2NiPhb/CtrpcwIQ7r6YeoHcBSheuV1Z
xZDHynC3KUpRQGx1+Z9QqPrDf180MaoqALT14+W6Pr2NJYrVUFGS/ivYshMg5741
CPU6r4wWZSKwEUXq4BInYX6z6iclp/p/J5QnJp2mAwyi6M+I13Y=
```

```
-----END CERTIFICATE-----
```

South America (São Paulo) — sa-east-1**DSA**

```

-----BEGIN CERTIFICATE-----
MIIC7TCCAqQCCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEiExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEiExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkJ00AQBMIIIBHwKBGQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buycU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmve5f8LIE/Gf
MNMp9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIIDITCCAOqgAwIBAgIUX4Bh4MQ86Roh37VDRRX1MN0B3TcwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0
BgNVBAClTB1N1YXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBxZWVzIGU2Vydm1jZXMgTEEx
MB4XDTE0MDQyOTE2NDYwOVVoXDTI1MDQyODE2NDYwOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClTB1N1YXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBxZWVzIGU2Vydm1jZXMgTEExDMIGFMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvP12anJ0+eIB
UqPfQG09kZ1wpWpmy08bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13h0IcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXXtvCcWdwUuizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXXtvCcWdwUuizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTA1VTMRkwFwYDVQQIEiExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdT
ZWF0dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQ4IUX4Bh4MQ8
6Roh37VDRRX1MN0B3TcwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AA0BgQBhocfH6ZIX6F5K9+Y9V4HFk8vSaaKL5ytw/P5td1h9ej94KF3xkZ5fyjN
URvGQv3kNmNJBONarcP9I7JIMjsNPmVzqWawyCEGCZImoARxSS3Fc5EAs2PyBfcD
9nCtzMTaK009Xyq0wqXVYn1xJsE5d5yBDsGrzaTHKjxo61+ezQ==
-----END CERTIFICATE-----

```



```
P2ZT52/sH9orlck1n2z08xPi7MItgPHQwu30xsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwH0/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJs1Fc190TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFa0dy0+00aECAwEAATANBgkqhkiG9w0BAQsFAA0CAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7z1+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/70ynRyfQetJVGichaaxLNM31cr6kcx0owb+WQQ84cwrB3keykH4gRX
KHB2r1WSxta+2panSE01JX2q5jhcFP90rD0tZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlInIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+611MVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19id0UqxPxWsasWxQX00azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALS0Mb0oU2svMA0GCSqGSIb3DQEBCwUAMFwx CzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXyXNoaw5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwx CzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
YXNoaw5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQzCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEA
uhhUN1qAZdcWWB/OSDVGk30A99EFz0n/mJlmcIQ/Xwu2dFJWmSCqEAE6gjuFjQ
q3voxAhC2CF+e1KtJW/C0Sz/LYo60PUqd6iXF4h+upB9Hk00GuWHXsHBTsvgkgGA
1CGge14U0Cdq+23eANr8N8m28Uz1jjSnT1rYCHtzN4sCAwEAa0B1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR00BBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIG0BgNVHSME
gYYwgY0AFBkZu3wT27NnYgrfH+xJz4HJaNJoWcKXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFd1YiBTZXJ2aWN1cyBMTE0CCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqzm117e8Hgb
oADS0ph+YUz5P/bUCm61wFj1xaTfwKcuTR3ytj7bFLow5Bm7Sa+TC1310Gb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUEk+MRiWu+0h5/1JGii3qw4YByx
SUD1RyNy1jJFstEZj0hs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJA0trM5XLDSjCMA0GCSqGSIb3DQEBCwUAMFwx CzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXyXNoaw5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTA4MTQx
MDAxNDJaGA8yMTk1MDExNzEwMDE0MDEwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWJgU2Vydm1jZXMGTEExDMIIIBjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
```

```
CgKCAQEAvVBz+wQNdPiM9S+aUUL0QEriTmNDUurjLWlr7Sfa0JScBzis5D5ju0jh1
+qJdkbuGKtFX50TWtm8pWhInX+hIOoS3exC4BaANoa1A3o6quoG+Rsv72qQf8LLH
sgEi6+LM1CN9TwnRK0ToEabmDKorss4zF17VSsbQJwcBSf0cIwbdRRaW9Ab6uJHu
79L+mBR3Ea+G7vSDrVIA8goAPkae6jY9WGw9Kxs0rcvNdQoEkqRVtHo4bs9fMRHU
Etphj2gh40bX1FN92VtvzD6QBs3CcoFWgyWGvzg+dNG5VCbsiiuRdmii3kciZ3H
Nv1wCcZoEAqH72etVhsuvNRC/xAP8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA8
ezx5LRjzUU9EYWyhyYIEShF1P1qDHs7F4L46/51c4pL8FPoQm5CZuAF31DJhYi/b
fcV7i3n++/ymQbCLC6kAg8DUB7NrcR0l15ag8d/JXGzcTCnLDXLXx1905fPNa+jI
0q5quTmdmiSi0taeaKZmyUdhrB+a7ohWdSdlokEI0tbH1P+g5y113bI21eYE6Tm8
LKbyfK/532xJPq09abx4Ddn89ZEC6vvWVNDgTsxERg992Wi+/xoSw3XxkgAryIv1
zQ4dQ6irFmXwCWJqc6kHg/M5W+z60S/94+wGTXmp+19U6Rkq5jVMLh16XJXrXwHe
4KcgIS/aQGVgjM6wivVA
-----END CERTIFICATE-----
```

China (Ningxia) — cn-northwest-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAhh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBCMQswCQYDVQQGEwJV
UzEZMBCGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk10TE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxZzA5BjBBAQsFAA0CAQEAASzN2+0E
aW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKEXdBbWF6b24g
V2ViIFN1cnZpY2VzIEExMQzCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/1
gsw9+QSnEJeYWnmivJW0Bdn9CyDpN7cpHVmeGgNjL2fvImWyWe2f2Kq/BL917N7C
P2ZT52/sH9orlck1n2z08xPi7MIItgPHQwu30xsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBmatPHhRBqhwH0/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJs1Fc190TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWfa0dy0+00aECAwEAATANBgkqhkiG9w0BAQsFAA0CAQEASzN2+0E
V1BfR3DPWJHWRf1b7z1+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6k7G6EtqhZUqteY7
zAceoLrVu/70ynRyfQetJVGichaaxLNM31cr6kcx0owb+WQQ84cwrB3keykH4gRX
KHB2r1WSxta+2panSE01JX2q5jhcFP90rD0tZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZ1nIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+611MVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFhbQp1peGC19id0UqxPxWsasWxQX00azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
```



```

MIIDCzCCAnSgAwIBAgIJALS0Mb0oU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWw6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEExB
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWw6
b24gV2ViIFN1cnZpY2VzIEExMQzCBnzANBghkqhkiG9w0BAQEFAA0BjQAwgYkCgYEA
uhhUN1qAZdcwWB/OSDvDgk30A99EFz0n/mJlmcIQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+e1KtJW/C0Sz/LYo60PUqd6iXF4h+upB9Hk00GuWHXsHBTsvgkgGA
1CGge14U0Cdq+23eANr8N8m28Uz1jjSnT1rYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR00BBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIG0BgNVHSME
gYYwgY0AFBkZu3wT27NnYgrfH+xJz4HJaNJoWcKXjBcMQswCQYDVQQGEwJVUzEZ
MBCGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFd1YiBTZXJ2aWw1cyBMTE0CCQ0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqzm117e8Hgb
oADS0ph+YUz5P/bUCm61wFj1xaTfwKcuTR3ytj7bFLow5Bm7Sa+TC1310Gb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUEK+MRiWu+0h5/lJGii3qw4YByx
SUD1RyNy1jJFstEZj0hs
-----END CERTIFICATE-----

```

RSA-2048

```

-----BEGIN CERTIFICATE-----
MIID0zCCAi0gAwIBAgIJAPu4ssY3B1zcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWw6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTEyMDMy
MTI5MzJaGA8yMTk1MDUwODIxMjkzMlowXDELMAKGA1UEBhMVCVVMxGTAXBgNVBAGT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClBTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBhZG9uIFd1YiBTZXJ2aWw1cyBMTE0CCQ0jjGzqFNrLzASBgNVBAoTF0Ft
CgKCAQEAs0iGi4A6+YTLzCdIyP8b8SCT2M/6PGKwzKJ5XbSBol3gsnSWiFYqPg9c
uJPNbiy9wSA9vlyfWmd90qvTfiNrT6vewP813QdJ3EENZ0x4ERcf/Wd22tV72kxD
yw1Q3I10MH4b0ItGQAxU50tXCjBZEEUZoo0kU8RoUQ0U2Pq14NTiUpzWacNutAn5
HHS7MDc4lUlsJqbN+5QW6fFrcNG/0Mrib3JbwdFUNhrQ5j+Yq5h78HarnUivnX/3
Ap+oPbentv1qd7wvPJu556LZuhfqI0TohiIT1Ah+yUdN5osoaMxTHKktf/CsSJ1F
w3qXqFJQA0VwsqjFyHXFI32I/G0upwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCn
Um00QHvUsJSN6KATbghowLynHn3wZSqsuS8E0C0pcFJFxp2SV0NYkERbXu0n/Vhi
yq5F8v4/bRA2/xpedLWmvFs7QWlomuXhSnYFkd33Z5gnXPb9vRkLwiMSw4uX1s35
qQrarczUJ9EXDhrv7VmngIk9H3YsxYr1DGEqh/oz4Ze4UL0gnfkauanHikk+BUeSg
/jTD+7e+niEzJPihHdsVKFD1ud5pakEzyxovHwNJ1GS2I//yxrJFIL91mehjqEk
RLPdNse7N6UvSnuXc0okwu616kfzigGkJBxkcq4gre3szZFdCQCuioj7Z4xtuTL8
YMqfiDtN5cbD8R8ojw9Y
-----END CERTIFICATE-----

```

AWS GovCloud (US-East) — us-gov-east-1**DSA**

```

-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkJ00AQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIEiExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEiExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgcqhkJ00AQBMIIbHwKBgQCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buycU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmeve5f8LIE/Gf
MNMp9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3Igb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----

```

RSA

```

-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAIE9Hnq8207UMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVQQIEiExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQKKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0yMTA3MTQx
NDI3NTdaFw0yNDA3MTMxNDI3NTdaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIEiExB
XYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQKKExdBbWF6
b24gV2ViIFN1cnZpY2VzIEExMQzCBnzANBgcqhkiG9w0BAQEFAA0BjQAwgYkCgYEA
qaIcGFFTx/S01W5G91jHvyQdGP25n1Y91aXCu00WAUTvSvNGpXrI4AXNrf+CmIO
C4beBASnHCx082jYudWBB19Wiza0psYc9f1rczSzVLMmN8w/c78F/95NfiQdnUQP
pvgqcMeJo82cgHkLR7XoFWgMrZJqrcUK0gnsQcb6kakCAwEAAa0B1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR00BBYEFNWV53gWJz72F5B1ZVY40/dfFYBPMIG0BgNVHSME
gYYwgY0AFNWV53gWJz72F5B1ZVY40/dfFYBPoWckXjBcMQswCQYDVQQGEwJVUzEZ
MBCaGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aW50cyBMTE0CCQCHvR56vNju1DASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBACrKjWj460GUPZCGm3/z0dIz
M2BPuH769wc0sqfFZcMKEysSFK91tVtUb1soFwH4/Lb/T0PqNrvtEwD1Nva5k0h2
xZhNNRmDuh0hW1K9wCcnHGRBwY5t41YL6hNV6hcrqYwGMjTjCAjBG2yMgzsnSNFle
Rwi/S3BFXISixNx9cILu
-----END CERTIFICATE-----

```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAi0gAwIBAgIJALPB6hxFhay8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTA1VTMRkwFwYDVoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVoQHEwdTZWF0
dGx1MSAwHgYDVoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xODA0MTAx
MjMyNDlaGA8yMTk3MDkxMzEyMzI0OVowXDELMAkGA1UEBhMCMVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAClB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBZXWV2VydmljZXMgTEExDMIIIBIjANBgkqhkiG9w0BAQEFAAOCQAQ8AMIIB
CgKCAQEAv9xsI9237KYb/SPWmeCVzi7giKNron8hoRDwlwwMC9+uHPd53UxzKLB
pTgtJWAPkZVxEdl2Gdhwr3SULoKcKmkqE61tVFrVuPT33La1UufguT9k8ZDDu09C
hQNHUdSVEuVrK3bLjaSsM0S7Uxmnn71YT990IREowvnnBNBsBlcabfQTBV04xfUG0
/m0XUiUFj0xDBqbNzkeIb1W7vK7ydSjTfMS1jga54UAVXibQt9EAI7B8k912iLa
mu9yEjyQy+ZQICTuAvPUEWe6va2CHVY9gYQLA31/zU0VBKZPTNExjaqK4j8bKs1/
7d0V1so39sIGBz21cUBec1o+yCS5SwIDAQAQMA0GCSqGSIb3DQEBCwUAA4IBAQBt
h02W/Lm+Nk0qsXW6mqQFsAou0cASc/vtGNCyBfoFNX6aKXsVCHxq2aq2TUKWENs+
mKmYu1lZVhB0mLshy1lh3RRoL30hp3jCwXytkWQ7E1cGjDzNGc0FArzB8xFyQNdK
MNvXDi/ErzgrHGSpvcvmGHi0hMf3UzChMwB1r6udoD1MbSI07+8F+jUJkh4X111Kb
YeN5fsLZp7T/6YvbFSPpmbn1YoE2vKtuGKx0bRrhU3h4JHdp1Ze11pZ61h5iM0ec
SD11SximGIYCjfZpRqI3q50mbxCd7ckULz+UUPwLrf0ds4VrVVSj+x0ZdY19Plv2
9shw5ez6Cn7E3IfzqNH0
-----END CERTIFICATE-----
```

AWS GovCloud (US-West) — us-gov-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgqhkiG9w0AQMDFwxCzAJBgNVBAYTA1VTMRkw
FwYDVoQIEExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYD
VoQKExdBbWF6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0xMjAxMDUxMjU2MTJhFw0z
ODA0MDUxMjU2MTJhMFwxCzAJBgNVBAYTA1VTMRkwFwYDVoQIEExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVoQHEwdTZWF0dGx1MSAwHgYDVoQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIEExMQzCCAbcwggEsBgqhkiG9w0AQMIIIBHwKBGCjKvcS2bb1VQ4yt/5e
ih5006kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7Eg1K9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwwHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyCU1CiYQk40KNHCcHfNiZbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZr0LBA4GEAAKBgEbmve5f8LIE/Gf
```

```
MNmP9CM5eovQ0Gx5ho8WqD+aTebS+k2tn92BBPqeZqpWRa5P/+jrdKm11qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm00bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDlwAwLAIUwXB1k40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6R0k0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAIe9Hnq8207UMA0GCSqGSIb3DQEBCwUAMFwx CzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXyXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWw6b24gV2ViIFN1cnZpY2VzIEExMQzAeFw0yMTA3MTQx
NDI3NTdaFw0yNDA3MTMxNDI3NTdaMFwx CzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBX
YXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWw6
b24gV2ViIFN1cnZpY2VzIEExMQzCBnzANBghkhiG9w0BAQEFAA0BjQAwwYkCgYEA
qaIcGFFTx/S01W5G91jHvyQdGP25n1Y91aXCu00WAutvSvNgpXrI4AXNrqF+CmIO
C4beBASnHCx082jYudWBB19Wiza0psYc9f1rczSzVLMmN8w/c78F/95NfiQdnUQP
pvqqcMeJo82cgHkLR7XoFwGMrZJqrcUK0gnsQcb6kakCAwEAAa0B1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR00BBYEFNWV53gWJz72F5B1ZVY40/dfFYBPMIG0BgNVHSME
gYyWgY0AFNWV53gWJz72F5B1ZVY40/dfFYBPoWckXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFd1YiBTZXJ2aWw1cyBMTE0CCQCHvR56vNju1DASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBACrKjWj460GUPZCGm3/z0dIz
M2BPuH769wc0sqfFZcMKEysSFK91tVtUb1soFwH4/Lb/T0PqNrvtEwD1Nva5k0h2
xZhNNRmDuh0hW1K9wCcnHGRBwY5t41YL6hNV6hcrqYwGMjTjCAjBG2yMgzsnSNFle
Rwi/S3BFXISixNx9cILu
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIID0zCCAiOgAwIBAgIJANCOF0Q6ohnuMA0GCSqGSIb3DQEBCwUAMFwx CzAJBgNV
BAYTA1VTMRkwFwYDVQQIEExBXyXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0
dGx1MSAwHgYDVQQKExdBbWw6b24gV2ViIFN1cnZpY2VzIEExMQzAgFw0xNTA5MTAx
OTQyNDdaGA8yMTk1MDIxMzE5NDI0N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDA0BgNVBAcTB1N1YXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWlgaU2Vydm1jZXMgTEExMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAzIcGTzNqie3f1o1rrqcFzGfbymSM2QfbTzDIOG6xXXeFrCDAm0q0wUhi
3fRCuoeh1K0WAPu76B9os71+zgF22dIDEVkpqHCjBrGzDQZXUw0zhm+PmBUI8Z1
qvbVD4ZYhjCujWzrsX6Z4yEK7PEFjtf4M4W8euw0RmiNwjy+knIFa+VxK6aQv94
1W98URFP2fD84xedHp6ozZ1r3+RZSIFZs0iyxYsgiwTbesRMI0Y7LnkKGCiHQ/XJ
0wSISWaCddbu59BZeAdnyh14f+pWaSQpQQ1DpXvZAVBYvCH97J1oAxLfh8xcwgsQ
/se3wtn095VBt5b7qTVj0vy6vKZazwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA/
```

```
S8+a9csfASkdtQUOLsBynAbsBCH9Gykq2m8JS7YE4TGvq1pnWehz78rFTzQwmz4D
fwq8byPk16DjdF9utqZ0JUo/Fxe1xom0h6oievtB1SkmZJNbgc2WYm1zi6ptViup
Y+4S2+vWZyg/X1PXD7wyRWuETmykk73uEyeWFBYKCHws09sI+6204Vf8Jkuj/cie
1NSJX8fkerVfLrZSHBYhxLbL+actVEo00tiyZz8GnhgWx5faCY38D/k4Y/j5Vz99
71UX/+fWHT3+1TL8ZZK7f0QWh6NQpI0wTP9KtWqf0UwMIbgFQPoxkP00TWRmdmPz
W0wT0bEf9ouTnjG90Z20
-----END CERTIFICATE-----
```

Instance identity roles

Each 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** – [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 `AWSsSystemsManagerDefaultEC2InstanceManagementRole` 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-0123456789example
```

For more information about ARNs, see [Amazon Resource Names \(ARNs\)](#) in the *IAM User Guide*.

Identify EC2 Linux instances

You might need to determine whether your application is running on an EC2 instance.

For information about identifying Windows instances, see [Identify EC2 Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Inspect the instance identity document

For a definitive and cryptographically verified method of identifying an EC2 instance, check the instance identity document, including its signature. These documents are available on every EC2 instance at the local, non-routable address `http://169.254.169.254/latest/dynamic/instance-identity/`. For more information, see [Instance identity documents](#).

Inspect the system UUID

You can get the system UUID and look for the presence of the characters "ec2" or "EC2" in the beginning octet of the UUID. This method to determine whether a system is an EC2 instance is quick but potentially inaccurate because there is a small chance that a system that is not an EC2 instance could have a UUID that starts with these characters. Furthermore, for EC2 instances that are not using Amazon Linux 2, the distribution's implementation of SMBIOS might represent the UUID in little-endian format, therefore the "EC2" characters do not appear at the beginning of the UUID.

Example : Get the UUID from 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
```

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
```

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
```

EC2 Fleet and Spot Fleet

EC2 Fleet and Spot Fleet are designed to be a useful way to launch a fleet—or group—of instances with AWS. Each instance in a fleet is based on a [launch template](#) or a set of launch parameters that you configure manually at launch.

Fleets provide the following features and benefits. These benefits make it possible for you to maximize cost savings and optimize availability and performance when running applications on multiple EC2 instances.

Multiple instance types and purchase options

In a single API call, a fleet can launch multiple instance types and 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 Plan discounts by using them in conjunction with On-Demand Instances in the fleet.

Distributing instances across Availability Zones

A fleet automatically attempts to distribute instances evenly across multiple Availability Zones for high availability. This provides resiliency in case an Availability Zone becomes unavailable.

Automated replacement of Spot Instances

If your fleet includes Spot Instances, it can automatically request replacement Spot capacity if your Spot Instances are interrupted or become impaired due to a change in instance health. Through Capacity Rebalancing, a fleet can also monitor and proactively replace your Spot Instances that are at an elevated risk of interruption.

EC2 Fleet is a good option if you require flexibility for managing aspects of the instance lifecycle or the scaling mechanisms. You can also use Spot Fleet, but we don't recommend that you do because it's a legacy API with no planned investment. However, if you are already using Spot Fleet, you can continue using it. Spot Fleet and EC2 Fleet offer the same core functionality.

Tip

As a general best practice, we recommend rather 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](#).

Topics

- [EC2 Fleet](#)
- [Spot Fleet](#)
- [Monitor fleet events using Amazon EventBridge](#)
- [Tutorials for EC2 Fleet and Spot Fleet](#)
- [Example configurations for EC2 Fleet and Spot Fleet](#)
- [Fleet quotas](#)

EC2 Fleet

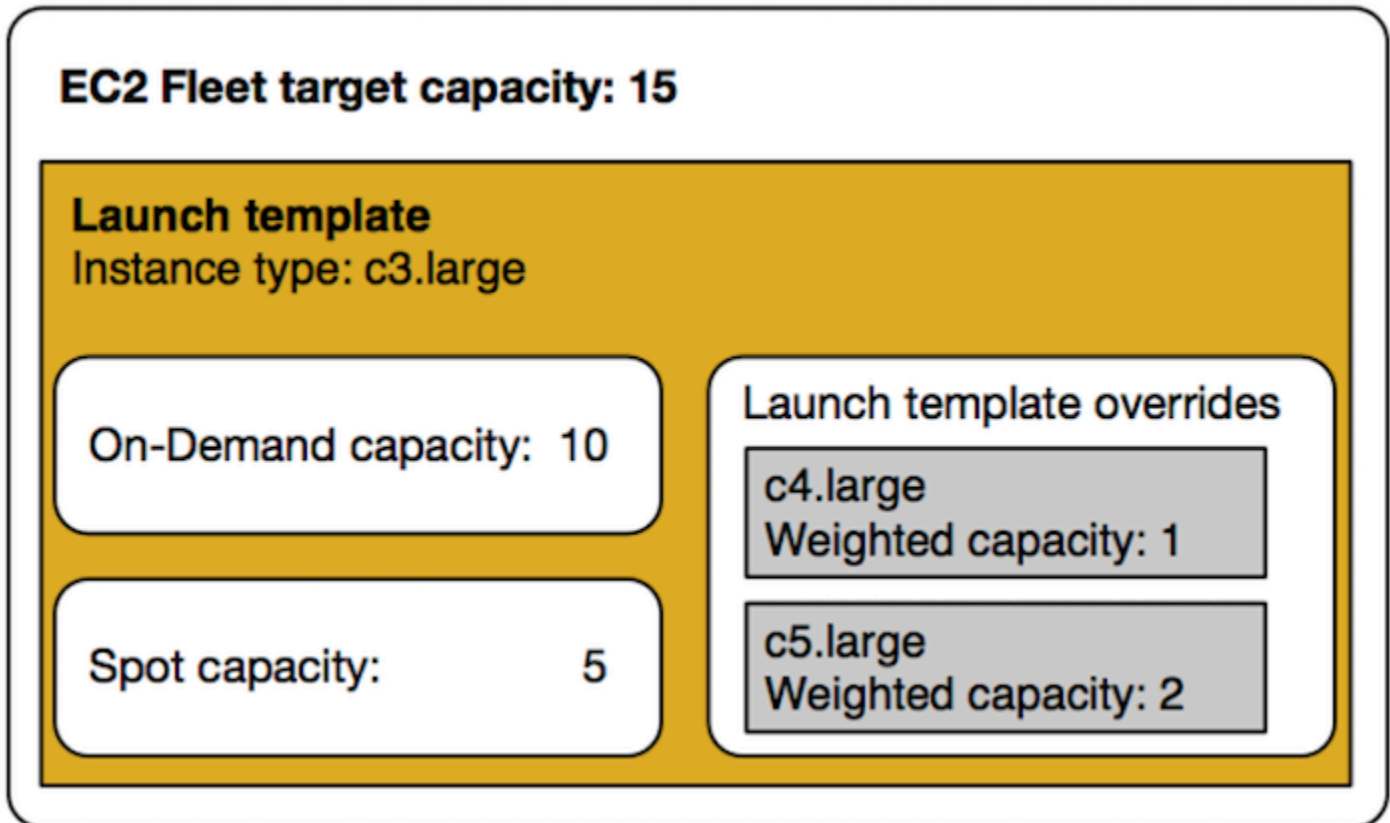
An *EC2 Fleet* contains the configuration information to launch a fleet of instances. In a single API call, a fleet can launch multiple instance types across multiple Availability Zones, using the Spot Instance, On-Demand Instance, Reserved Instance, and Savings Plan purchasing options together. Using EC2 Fleet, you can:

- Define separate Spot and On-Demand capacity targets and the maximum amount you're willing to pay per hour
- Specify the instance types that work best for your applications
- Specify how Amazon EC2 should distribute your fleet capacity within each purchasing option

You can also set a maximum amount per hour that you're willing to pay for your fleet, and EC2 Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity specified in your request. If you specified a total maximum price per hour, it fulfills the

capacity until it reaches the maximum amount that you're willing to pay. The fleet can also attempt to maintain its target Spot capacity if your Spot Instances are interrupted. For more information, see [How Spot Instances work](#).



You can specify an unlimited number of instance types per EC2 Fleet. Those instance types can be provisioned using both Spot and On-Demand purchasing options. You can also specify multiple Availability Zones, specify different maximum Spot prices for each instance, and choose additional Spot options for each fleet. Amazon EC2 uses the specified options to provision capacity when the fleet launches.

While the fleet is running, if Amazon EC2 reclaims a Spot Instance because of a price increase or instance failure, EC2 Fleet can try to replace the instances with any of the instance types that you specify. This makes it easier to regain capacity during a spike in Spot pricing. You can develop a flexible and elastic resourcing strategy for each fleet. For example, within specific fleets, your primary capacity can be On-Demand supplemented with less-expensive Spot capacity if available.

If you have Reserved Instances and you specify On-Demand Instances in your fleet, EC2 Fleet uses your Reserved Instances. For example, if your fleet specifies an On-Demand Instance as `c4.large`, and you have Reserved Instances for `c4.large`, you receive the Reserved Instance pricing. The same applies if you use a Savings Plan.

There is no additional charge for using EC2 Fleet. You pay only for the EC2 instances that the fleet launches for you.

Contents

- [EC2 Fleet limitations](#)
- [Burstable performance instances](#)
- [EC2 Fleet request types](#)
- [EC2 Fleet configuration strategies](#)
- [Work with EC2 Fleets](#)

EC2 Fleet limitations

The following limitations apply to EC2 Fleet:

- 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.

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.

EC2 Fleet request types

There are three types of EC2 Fleet requests:

`instant`

If you configure the request type as `instant`, EC2 Fleet places a synchronous one-time request for your desired capacity. In the API response, it returns the instances that launched, along with errors for those instances that could not be launched. For more information, see [Use an EC2 Fleet of type 'instant'](#).

`request`

If you configure the request type as `request`, EC2 Fleet places an asynchronous one-time request for your desired capacity. Thereafter, if capacity is diminished because of Spot interruptions, the fleet does not attempt to replenish Spot Instances, nor does it submit requests in alternative Spot capacity pools if capacity is unavailable.

`maintain`

(Default) If you configure the request type as `maintain`, EC2 Fleet places an asynchronous request for your desired capacity, and maintains capacity by automatically replenishing any interrupted Spot Instances.

All three types of requests benefit from an allocation strategy. For more information, see [Allocation strategies for Spot Instances](#).

Use an EC2 Fleet of type 'instant'

The EC2 Fleet of type *instant* is a synchronous one-time request that makes only one attempt to launch your desired capacity. The API response lists the instances that launched, along with errors for those instances that could not be launched. There are several benefits to using an EC2 Fleet of type *instant*, which are described in this article. Example configurations are provided at the end of the article.

For workloads that need a launch-only API to launch EC2 instances, you can use the `RunInstances` API. However, with `RunInstances`, you can only launch On-Demand Instances or Spot Instances, but not both in the same request. Furthermore, when you use `RunInstances` to launch Spot Instances, your Spot Instance request is limited to one instance type and one Availability Zone. This targets a single Spot capacity pool (a set of unused instances with the same instance type and Availability Zone). If the Spot capacity pool does not have sufficient Spot Instance capacity for your request, the `RunInstances` call fails.

Instead of using `RunInstances` to launch Spot Instances, we recommend that you rather use the `CreateFleet` API with the `type` parameter set to `instant` for the following benefits:

- **Launch On-Demand Instances and Spot Instances in one request.** An EC2 Fleet can launch On-Demand Instances, Spot Instances, or both. The request for Spot Instances is fulfilled if there is available capacity and the maximum price per hour for your request exceeds the Spot price.
- **Increase the availability of Spot Instances.** By using an EC2 Fleet of type `instant`, you can launch Spot Instances following [Spot best practices](#) with the resulting benefits:
 - **Spot best practice: Be flexible about instance types and Availability Zones.**

Benefit: By specifying several instance types and Availability Zones, you increase the number of Spot capacity pools. This gives the Spot service a better chance of finding and allocating your desired Spot compute capacity. A good rule of thumb is to be flexible across at least 10 instance types for each workload and make sure that all Availability Zones are configured for use in your VPC.

- **Spot best practice: Use the capacity-optimized allocation strategy.**

Benefit: The capacity-optimized allocation strategy automatically provisions instances from the most-available Spot capacity pools. Because your Spot Instance capacity is sourced from pools with optimal capacity, this decreases the possibility that your Spot Instances will be interrupted when Amazon EC2 needs the capacity back.

- **Get access to a wider set of capabilities.** For workloads that need a launch-only API, and where you prefer to manage the lifecycle of your instance rather than let EC2 Fleet manage it for you, use the EC2 Fleet of type `instant` instead of the [RunInstances](#) API. EC2 Fleet provides a wider set of capabilities than `RunInstances`, as demonstrated in the following examples. For all other workloads, you should use Amazon EC2 Auto Scaling because it supplies a more comprehensive feature set for a wide variety of workloads, like ELB-backed applications, containerized workloads, and queue processing jobs.

You can use EC2 Fleet of type *instant* to launch instances into Capacity Blocks. For more information, see [Tutorial: 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 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. When you make the API call, EC2 Fleet places a synchronous one-time request for your desired capacity.
3. The API response lists the instances that launched, along with errors for those instances that could not be launched.
4. You can describe your EC2 Fleet, list the instances associated with your EC2 Fleet, and view the history of your EC2 Fleet.
5. After your instances have launched, you can [delete the fleet request](#). When deleting the fleet request, you can also choose to terminate the associated instances, or leave them running.
6. You can terminate the instances at any time.

Examples

The following examples show how to use EC2 Fleet of type *instant* for different use cases. For more information about using the EC2 CreateFleet API parameters, see [CreateFleet](#) in the *Amazon EC2 API Reference*.

Examples

- [Example 1: Launch Spot Instances with the capacity-optimized allocation strategy](#)
- [Example 2: Launch a single Spot Instance with the capacity-optimized allocation strategy](#)
- [Example 3: Launch Spot Instances using instance weighting](#)
- [Example 4: Launch Spot Instances within single Availability zone](#)
- [Example 5: Launch Spot Instances of single instance type within single Availability zone](#)

- [Example 6: Launch Spot Instances only if minimum target capacity can be launched](#)
- [Example 7: Launch Spot Instances only if minimum target capacity can be launched of same Instance Type in a single Availability Zone](#)
- [Example 8: Launch instances with multiple Launch Templates](#)
- [Example 9: Launch Spot Instance with a base of On-Demand Instances](#)
- [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](#)
- [Example 11: Launch Spot Instances using capacity-optimized-prioritized 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-1t1",
        "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"
    }
  ],
  {
    "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 [EC2 Fleet instance weighting](#).

```
{
  "SpotOptions":{
    "AllocationStrategy":"capacity-optimized"
  },
  "LaunchTemplateConfigs":[
    {
      "LaunchTemplateSpecification":{
        "LaunchTemplateName":"ec2-fleet-1t1",
        "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-1t1",
                "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.

The 12 launch template overrides have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity and the minimum target capacity are both set to 20 instances, the default purchasing option is spot, the Spot allocation strategy is capacity-optimized. The EC2 Fleet launches 20 Spot Instances from the Spot capacity pool with optimal capacity using the launch template overrides, only if it can launch all 20 instances at the same time.

```

{
  "SpotOptions": {
    "AllocationStrategy": "capacity-optimized",
    "MinTargetCapacity": 20
  },
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "ec2-fleet-lt1",
        "Version": "$Latest"
      },
      "Overrides": [
        {
          "InstanceType": "c5.4xlarge",
          "SubnetId": "subnet-fae8c380"
        },
        {
          "InstanceType": "c5.4xlarge",
          "SubnetId": "subnet-e7188bab"
        },
        {

```

```
        "InstanceType": "c5.4xlarge",
        "SubnetId": "subnet-49e41922"
    },
    {
        "InstanceType": "c5d.4xlarge",
        "SubnetId": "subnet-fae8c380"
    },
    {
        "InstanceType": "c5d.4xlarge",
        "SubnetId": "subnet-e7188bab"
    },
    {
        "InstanceType": "c5d.4xlarge",
        "SubnetId": "subnet-49e41922"
    },
    {
        "InstanceType": "m5.4xlarge",
        "SubnetId": "subnet-fae8c380"
    },
    {
        "InstanceType": "m5.4xlarge",
        "SubnetId": "subnet-e7188bab"
    },
    {
        "InstanceType": "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-9x1",
      "Version":"$Latest"
    },
    "Overrides":[
      {
        "InstanceType":"c5n.9xlarge",
        "SubnetId":"subnet-fae8c380",
        "WeightedCapacity":36
      },
      {
        "InstanceType":"c5n.9xlarge",
        "SubnetId":"subnet-e7188bab",
        "WeightedCapacity":36
      },
      {
        "InstanceType":"c5n.9xlarge",
        "SubnetId":"subnet-49e41922",
        "WeightedCapacity":36
      }
    ]
  },
  {
    "LaunchTemplateSpecification":{
      "LaunchTemplateName":"ec2-fleet-lt-4x1",
      "Version":"$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 $\text{TotalTargetCapacity} - \text{OnDemandTargetCapacity} = \text{DefaultTargetCapacityType}$, which results in the fleet launching 15 Spot Instances from one of the 12 Spot capacity pools based on the capacity-optimized allocation strategy.

```

{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "ec2-fleet-1t1",
                "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-1t1",
        "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](#) 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"
}
```

EC2 Fleet configuration strategies

An *EC2 Fleet* is a group of On-Demand Instances and Spot Instances. EC2 Fleet can also be a group of Capacity Blocks instances.

On-Demand Instances and Spot Instances

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity that you specify in the fleet request. The fleet can comprise only On-Demand Instances, only Spot Instances, or a combination of both On-Demand Instances and Spot Instances. The request for Spot Instances is fulfilled if there is available capacity and the maximum price per hour for your request exceeds the Spot price. The fleet also attempts to maintain its target capacity if your Spot Instances are interrupted.

You can also set a maximum amount per hour that you're willing to pay for your fleet, and EC2 Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

A *Spot capacity pool* is a set of unused EC2 instances with the same instance type and Availability Zone. When you create an EC2 Fleet, you can include multiple launch specifications, which vary by instance type, Availability Zone, subnet, and maximum price. The fleet selects the Spot capacity pools that are used to fulfill the request, based on the launch specifications included in your request, and the configuration of the request. The Spot Instances come from the selected pools.

An EC2 Fleet enables you to provision large amounts of EC2 capacity that makes sense for your application based on number of cores or instances, or amount of memory. For example, you can specify an EC2 Fleet to launch a target capacity of 200 instances, of which 130 are On-Demand Instances and the rest are Spot Instances.

Capacity Blocks instances

Capacity Blocks for ML allow you to reserve GPU instances on a future date to support your short duration machine learning (ML) workloads. Instances that run in a Capacity Block are automatically

placed close together inside [Amazon EC2 UltraClusters](#). For more information about Capacity Blocks, see [Capacity Blocks for ML](#).

Use the appropriate configuration strategies to create an EC2 Fleet that meets your needs.

Contents

- [Plan an EC2 Fleet](#)
- [Allocation strategies for Spot Instances](#)
- [Attribute-based instance type selection for EC2 Fleet](#)
- [Configure EC2 Fleet for On-Demand backup](#)
- [Capacity Rebalancing](#)
- [Maximum price overrides](#)
- [Control spending](#)
- [EC2 Fleet instance weighting](#)

Plan an EC2 Fleet

When planning your EC2 Fleet, we recommend that you do the following:

- Determine whether you want to create an EC2 Fleet that submits a synchronous or asynchronous one-time request for the desired target capacity, or one that maintains a target capacity over time. For more information, see [EC2 Fleet request types](#).
- Determine the instance types that meet your application requirements.
- If you plan to include Spot Instances in your EC2 Fleet, review [Spot Best Practices](#) before you create the fleet. Use these best practices when you plan your fleet so that you can provision the instances at the lowest possible price.
- Determine the target capacity for your EC2 Fleet. You can set target capacity in instances or in custom units. For more information, see [EC2 Fleet instance weighting](#).
- Determine what portion of the EC2 Fleet target capacity must be On-Demand capacity and Spot capacity. You can specify 0 for On-Demand capacity or Spot capacity, or both.
- Determine your price per unit, if you are using instance weighting. To calculate the price per unit, divide the price per instance hour by the number of units (or weight) that this instance represents. If you are not using instance weighting, the default price per unit is the price per instance hour.

- Determine the maximum amount per hour that you're willing to pay for your fleet. For more information, see [Control spending](#).
- Review the possible options for your EC2 Fleet. For information about the fleet parameters, see [create-fleet](#) in the *AWS CLI Command Reference*. For EC2 Fleet configuration examples, see [EC2 Fleet example configurations](#).

Allocation strategies for Spot Instances

Your launch configuration determines all the possible Spot capacity pools (instance types and Availability Zones) from which EC2 Fleet can launch Spot Instances. However, when launching instances, EC2 Fleet uses the allocation strategy that you specify to pick the specific pools from all your possible pools.

Note

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](#) of the selected instance type. If the allocation strategy uses price as an input, EC2 Fleet does not include this additional fee; only the Spot price is used.

Allocation strategies

You can specify one of the following allocation strategies for Spot Instances:

`price-capacity-optimized` (recommended)

EC2 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. EC2 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.

`capacity-optimized`

EC2 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 using `capacity-optimized-prioritized`. EC2 Fleet optimizes for capacity first, but honors instance type priorities on a best-effort basis.

With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time. The `capacity-optimized` strategy automatically launches Spot Instances into the most available pools by looking at real-time capacity data and predicting which are the most available. This works well for workloads 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. EC2 Fleet will optimize for capacity first, but will honor instance type priorities on a best-effort basis (for example, if honoring the priorities will not significantly affect EC2 Fleet's ability to provision optimal capacity). This is a good option for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters. Using priorities is supported only if your fleet uses a launch template. Note that when you set the priority for `capacity-optimized-prioritized`, the same priority is also applied to your On-Demand Instances if the `On-Demand AllocationStrategy` is set to `prioritized`.

diversified

The Spot Instances are distributed across all Spot capacity pools.

lowest-price

The Spot Instances come from the lowest priced pool that has available capacity. This is the default strategy. However, we recommend that you override the default by specifying the `price-capacity-optimized` allocation 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, EC2 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.

InstancePoolsToUseCount

The number of Spot pools across which to allocate your target Spot capacity. Valid only when the allocation strategy is set to `lowest-price`. EC2 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 EC2 Fleet attempts to draw Spot Instances from the number of pools that you specify on a best effort basis. If a pool runs out of Spot capacity before fulfilling your target capacity, EC2 Fleet will continue to fulfill your request by drawing from the next 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.

Choose the appropriate allocation strategy

You can optimize your fleet for your use case by choosing the appropriate Spot allocation strategy. For On-Demand Instance target capacity, EC2 Fleet always selects the least expensive instance type based on the public On-Demand price, while following the allocation strategy—either `price-capacity-optimized`, `capacity-optimized`, `diversified`, or `lowest-price`—for Spot Instances.

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 an example configuration that uses the price-capacity-optimized allocation strategy, see [Example 11: Launch Spot Instances in a price-capacity-optimized 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. For an example configuration that uses the capacity-optimized allocation strategy, see [Example 9: Launch Spot Instances in a capacity-optimized fleet](#).

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. For an example configuration, see [Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities](#).

Note that using priorities is supported only if your fleet uses a launch template. Also note that when you set priorities for capacity-optimized-prioritized, the same priorities are also applied to your On-Demand Instances if the On-Demand AllocationStrategy is set to prioritized.

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.

If your workload is time flexible and capacity availability is not a factor, you can optionally use the lowest-price allocation strategy to maximize cost savings. Note, however, that because the lowest-price allocation strategy only considers instance price and not capacity availability, it might lead to high Spot Instance interruption rates.

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 EC2 Fleet specifies 10 pools and a target capacity of 100 instances, the fleet launches 10 Spot Instances in each pool. If the Spot price for one pool exceeds your maximum price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time. With the `diversified` strategy, the EC2 Fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the [On-Demand price](#).

To create an inexpensive and diversified fleet, use the `lowest-price` strategy in combination with `InstancePoolsToUseCount`. For example, if your target capacity is 10 Spot Instances, and you specify 2 Spot capacity pools (for `InstancePoolsToUseCount`), EC2 Fleet will draw on the two lowest priced pools to fulfill your Spot capacity.

You can use a low or high number of Spot capacity pools across which to allocate your Spot Instances. For example, if you run batch processing, we recommend specifying a low number of Spot capacity pools (for example, `InstancePoolsToUseCount=2`) to ensure that your queue always has compute capacity while maximizing savings. If you run a web service, we recommend specifying a high number of Spot capacity pools (for example, `InstancePoolsToUseCount=10`) to minimize the impact if a Spot capacity pool becomes temporarily unavailable.

Note that EC2 Fleet attempts to draw Spot Instances from the number of pools that you specify on a best effort basis. If a pool runs out of Spot capacity before fulfilling your target capacity, EC2 Fleet will continue to fulfill your request by drawing from the next 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.

Maintain target capacity

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot capacity pool, an EC2 Fleet of type `maintain` launches replacement Spot Instances. 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.
- If the allocation strategy is `lowest-price`, the fleet launches replacement instances in the pool where the Spot price is currently the lowest.
- If the allocation strategy is `lowest-price` in combination with `InstancePoolsToUseCount`, the fleet selects the Spot capacity pools with the lowest price and launches Spot Instances across the number of Spot capacity pools that you specify.

Attribute-based instance type selection for EC2 Fleet

When you create an EC2 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 EC2 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 an EC2 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, EC2 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](#)
- [Price protection](#)
- [Considerations](#)
- [Create an EC2 Fleet with attribute-based instance type selection](#)
- [Examples of configurations that are valid and not valid](#)
- [Preview instance types with specified attributes](#)

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 will launch instances on any available instance types that have the specified instance attributes.

Topics

- [Types of instance attributes](#)
- [Where to configure attribute-based instance type selection](#)
- [How EC2 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 component:

- In a launch template, and then reference the launch template 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.

- In a launch specification

How EC2 Fleet uses attribute-based instance type selection when provisioning a fleet

EC2 Fleet provisions a fleet in the following way:

- EC2 Fleet identifies the instance types that have the specified attributes.
- EC2 Fleet uses price protection to determine which instance types to exclude.
- EC2 Fleet 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.

- EC2 Fleet 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. There might be a large number of instance types with the specified attributes, and some of them might be expensive. The default allocation strategy of `lowest-price` for Spot and On-Demand guarantees that EC2 Fleet will launch instances from the least expensive capacity pools.

If you specify an allocation strategy, EC2 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.
- 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 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](#)
- [On-Demand Instance price protection](#)
- [Spot Instance price protection](#)
- [Specify the price protection threshold](#)

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

While creating the EC2 Fleet, 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 about creating the fleet, see [Create an EC2 Fleet with attribute-based instance type selection](#).

Note

When creating the EC2 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.

Considerations

- You can specify either instance types or instance attributes in an EC2 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 a fleet to use attribute-based instance type selection by using the AWS CLI.

To create an EC2 Fleet with attribute-based instance type selection (AWS CLI)

Use the [create-fleet](#) (AWS CLI) 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
```

Example *file_name*.json file

The following example contains the parameters that configure an EC2 Fleet to use attribute-based instance type selection, and is followed by a text explanation.

```
{  
  "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"  
}
```

The attributes for attribute-based instance type selection are specified in the InstanceRequirements structure. In this example, two 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*.

Note

When `InstanceRequirements` is included in the fleet configuration, `InstanceType` and `WeightedCapacity` must be excluded; they cannot determine the fleet configuration at the same time as instance attributes.

The JSON also contains the following fleet configuration:

- `"AllocationStrategy": "price-capacity-optimized"` – The allocation strategy for the Spot Instances in the fleet.
- `"LaunchTemplateName": "my-launch-template", "Version": "1"` – The launch template contains some instance configuration information, but if any instance types are specified, they will be overridden by the attributes that are specified in `InstanceRequirements`.
- `"TotalTargetCapacity": 20` – The target capacity is 20 instances.
- `"DefaultTargetCapacityType": "spot"` – The default capacity is Spot Instances.
- `"Type": "instant"` – The request type for the fleet is instant.

Examples of 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
        }
      }
    }
  ]
}

```

Preview instance types with specified attributes

You can use the [get-instance-types-from-instance-requirements](#) AWS CLI 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](#) in the *Amazon EC2 Command Line Reference*.

- 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.

Configure EC2 Fleet for On-Demand backup

If you have urgent, unpredictable scaling needs, such as a news website that must scale during a major news event or game launch, we recommend that you specify alternative instance types for your On-Demand Instances, in the event that your preferred option does not have sufficient available capacity. For example, you might prefer `c5.2xlarge` On-Demand Instances, but if there is insufficient available capacity, you'd be willing to use some `c4.2xlarge` instances during peak load. In this case, EC2 Fleet attempts to fulfill all of your target capacity using `c5.2xlarge` instances, but if there is insufficient capacity, it automatically launches `c4.2xlarge` instances to fulfill the target capacity.

Topics

- [Prioritize instance types for On-Demand capacity](#)
- [Use Capacity Reservations for On-Demand Instances](#)

Prioritize instance types for On-Demand capacity

When EC2 Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest priced instance type first. If `AllocationStrategy` is set to `prioritized`, EC2 Fleet uses priority to determine which instance type to use first in fulfilling On-Demand capacity. The priority is assigned to the launch template override, and the highest priority is launched first.

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 Reservations for On-Demand Instances

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.

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).

For examples of how to configure a fleet to use Capacity Reservations to fulfil On-Demand capacity, see [EC2 Fleet example configurations](#), specifically Examples 5 through 7.

For information about configuring Capacity Reservations, see [On-Demand Capacity Reservations](#) and the [On-Demand Capacity Reservation FAQs](#).

Capacity Rebalancing

You can configure EC2 Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance recommendation to notify you that a Spot Instance is at an elevated risk of interruption. Capacity Rebalancing helps you maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running instance is interrupted by Amazon EC2. For more information, see [EC2 instance rebalance recommendations](#).

To configure EC2 Fleet to launch a replacement Spot Instance, use the [create-fleet](#) (AWS CLI) command and the relevant parameters in the `MaintenanceStrategies` structure. For more information, see the [example launch configuration](#).

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 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` 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 for Capacity Rebalancing, consider the following:

Provide as many Spot capacity pools in the request as possible

Configure your EC2 Fleet to use multiple instance types and Availability Zones. This provides the flexibility to launch Spot Instances in various Spot capacity pools. For more information, see [Be flexible about instance types and Availability Zones](#).

Avoid an elevated risk of interruption of replacement Spot Instances

Your replacement Spot Instances may be at an elevated risk of interruption if you use the `lowest-price` allocation strategy. 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. To avoid an elevated risk of interruption, we strongly recommend against using the `lowest-price` allocation strategy, and instead 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](#).

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 an EC2 Fleet is configured for Capacity Rebalancing, the fleet attempts to launch a new replacement Spot Instance for every Spot Instance that receives a rebalance recommendation. After a Spot Instance receives a rebalance recommendation, it is no longer counted as part of the fulfilled capacity. 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 an EC2 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 cannot exceed double its target capacity.

Note that you are charged for all of the instances while they are running.

We recommend that you configure EC2 Fleet to terminate Spot Instances that receive a rebalance recommendation

If you configure your EC2 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](#).

EC2 Fleet does not count instances that receive a rebalance recommendation when calculating fulfilled capacity during scale in or out

If your EC2 Fleet is configured for Capacity Rebalancing, and you change the target capacity to either scale in or scale out, the fleet does not count the instances that are marked for rebalance as part of the fulfilled capacity, as follows:

- **Scale in** – If you decrease your desired target capacity, 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 an EC2 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 an EC2 Fleet with a target capacity of 100 Spot Instances. 10 instances receive a rebalance recommendation, so the fleet launches 10 new replacement instances, resulting in a fulfilled capacity of 110 instances. You then increase the target capacity to 200 (scale out), but the fulfilled capacity is actually 210 instances because the 10 instances that are marked for rebalance are not counted by the fleet as part of the target capacity. You need to manually terminate these instances, or you can leave them running.

Maximum price overrides

Each EC2 Fleet can either include a global maximum price, or use the default (the On-Demand price). The fleet uses this as the default maximum price for each of its launch specifications.

You can optionally specify a maximum price in one or more launch specifications. This price is specific to the launch specification. If a launch specification includes a specific price, the EC2 Fleet uses this maximum price, overriding the global maximum price. Any other launch specifications that do not include a specific maximum price still use the global maximum price.

Control spending

EC2 Fleet stops launching instances when it has met one of the following parameters: the `TotalTargetCapacity` or the `MaxTotalPrice` (the maximum amount you're willing to pay). To control the amount you pay per hour for your fleet, you can specify the `MaxTotalPrice`. When the maximum total price is reached, EC2 Fleet stops launching instances even if it hasn't met the target capacity.

The following examples show two different scenarios. In the first, EC2 Fleet stops launching instances when it has met the target capacity. In the second, EC2 Fleet stops launching instances when it has reached the maximum amount you're willing to pay (`MaxTotalPrice`).

Example: Stop launching instances when target capacity is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: \$1.50

EC2 Fleet launches 10 On-Demand Instances because the total of \$1.00 (10 instances x \$0.10) does not exceed the `MaxTotalPrice` of \$1.50 for On-Demand Instances.

Example: Stop launching instances when maximum total price is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- `OnDemandTargetCapacity`: 10

- `MaxTotalPrice`: \$0.80

If EC2 Fleet launches the On-Demand target capacity (10 On-Demand Instances), the total cost per hour would be \$1.00. This is more than the amount (\$0.80) specified for `MaxTotalPrice` for On-Demand Instances. To prevent spending more than you're willing to pay, EC2 Fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the `MaxTotalPrice` for On-Demand Instances.

EC2 Fleet instance weighting

When you create an EC2 Fleet, you can define the capacity units that each instance type would contribute to your application's performance. You can then adjust your maximum price for each launch specification by using *instance weighting*.

By default, the price that you specify is *per instance hour*. When you use the instance weighting feature, the price that you specify is *per unit hour*. You can calculate your price per unit hour by dividing your price for an instance type by the number of units that it represents. EC2 Fleet calculates the number of instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity. The fleet can select any pool that you specify in your launch specification, even if the capacity of the instances launched exceeds the requested target capacity.

The following table includes examples of calculations to determine the price per unit for an EC2 Fleet with a target capacity of 10.

Instance type	Instance weight	Target capacity	Number of instances launched	Price per instance hour	Price per unit hour
r3.xlarge	2	10	5 (10 divided by 2)	\$0.05	\$0.025 (.05 divided by 2)
r3.8xlarge	8	10	2	\$0.10	\$0.0125 (.10 divided by 8)

Instance type	Instance weight	Target capacity	Number of instances launched	Price per instance hour	Price per unit hour
			(10 divided by 8, result rounded up)		

Use EC2 Fleet instance weighting as follows to provision the target capacity that you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your EC2 Fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the price per unit.
3. For each launch specification, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance weighting example

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest price per unit (price for `r3.2xlarge` per instance hour divided by 6), the EC2 Fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest price per unit (price for `c3.xlarge` per instance hour divided by 5), the EC2 Fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 30 Spot Instances
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The EC2 Fleet would launch four instances (30 divided by 8, result rounded up). With the `lowest-price` strategy, all four instances come from the pool that provides the lowest price per unit. With the `diversified` strategy, the fleet launches one instance in each of the three pools, and the fourth instance in whichever of the three pools provides the lowest price per unit.

Work with EC2 Fleets

To start using an EC2 Fleet, you create a request that includes the total target capacity, On-Demand capacity, Spot capacity, one or more launch specifications for the instances, and the maximum price that you are willing to pay. The fleet request must include a launch template that defines the information that the fleet needs to launch an instance, such as an AMI, instance type, subnet or Availability Zone, and one or more security groups. You can specify launch specification overrides for the instance type, subnet, Availability Zone, and maximum price you're willing to pay, and you can assign weighted capacity to each launch specification override.

The EC2 Fleet launches On-Demand Instances when there is available capacity, and launches Spot Instances when your maximum price exceeds the Spot price and capacity is available.

If your fleet includes Spot Instances, Amazon EC2 can attempt to maintain your fleet target capacity as Spot prices change.

An EC2 Fleet request of type `maintain` or `request` remains active until it expires or you delete it. When you delete a fleet of type `maintain` or `request`, you can specify whether deletion terminates the instances in that fleet. Otherwise, the On-Demand Instances run until you terminate them, and the Spot Instances run until they are interrupted or you terminate them.

Contents

- [EC2 Fleet request states](#)
- [EC2 Fleet prerequisites](#)
- [EC2 Fleet health checks](#)

- [Generate an EC2 Fleet JSON configuration file](#)
- [Create an EC2 Fleet](#)
- [Tag an EC2 Fleet](#)
- [Describe your EC2 Fleet](#)
- [Modify an EC2 Fleet](#)
- [Delete an EC2 Fleet](#)

EC2 Fleet request states

An EC2 Fleet request can be in one of the following states:

submitted

The EC2 Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances. The request can include On-Demand Instances, Spot Instances, or both. 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 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 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 includes information about the instances to launch, such as the instance type, Availability Zone, and the maximum price that you are willing to pay. For more information, see [Launch an instance from a launch template](#).

Service-linked role for EC2 Fleet

The `AWSServiceRoleForEC2Fleet` role grants the EC2 Fleet permission to request, launch, terminate, and tag instances on your behalf. Amazon EC2 uses this service-linked role to complete the following actions:

- `ec2:RunInstances` – Launch instances.
- `ec2:RequestSpotInstances` – Request Spot Instances.
- `ec2:TerminateInstances` – Terminate instances.
- `ec2:DescribeImages` – Describe Amazon Machine Images (AMIs) for the Spot Instances.
- `ec2:DescribeInstanceStatus` – Describe the status of the Spot Instances.
- `ec2:DescribeSubnets` – Describe the subnets for Spot Instances.
- `ec2:CreateTags` – Add tags to the EC2 Fleet, instances, and volumes.

Ensure that this role exists before you use the AWS CLI or an API to create an EC2 Fleet.

Note

An instant EC2 Fleet does not require this role.

To create the role, use the IAM console as follows.

To create the `AWSServiceRoleForEC2Fleet` role for EC2 Fleet

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create role**.
3. On the **Select type of 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**.
4. On the **Add permissions** page, choose **Next**.
 5. 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 [Using service-linked roles](#) in the *IAM User Guide*.

Grant access to customer managed keys for use with encrypted AMIs and EBS snapshots

If you specify an [encrypted AMI](#) 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**.

```
{  
  "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 [Creating 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 [Creating 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*.

EC2 Fleet health checks

EC2 Fleet checks the health status of the instances in the fleet every two minutes. The health status of an instance is either `healthy` or `unhealthy`.

EC2 Fleet determines the health status of an instance by using the status checks provided by Amazon EC2. An instance is determined as `unhealthy` when the status of either the instance status check or the system status check is `impaired` for three consecutive health status checks. For more information, see [Status checks for your instances](#).

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. Follow the steps for creating an EC2 Fleet. For more information, see [Create an EC2 Fleet](#).
2. To configure the fleet to replace unhealthy Spot Instances, in the JSON file, for `ReplaceUnhealthyInstances`, enter `true`.

Generate an EC2 Fleet JSON configuration file

To view the full list of EC2 Fleet configuration parameters, you can generate a JSON file. For a description of each parameter, see [create-fleet](#) in the AWS CLI Command Reference.

To generate a JSON file with all possible EC2 Fleet parameters using the command line

- Use the [create-fleet](#) (AWS CLI) command and the `--generate-cli-skeleton` parameter to generate an EC2 Fleet JSON file, and direct the output to a file to save it.

```
aws ec2 create-fleet \  
  --generate-cli-skeleton input > ec2createfleet.json
```

Example output

```
{  
  "DryRun": true,  
  "ClientToken": "",  
  "SpotOptions": {  
    "AllocationStrategy": "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": ""
}

```

Create an EC2 Fleet

To create an EC2 Fleet, you need only specify the following parameters:

- `LaunchTemplateId` or `LaunchTemplateName` – Specifies the launch template to use (which contains the parameters for the instances to launch, such as the instance type, Availability Zone, and the maximum price you're willing to pay)
- `TotalTargetCapacity` – Specifies the total target capacity for the fleet
- `DefaultTargetCapacityType` – Specifies whether the default purchasing option is On-Demand or Spot

You can specify multiple launch specifications that override the launch template. The launch specifications can vary by instance type, Availability Zone, subnet, and maximum price, and can include a different weighted capacity. Alternatively, 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 [Attribute-based instance type selection for EC2 Fleet](#).

If you do not specify a parameter, the fleet uses the default value for the parameter.

Specify the fleet parameters in a JSON file. For more information, see [Generate an EC2 Fleet JSON configuration file](#).

There is currently no console support for creating an EC2 Fleet.

To create an EC2 Fleet (AWS CLI)

- Use the [create-fleet](#) (AWS CLI) command to create an EC2 Fleet and specify the JSON file that contains the fleet configuration parameters.

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

For example configuration files, see [EC2 Fleet example configurations](#).

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": []
}
```

Tag an EC2 Fleet

To help categorize and manage your EC2 Fleet requests, you can tag them with custom metadata. You can assign a tag to an EC2 Fleet request when you create it, or afterward.

When you tag a fleet request, the instances and volumes that are launched by the fleet are not automatically tagged. You need to explicitly tag the instances and volumes launched by the fleet. You can choose to assign tags to only the fleet request, or to only the instances launched by the fleet, or to only the volumes attached to the instances launched by the fleet, or to all three.

Note

For instant fleet types, you can tag volumes that are attached to On-Demand Instances and Spot Instances. For request or maintain fleet types, you can only tag volumes that are attached to On-Demand Instances.

For more information about how tags work, see [Tag your Amazon EC2 resources](#).

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "TagEC2FleetRequest",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2:CreateFleet"
      ],
      "Resource": "*"
    }
  ]
}
```

⚠ Important

We currently do not support resource-level permissions for the `create-fleet` resource. If you specify `create-fleet` as a resource, you will get an unauthorized exception when you try to tag the fleet. The following example illustrates how *not* to set the policy.

```
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags",
    "ec2:CreateFleet"
  ],
  "Resource": "arn:aws:ec2:us-east-1:111122223333:create-fleet/*"
}
```

To 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 [Creating 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 [Creating 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 (AWS CLI)

Use the [create-tags](#) command to tag existing resources.

```
aws ec2 create-tags \  
  --resources fleet-12a34b55-67cd-8ef9-  
ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE \  
  --tags Key=purpose,Value=test
```

Describe your EC2 Fleet

You can describe your EC2 Fleet configuration, the instances in your EC2 Fleet, and the event history of your EC2 Fleet.

To describe your EC2 Fleets (AWS CLI)

Use the [describe-fleets](#) command to describe your EC2 Fleets.

```
aws ec2 describe-fleets
```

Important

If a fleet is of type `instant`, you must specify the fleet ID, otherwise it does not appear in the response. Include `--fleet-ids` as follows:

```
aws ec2 describe-fleets --fleet-ids fleet-8a22eee4-f489-ab02-06b8-832a7EXAMPLE
```

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"
      }
    }
  ]
}
```

Use the [describe-fleet-instances](#) command to describe the instances for the specified EC2 Fleet. The returned list of running instances is refreshed periodically and might be out of date.

```
aws ec2 describe-fleet-instances --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

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"
}
```

Use the [describe-fleet-history](#) command to describe the history for the specified EC2 Fleet for the specified time.

```
aws ec2 describe-fleet-history --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --
start-time 2018-04-10T00:00:00Z
```

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": "2018-04-09T23:53:20.000Z"
}

```

Modify an EC2 Fleet

You can modify an EC2 Fleet that is in the submitted or active state. When you modify a fleet, it enters the modifying state.

You can only modify an EC2 Fleet that is of type `maintain`. You cannot modify an EC2 Fleet of type `request` or `instant`.

You can modify the following parameters of an EC2 Fleet:

- `target-capacity-specification` – Increase or decrease the target capacity for `TotalTargetCapacity`, `OnDemandTargetCapacity`, and `SpotTargetCapacity`.
- `excess-capacity-termination-policy` – Whether running instances should be terminated if the total target capacity of the EC2 Fleet is decreased below the current size of the fleet. Valid values are `no-termination` and `termination`.

When you increase the target capacity, the EC2 Fleet launches the additional instances according to the instance purchasing option specified for `DefaultTargetCapacityType`, which are either On-Demand Instances or Spot Instances.

If the `DefaultTargetCapacityType` is `spot`, the EC2 Fleet launches the additional Spot Instances according to its allocation strategy. If the allocation strategy is `lowest-price`, the fleet launches the instances from the lowest priced Spot capacity pool in the request. If the allocation strategy is `diversified`, the fleet distributes the instances across the pools in the request.

When you decrease the target capacity, the EC2 Fleet deletes any open requests that exceed the new target capacity. You can request that the fleet terminate instances until the size of the fleet reaches the new target capacity. If the allocation strategy is `lowest-price`, the fleet terminates the instances with the highest price per unit. If the allocation strategy is `diversified`, the fleet terminates instances across the pools. Alternatively, you can request that EC2 Fleet keep the fleet at its current size, but not replace any Spot Instances that are interrupted or any instances that you terminate manually.

When an EC2 Fleet terminates a Spot Instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

To modify an EC2 Fleet (AWS CLI)

Use the [modify-fleet](#) command to update the target capacity of the specified EC2 Fleet.

```
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 command as follows.

```
aws ec2 modify-fleet \  
  --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --target-capacity-specification TotalTargetCapacity=20
```

```
--target-capacity-specification TotalTargetCapacity=10 \  
--excess-capacity-termination-policy no-termination
```

Delete an EC2 Fleet

If you no longer require an EC2 Fleet, you can delete it. After you delete a fleet, all Spot requests associated with the fleet are canceled, so that no new Spot Instances are launched.

When you delete an EC2 Fleet, 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 is deleted, the fleet 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 request.
- You can delete up to 100 fleets of type `maintain` or `request` in a single request.
- You can delete up to 125 fleets in a single request, 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.
- Up to 1000 instances can be terminated in a single request to delete `instant` fleets.

To delete an EC2 Fleet and terminate its instances (AWS CLI)

Use the [delete-fleets](#) command and the `--terminate-instances` parameter to delete the specified EC2 Fleet and terminate its associated instances.

```
aws ec2 delete-fleets \  
--fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
--terminate-instances
```

Example output

```
{
```

```

"UnsuccessfulFleetDeletions": [],
"SuccessfulFleetDeletions": [
  {
    "CurrentFleetState": "deleted_terminating",
    "PreviousFleetState": "active",
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
  }
]
}

```

To delete an EC2 Fleet without terminating its instances (AWS CLI)

You can modify the previous command using the `--no-terminate-instances` parameter to delete the specified EC2 Fleet without terminating its associated instances.

Note

`--no-terminate-instances` is not supported for instant fleets.

```

aws ec2 delete-fleets \
  --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --no-terminate-instances

```

Example output

```

{
  "UnsuccessfulFleetDeletions": [],
  "SuccessfulFleetDeletions": [
    {
      "CurrentFleetState": "deleted_running",
      "PreviousFleetState": "active",
      "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"
    }
  ]
}

```

Troubleshoot when a fleet fails to delete

If an EC2 Fleet fails to delete, `UnsuccessfulFleetDeletions` in the output returns the ID of the EC2 Fleet, an error code, and an error message.

The error codes are:

- `ExceededInstantFleetNumForDeletion`
- `fleetIdDoesNotExist`
- `fleetIdMalformed`
- `fleetNotInDeletableState`
- `NoTerminateInstancesNotSupported`
- `UnauthorizedOperation`
- `unexpectedError`

Troubleshooting `ExceededInstantFleetNumForDeletion`

If you try to delete more than 25 instant fleets in a single request, the `ExceededInstantFleetNumForDeletion` error is returned. The following is example output for this error.

```
{
  "UnsuccessfulFleetDeletions": [
    {
      "FleetId": " fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
      "Error": {
        "Message": "Can't delete more than 25 instant fleets in a single
request.",
        "Code": "ExceededInstantFleetNumForDeletion"
      }
    },
    {
      "FleetId": "fleet-9a941b23-0286-5bf4-2430-03a029a07e31",
      "Error": {
        "Message": "Can't delete more than 25 instant fleets in a single
request.",
        "Code": "ExceededInstantFleetNumForDeletion"
      }
    }
  ]
  "SuccessfulFleetDeletions": []
}
```

}

Troubleshoot NoTerminateInstancesNotSupported

If you specify that the instances in an instant fleet must not be terminated when you delete the fleet, the `NoTerminateInstancesNotSupported` error is returned. `--no-terminate-instances` is not supported for instant fleets. The following is example output for this error.

```
{
  "UnsuccessfulFleetDeletions": [
    {
      "FleetId": "fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
      "Error": {
        "Message": "NoTerminateInstances option is not supported for
instant fleet",
        "Code": "NoTerminateInstancesNotSupported"
      }
    }
  ],
  "SuccessfulFleetDeletions": []
}
```

Troubleshoot UnauthorizedOperation

If you do not have permission to terminate instances, you get the `UnauthorizedOperation` error when deleting a fleet that must terminate its instances. The following is the error response.

```
<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not
authorized to perform this
operation. Encoded authorization failure message: VvuncIxj7Z_CPGNYXWqnuFV-
YjByeAU66Q9752NtQ-I3-qnDLWs6JLFd
KnSMMiq5s6cGqjjPtEDpsnGHzyHasFH0aRYJpaDVravoW25azn6KNkUQQLFwhJyujt2dtNCdduJfrqcFYAj1EiRMkfdHt7
BHturzDK6A560Y2nDSUiMmAB1y9UNTqaZJ9SNe5sNxKMqZaqKtjRbk02RZu5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtJ76
VPiU5v2s-
UgZ7h0p2yth6ysUdh10Ng6dBYu8_y_HtEI54invCj4CoK0qawqzMNe6rcmCQHvtCxtXsbkgyaEbcwmim2m01-
EMhekLFZeJLr
DtY0pYcE14_nWFX1wtQDCnNNCmxnJZAoJvb3VMDYpDTsxjQv1Px0DZuqWHs23YXWVyzgnLtHeRf2o4lUhGBw17mXsS07k7
PT9vrHtQiILor5VVTsjSPWg7edj__1rsnXhwPSu8gI48ZLRGrPQqFq0RmKO_QIE8N8s6NWzCK4yoX-9gDcheur0GpkprPIC
</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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DeleteFleetsAndTerminateInstances",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteFleets",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

Spot Fleet

A Spot Fleet is a set of Spot Instances and optionally On-Demand Instances that is launched based on criteria that you specify. The Spot Fleet selects the Spot capacity pools that meet your needs and launches Spot Instances to meet the target capacity for the fleet. By default, Spot Fleets are set to *maintain* target capacity by launching replacement instances after Spot Instances in the fleet are terminated. You can submit a Spot Fleet as a one-time *request*, which does not persist after the instances have been terminated. You can include On-Demand Instance requests in a Spot Fleet request.

Note

If you want to use a console to create a fleet that includes Spot Instances, we recommend using an Auto Scaling group rather than Spot Fleet. For more information, see [Auto Scaling groups with multiple instance types and purchase options](#) in the *Amazon EC2 Auto Scaling User Guide*.

If you want to use the AWS CLI to create a fleet that includes Spot Instances, we recommend using either an Auto Scaling group or EC2 Fleet rather than Spot Fleet. The [RequestSpotFleet](#) API, on which Spot Fleet is based, is a legacy API with no planned investment.

For more information about the recommended APIs to use, see [Which is the best Spot request method to use?](#)

Topics

- [Spot Fleet request types](#)
- [Spot Fleet configuration strategies](#)
- [Work with Spot Fleets](#)
- [CloudWatch metrics for Spot Fleet](#)
- [Automatic scaling for Spot Fleet](#)

Spot Fleet request types

There are two types of Spot Fleet requests:

request

If you configure the request type as `request`, Spot Fleet places an asynchronous one-time request for your desired capacity. Thereafter, if capacity is diminished because of Spot interruptions, the fleet does not attempt to replenish Spot Instances, nor does it submit requests in alternative Spot capacity pools if capacity is unavailable.

maintain

If you configure the request type as `maintain`, Spot Fleet places an asynchronous request for your desired capacity, and maintains capacity by automatically replenishing any interrupted Spot Instances.

To specify the type of request in the Amazon EC2 console, do the following when creating a Spot Fleet request:

- To create a Spot Fleet of type `request`, clear the **Maintain target capacity** check box.
- To create a Spot Fleet of type `maintain`, select the **Maintain target capacity** check box.

For more information, see [Create a Spot Fleet request using defined parameters \(console\)](#).

Both types of requests benefit from an allocation strategy. For more information, see [Allocation strategies for Spot Instances](#).

Spot Fleet configuration strategies

A *Spot Fleet* is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances.

The Spot Fleet attempts to launch the number of Spot Instances and On-Demand Instances to meet the target capacity that you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if there is available capacity and the maximum price you specified in the request exceeds the current Spot price. The Spot Fleet also attempts to maintain its target capacity fleet if your Spot Instances are interrupted.

You can also set a maximum amount per hour that you're willing to pay for your fleet, and Spot Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

A *Spot capacity pool* is a set of unused EC2 instances with the same instance type (for example, `m5.large`), operating system, Availability Zone, and network platform. When you make a Spot Fleet request, you can include multiple launch specifications, that vary by instance type, AMI, Availability Zone, or subnet. The Spot Fleet selects the Spot capacity pools that are used to fulfill the request, based on the launch specifications included in your Spot Fleet request, and the configuration of the Spot Fleet request. The Spot Instances come from the selected pools.

Contents

- [Plan a Spot Fleet request](#)
- [Allocation strategies for Spot Instances](#)
- [Attribute-based instance type selection for Spot Fleet](#)
- [On-Demand in Spot Fleet](#)
- [Capacity Rebalancing](#)
- [Spot price overrides](#)
- [Control spending](#)
- [Spot Fleet instance weighting](#)

Plan a Spot Fleet request

Before you create a Spot Fleet request, review [Spot Best Practices](#). Use these best practices when you plan your Spot Fleet request so that you can provision the type of instances you want at the lowest possible price. We also recommend that you do the following:

- Determine whether you want to create a Spot Fleet that submits a one-time request for the desired target capacity, or one that maintains a target capacity over time.
- Determine the instance types that meet your application requirements.
- Determine the target capacity for your Spot Fleet request. You can set the target capacity in instances or in custom units. For more information, see [Spot Fleet instance weighting](#).
- Determine what portion of the Spot Fleet target capacity must be On-Demand capacity. You can specify 0 for On-Demand capacity.
- Determine your price per unit, if you are using instance weighting. To calculate the price per unit, divide the price per instance hour by the number of units (or weight) that this instance represents. If you are not using instance weighting, the default price per unit is the price per instance hour.

- Review the possible options for your Spot Fleet request. For more information, see the [request-spot-fleet](#) command in the *AWS CLI Command Reference*. For additional examples, see [Spot Fleet example configurations](#).

Allocation strategies for Spot Instances

Your launch configuration determines all the possible Spot capacity pools (instance types and Availability Zones) from which Spot Fleet can launch Spot Instances. However, when launching instances, Spot Fleet uses the allocation strategy that you specify to pick the specific pools from all your possible pools.

Note

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](#) 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.

Allocation strategies

You can specify one of the following allocation strategies for your Spot Instances:

`priceCapacityOptimized` (recommended)

Spot 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. Spot Fleet then requests Spot Instances from the lowest priced of these pools.

The `priceCapacityOptimized` 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.

`capacityOptimized`

Spot 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 using `capacityOptimizedPrioritized`. Spot Fleet optimizes for capacity first, but honors instance type priorities on a best-effort basis.

With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time. The `capacityOptimized` strategy automatically launches Spot Instances into the most available pools by looking at real-time capacity data and predicting which are the most available. This works well for workloads 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 `capacityOptimized` strategy can lower the overall cost of your workload.

Alternatively, you can use the `capacityOptimizedPrioritized` allocation strategy with a `priority` parameter to order instance types from highest to lowest priority. You can set the same priority for different instance types. Spot Fleet will optimize for capacity first, but will honor instance type priorities on a best-effort basis (for example, if honoring the priorities will not significantly affect Spot Fleet's ability to provision optimal capacity). This is a good option for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters. Using priorities is supported only if your fleet uses a launch template. Note that when you set the priority for `capacityOptimizedPrioritized`, the same priority is also applied to your On-Demand Instances if the `On-Demand AllocationStrategy` is set to `prioritized`.

diversified

The Spot Instances are distributed across all pools.

lowestPrice

The Spot Instances come from the lowest priced pool that has available capacity. This is the default strategy. However, we recommend that you override the default by specifying the `priceCapacityOptimized` allocation 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, Spot 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.

InstancePoolsToUseCount

The number of Spot pools across which to allocate your target Spot capacity. Valid only when the allocation strategy is set to `lowestPrice`. Spot 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 Spot Fleet attempts to draw Spot Instances from the number of pools that you specify on a best effort basis. If a pool runs out of Spot capacity before fulfilling your target capacity, Spot Fleet will continue to fulfill your request by drawing from the next 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.

Choose an appropriate allocation strategy

You can optimize your fleet for your use case by choosing the appropriate Spot allocation strategy. For On-Demand Instance target capacity, Spot Fleet always selects the least expensive instance type based on the public On-Demand price, while following the allocation strategy—either `priceCapacityOptimized`, `capacityOptimized`, `diversified`, or `lowestPrice`—for Spot Instances.

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 `priceCapacityOptimized` 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 `priceCapacityOptimized` 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 `priceCapacityOptimized` 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 an example configuration that uses the `priceCapacityOptimized` allocation strategy, see [Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities](#).

When workloads have a high cost of interruption

You can optionally use the `capacityOptimized` 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. For an example configuration that uses the `capacityOptimized` allocation strategy, see [Example 8: Configure Capacity Rebalancing to launch replacement Spot Instances](#).

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 `capacityOptimizedPrioritized` allocation strategy and then setting the order of instance types to use from highest to lowest priority. For an example configuration, see [Example 9: Launch Spot Instances in a capacity-optimized fleet](#).

Note that using priorities is supported only if your fleet uses a launch template. Also note that when you set priorities for `capacityOptimizedPrioritized`, the same priorities are also applied to your On-Demand Instances if the `On-Demand AllocationStrategy` is set to `prioritized`.

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 `priceCapacityOptimized` to maximize cost savings while still considering capacity availability.

If your workload is time flexible and capacity availability is not a factor, you can optionally use the `lowestPrice` allocation strategy to maximize cost savings. Note, however, that because the `lowestPrice` allocation strategy only considers instance price and not capacity availability, it might lead to high Spot Instance interruption rates.

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 Spot 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 Spot Fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the [On-Demand price](#).

To create an inexpensive and diversified fleet, use the `lowestPrice` strategy in combination with `InstancePoolsToUseCount`. For example, if your target capacity is 10 Spot Instances, and you specify 2 Spot capacity pools (for `InstancePoolsToUseCount`), Spot Fleet will draw on the two lowest priced pools to fulfill your Spot capacity.

You can use a low or high number of Spot capacity pools across which to allocate your Spot Instances. For example, if you run batch processing, we recommend specifying a low number of Spot capacity pools (for example, `InstancePoolsToUseCount=2`) to ensure that your queue always has compute capacity while maximizing savings. If you run a web service, we recommend specifying a high number of Spot capacity pools (for example, `InstancePoolsToUseCount=10`) to minimize the impact if a Spot capacity pool becomes temporarily unavailable.

Note that Spot Fleet attempts to draw Spot Instances from the number of pools that you specify on a best effort basis. If a pool runs out of Spot capacity before fulfilling your target capacity, Spot Fleet will continue to fulfill your request by drawing from the next 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.

Maintain target capacity

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot capacity pool, a Spot Fleet of type `maintain` launches replacement Spot Instances. The allocation strategy determines the pools from which the replacement instances are launched, as follows:

- If the allocation strategy is `priceCapacityOptimized`, 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 `capacityOptimized`, 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.
- If the allocation strategy is `lowestPrice`, the fleet launches replacement instances in the pool where the Spot price is currently the lowest.
- If the allocation strategy is `lowestPrice` in combination with `InstancePoolsToUseCount`, the fleet selects the Spot capacity pools with the lowest price and launches Spot Instances across the number of Spot capacity pools that you specify.

Attribute-based instance type selection for Spot Fleet

When you create a 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 Spot 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 Spot 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, Spot 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](#)
- [Price protection](#)
- [Considerations](#)
- [Create a Spot Fleet with attribute-based instance type selection](#)
- [Examples of configurations that are valid and not valid](#)
- [Preview instance types with specified attributes](#)

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. Spot Fleet will launch instances on any available instance types that have the specified instance attributes.

Topics

- [Types of instance attributes](#)
- [Where to configure attribute-based instance type selection](#)
- [How 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 one or both of the following fleet configuration components:

- In a launch template, and then reference the launch template in the fleet request
- 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 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.

- In a launch specification

How Spot Fleet uses attribute-based instance type selection when provisioning a fleet

Spot Fleet provisions a fleet in the following way:

- Spot Fleet identifies the instance types that have the specified attributes.
- Spot Fleet uses price protection to determine which instance types to exclude.
- Spot Fleet 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.

- Spot Fleet 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. There might be a large number of instance types with the specified attributes, and some of them might be expensive. The default allocation strategy of `lowestPrice` for Spot and On-Demand guarantees that Spot Fleet will launch instances from the least expensive capacity pools.

If you specify an allocation strategy, Spot Fleet will launch instances according to the specified allocation strategy.

- For Spot Instances, attribute-based instance type selection supports the `capacityOptimizedPrioritized`, `capacityOptimized` and `lowestPrice` allocation strategies.
- For On-Demand Instances, attribute-based instance type selection supports the `lowestPrice` 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 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](#)
- [On-Demand Instance price protection](#)
- [Spot Instance price protection](#)
- [Specify the price protection threshold](#)

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

While creating the Spot Fleet, configure the fleet for attribute-based instance type selection, and then do the following:

- Console

To specify the On-Demand Instance price protection threshold, under **Additional instance attribute**, choose **On-demand price protection**, and then choose **Add attribute**. For **On-Demand price protection percentage**, enter the price protection threshold as a percentage.

To specify the Spot Instance price protection threshold, under **Additional instance attribute**, choose **Spot price protection**, and then choose **Add attribute**. Choose a parameter and enter the price protection threshold as a percentage.

- AWS CLI

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 about creating the fleet, see [Create a Spot Fleet with attribute-based instance type selection](#).

 **Note**

When creating the Spot Fleet, if you set **Total target capacity** type to **vCPUs** or **Memory (MiB)** (console) or TargetCapacityUnitType to vcpu or memory-mib (AWS CLI), the price protection threshold is applied based on the per-vCPU or per-memory price instead of the per-instance price.

Considerations

- You can specify either instance types or instance attributes in a 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 a Spot Fleet with attribute-based instance type selection

You can configure a fleet to use attribute-based instance type selection by using the Amazon EC2 console or the AWS CLI.

Topics

- [Create a Spot Fleet using the console](#)
- [Create a Spot Fleet using the AWS CLI](#)

Create a Spot Fleet using the console

To configure a Spot Fleet for attribute-based instance type selection (console)

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 \(console\)](#).

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.

Create a Spot Fleet using the AWS CLI

To configure a Spot Fleet for attribute-based instance type selection (AWS CLI)

Use the [request-spot-fleet](#) (AWS CLI) 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
```

Example *file_name*.json file

The following example contains the parameters that configure a Spot Fleet to use attribute-based instance type selection, and is followed by a text explanation.

```
{  
  "AllocationStrategy": "priceCapacityOptimized",  
  "TargetCapacity": 20,  
  "Type": "request",  
  "LaunchTemplateConfigs": [{  
    "LaunchTemplateSpecification": {  
      "LaunchTemplateName": "my-launch-template",  
      "Version": "1"  
    },  
  },  
  "Overrides": [{  
    "InstanceRequirements": {  
      "VCpuCount": {  
        "Min": 2  
      },  
      "MemoryMiB": {  
        "Min": 4  
      }  
    }  
  }  
}
```

```
    }]  
  }]  
}
```

The attributes for attribute-based instance type selection are specified in the `InstanceRequirements` structure. In this example, two 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*.

Note

When `InstanceRequirements` is included in the fleet configuration, `InstanceType` and `WeightedCapacity` must be excluded; they cannot determine the fleet configuration at the same time as instance attributes.

The JSON also contains the following fleet configuration:

- `"AllocationStrategy"`: `"priceCapacityOptimized"` – The allocation strategy for the Spot Instances in the fleet.
- `"LaunchTemplateName"`: `"my-launch-template"`, `"Version"`: `"1"` – The launch template contains some instance configuration information, but if any instance types are specified, they will be overridden by the attributes that are specified in `InstanceRequirements`.
- `"TargetCapacity"`: `20` – The target capacity is 20 instances.
- `"Type"`: `"request"` – The request type for the fleet is request.

Examples of 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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::000000000000: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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::000000000000: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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::000000000000: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": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::000000000000: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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::000000000000: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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::000000000000: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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::000000000000: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](#) AWS CLI 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](#) in the *Amazon EC2 Command Line Reference*.

- 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                       ||
||  c6a.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.

On-Demand in Spot Fleet

To ensure that you always have instance capacity, you can include a request for On-Demand capacity in your Spot Fleet request. In your Spot Fleet request, you specify your desired target capacity and how much of that capacity must be On-Demand. The balance comprises Spot capacity, which is launched if there is available Amazon EC2 capacity and availability. For example, if in your Spot Fleet request you specify the target capacity as 10 and the On-Demand capacity as 8, Amazon EC2 launches 8 capacity units as On-Demand, and 2 capacity units (10-8=2) as Spot.

Prioritize instance types for On-Demand capacity

When Spot Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest priced instance type first. If `OnDemandAllocationStrategy` is set to `prioritized`, Spot Fleet uses priority to determine which instance type to use first in fulfilling On-Demand capacity.

The priority is assigned to the launch template override, and the highest priority is launched first.

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 cheapest instance type:

- `m4.large` – cheapest
- `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 cheapest 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

Capacity Rebalancing

You can configure Spot Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance recommendation to notify you that a Spot Instance is at an elevated risk of interruption. Capacity Rebalancing helps you maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running instance is interrupted by Amazon EC2. For more information, see [EC2 instance rebalance recommendations](#).

To configure Spot Fleet to launch a replacement Spot Instance, you can use the Amazon EC2 console or the AWS CLI.

- Amazon EC2 console: You must select the **Capacity rebalance** check box when you create the Spot Fleet. For more information, see step 6.d. in [Create a Spot Fleet request using defined parameters \(console\)](#).

- AWS CLI: Use the [request-spot-fleet](#) command and the relevant parameters in the `SpotMaintenanceStrategies` structure. For more information, see the [example launch configuration](#).

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 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 `lowestPrice` 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 a Spot Fleet for Capacity Rebalancing, consider the following:

Provide as many Spot capacity pools in the request as possible

Configure your Spot Fleet to use multiple instance types and Availability Zones. This provides the flexibility to launch Spot Instances in various Spot capacity pools. For more information, see [Be flexible about instance types and Availability Zones](#).

Avoid an elevated risk of interruption of replacement Spot Instances

Your replacement Spot Instances may be at an elevated risk of interruption if you use the `LowestPrice` allocation strategy. 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. To avoid an elevated risk of interruption, we strongly recommend against using the `LowestPrice` allocation strategy, and instead recommend the `capacityOptimized` or `capacityOptimizedPrioritized` 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](#).

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 Spot Fleet is configured for Capacity Rebalancing, Amazon EC2 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 Spot 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 cannot exceed double its target capacity.

Note that you are charged for all of the instances while they are running.

We recommend that you configure Spot Fleet to terminate Spot Instances that receive a rebalance recommendation

If you configure your Spot 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](#).

Spot Fleet does not count instances that receive a rebalance recommendation when calculating fulfilled capacity during scale in or out

If your Spot Fleet is configured for Capacity Rebalancing, and you change the target capacity to either scale in or scale out, the fleet does not count the instances that are marked for rebalance as part of the fulfilled capacity, as follows:

- **Scale in** – If you decrease your desired target capacity, 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 Spot 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 Spot 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 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.

Spot price overrides

Each Spot Fleet request can include a global maximum price, or use the default (the On-Demand price). Spot Fleet uses this as the default maximum price for each of its launch specifications.

You can optionally specify a maximum price in one or more launch specifications. This price is specific to the launch specification. If a launch specification includes a specific price, the Spot Fleet uses this maximum price, overriding the global maximum price. Any other launch specifications that do not include a specific maximum price still use the global maximum price.

Control spending

Spot Fleet stops launching instances when it has either reached the target capacity or the maximum amount you're willing to pay. To control the amount you pay per hour for your fleet, you can specify the `SpotMaxTotalPrice` for Spot Instances and the `OnDemandMaxTotalPrice` for On-Demand Instances. When the maximum total price is reached, Spot Fleet stops launching instances even if it hasn't met the target capacity.

The following examples show two different scenarios. In the first, Spot Fleet stops launching instances when it has met the target capacity. In the second, Spot Fleet stops launching instances when it has reached the maximum amount you're willing to pay.

Example: Stop launching instances when target capacity is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- OnDemandTargetCapacity: 10
- OnDemandMaxTotalPrice: \$1.50

Spot Fleet launches 10 On-Demand Instances because the total of \$1.00 (10 instances x \$0.10) does not exceed the OnDemandMaxTotalPrice of \$1.50.

Example: Stop launching instances when maximum total price is reached

Given a request for m4.large On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- OnDemandTargetCapacity: 10
- OnDemandMaxTotalPrice: \$0.80

If Spot Fleet launches the On-Demand target capacity (10 On-Demand Instances), the total cost per hour would be \$1.00. This is more than the amount (\$0.80) specified for OnDemandMaxTotalPrice. To prevent spending more than you're willing to pay, Spot Fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the OnDemandMaxTotalPrice.

Spot Fleet instance weighting

When you request a fleet of Spot Instances, you can define the capacity units that each instance type would contribute to your application's performance, and adjust your maximum price for each Spot capacity pool accordingly using *instance weighting*.

By default, the price that you specify is *per instance hour*. When you use the instance weighting feature, the price that you specify is *per unit hour*. You can calculate your price per unit hour by dividing your price for an instance type by the number of units that it represents. Spot Fleet calculates the number of Spot Instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot Fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity. Spot Fleet can select any pool that you specify in your launch specification, even if the capacity of the instances launched exceeds the requested target capacity.

The following tables provide examples of calculations to determine the price per unit for a Spot Fleet request with a target capacity of 10.

Instance type	Instance weight	Price per instance hour	Price per unit hour	Number of instances launched
r3.xlarge	2	\$0.05	.025 (.05 divided by 2)	5 (10 divided by 2)

Instance type	Instance weight	Price per instance hour	Price per unit hour	Number of instances launched
r3.8xlarge	8	\$0.10	.0125 (.10 divided by 8)	2 (10 divided by 8, result rounded up)

Use Spot Fleet instance weighting as follows to provision the target capacity that you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your Spot Fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the price per unit.
3. For each launch configuration, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance weighting example

Consider a Spot Fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest price per unit (price for `r3.2xlarge` per instance hour divided by 6), the Spot Fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest price per unit (price for `c3.xlarge` per instance hour divided by 5), the Spot Fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider a Spot Fleet request with the following configuration:

- A target capacity of 30
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The Spot Fleet would launch four instances (30 divided by 8, result rounded up). With the `LowestPrice` strategy, all four instances come from the pool that provides the lowest price per unit. With the `diversified` strategy, the Spot Fleet launches one instance in each of the three pools, and the fourth instance in whichever pool provides the lowest price per unit.

Work with Spot Fleets

To start using a Spot Fleet, you create a Spot Fleet request that includes the target capacity, an optional On-Demand portion, one or more launch specifications for the instances, and the maximum price that you are willing to pay. The fleet request must include a launch specification that defines the information that the fleet needs to launch an instance, such as an AMI, instance type, subnet or Availability Zone, and one or more security groups.

If your fleet includes Spot Instances, Amazon EC2 can attempt to maintain your fleet target capacity as Spot prices change.

It is not possible to modify the target capacity of a one-time request after it's been submitted. To change the target capacity, cancel the request and submit a new one.

A Spot Fleet request remains active until it expires or you cancel it. When you cancel a fleet request, you can specify whether canceling the request terminates the Spot Instances in that fleet.

Contents

- [Spot Fleet request states](#)
- [Spot Fleet health checks](#)
- [Spot Fleet permissions](#)
- [Create a Spot Fleet request](#)
- [Tag a Spot Fleet](#)
- [Describe your Spot Fleet](#)
- [Modify a Spot Fleet request](#)
- [Cancel a Spot Fleet request](#)

Spot Fleet request states

A Spot Fleet request can be in one of the following states:

- **submitted** – The Spot Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances. If a request would exceed your Spot Fleet limits, 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 Spot Fleet is canceled. A one-time request cannot be modified, and this state does not apply to such Spot requests.
- **cancelled_running** – The Spot Fleet is canceled and does not launch additional Spot Instances. Its existing Spot Instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- **cancelled_terminating** – The Spot Fleet is canceled and its Spot Instances are terminating. The request remains in this state until all instances are terminated.
- **cancelled** – The Spot Fleet is canceled and has no running Spot Instances. The Spot Fleet request is deleted two days after its instances were terminated.

Spot Fleet health checks

Spot Fleet checks the health status of the Spot Instances in the fleet every two minutes. The health status of an instance is either healthy or unhealthy.

Spot Fleet determines the health status of an instance by using the status checks provided by Amazon EC2. An instance is determined as unhealthy when the status of either the instance status check or the system status check is impaired for three consecutive health checks. For more information, see [Status checks for your instances](#).

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 using the console

1. Follow the steps for creating a Spot Fleet. For more information, see [Create a Spot Fleet request using defined parameters \(console\)](#).
2. To configure the fleet to replace unhealthy Spot Instances, for **Health check**, choose **Replace unhealthy instances**. To enable this option, you must first choose **Maintain target capacity**.

AWS CLI

To configure a Spot Fleet to replace unhealthy Spot Instances using the AWS CLI

1. Follow the steps for creating a Spot Fleet. For more information, see [Create a Spot Fleet using the AWS CLI](#).
2. To configure the fleet to replace unhealthy Spot Instances, for `ReplaceUnhealthyInstances`, enter `true`.

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.

```
{
```

```

"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 [Creating 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 [Creating 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 [Using service-linked roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2SpotFleet** to launch and manage instances on your behalf.

Important

If you specify an [encrypted AMI](#) 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`

Amazon EC2 uses `AWSServiceRoleForEC2SpotFleet` to complete the following actions:

- `ec2:RequestSpotInstances` - Request Spot Instances
- `ec2:RunInstances` - Launch instances
- `ec2:TerminateInstances` - Terminate instances
- `ec2:DescribeImages` - Describe Amazon Machine Images (AMIs) for the instances
- `ec2:DescribeInstanceStatus` - Describe the status of the instances
- `ec2:DescribeSubnets` - Describe the subnets for the instances
- `ec2:CreateTags` - Add tags to the Spot Fleet request, instances, and volumes
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer` - Add the specified instances to the specified load balancer
- `elasticloadbalancing:RegisterTargets` - Register the specified targets with the specified target group

Create the service-linked role

Under most circumstances, you don't need to manually create a service-linked role. Amazon EC2 creates the `AWSServiceRoleForEC2SpotFleet` service-linked role the first time you create a Spot Fleet using the console.

If you had an active Spot Fleet request before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the `AWSServiceRoleForEC2SpotFleet` role in your AWS account. For more information, see [A new role appeared in my AWS account](#) in the *IAM User Guide*.

If you use the AWS CLI or an API to create a Spot Fleet, you must first ensure that this role exists.

To create `AWSServiceRoleForEC2SpotFleet` using the console

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**.
 - d. Choose **Next**.
5. On the **Add permissions** page, choose **Next**.
6. On the **Name, review, and create** page, choose **Create role**.

To create `AWSServiceRoleForEC2SpotFleet` using the AWS CLI

Use the [create-service-linked-role](#) command as follows.

```
aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

If you no longer need to use Spot Fleet, we recommend that you delete the `AWSServiceRoleForEC2SpotFleet` role. After this role is deleted from your account, Amazon EC2 will create the role again if you request a Spot Fleet using the console. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

Grant access to CMKs for use with encrypted AMIs and EBS snapshots

If you specify an [encrypted AMI](#) 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/  
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 Spot Fleet](#).

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 check box).
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:account_id:spot-fleet-request/sfr-
*"
    },
    "StringEquals": {
      "aws:SourceAccount": "account_id"
    }
  }

```

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:

```

{
  "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:account_id:spot-fleet-request/sfr-
*"
      },
      "StringEquals": {
        "aws:SourceAccount": "account_id"
      }
    }
  }
}

```

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	<code>aws:SourceArn</code>
<code>RequestSpotFleet</code>	AWS STS (<code>AssumeRole</code>)	Limit the <code>AssumeRole</code> capability on <code>aws-ec2-spot-fleet-tagging-role</code> to spot-fleet-requests in the specified account.	<code>arn:aws:ec2:*:123456789012:spot-fleet-request/sfr-*</code>
<code>RequestSpotFleet</code>	AWS STS (<code>AssumeRole</code>)	Limit the <code>AssumeRole</code> capability on <code>aws-ec2-spot-fleet-tagging-role</code> to spot-fleet-requests in the specified account and specified Region. Note that this role will not be usable in other Regions.	<code>arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-*</code>
<code>RequestSpotFleet</code>	AWS STS (<code>AssumeRole</code>)	Limit the <code>AssumeRole</code> capability on <code>aws-ec2-spot-fleet-tagging-role</code> to only actions affecting the fleet <code>sfr-11111111-1111-1111-1111-1111-1111111111</code> . Note that this role may not be usable for other Spot Fleets.	<code>arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-11111111-1111-1111-1111-11111111</code>

API operation	Called service	Scope	aws:SourceArn
		Also, this role cannot be used to launch any new Spot Fleets through request-spot-fleet.	

Create a Spot Fleet request

Using the AWS Management Console, quickly create a Spot Fleet request by choosing only your application or task need and minimum compute specs. Amazon EC2 configures a fleet that best meets your needs and follows Spot best practice. For more information, see [Quickly create a Spot Fleet request \(console\)](#). Otherwise, you can modify any of the default settings. For more information, see [Create a Spot Fleet request using defined parameters \(console\)](#) and [Create a Spot Fleet using the AWS CLI](#).

Options for creating a Spot Fleet

- [Quickly create a Spot Fleet request \(console\)](#)
- [Create a Spot Fleet request using defined parameters \(console\)](#)
- [Create a Spot Fleet using the AWS CLI](#)

Quickly create a Spot Fleet request (console)

Follow these steps to quickly create a Spot Fleet request.

To create a Spot Fleet request using the recommended settings (console)

1. Open the 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 **Request Spot Instances**.
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 (MiB)**.

7. For **Your fleet request at a glance**, review your fleet configuration, and choose **Launch**.

Create a Spot Fleet request using defined parameters (console)

You can create a Spot Fleet by using parameters that you define.

To create a Spot Fleet request using defined parameters (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
4. For **Launch parameters**, do the following:
 - a. To define the launch parameters in the Spot console, choose **Manually configure launch parameters**.
 - b. For **AMI**, choose one of the basic AMIs provided by AWS, or choose **Search for AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.

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.

- c. (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.

- d. (Optional) Expand **Additional launch parameters**, and do the following:
 - i. (Optional) To enable Amazon EBS optimization, for **EBS-optimized**, select **Launch EBS-optimized instances**.
 - ii. (Optional) To add temporary block-level storage for your instances, for **Instance store**, choose **Attach at launch**.

- iii. (Optional) To add storage, choose **Add new volume**, and specify additional instance store volumes or Amazon EBS volumes, depending on the instance type.
- iv. (Optional) By default, basic monitoring is enabled for your instances. To enable detailed monitoring, for **Monitoring**, select **Enable CloudWatch detailed monitoring**.
- v. (Optional) To run a Dedicated Spot Instance, for **Tenancy**, choose **Dedicated - run a dedicated instance**.
- vi. (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.

- vii. (Optional) To make your instances reachable from the internet, for **Auto-assign IPv4 Public IP**, choose **Enable**.
- viii. (Optional) To launch your Spot Instances with an IAM role, for **IAM instance profile**, choose the role.
- ix. (Optional) To run a start-up script, copy it to **User data**.
- x. (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**.

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) For **Maximum price**, you can use the default maximum price (the On-Demand price) or specify the maximum price you are willing to pay. If your maximum price is lower than the Spot price for the instance types that you selected, your Spot Instances are not launched.
- d. (Optional) To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid until**.

- e. (Optional) By default, we terminate your Spot Instances when the Spot Fleet request expires. To keep them running after your request expires, clear **Terminate the instances when the request expires**.
 - f. (Optional) To register your Spot Instances with a load balancer, choose **Receive traffic from one or more load balancers** and choose one or more Classic Load Balancers or target groups.
6. For **Minimum compute unit**, choose the minimum hardware specifications (vCPUs, memory, and storage) that you need for your application or task, either **as specs** or **as an instance type**.
- For **as specs**, specify the required number of vCPUs and amount of memory.
 - For **as an instance type**, accept the default instance type, or choose **Change instance type** to choose a different instance type.
7. 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, choose **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. 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 Spot Fleet terminates the old instances. For more information, see [Capacity Rebalancing](#).
 - 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 [Control spending](#).
8. 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 to the Amazon VPC console. When you are done, return to the wizard and refresh the list.
 - b. (Optional) For **Availability Zone**, let AWS choose the Availability Zones for your Spot Instances, or specify one or more Availability Zones.

If you have more than one subnet in an Availability Zone, choose the appropriate subnet from **Subnet**. To add subnets, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.
9. 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 [Attribute-based instance type selection for 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**, **No maximum**, or both.
 - ii. For **Memory (GiB)**, enter the desired minimum and maximum amount of memory. To specify no limit, select **No minimum**, **No maximum**, or both.
 - iii. (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](#) in the *Amazon EC2 Command Line Reference*.

- 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**.
10. For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Allocation strategies for Spot Instances](#).
 11. For **Your fleet request at a glance**, review your fleet configuration, and make any adjustments if necessary.
 12. (Optional) To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**.
 13. Choose **Launch**.

The Spot Fleet request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

Create a Spot Fleet using the AWS CLI

To create a Spot Fleet request using the AWS CLI

- Use the [request-spot-fleet](#) command to create a Spot Fleet request.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For example configuration files, see [Spot Fleet example configurations](#).

The following is example output:

```
{  
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
```

```
}
```

Tag a Spot Fleet

To help categorize and manage your Spot Fleet requests, you can tag them with custom metadata. You can assign a tag to a Spot Fleet request when you create it, or afterward. You can assign tags using the Amazon EC2 console or a command line tool.

When you tag a Spot Fleet request, the instances and volumes that are launched by the Spot Fleet are not automatically tagged. You need to explicitly tag the instances and volumes launched by the Spot Fleet. You can choose to assign tags to only the Spot Fleet request, or to only the instances launched by the fleet, or to only the volumes attached to the instances launched by the fleet, or to all three.

Note

Volume tags are only supported for volumes that are attached to On-Demand Instances. You can't tag volumes that are attached to Spot Instances.

For more information about how tags work, see [Tag your Amazon EC2 resources](#).

Contents

- [Prerequisite](#)
- [Tag a new Spot Fleet](#)
- [Tag a new Spot Fleet and the instances and volumes that it launches](#)
- [Tag an existing Spot Fleet](#)
- [View Spot Fleet request tags](#)

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.

```
{
  "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 [Creating 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 [Creating 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

To tag a new Spot Fleet request using the console

1. Follow the [Create a Spot Fleet request using defined parameters \(console\)](#) procedure.
2. To add a tag, expand **Additional configurations**, choose **Add new tag**, and enter the key and value for the tag. Repeat for each tag.

For each tag, you can tag the Spot Fleet request and the instances with the same tag. To tag both, ensure that both **Instance tags** and **Fleet tags** are selected. To tag only the Spot Fleet request, clear **Instance tags**. To tag only the instances launched by the fleet, clear **Fleet tags**.

3. Complete the required fields to create a Spot Fleet request, and then choose **Launch**. For more information, see [Create a Spot Fleet request using defined parameters \(console\)](#).

To tag a new Spot Fleet request using the AWS CLI

To tag a Spot Fleet request when you create it, configure the Spot Fleet request configuration as follows:

- Specify the tags for the Spot Fleet request in `SpotFleetRequestConfig`.
- For `ResourceType`, specify `spot-fleet-request`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the Spot Fleet request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123.

```
{
  "SpotFleetRequestConfig": {
    "AllocationStrategy": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
    "LaunchSpecifications": [
      {
        "ImageId": "ami-0123456789EXAMPLE",
        "InstanceType": "c4.large"
      }
    ],
    "SpotPrice": "5",
    "TargetCapacity": 2,
    "TerminateInstancesWithExpiration": true,
    "Type": "maintain",
    "ReplaceUnhealthyInstances": true,
    "InstanceInterruptionBehavior": "terminate",
    "InstancePoolsToUseCount": 1,
    "TagSpecifications": [
      {
        "ResourceType": "spot-fleet-request",
        "Tags": [
          {
            "Key": "Environment",
            "Value": "Production"
          },
          {
            "Key": "Cost-Center",
            "Value": "123"
          }
        ]
      }
    ]
  }
}
```

Tag a new Spot Fleet and the instances and volumes that it launches

To tag a new Spot Fleet request and the instances and volumes that it launches using the AWS CLI

To tag a Spot Fleet request when you create it, and to tag the instances and volumes when they are launched by the fleet, configure the Spot Fleet request configuration as follows:

Spot Fleet request tags:

- Specify the tags for the Spot Fleet request in `SpotFleetRequestConfig`.
- For `ResourceType`, specify `spot-fleet-request`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

Instance tags:

- Specify the tags for the instances in `LaunchSpecifications`.
- For `ResourceType`, specify `instance`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

Alternatively, you can specify the tags for the instance in the [launch template](#) 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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
```

```
"IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
"LaunchSpecifications": [
  {
    "ImageId": "ami-0123456789EXAMPLE",
    "InstanceType": "c4.large",
    "TagSpecifications": [
      {
        "ResourceType": "instance",
        "Tags": [
          {
            "Key": "Cost-Center",
            "Value": "123"
          }
        ]
      }
    ]
  }
],
"SpotPrice": "5",
"TargetCapacity": 2,
"TerminateInstancesWithExpiration": true,
"Type": "maintain",
"ReplaceUnhealthyInstances": true,
"InstanceInterruptionBehavior": "terminate",
"InstancePoolsToUseCount": 1,
"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": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
    "LaunchSpecifications": [
      {
        "ImageId": "ami-0123456789EXAMPLE",
        "InstanceType": "c4.large",
        "TagSpecifications": [
          {
            "ResourceType": "instance",
            "Tags": [
              {
                "Key": "Cost-Center",
                "Value": "123"
              }
            ]
          }
        ]
      }
    ],
    "SpotPrice": "5",
    "TargetCapacity": 2,
    "TerminateInstancesWithExpiration": true,
    "Type": "maintain",
    "ReplaceUnhealthyInstances": true,
  }
}
```



```
    "InstanceInterruptionBehavior": "terminate",
    "InstancePoolsToUseCount": 1
  }
}
```

To tag volumes attached to On-Demand Instances launched by a Spot Fleet using the AWS CLI

To tag volumes when they are created by the fleet, you must specify the tags in the [launch template](#) that is referenced in the Spot Fleet request.

Note

Volume tags are only supported for volumes that are attached to On-Demand Instances. You can't tag volumes that are attached to Spot Instances. Volume tagging in `LaunchSpecifications` is not supported.

Tag an existing Spot Fleet

To tag an existing Spot Fleet request using the console

After you have created a Spot Fleet request, you can add tags to the fleet request using the console.

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 your Spot Fleet

The Spot Fleet launches Spot Instances when your maximum price exceeds the Spot price and capacity is available. The Spot Instances run until they are interrupted or you terminate them.

To describe your Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request. To see the configuration details, choose **Description**.
4. To list the Spot Instances for the Spot Fleet, choose **Instances**.
5. To view the history for the Spot Fleet, choose **History**.

To describe your Spot Fleet (AWS CLI)

Use the [describe-spot-fleet-requests](#) command to describe your Spot Fleet requests.

```
aws ec2 describe-spot-fleet-requests
```

Use the [describe-spot-fleet-instances](#) command to describe the Spot Instances for the specified Spot Fleet.

```
aws ec2 describe-spot-fleet-instances \  
  --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Use the [describe-spot-fleet-request-history](#) command to describe the history for the specified Spot Fleet request.

```
aws ec2 describe-spot-fleet-request-history \  
  --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --start-time 2015-05-18T00:00:00Z
```

Modify a Spot Fleet request

You can modify an active Spot Fleet request to complete the following tasks:

- Increase the target capacity and On-Demand portion
- Decrease the target capacity and On-Demand portion

Note

You can't modify a one-time Spot Fleet request. You can only modify a Spot Fleet request if you selected **Maintain target capacity** when you created the Spot Fleet request.

When you increase the target capacity, the Spot Fleet launches additional Spot Instances. When you increase the On-Demand portion, the Spot Fleet launches additional On-Demand Instances.

When you increase the target capacity, the Spot Fleet launches the additional Spot Instances according to the allocation strategy for its Spot Fleet request. If the allocation strategy is `LowestPrice`, the Spot Fleet launches the instances from the lowest priced Spot capacity pool in the Spot Fleet request. If the allocation strategy is `diversified`, the Spot Fleet distributes the instances across the pools in the Spot Fleet request.

When you decrease the target capacity, the Spot Fleet cancels any open requests that exceed the new target capacity. You can request that the Spot Fleet terminate Spot Instances until the size of the fleet reaches the new target capacity. If the allocation strategy is `LowestPrice`, the Spot Fleet terminates the instances with the highest price per unit. If the allocation strategy is `diversified`, the Spot Fleet terminates instances across the pools. Alternatively, you can request that the Spot Fleet keep the fleet at its current size, but not replace any Spot Instances that are interrupted or that you terminate manually.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

To modify a Spot Fleet request (console)

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**.

To modify a Spot Fleet request using the AWS CLI

Use the [modify-spot-fleet-request](#) command to update the target capacity of the specified Spot Fleet request.

```
aws ec2 modify-spot-fleet-request \  
  --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --target-capacity 20
```

You can modify the previous command as follows to decrease the target capacity of the specified Spot Fleet without terminating any Spot Instances as a result.

```
aws ec2 modify-spot-fleet-request \  
  --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --target-capacity 10 \  
  --excess-capacity-termination-policy NoTermination
```

Cancel a Spot Fleet request

If you no longer require a Spot Fleet, you can cancel the Spot Fleet 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, the fleet request enters the `cancelled_running` state and the instances continue to run until they are interrupted or you terminate them manually.

Restrictions

- You can delete up to 100 fleets in a single request. If you exceed the specified number, no fleets are deleted.

To cancel a Spot Fleet request (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 **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** check box selected. To cancel the Spot Fleet request without terminating the associated instances, clear the **Terminate instances** check box.
 - b. Choose **Confirm**.

To cancel a Spot Fleet request and terminate its instances using the AWS CLI

Use the [cancel-spot-fleet-requests](#) command to cancel the specified Spot Fleet request and terminate its On-Demand Instances and Spot Instances.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --terminate-instances
```

Example output

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_terminating",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

To cancel a Spot Fleet request without terminating its instances using the AWS CLI

You can modify the previous command using the `--no-terminate-instances` parameter to cancel the specified Spot Fleet request without terminating its On-Demand Instances and Spot Instances.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --no-terminate-instances
```

```
--no-terminate-instances
```

Example output

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_running",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

CloudWatch metrics for Spot Fleet

Amazon EC2 provides Amazon CloudWatch metrics that you can use to monitor your Spot Fleet.

Important

To ensure accuracy, we recommend that you enable detailed monitoring when using these metrics. For more information, see [Enable or turn off detailed monitoring for your instances](#).

For more information about CloudWatch metrics provided by Amazon EC2, see [Monitor your instances using CloudWatch](#).

Spot Fleet metrics

The AWS/EC2Spot namespace includes the following metrics, plus the CloudWatch metrics for the Spot Instances in your fleet. For more information, see [Instance metrics](#).

Metric	Description
AvailableInstancePoolsCount	The Spot capacity pools specified in the Spot Fleet request. Units: Count

Metric	Description
BidsSubmittedForCapacity	<p>The capacity for which Amazon EC2 has submitted Spot Fleet requests.</p> <p>Units: Count</p>
EligibleInstancePoolCount	<p>The Spot capacity pools specified in the Spot Fleet request where Amazon EC2 can fulfill requests. Amazon EC2 does not fulfill requests in pools where the maximum price you're willing to pay for Spot Instances is less than the Spot price or the Spot price is greater than the price for On-Demand Instances.</p> <p>Units: Count</p>
FulfilledCapacity	<p>The capacity that Amazon EC2 has fulfilled.</p> <p>Units: Count</p>
MaxPercentCapacityAllocation	<p>The maximum value of PercentCapacityAllocation across all Spot Fleet pools specified in the Spot Fleet request.</p> <p>Units: Percent</p>
PendingCapacity	<p>The difference between TargetCapacity and FulfilledCapacity .</p> <p>Units: Count</p>

Metric	Description
PercentCapacityAllocation	<p>The capacity allocated for the Spot capacity pool for the specified dimensions. To get the maximum value recorded across all Spot capacity pools, use <code>MaxPercentCapacityAllocation</code> .</p> <p>Units: Percent</p>
TargetCapacity	<p>The target capacity of the Spot Fleet request.</p> <p>Units: Count</p>
TerminatingCapacity	<p>The capacity that is being terminated because the provisioned capacity is greater than the target capacity.</p> <p>Units: Count</p>

If the unit of measure for a metric is Count, the most useful statistic is Average.

Spot Fleet dimensions

To filter the data for your Spot Fleet, use the following dimensions.

Dimensions	Description
AvailabilityZone	Filter the data by Availability Zone.
FleetRequestId	Filter the data by Spot Fleet request.
InstanceType	Filter the data by instance type.

View the CloudWatch metrics for your Spot Fleet

You can view the CloudWatch metrics for your Spot Fleet using the Amazon CloudWatch console. These metrics are displayed as monitoring graphs. These graphs show data points if the Spot Fleet is active.

Metrics are grouped first by namespace, and then by the various combinations of dimensions within each namespace. For example, you can view all Spot Fleet metrics or Spot Fleet metrics groups by Spot Fleet request ID, instance type, or Availability Zone.

To view Spot Fleet metrics

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2 Spot** namespace.

Note

If the **EC2 Spot** namespace is not displayed, there are two reasons for this. Either you've not yet used Spot Fleet—only the AWS services that you're using send metrics to Amazon CloudWatch. Or, if you've not used Spot Fleet for the past two weeks, the namespace does not appear.

4. (Optional) To filter the metrics by dimension, select one of the following:
 - **Fleet Request Metrics** – Group by Spot Fleet request
 - **By Availability Zone** – Group by Spot Fleet request and Availability Zone
 - **By Instance Type** – Group by Spot Fleet request and instance type
 - **By Availability Zone/Instance Type** – Group by Spot Fleet request, Availability Zone, and instance type
5. To view the data for a metric, select the check box next to the metric.

EC2 Spot Search Metrics 1 to 18 of 18 Metrics

Fleet Request Metrics x By Availability Zone By Instance Type By Availability Zone/Instance Type

Showing all results (18) for EC2 Spot > Fleet Request Metrics. For more results expand your search to All EC2 Spot Metrics. Select All | Clear

EC2 Spot > Fleet Request Metrics

FleetRequestId	Metric Name
<input type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	AvailableInstancePoolsCount
<input type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	BidsSubmittedForCapacity
<input checked="" type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	CPUUtilization
<input type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	DiskReadBytes

Automatic scaling for Spot Fleet

Automatic scaling is the ability to increase or decrease the target capacity of your Spot Fleet automatically based on demand. A Spot Fleet can either launch instances (scale out) or terminate instances (scale in), within the range that you choose, in response to one or more scaling policies.

Spot Fleet supports the following types of automatic scaling:

- [Target tracking scaling](#) – Increase or decrease the current capacity of the fleet based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home—you select temperature and the thermostat does the rest.
- [Step scaling](#) – 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.

If you are using [instance weighting](#), keep in mind that Spot Fleet can exceed the target capacity as needed. Fulfilled capacity can be a floating-point number but target capacity must be an integer, so Spot Fleet rounds up to the next integer. You must take these behaviors into account when you look at the outcome of a scaling policy when an alarm is triggered. For example, suppose that the target capacity is 30, the fulfilled capacity is 30.1, and the scaling policy subtracts 1. When the alarm is triggered, the automatic scaling process subtracts 1 from 30.1 to get 29.1 and then rounds it up to 30, so no scaling action is taken. As another example, suppose that you selected instance weights of 2, 4, and 8, and a target capacity of 10, but no weight 2 instances were available so

Spot Fleet provisioned instances of weights 4 and 8 for a fulfilled capacity of 12. If the scaling policy decreases target capacity by 20% and an alarm is triggered, the automatic scaling process subtracts 12×0.2 from 12 to get 9.6 and then rounds it up to 10, so no scaling action is taken.

The scaling policies that you create for Spot Fleet support a cooldown period. This is the number of seconds after a scaling activity completes where previous trigger-related scaling activities can influence future scaling events. For scale-out policies, while the cooldown period is in effect, the capacity that has been added by the previous scale-out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. The intention is to continuously (but not excessively) scale out. For scale in policies, the cooldown period is used to block subsequent scale in requests until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in, automatic scaling scales out your scalable target immediately.

We recommend that you scale based on instance metrics with a 1-minute frequency because that ensures a faster response to utilization changes. Scaling on metrics with a 5-minute frequency can result in slower response time and scaling on stale metric data. To send metric data for your instances to CloudWatch in 1-minute periods, you must specifically enable detailed monitoring. For more information, see [Enable or turn off detailed monitoring for your instances](#) and [Create a Spot Fleet request using defined parameters \(console\)](#).

For more information about configuring scaling for Spot Fleet, see the following resources:

- [application-autoscaling](#) section of the *AWS CLI Command Reference*
- [Application Auto Scaling API Reference](#)
- [Application Auto Scaling User Guide](#)

IAM permissions required for Spot Fleet automatic scaling

Automatic scaling for Spot Fleet is made possible by a combination of the Amazon EC2, Amazon CloudWatch, and Application Auto Scaling APIs. Spot Fleet requests are created with Amazon EC2, alarms are created with CloudWatch, and scaling policies are created with Application Auto Scaling.

In addition to the [IAM permissions for Spot Fleet](#) and Amazon EC2, the user that accesses fleet scaling settings must have the appropriate permissions for the services that support dynamic scaling. Users must have permissions to use the actions shown in the following example policy.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "application-autoscaling:*",
      "ec2:DescribeSpotFleetRequests",
      "ec2:ModifySpotFleetRequest",
      "cloudwatch:DeleteAlarms",
      "cloudwatch:DescribeAlarmHistory",
      "cloudwatch:DescribeAlarms",
      "cloudwatch:DescribeAlarmsForMetric",
      "cloudwatch:GetMetricStatistics",
      "cloudwatch:ListMetrics",
      "cloudwatch:PutMetricAlarm",
      "cloudwatch:DisableAlarmActions",
      "cloudwatch:EnableAlarmActions",
      "iam:CreateServiceLinkedRole",
      "sns:CreateTopic",
      "sns:Subscribe",
      "sns:Get*",
      "sns:List*"
    ],
    "Resource": "*"
  }
]
```

You can also create your own IAM policies that allow more fine-grained permissions for calls to the Application Auto Scaling API. For more information, see [Authentication and Access Control](#) in the *Application Auto Scaling User Guide*.

The Application Auto Scaling service also needs permission to describe your Spot Fleet and CloudWatch alarms, and permissions to modify your Spot Fleet target capacity on your behalf. If you enable automatic scaling for your Spot Fleet, it creates a service-linked role named `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`. This service-linked role grants Application Auto Scaling permission to describe the alarms for your policies, to monitor the current capacity of the fleet, and to modify the capacity of the fleet. The original managed Spot Fleet role for Application Auto Scaling was `aws-ec2-spot-fleet-autoscale-role`, but it is no longer required. The service-linked role is the default role for Application Auto Scaling. For more information, see [Service-Linked Roles](#) in the *Application Auto Scaling User Guide*.

Scale Spot Fleet using a target tracking policy

With target tracking scaling policies, you select a metric and set a target value. Spot Fleet creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the fluctuations in the metric due to a fluctuating load pattern and minimizes rapid fluctuations in the capacity of the fleet.

You can create multiple target tracking scaling policies for a Spot Fleet, provided that each of them uses a different metric. The fleet scales based on the policy that provides the largest fleet capacity. This enables you to cover multiple scenarios and ensure that there is always enough capacity to process your application workloads.

To ensure application availability, the fleet scales out proportionally to the metric as fast as it can, but scales in more gradually.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Do not edit or delete the CloudWatch alarms that Spot Fleet manages for a target tracking scaling policy. Spot Fleet deletes the alarms automatically when you delete the target tracking scaling policy.

Limitation

The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for requests of type `request`.

To configure a target tracking policy (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Auto Scaling**.
4. If automatic scaling is not configured, choose **Configure**.
5. Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.

6. For **Policy name**, enter a name for the policy.
7. Choose a **Target metric**.
8. Enter a **Target value** for the metric.
9. For **Cooldown period**, specify a new value (in seconds) or keep the default.
10. (Optional) Select **Disable scale-in** to omit creating a scale-in policy based on the current configuration. You can create a scale-in policy using a different configuration.
11. Choose **Save**.

To configure a target tracking policy using the AWS CLI

1. Register the Spot Fleet request as a scalable target using the [register-scalable-target](#) command.
2. Create a scaling policy using the [put-scaling-policy](#) command.

Scale Spot Fleet using step scaling policies

With step scaling policies, you specify CloudWatch alarms to trigger the scaling process. For example, if you want to scale out when CPU utilization reaches a certain level, create an alarm using the `CPUUtilization` metric provided by Amazon EC2.

When you create a step scaling policy, you must specify one of the following scaling adjustment types:

- **Add** – Increase the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Remove** – Decrease the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Set to** – Set the target capacity of the fleet to the specified number of capacity units.

When an alarm is triggered, the automatic scaling process calculates the new target capacity using the fulfilled capacity and the scaling policy, and then updates the target capacity accordingly. For example, suppose that the target capacity and fulfilled capacity are 10 and the scaling policy adds 1. When the alarm is triggered, the automatic scaling process adds 1 to 10 to get 11, so Spot Fleet launches 1 instance.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Limitation

The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for requests of type `request`, or Spot blocks.

Prerequisites

- Consider which CloudWatch metrics are important to your application. You can create CloudWatch alarms based on metrics provided by AWS or your own custom metrics.
- For the AWS metrics that you will use in your scaling policies, enable CloudWatch metrics collection if the service that provides the metrics does not enable it by default.

To create a CloudWatch alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Choose **Create alarm**.
4. On the **Specify metric and conditions** page, choose **Select metric**.
5. Choose **EC2 Spot, Fleet Request Metrics**, select a metric (for example, `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 a description**, enter a name and description for the alarm and choose **Next**.
 14. Choose **Create alarm**.

To configure a step scaling policy for your Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Auto Scaling**.
4. If automatic scaling is not configured, choose **Configure**.
5. Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Scaling policies do not scale your fleet below the minimum capacity or above the maximum capacity.
6. For **Scaling policies**, **Policy type**, choose **Step scaling policy**.
7. 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**.
8. 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, type 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.
9. Choose **Save**.

To configure step scaling policies for your Spot Fleet using the AWS CLI

1. Register the Spot Fleet request as a scalable target using the [register-scalable-target](#) command.
2. Create a scaling policy using the [put-scaling-policy](#) command.
3. Create an alarm that triggers the scaling policy using the [put-metric-alarm](#) command.

Scale Spot Fleet using scheduled scaling

Scaling based on a schedule enables you to scale your application in response to predictable changes in demand. To use scheduled scaling, you create *scheduled actions*, which tell Spot Fleet to perform scaling activities at specific times. When you create a scheduled action, you specify an existing Spot Fleet, when the scaling activity should occur, minimum capacity, and maximum capacity. You can create scheduled actions that scale one time only or that scale on a recurring schedule.

You can only create a scheduled action for Spot Fleets that already exist. You can't create a scheduled action at the same time that you create a Spot Fleet.

Limitation

The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for requests of type `request`, or Spot blocks.

To create a one-time scheduled action

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose the **Scheduled Scaling** tab near the bottom of the screen.
4. Choose **Create Scheduled Action**.
5. For **Name**, specify a name for the scheduled action.
6. Enter a value for **Minimum capacity**, **Maximum capacity**, or both.
7. For **Recurrence**, choose **Once**.
8. (Optional) Choose a date and time for **Start time**, **End time**, or both.
9. Choose **Submit**.

To scale on a recurring schedule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose the **Scheduled Scaling** tab near the bottom of the screen.
4. For **Recurrence**, choose one of the predefined schedules (for example, **Every day**), or choose **Custom** and enter a cron expression. For more information about the cron expressions supported by scheduled scaling, see [Cron Expressions](#) in the *Amazon CloudWatch Events User Guide*.
5. (Optional) Choose a date and time for **Start time**, **End time**, or both.
6. Choose **Submit**.

To edit a scheduled action

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose the **Scheduled Scaling** tab near the bottom of the screen.
4. Select the scheduled action and choose **Actions, Edit**.
5. Make the needed changes and choose **Submit**.

To delete a scheduled action

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose the **Scheduled Scaling** tab near the bottom of the screen.
4. Select the scheduled action and choose **Actions, Delete**.
5. When prompted for confirmation, choose **Delete**.

To manage scheduled scaling using the AWS CLI

Use the following commands:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Monitor fleet events using Amazon EventBridge

When the state of an EC2 Fleet or Spot Fleet changes, the fleet emits a notification. The notification is made available as an event that is sent to Amazon EventBridge (formerly known as Amazon CloudWatch Events). Events are emitted on a best effort basis.

With Amazon EventBridge, you can create rules that trigger programmatic actions in response to an event. For example, you can create two EventBridge rules, one that's triggered when a fleet state changes, and one that's triggered when an instance in the fleet is terminated. You can configure the first rule so that, if the fleet state changes, the rule invokes an SNS topic to send an email notification to you. You can configure the second rule so that, if an instance is terminated, the rule invokes a Lambda function to launch a new instance.

Topics

- [EC2 Fleet event types](#)
- [Spot Fleet event types](#)
- [Create Amazon EventBridge rules](#)

EC2 Fleet event types

Note

Only fleets of type `maintain` and `request` emit events. Fleets of type `instant` do not emit events because they submit synchronous one-time requests, and the state of the fleet is known immediately in the response.

There are five EC2 Fleet event types. For each event type, there are several sub-types.

The events are sent to EventBridge in JSON format. The following fields in the event form the event pattern that is defined in the rule, and which trigger an action:

```
"source": "aws.ec2fleet"
```

Identifies that the event is from EC2 Fleet.

```
"detail-type": "EC2 Fleet State Change"
```

Identifies the event type.

```
"detail": { "sub-type": "submitted" }
```

Identifies the event sub-type.

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-6b0809bffff0a"
  ],
}
```

```
"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 fulfill its target capacity.

The following is example data for this event.

```
{
  "version": "0",
  "id": "76529817-d605-4571-7224-d36cc1b2c0c4",
  "detail-type": "EC2 Fleet Information",
  "source": "aws.ec2fleet",
  "account": "123456789012",
  "time": "2020-11-09T08:17:07Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-8becf5fe-
    bb9e-415d-8f54-3fa5a8628b91"
  ]
}
```

```
    ],
    "detail": {
      "description": "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.

The events are sent to EventBridge in JSON format. The following fields in the event form the event pattern that is defined in the rule, and which trigger an action:

```
"source": "aws.ec2spotfleet"
```

Identifies that the event is from Spot Fleet.

```
"detail-type": "EC2 Spot Fleet State Change"
```

Identifies the event type.

```
"detail": { "sub-type": "submitted" }
```

Identifies the event sub-type.

Event types

- [EC2 Spot Fleet State Change](#)
- [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.

Create Amazon EventBridge rules

When a notification of a state change is emitted for an EC2 Fleet or Spot Fleet, the event for the notification is sent to Amazon EventBridge. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes a target (or targets) specified in the rule.

You can write an EventBridge rule and automate what actions to take when the event pattern matches the rule.

Topics

- [Create Amazon EventBridge rules to monitor EC2 Fleet events](#)
- [Create Amazon EventBridge rules to monitor Spot Fleet events](#)

Create Amazon EventBridge rules to monitor EC2 Fleet events

When a notification of a state change is emitted for an EC2 Fleet, the event for the notification is sent to Amazon EventBridge in the form of a JSON file. You can write an EventBridge rule to automate what actions to take when an event pattern matches the rule. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes the target (or targets) specified in the rule.

The following fields form the event pattern that is defined in the rule:

```
"source": "aws.ec2fleet"
```

Identifies that the event is from EC2 Fleet.

```
"detail-type": "EC2 Fleet State Change"
```

Identifies the event type.

```
"detail": { "sub-type": "submitted" }
```

Identifies the event sub-type.

For the list of EC2 Fleet events and example event data, see [the section called “EC2 Fleet event types”](#).

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.

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 about using Lambda, see [Create a Lambda function with the console](#) 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*.

Create Amazon EventBridge rules to monitor Spot Fleet events

When a notification of a state change is emitted for a Spot Fleet, the event for the notification is sent to Amazon EventBridge in the form of a JSON file. You can write an EventBridge rule to automate what actions to take when an event pattern matches the rule. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes the target (or targets) specified in the rule.

The following fields form the event pattern that is defined in the rule:


```
"source": "aws.ec2spotfleet"
```

Identifies that the event is from Spot Fleet.

```
"detail-type": "EC2 Spot Fleet State Change"
```

Identifies the event type.

```
"detail": { "sub-type": "submitted" }
```

Identifies the event sub-type.

For the list of Spot Fleet events and example event data, see [the section called “Spot Fleet event types”](#).

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 a Spot Fleet state change notification. The signal in this example is emitted as an EC2 Spot Fleet State Change event, which triggers the action defined by the rule. Before creating the EventBridge rule, you must create the Amazon SNS topic for the email, text message, or mobile push notification.

To create an EventBridge rule to send a notification when a Spot Fleet state changes

1. Open the Amazon EventBridge console at <https://console.aws.amazon.com/events/>.
2. Choose **Create rule**.
3. 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 Spot Fleet Instance Change event.

```
{
  "source": ["aws.ec2spotfleet"],
  "detail-type": ["EC2 Spot 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 Spot Fleet**.
 - D. For **Event type**, choose **EC2 Spot 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 a Spot Fleet instance change notification for when an instance is launched. The signal in this example is emitted as an EC2 Spot Fleet Instance Change event, sub-type launched, which triggers the action defined by the rule.

Before creating the EventBridge rule, you must create the Lambda function.

To create 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 about using Lambda, see [Create a Lambda function with the console](#) in the *AWS Lambda Developer Guide*.

To create an EventBridge rule to trigger a Lambda function when an instance in a Spot 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 Spot Fleet Instance Change event and `launched` sub-type.

```
{
  "source": ["aws.ec2spotfleet"],
  "detail-type": ["EC2 Spot 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 Spot Fleet**.
 - D. For **Event type**, choose **EC2 Spot 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*.

Tutorials for EC2 Fleet and Spot Fleet

The following tutorials take you through the common processes for creating EC2 Fleets and Spot Fleets.

Tutorials

- [Tutorial: Use EC2 Fleet with instance weighting](#)
- [Tutorial: Use EC2 Fleet with On-Demand as the primary capacity](#)
- [Tutorial: Launch On-Demand Instances using targeted Capacity Reservations](#)

- [Tutorial: Launch instances into Capacity Blocks](#)
- [Tutorial: Use Spot Fleet with instance weighting](#)

Tutorial: Use EC2 Fleet with instance weighting

This tutorial uses a fictitious company called Example Corp to illustrate the process of requesting an EC2 Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to use the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the requirements for their EC2 Fleet.

Instance types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

Target capacity in units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the

base for their application (60 GB of RAM and eight vCPUs) as one unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their EC2 Fleet request to 20.

Instance weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their EC2 Fleet request.

Price per unit hour

Example Corp uses the [On-Demand price](#) per instance hour as a starting point for their price. They could also use recent Spot prices, or a combination of the two. To calculate the price per unit hour, they divide their starting price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp could use a global price per unit hour of \$0.7 and be competitive for all three instance types. They could also use a global price per unit hour of \$0.7 and a specific price per unit hour of \$0.9 in the r3.8xlarge launch specification.

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 a 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: Use EC2 Fleet with On-Demand as the primary capacity

This tutorial uses a fictitious company called ABC Online to illustrate the process of requesting an EC2 Fleet with On-Demand as the primary capacity, and Spot capacity if available.

Objective

ABC Online, a restaurant delivery company, wants to be able to provision Amazon EC2 capacity across EC2 instance types and purchasing options to achieve their desired scale, performance, and cost.

Plan

ABC Online requires a fixed capacity to operate during peak periods, but would like to benefit from increased capacity at a lower price. ABC Online determines the following requirements for their EC2 Fleet:

- On-Demand Instance capacity – ABC Online requires 15 On-Demand Instances to ensure that they can accommodate traffic at peak periods.
- Spot Instance capacity – ABC Online would like to improve performance, but at a lower price, by provisioning 5 Spot Instances.

Verify permissions

Before creating an EC2 Fleet, ABC Online verifies that it has an IAM role with the required permissions. For more information, see [EC2 Fleet prerequisites](#).

Create a launch template

Next, ABC Online creates a launch template. The launch template ID is used in the following step. For more information, see [Create a launch template](#).

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 capacity and availability.

Tutorial: 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 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.

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 for On-Demand Instances](#).

```
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](#) 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: 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. For more information about Capacity Blocks, see [Capacity Blocks for ML](#).

You can use EC2 Fleet of type *instant* to launch instances into Capacity Blocks. For more information, see [Use an EC2 Fleet of type 'instant'](#).

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 stop or terminate them through a separate Amazon EC2 API call or until Amazon EC2 terminates the instances when the Capacity Block reservation ends.

Considerations

- Multiple Capacity Blocks in the same `CreateFleet` 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.

Create a launch template

The launch template ID is used in the following step. For more information, see [Create a launch template](#).

To configure the launch template, for `InstanceMarketOptionsRequest`, set `MarketType` to `capacity-block`. Specify the Capacity Block reservation ID that you are targeting by setting the `CapacityReservationID` parameter.

Create the EC2 Fleet

Create 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": {
        "LaunchTemplateName": "CBR-launch-template",
        "Version": "1"
      },
      "Overrides": [
        {
          "InstanceType": "p5.48xlarge",
          "AvailabilityZone": "us-east-1a"
        }
      ]
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 10,
    "DefaultTargetCapacityType": "capacity-block"
  },
  "Type": "instant"
}
```

Use the following [create-fleet](#) command.

```
aws ec2 create-fleet \
```

```
--cli-input-json file://config.json
```

For more information, see [Create an EC2 Fleet](#).

Tutorial: Use Spot Fleet with instance weighting

This tutorial uses a fictitious company called Example Corp to illustrate the process of requesting a Spot Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to leverage the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the following requirements for their Spot Fleet.

Instance types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

Target capacity in units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as 1 unit, Example Corp decides that 20

times this amount would meet their needs. So the company sets the target capacity of their Spot Fleet request to 20.

Instance weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their Spot Fleet request.

Price per unit hour

Example Corp uses the [On-Demand price](#) per instance hour as a starting point for their price. They could also use recent Spot prices, or a combination of the two. To calculate the price per unit hour, they divide their starting price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp could use a global price per unit hour of \$0.7 and be competitive for all three instance types. They could also use a global price per unit hour of \$0.7 and a specific price per unit hour of \$0.9 in the r3.8xlarge launch specification.

Verify permissions

Before creating a Spot Fleet request, Example Corp verifies that it has an IAM role with the required permissions. For more information, see [Spot Fleet permissions](#).

Create the request

Example Corp creates a file, `config.json`, with the following configuration for its Spot Fleet request:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 1
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.4xlarge",
      "SubnetId": "subnet-482e4972",
      "WeightedCapacity": 2
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.8xlarge",
      "SubnetId": "subnet-482e4972",
      "SpotPrice": "0.90",
      "WeightedCapacity": 4
    }
  ]
}
```

Example Corp creates the Spot Fleet request using the [request-spot-fleet](#) command.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For more information, see [Spot Fleet request types](#).

Fulfillment

The allocation strategy determines which Spot capacity pools your Spot Instances come from.

With the `lowestPrice` strategy (which is the default strategy), the Spot Instances come from the pool with the lowest price per unit at the time of fulfillment. To provide 20 units of capacity, the Spot Fleet launches either 20 `r3.2xlarge` instances (20 divided by 1), 10 `r3.4xlarge` instances (20 divided by 2), or 5 `r3.8xlarge` instances (20 divided by 4).

If Example Corp used the `diversified` strategy, the Spot Instances would come from all three pools. The Spot Fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 units.

Example configurations for EC2 Fleet and Spot Fleet

The following examples show launch configurations that you can use to create EC2 Fleets and Spot Fleets.

Topics

- [EC2 Fleet example configurations](#)
- [Spot Fleet example configurations](#)

EC2 Fleet example configurations

The following examples show launch configurations that you can use with the [create-fleet](#) command to create an EC2 Fleet. For more information about the parameters, see [create-fleet](#) in the *AWS CLI Command Reference*.

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 Spot Instances using the lowest-price allocation strategy](#)
- [Example 5: Launch On-Demand Instances using multiple Capacity Reservations](#)
- [Example 6: Launch On-Demand Instances using Capacity Reservations when the total target capacity exceeds the number of unused Capacity Reservations](#)
- [Example 7: Launch On-Demand Instances using targeted Capacity Reservations](#)
- [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 11: Launch Spot Instances in a price-capacity-optimized fleet](#)
- [Example 12: Configure attribute-based instance type selection](#)

Example 1: Launch Spot Instances as the default purchasing option

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing option. The launch template is identified by its launch template ID and version number. The target capacity for the fleet is 2 instances, and the default purchasing option is spot, which results in the fleet launching 2 Spot Instances.

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "spot"
  }
}
```

Example 2: Launch On-Demand Instances as the default purchasing option

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing option. The launch template is identified by its launch template ID and version number. The target capacity for the fleet is 2 instances, and the default purchasing option is on-demand, which results in the fleet launching 2 On-Demand Instances.

```
{
  "LaunchTemplateConfigs": [
    {
```



```
        "LaunchTemplateSpecification": {
            "LaunchTemplateId": "lt-0e8c754449b27161c",
            "Version": "1"
        }
    },
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "on-demand"
    }
}
```

Example 3: Launch On-Demand Instances as the primary capacity

The following example specifies the total target capacity of 2 instances for the fleet, and a target capacity of 1 On-Demand Instance. The default purchasing option is spot. The fleet launches 1 On-Demand Instance as specified, but needs to launch one more instance to fulfill the total target capacity. The purchasing option for the difference is calculated as $TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType$, which results in the fleet launching 1 Spot Instance.

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "OnDemandTargetCapacity": 1,
    "DefaultTargetCapacityType": "spot"
  }
}
```

Example 4: Launch Spot Instances using the lowest-price allocation strategy

If the allocation strategy for Spot Instances is not specified, the default allocation strategy, which is lowest-price, is used. The following example uses the lowest-price allocation strategy. The three launch specifications, which override the launch template, have different instance types but the same weighted capacity and subnet. The total target capacity is 2 instances and the default purchasing option is spot. The EC2 Fleet launches 2 Spot Instances using the instance type of the launch specification with the lowest price.

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "Overrides": [
    {
      "InstanceType": "c4.large",
      "WeightedCapacity": 1,
      "SubnetId": "subnet-a4f6c5d3"
    },
    {
      "InstanceType": "c3.large",
      "WeightedCapacity": 1,
      "SubnetId": "subnet-a4f6c5d3"
    },
    {
      "InstanceType": "c5.large",
      "WeightedCapacity": 1,
      "SubnetId": "subnet-a4f6c5d3"
    }
  ]
},
"TargetCapacitySpecification": {
  "TotalTargetCapacity": 2,
  "DefaultTargetCapacityType": "spot"
}
}
```

Example 5: Launch On-Demand Instances using 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 6: 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 7: 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: 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 8: 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 [Capacity Rebalancing](#).

```
{
  "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 9: 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 10: 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 11: 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 12: 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"
}

```

Spot Fleet example configurations

The following examples show launch configurations that you can use with the [request-spot-fleet](#) command to create a Spot Fleet request. For more information, see [Create a Spot Fleet request](#).

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 or subnet in the Region](#)

- [Example 2: Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list](#)
- [Example 3: Launch Spot Instances using the lowest-priced instance type in a specified 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 with 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 11: Launch Spot Instances in a priceCapacityOptimized fleet](#)
- [Example 12: Configure attribute-based instance type selection](#)

Example 1: Launch Spot Instances using the lowest-priced Availability Zone or subnet in the Region

The following example specifies a single launch specification without an Availability Zone or subnet. The Spot Fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. The price you pay does not exceed the On-Demand price.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

```
]
}
```

Example 2: Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list

The following examples specify two launch specifications with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

The Spot Fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "Placement": {
        "AvailabilityZone": "us-west-2a, us-west-2b"
      },
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Subnets

You can specify default subnets or nondefault subnets, and the nondefault subnets can be from a default VPC or a nondefault VPC. The Spot service launches the instances in whichever subnet is in the lowest-priced Availability Zone.

You can't specify different subnets from the same Availability Zone in a Spot Fleet request.

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

If the instances are launched in a default VPC, they receive a public IPv4 address by default. If the instances are launched in a nondefault VPC, they do not receive a public IPv4 address by default. Use a network interface in the launch specification to assign a public IPv4 address to instances launched in a nondefault VPC. When you specify a network interface, you must include the subnet ID and security group ID using the network interface.

```
...
  {
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "InstanceType": "m3.medium",
    "NetworkInterfaces": [
      {
        "DeviceIndex": 0,
        "SubnetId": "subnet-1a2b3c4d",
        "Groups": [ "sg-1a2b3c4d" ],
        "AssociatePublicIpAddress": true
      }
    ],
  },
```

```
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
    }
  }
  ...
```

Example 3: Launch Spot Instances using the lowest-priced instance type in a specified list

The following examples specify two launch configurations with different instance types, but the same AMI and Availability Zone or subnet. The Spot Fleet launches the instances using the specified instance type with the lowest price.

Availability Zone

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "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 [Allocation strategies for Spot Instances](#).

Availability Zone

```

{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c4.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "m3.2xlarge",
      "Placement": {

```

```

        "AvailabilityZone": "us-west-2b"
    }
},
{
    "ImageId": "ami-1a2b3c4d",
    "InstanceType": "r3.2xlarge",
    "Placement": {
        "AvailabilityZone": "us-west-2b"
    }
}
]
}

```

Subnet

```

{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}

```

A best practice to increase the chance that a spot request can be fulfilled by EC2 capacity in the event of an outage in one of the Availability Zones is to diversify across zones. For this scenario,

include each Availability Zone available to you in the launch specification. And, instead of using the same subnet each time, use three unique subnets (each mapping to a different zone).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c4.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2a"
      }
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "m3.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2c"
      }
    }
  ]
}
```

Subnet

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
```

```
{
  "ImageId": "ami-1a2b3c4d",
  "InstanceType": "c4.2xlarge",
  "SubnetId": "subnet-1a2b3c4d"
},
{
  "ImageId": "ami-1a2b3c4d",
  "InstanceType": "m3.2xlarge",
  "SubnetId": "subnet-2a2b3c4d"
},
{
  "ImageId": "ami-1a2b3c4d",
  "InstanceType": "r3.2xlarge",
  "SubnetId": "subnet-3a2b3c4d"
}
]
```

Example 6: Launch a Spot Fleet using instance weighting

The following examples use instance weighting, which means that the price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight. The Spot Fleet selects the instance type with the lowest price per unit hour. The Spot Fleet calculates the number of Spot Instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot Fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

If the `r3.2xlarge` request is successful, Spot provisions 4 of these instances. Divide 20 by 6 for a total of 3.33 instances, then round up to 4 instances.

If the `c3.xlarge` request is successful, Spot provisions 7 of these instances. Divide 20 by 3 for a total of 6.66 instances, then round up to 7 instances.

For more information, see [Spot Fleet instance weighting](#).

Availability Zone

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
```

```
"LaunchSpecifications": [  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "r3.2xlarge",  
    "Placement": {  
      "AvailabilityZone": "us-west-2b"  
    },  
    "WeightedCapacity": 6  
  },  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "c3.xlarge",  
    "Placement": {  
      "AvailabilityZone": "us-west-2b"  
    },  
    "WeightedCapacity": 3  
  }  
]  
}
```

Subnet

```
{  
  "SpotPrice": "0.70",  
  "TargetCapacity": 20,  
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
  "LaunchSpecifications": [  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "r3.2xlarge",  
      "SubnetId": "subnet-1a2b3c4d",  
      "WeightedCapacity": 6  
    },  
    {  
      "ImageId": "ami-1a2b3c4d",  
      "InstanceType": "c3.xlarge",  
      "SubnetId": "subnet-1a2b3c4d",  
      "WeightedCapacity": 3  
    }  
  ]  
}
```

Example 7: Launch a Spot Fleet with On-Demand capacity

To ensure that you always have instance capacity, you can include a request for On-Demand capacity in your Spot Fleet request. If there is capacity, the On-Demand request is always fulfilled. The balance of the target capacity is fulfilled as Spot if there is capacity and availability.

The following example specifies the desired target capacity as 10, of which 5 must be On-Demand capacity. Spot capacity is not specified; it is implied in the balance of the target capacity minus the On-Demand capacity. Amazon EC2 launches 5 capacity units as On-Demand, and 5 capacity units (10-5=5) as Spot if there is available Amazon EC2 capacity and availability.

For more information, see [On-Demand in Spot Fleet](#).

```
{
  "IamFleetRole": "arn:aws:iam::781603563322:role/aws-ec2-spot-fleet-tagging-role",
  "AllocationStrategy": "lowestPrice",
  "TargetCapacity": 10,
  "SpotPrice": null,
  "ValidFrom": "2018-04-04T15:58:13Z",
  "ValidUntil": "2019-04-04T15:58:13Z",
  "TerminateInstancesWithExpiration": true,
  "LaunchSpecifications": [],
  "Type": "maintain",
  "OnDemandTargetCapacity": 5,
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",
        "Version": "2"
      },
      "Overrides": [
        {
          "InstanceType": "t2.medium",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-d0dc51fb"
        }
      ]
    }
  ]
}
```


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 [Capacity Rebalancing](#).

```
{
  "SpotFleetRequestConfig": {
    "AllocationStrategy": "capacityOptimized",
    "IamFleetRole": "arn:aws:iam::000000000000: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
      }
    }
  }
]}
}
```

Fleet quotas

The usual Amazon EC2 quotas (formerly referred to as limits) apply to instances launched by an EC2 Fleet or a Spot Fleet, such as [Spot Instance limits](#) and [volume limits](#).

In addition, the following quotas apply:

Quota description	Quota
The number of EC2 Fleets and Spot Fleets per Region of type <code>maintain</code> and <code>request</code> in the <code>active</code> , <code>deleted_running</code> , and <code>cancelled_running</code> states	1,000 ^{1 2 3}
The number of EC2 Fleets of type <code>instant</code>	Unlimited
The number of Spot capacity pools (unique combination of instance type and subnet) for EC2 Fleets and Spot Fleets of type <code>maintain</code> and <code>request</code>	300 ^{1 4}
The number of Spot capacity pools (unique combination of instance type and subnet) for EC2 Fleets of type <code>instant</code>	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 cannot 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](#) and [Cancel 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 AWS 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**.

Monitor Amazon EC2

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions. You should collect monitoring data from all of the parts in your AWS solutions so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Amazon EC2, however, you should create a monitoring plan that should include:

- What are your goals for monitoring?
- What resources will you monitor?
- How often will you monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you have defined your monitoring goals and have created your monitoring plan, the next step is to establish a baseline for normal Amazon EC2 performance in your environment. You should measure Amazon EC2 performance at various times and under different load conditions. As you monitor Amazon EC2, you should store a history of monitoring data that you collect. You can compare current Amazon EC2 performance to this historical data to help you to identify normal performance patterns and performance anomalies, and devise methods to address them. For example, you can monitor CPU utilization, disk I/O, and network utilization for your EC2 instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

Item to monitor	Amazon EC2 metric	Monitoring agent/CloudWatch Logs
CPU utilization	CPUUtilization	
Network utilization	NetworkIn NetworkOut	

Item to monitor	Amazon EC2 metric	Monitoring agent/CloudWatch Logs
Disk performance	DiskReadOps DiskWriteOps	
Disk Reads/Writes	DiskReadBytes DiskWriteBytes	
Memory utilization, disk swap utilization, disk space utilization, page file utilization, log collection		<p>[Linux and Windows Server instances] Collect Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent</p> <p>[Migration from previous CloudWatch Logs agent on Windows Server instances] Migrate Windows Server Instance Log Collection to the CloudWatch Agent</p>

Automated and manual monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Monitoring tools

- [Automated monitoring tools](#)
- [Manual monitoring tools](#)

Automated monitoring tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System status checks** – monitor the AWS systems required to use your instance to ensure that they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
 - Loss of network connectivity
 - Loss of system power
 - Software issues on the physical host
 - Hardware issues on the physical host that impact network reachability

For more information, see [Status checks for your instances](#).

- **Instance status checks** – monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:
 - Failed system status checks
 - Misconfigured networking or startup configuration
 - Exhausted memory
 - Corrupted file system
 - Incompatible kernel

For more information, see [Status checks for your instances](#).

- **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** – automate your AWS services and respond automatically to system events. Events from AWS services are delivered to EventBridge in near real time, and you can specify automated actions to take when an event matches a rule you write. For more information, see [What is Amazon EventBridge?](#).
- **Amazon CloudWatch Logs** – monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. For more information, see the [Amazon CloudWatch Logs User Guide](#).
- **CloudWatch agent** – collect logs and system-level metrics from both hosts and guests on your EC2 instances and on-premises servers. For more information, see [Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent](#) in the *Amazon CloudWatch User Guide*.

Manual monitoring tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
 - Service Health and Scheduled Events by Region
 - Instance state
 - Status checks
 - Alarm status
 - Instance metric details (In the navigation pane choose **Instances**, select an instance, and choose the **Monitoring** tab)
 - Volume metric details (In the navigation pane choose **Volumes**, select a volume, and choose the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems
- See at-a-glance overviews of your alarms and AWS resources

Best practices for monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.
- Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
 - What are your goals for monitoring?
 - What resources will you monitor?
 - How often will you monitor these resources?
 - What monitoring tools will you use?
 - Who will perform the monitoring tasks?
 - Who should be notified when something goes wrong?
- Automate monitoring tasks as much as possible.
- Check the log files on your EC2 instances.

Monitor the status of your instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances.

A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status of specific events that are scheduled for your instances. The status of events provides information about upcoming activities that are planned for your instances, such as rebooting or retirement. They also provide the scheduled start and end time of each event.

Contents

- [Status checks for your instances](#)
- [State change events for your instances](#)
- [Scheduled events for your instances](#)

Status checks for your instances

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. The event status data augments the information that Amazon EC2 already provides about the state of each instance (such as pending, running, stopping) and the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute, returning a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted.

When a status check fails, the corresponding CloudWatch metric for status checks is incremented. For more information, see [Status check metrics](#). You can use these metrics to create CloudWatch alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see [Create and edit status check alarms](#).

You can also create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying issue. For more information, see [Recover your instance](#).

Contents

- [Types of status checks](#)
- [Working with status checks](#)

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 Linux instances backed by instance store, you can terminate and replace the instance. For Windows instances, the root volume must be an Amazon EBS volume; instance store is not supported for the root volume. Note that instance store volumes are ephemeral and all data is lost when the instance is stopped.

The following are examples of problems that can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

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 configuration of your individual instance. Amazon EC2 checks the health of the instance by sending an address resolution protocol (ARP) request to the network interface (NIC). These checks detect problems that require your

involvement to repair. When an instance status check fails, you typically must address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel

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 of the EBS volumes attached to your instance, and the attached EBS status check indicates an impairment, you can stop and start the instance to address issues with the physical host that is impacting the reachability of the EBS volumes. For more information, see [Amazon CloudWatch metrics for Amazon EBS](#)

Note

- The attached EBS status check metric is available only for Nitro instances.
- You can monitor the attached EBS status check metric by [creating a CloudWatch alarm](#) based on the `StatusCheckFailed_AttachedEBS` metric. You can't view this status check by using the [describe-instance-status](#) AWS CLI command.

Working with status checks

You can work with status checks using the console and command line tools, such as the AWS CLI.

Topics

- [View status checks](#)
- [Create and edit status check alarms](#)

View status checks

To view status checks, use one of the following methods.

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.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availi
spot-instance-2	i-01aeed690c9fb5322	Running	t3.nano	1/2 checks ...	View alarms +	eu-w
spot-instance-1	i-0ba5e5bbc9d634fa6	Stopped	t3.nano	-	View alarms +	eu-w
EIC-RHEL	i-08e66e73da739c7f4	Running	t2.micro	2/2 checks passed	View alarms +	eu-w
Windows	i-0cb952751a0d8388b	Running	t3.nano	2/2 checks passed	View alarms +	eu-w

Instance: i-01aeed690c9fb5322 (spot-instance-2)

Details | **Status and alarms New** | Monitoring | Security | Networking | Storage | Tags

Status checks Info

Status checks detect problems that may impair i-01aeed690c9fb5322 (spot-instance-2) from running your applications.

System status checks

- System reachability check passed
- Metrics
- Alarms

Instance status checks

- Instance reachability check failed
- Check failure at 2020/12/16 17:30 GMT+2 (about 1 month)

Name	State	Description	Metric name	State reason
Instance has no associated alarms				

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 instances with failed status checks](#).

- 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**

For more information, see [the section called "Status check metrics"](#).

Command line

You can view status checks for running instances by using the [describe-instance-status](#) (AWS CLI) command.

To view the status of all instances, use the following command.

```
aws ec2 describe-instance-status
```

To get the status of all instances with an instance status of `impaired`, use the following command.

```
aws ec2 describe-instance-status \
  --filters Name=instance-status.status,Values=impaired
```

To get the status of a single instance, use the following command.

```
aws ec2 describe-instance-status \
  --instance-ids i-1234567890abcdef0
```

Alternatively, use the following commands:

- [Get-EC2InstanceStatus](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceStatus](#) (Amazon EC2 Query API)

If you have an instance with a failed status check, see [Troubleshoot instances with failed status checks](#).

Create and edit status check alarms

You can use the [status check metrics](#) to create CloudWatch alarms to notify you when an instance has a failed status check.

To create a status check alarm, use one of the following methods:

Console

Use the following procedure to configure an alarm that sends you a notification by email, or stops, terminates, or recovers an instance when it fails a status check.

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 created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by choosing the link contained in that message. Alert notifications are sent only to confirmed addresses.

6. For **Alarm action**, turn the toggle on to specify an action to take when the alarm is triggered. Select the action.
7. For **Alarm thresholds**, specify the metric and criteria for the alarm.

You can leave the default settings for **Group samples by (Average)** and **Type of data to sample (Status check failed:either)**, or you can change them to suit your needs.

For **Consecutive period**, set the number of periods to evaluate and, in **Period**, enter the evaluation period duration before triggering the alarm and sending an email.

8. (Optional) For **Sample metric data**, choose **Add to dashboard**.
9. Choose **Create**.

If you need to make changes to an instance status alarm, you can edit it.

To edit a status check alarm

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance 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**.

Command line

In the following example, the alarm publishes a notification to an SNS topic, `arn:aws:sns:us-west-2:111122223333:my-sns-topic`, when the instance fails either the instance check or system status check for at least two consecutive periods. The CloudWatch metric used is `StatusCheckFailed`.

To create a status check alarm using the AWS CLI

1. Select an existing SNS topic or create a new one. For more information, see [Using the AWS CLI with Amazon SNS](#) in the *AWS Command Line Interface User Guide*.
2. Use the following [list-metrics](#) command to view the available Amazon CloudWatch metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following [put-metric-alarm](#) command to create the alarm.

```
aws cloudwatch put-metric-alarm \  
  --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 \  
  --metric-name StatusCheckFailed \  
  --namespace AWS/EC2 \  
  --statistic Maximum \  
  --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \  
  --unit Count \  
  --period 300 \  
  --evaluation-periods 2 \  
  --threshold 1 \  
  --comparison-operator GreaterThanOrEqualToThreshold \  
  --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

The period is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes. The evaluation period is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2. The alarm actions are the actions to perform when this alarm is triggered. This example configures the alarm to send an email using Amazon SNS.

State change events for your 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-abcd1111"
  ],
  "detail": {
    "instance-id": "i-abcd1111",
    "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.

Get an email notification when an 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 your instances

AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account prior to the scheduled event. The email provides details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event. AWS also sends an AWS Health event, which you can monitor and manage by using Amazon CloudWatch Events. For more information about monitoring AWS Health events with CloudWatch, see [Monitoring AWS Health events with CloudWatch Events](#).

Scheduled events are managed by AWS; you cannot schedule events for your instances. You can view the events scheduled by AWS, customize scheduled event notifications to include or remove tags from the email notification, and perform actions when an instance is scheduled to reboot, retire, or stop.

To update the contact information for your account so that you can be sure to be notified about scheduled events, go to the [Account Settings](#) page.

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 Auto Scaling instances](#) in the *Amazon EC2 Auto Scaling User Guide*.

Contents

- [Types of scheduled events](#)
- [View scheduled events](#)
- [Customize scheduled event notifications](#)
- [Work with instances scheduled to stop or retire](#)
- [Work with instances scheduled for reboot](#)
- [Work with instances scheduled for maintenance](#)
- [Reschedule a scheduled event](#)
- [Define event windows for scheduled events](#)

Types of scheduled events

Amazon EC2 can create the following types of events for your instances, where the event occurs at a scheduled time:

- **Instance stop:** At the scheduled time, the instance is stopped. When you start it again, it's migrated to a new host. Applies only to instances backed by Amazon EBS.
- **Instance retirement:** At the scheduled time, the instance is stopped if it is backed by Amazon EBS, or terminated if it is backed by instance store.
- **Instance reboot:** At the scheduled time, the instance is rebooted.
- **System reboot:** At the scheduled time, the host for the instance is rebooted.
- **System maintenance:** At the scheduled time, the instance might be temporarily affected by network maintenance or power maintenance.

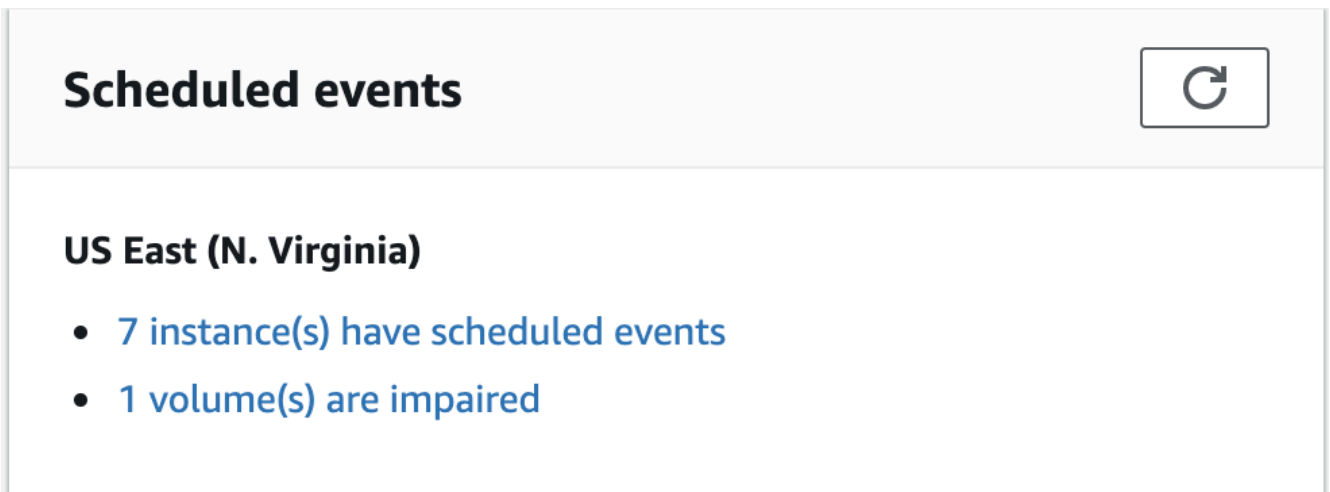
View scheduled events

In addition to receiving notification of scheduled events in email, you can check for scheduled events by using one of the following methods.

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.

Resource ID	Event status	Event type	Description	Progress	Duration	Start time
I-02c48ffba61cd16f	Scheduled	instance-stop	The instance is running on ...	Starts in 13 days		2019/07/22 13:00 GMT+2

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 using the AWS Tools for Windows PowerShell

Use the following [Get-EC2InstanceStatus](#) command.

```
PS C:\> (Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0).Events
```

The following example output shows an instance retirement event.

```
Code       : instance-stop
Description : The instance is running on degraded hardware
NotBefore  : 5/23/2015 12:00:00 AM
```

Instance metadata

To view scheduled events for your instances using instance metadata

You can retrieve information about active maintenance events for your instances from the [instance metadata](#) 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 the AWS Health Dashboard](#) in the *AWS Health User Guide*.

Customize scheduled event notifications

You can customize scheduled event notifications to include tags in the email notification. This makes it easier to identify the affected resource (instances or Dedicated Hosts) and to prioritize actions for the upcoming event.

When you customize event notifications to include tags, you can choose to include:

- All of the tags that are associated with the affected resource
- Only specific tags that are associated with the affected resource

For example, suppose that you assign `application`, `costcenter`, `project`, and `owner` tags to all of your instances. You can choose to include all of the tags in event notifications. Alternatively, if you'd like to see only the `owner` and `project` tags in event notifications, then you can choose to include only those tags.

After you select the tags to include, the event notifications will include the resource ID (instance ID or Dedicated Host ID) and the tag key and value pairs that are associated with the affected resource.

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.

You can include tags in event notifications by using one of the following methods.

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](#) AWS CLI command and set the `IncludeAllTagsOfInstance` parameter to `true`.

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute "IncludeAllTagsOfInstance=true"
```

To include specific tags in event notifications

Use the [register-instance-event-notification-attributes](#) AWS CLI command and specify the tags to include by using the `InstanceTagKeys` parameter.

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute 'InstanceTagKeys=["tag_key_1", "tag_key_2",  
  "tag_key_3"]'
```

Remove tags from event notifications

You can remove tags from event notifications by using one of the following methods.

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](#) AWS CLI command and set the `IncludeAllTagsOfInstance` parameter to `false`.

```
aws ec2 deregister-instance-event-notification-attributes \  
  --instance-tag-attribute "IncludeAllTagsOfInstance=false"
```

To remove specific tags from event notifications

Use the [deregister-instance-event-notification-attributes](#) AWS CLI command and specify the tags to remove by using the `InstanceTagKeys` parameter.

```
aws ec2 deregister-instance-event-notification-attributes \  
  --instance-tag-attribute 'InstanceTagKeys=["tag_key_1", "tag_key_2",  
  "tag_key_3"]'
```


View the tags to be included in event notifications

You can view the tags that are to be included in event notifications by using one of the following methods.

Console

To view the tags that are to be included in event notifications

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Actions, Manage event notifications**.

AWS CLI

To view the tags that are to be included in event notifications

Use the [describe-instance-event-notification-attributes](#) AWS CLI command.

```
aws ec2 describe-instance-event-notification-attributes
```

Work with instances scheduled to stop or retire

When AWS detects irreparable failure of the underlying host for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate. For more information, see [Instance retirement](#).

Important

Any data stored on instance store volumes is lost when an instance is stopped, hibernated, or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. Be sure to save data from your instance store volumes that you might need later before the instance is stopped, hibernated, or terminated.

Actions for Instances Backed by Amazon EBS

You can wait for the instance to stop as scheduled. Alternatively, you can stop and start the instance yourself, which migrates it to a new host. For more information about stopping your instance, in addition to information about the changes to your instance configuration when it's stopped, see [Stop and start Amazon EC2 instances](#).

You can automate an immediate stop and start in response to a scheduled instance stop event. For more information, see [Automating actions for Amazon EC2 instances](#) in the *AWS Health User Guide*.

Actions for Instances Backed by Instance Store

We recommend that you launch a replacement instance from your most recent AMI and migrate all necessary data to the replacement instance before the instance is scheduled to terminate. Then, you can terminate the original instance, or wait for it to terminate as scheduled.

Work with instances scheduled for reboot

When AWS must perform tasks such as installing updates or maintaining the underlying host, it can schedule the instance or the underlying host for a reboot. You can [reschedule most reboot events](#) so that your instance is rebooted at a specific date and time that suits you.

View the reboot event type

You can view whether a reboot event is an instance reboot or a system reboot by using one of the following methods.

Console

To view the type of scheduled reboot 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. For each instance, view the value in the **Event type** column. The value is either **system-reboot** or **instance-reboot**.

AWS CLI

To view the type of scheduled reboot event

Use the [describe-instance-status](#) command.

```
aws ec2 describe-instance-status \  
  --instance-id i-1234567890abcdef0
```

For scheduled reboot events, the value for Code is either `system-reboot` or `instance-reboot`. The following example output shows a `system-reboot` event.

```
[  
  "Events": [  
    {  
      "InstanceEventId": "instance-event-0d59937288b749b32",  
      "Code": "system-reboot",  
      "Description": "The instance is scheduled for a reboot",  
      "NotAfter": "2019-03-14T22:00:00.000Z",  
      "NotBefore": "2019-03-14T20:00:00.000Z",  
      "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"  
    }  
  ]  
]
```

Actions for instance reboot

You can wait for the instance reboot to occur within its scheduled maintenance window, [reschedule](#) the instance reboot to a date and time that suits you, or [reboot](#) the instance yourself at a time that is convenient for you.

After your instance is rebooted, the scheduled event is cleared and the event's description is updated. The pending maintenance to the underlying host is completed, and you can begin using your instance again after it has fully booted.

Actions for system reboot

It is not possible for you to reboot the system yourself. You can wait for the system reboot to occur during its scheduled maintenance window, or you can [reschedule](#) the system reboot to a date and time that suits you. A system reboot typically completes in a matter of minutes. After the system reboot has occurred, the instance retains its IP address and DNS name, and any data on local instance store volumes is preserved. After the system reboot is complete, the scheduled event for the instance is cleared, and you can verify that the software on your instance is operating as expected.

Alternatively, if it is necessary to maintain the instance at a different time and you can't reschedule the system reboot, then you can stop and start an Amazon EBS-backed instance, which migrates it to a new host. However, the data on the local instance store volumes is not preserved. You can also automate an immediate instance stop and start in response to a scheduled system reboot event. For more information, see [Automating Actions for EC2 Instances](#) in the *AWS Health User Guide*. For an instance store-backed instance, if you can't reschedule the system reboot, then you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Work with instances scheduled for maintenance

When AWS must maintain the underlying host for an instance, it schedules the instance for maintenance. There are two types of maintenance events: network maintenance and power maintenance.

During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance is restored after maintenance is complete.

During power maintenance, scheduled instances are taken offline for a brief period, and then rebooted. When a reboot is performed, all of your instance's configuration settings are retained.

After your instance has rebooted (this normally takes a few minutes), verify that your application is working as expected. At this point, your instance should no longer have a scheduled event associated with it, or if it does, the description of the scheduled event begins with **[Completed]**. It sometimes takes up to 1 hour for the instance status description to refresh. Completed maintenance events are displayed on the Amazon EC2 console dashboard for up to a week.

Actions for Instances Backed by Amazon EBS

You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host. For more information about stopping your instance, in addition to information about the changes to your instance configuration when it's stopped, see [Stop and start Amazon EC2 instances](#).

You can automate an immediate stop and start in response to a scheduled maintenance event. For more information, see [Automating Actions for EC2 Instances](#) in the *AWS Health User Guide*.

Actions for instances backed by instance store

You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance

from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Reschedule a scheduled event

You can reschedule an event so that it occurs at a specific date and time that suits you. Only events that have a deadline date can be rescheduled. There are other [limitations for rescheduling an event](#).

You can reschedule an event by using one of the following methods.

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

1. Only events that have an event deadline date, indicated by a value for `NotBeforeDeadline`, can be rescheduled. Use the [describe-instance-status](#) command to view the `NotBeforeDeadline` parameter value.

```
aws ec2 describe-instance-status \
```

```
--instance-id i-1234567890abcdef0
```

The following example output shows a system-reboot event that can be rescheduled because `NotBeforeDeadline` contains a value.

```
[
  "Events": [
    {
      "InstanceEventId": "instance-event-0d59937288b749b32",
      "Code": "system-reboot",
      "Description": "The instance is scheduled for a reboot",
      "NotAfter": "2019-03-14T22:00:00.000Z",
      "NotBefore": "2019-03-14T20:00:00.000Z",
      "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
    }
  ]
]
```

- To reschedule the event, use the [modify-instance-event-start-time](#) command. Specify the new event start time by using the `not-before` parameter. The new event start time must fall before the `NotBeforeDeadline`.

```
aws ec2 modify-instance-event-start-time \
  --instance-id i-1234567890abcdef0 \
  --instance-event-id instance-event-0d59937288b749b32 \
  --not-before 2019-03-25T10:00:00.000
```

It might take a minute or two before the [describe-instance-status](#) command returns the updated `not-before` parameter value.

Limitations

- Only events with an event deadline date can be rescheduled. The event can be rescheduled up to the event deadline date. The **Deadline** column in the console and the `NotBeforeDeadline` field in the AWS CLI indicate if the event has a deadline date.
- Only events that have not yet started can be rescheduled. The **Start time** column in the console and the `NotBefore` field in the AWS CLI indicate the event start time. Events that are scheduled to start in the next 5 minutes cannot be rescheduled.
- The new event start time must be at least 60 minutes from the current time.

- If you reschedule multiple events using the console, the event deadline date is determined by the event with the earliest event deadline date.

Define event windows for scheduled events

You can define custom event windows that recur weekly for scheduled events that reboot, stop, or terminate your Amazon EC2 instances. You can associate one or more instances with an event window. If a scheduled event for those instances is planned, AWS will schedule the events within the associated event window.

You can use event windows to maximize workload availability by specifying event windows that occur during off-peak periods for your workload. You can also align the event windows with your internal maintenance schedules.

You define an event window by specifying a set of time ranges. The minimum time range is 2 hours. The combined time ranges must total at least 4 hours.

You can associate one or more instances with an event window by using either instance IDs or instance tags. You can also associate Dedicated Hosts with an event window by using the host ID.

Warning

Event windows are applicable only for scheduled events that stop, reboot, or terminate instances.

Event windows are *not* applicable for:

- Expedited scheduled events and network maintenance events.
- Unscheduled maintenance such as AutoRecovery and unplanned reboots.

Work with event windows

- [Considerations](#)
- [View event windows](#)
- [Create event windows](#)
- [Modify event windows](#)
- [Delete event windows](#)
- [Tag event windows](#)

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 prior to the scheduled event, which migrates the instance to a new host, and the scheduled event will no longer take place.

View event windows

You can view event windows by using one of the following methods.

Console

To view event windows

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Actions, Manage event windows**.
4. Select an event window to view its details.

AWS CLI

To describe all event windows

Use the [describe-instance-event-windows](#) command.

```
aws ec2 describe-instance-event-windows \  
  --region us-east-1
```

Expected output

```
{  
  "InstanceEventWindows": [  
    {  
      "InstanceEventWindowId": "iew-0abcdef1234567890",  
      "Name": "myEventWindowName",  
      "CronExpression": "* 21-23 * * 2,3",  
      "AssociationTarget": {  
        "InstanceIds": [  
          "i-1234567890abcdef0",  
          "i-0598c7d356eba48d7"  
        ],  
        "Tags": [],  
        "DedicatedHostIds": []  
      },  
      "State": "active",  
      "Tags": []  
    }  
    ...  
  ],  
  "NextToken": "9d624e0c-388b-4862-a31e-a85c64fc1d4a"  
}
```

To describe a specific event window

Use the [describe-instance-event-windows](#) command with the `--instance-event-window-id` parameter to describe a specific event window.

```
aws ec2 describe-instance-event-windows \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890
```

To describe event windows that match one or more filters

Use the [describe-instance-event-windows](#) command with the `--filters` parameter. In the following example, the `instance-id` filter is used to describe all of the event windows that are associated with the specified instance.

When a filter is used, it performs a direct match. However, the `instance-id` filter is different. If there is no direct match to the instance ID, then it falls back to indirect associations with the event window, such as the instance's tags or Dedicated Host ID (if the instance is on a Dedicated Host).

For the list of supported filters, see [describe-instance-event-windows](#) in the *AWS CLI Reference*.

```
aws ec2 describe-instance-event-windows \
  --region us-east-1 \
  --filters Name=instance-id,Values=i-1234567890abcdef0 \
  --max-results 100 \
  --next-token <next-token-value>
```

Expected output

In the following example, the instance is on a Dedicated Host, which is associated with the event window.

```
{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0dbc0adb66f235982",
      "TimeRanges": [
        {
          "StartWeekDay": "sunday",
          "StartHour": 2,
          "EndWeekDay": "sunday",
          "EndHour": 8
        }
      ],
      "Name": "myEventWindowName",
      "AssociationTarget": {
        "InstanceIds": [],
        "Tags": [],
        "DedicatedHostIds": [
          "h-0140d9a7ecbd102dd"
        ]
      }
    }
  ],
}
```

```
        "State": "active",
        "Tags": []
    }
]
}
```

Create event windows

You can create one or more event windows. For each event window, you specify one or more blocks of time. For example, you can create an event window with blocks of time that occur every day at 4 AM for 2 hours. Or you can create an event window with blocks of time that occur on Sundays from 2 AM to 4 AM and on Wednesdays from 3 AM to 5 AM.

For the event window constraints, see [Considerations](#) earlier in this topic.

Event windows recur weekly until you delete them.

Use one of the following methods to create an event window.

Console

To create an event window

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 so that if the instances are scheduled for maintenance, the scheduled event will occur during the associated event window. You can associate one or more instances with an event window by using instance IDs or instance tags. You can associate Dedicated Hosts with an event window by using the host ID.

Note that you can create the event window without associating a target with the window. Later, you can modify the window to associate one or more targets.

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 using the AWS CLI, you first create the event window, and then you associate one or more targets with the event window.

Create an event window

You can define either a set of time ranges or a cron expression when creating the event window, but not both.

To create an event window with a time range

Use the [create-instance-event-window](#) command and specify the `--time-range` parameter. You can't also specify the `--cron-expression` parameter.

```
aws ec2 create-instance-event-window \
  --region us-east-1 \
  --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
  --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName
```

Expected output

```
{
  "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 and specify the `--cron-expression` parameter. You can't also specify the `--time-range` parameter.

```
aws ec2 create-instance-event-window \
  --region us-east-1 \
  --cron-expression "* 21-23 * * 2,3" \
  --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName
```

Expected output

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "State": "creating",
```

```

    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}

```

Associate a target with an event window

You can associate only one type of target (instance IDs, Dedicated Host IDs, or instance tags) with an event window.

To associate instance tags with an event window

Use the [associate-instance-event-window](#) command and specify the `instance-event-window-id` parameter to specify the event window. To associate instance tags, specify the `--association-target` parameter, and for the parameter values, specify one or more tags.

```

aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"

```

Expected output

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [
        {
          "Key": "k2",
          "Value": "v2"
        },
        {
          "Key": "k1",
          "Value": "v1"
        }
      ]
    }
  }
}

```

```

    ],
    "DedicatedHostIds": []
  },
  "State": "creating"
}
}

```

To associate one or more instances with an event window

Use the [associate-instance-event-window](#) command and specify the `instance-event-window-id` parameter to specify the event window. To associate instances, specify the `--association-target` parameter, and for the parameter values, specify one or more instance IDs.

```

aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"

```

Expected output

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-0598c7d356eba48d7"
      ],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

To associate a Dedicated Host with an event window

Use the [associate-instance-event-window](#) command and specify the `instance-event-window-id` parameter to specify the event window. To associate a Dedicated Host, specify

the `--association-target` parameter, and for the parameter values, specify one or more Dedicated Host IDs.

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

Expected output

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [],  
      "DedicatedHostIds": [  
        "h-029fa35a02b99801d"  
      ]  
    },  
    "State": "creating"  
  }  
}
```

Modify event windows

You can modify all of the fields of an event window except its ID. For example, when daylight savings begin, you might want to modify the event window schedule. For existing event windows, you might want to add or remove targets.

Use one of the following methods to modify an event window.

Console

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 an event window using the AWS CLI, you can modify the time range or cron expression, and associate or disassociate one or more targets with the event window.

Modify the event window time

You can modify either a time range or a cron expression when modifying the event window, but not both.

To modify the time range of an event window

Use the [modify-instance-event-window](#) command and specify the event window to modify. Specify the `--time-range` parameter to modify the time range. You can't also specify the `--cron-expression` parameter.

```
aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

Expected 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 and specify the event window to modify. Specify the `--time-range` parameter to modify the time range. You can't also specify the `--cron-expression` parameter in the same call.

```

aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
"wednesday", "EndHour": 8},
{"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
"EndHour": 8}]'

```

Expected output

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "TimeRanges": [
      {
        "StartWeekDay": "monday",
        "StartHour": 2,
        "EndWeekDay": "wednesday",
        "EndHour": 8
      }
    ]
  }
}

```

```

    },
    {
      "StartWeekDay": "thursday",
      "StartHour": 2,
      "EndWeekDay": "friday",
      "EndHour": 8
    }
  ],
  "Name": "myEventWindowName",
  "AssociationTarget": {
    "InstanceIds": [
      "i-0abcdef1234567890",
      "i-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 and specify the event window to modify. Specify the `--cron-expression` parameter to modify the cron expression. You can't also specify the `--time-range` parameter.

```

aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --cron-expression "* 21-23 * * 2,3"

```

Expected 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"
      }
    ]
  }
}

```

Modify the targets associated with an event window

You can associate additional targets with an event window. You can also disassociate existing targets from an event window. However, only one type of target (instance IDs, Dedicated Host IDs, or instance tags) can be associated with an event window.

To associate additional targets with an event window

For the instructions on how to associate targets with an event window, see [Associate a target with an event window](#).

To disassociate instance tags from an event window

Use the [disassociate-instance-event-window](#) command and specify the `instance-event-window-id` parameter to specify the event window. To disassociate instance tags, specify the `--association-target` parameter, and for the parameter values, specify one or more tags.

```

aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"

```

Expected output

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

To disassociate one or more instances from an event window

Use the [disassociate-instance-event-window](#) command and specify the `instance-event-window-id` parameter to specify the event window. To disassociate instances, specify the `--association-target` parameter, and for the parameter values, specify one or more instance IDs.

```
aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Expected output

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

```
}
```

To disassociate a Dedicated Host from an event window

Use the [disassociate-instance-event-window](#) command and specify the `instance-event-window-id` parameter to specify the event window. To disassociate a Dedicated Host, specify the `--association-target` parameter, and for the parameter values, specify one or more Dedicated Host IDs.

```
aws ec2 disassociate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target DedicatedHostIds=h-029fa35a02b99801d
```

Expected output

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

Delete event windows

You can delete one event window at a time by using one of the following methods.

Console

To delete an event window

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 \  
  --region us-east-1 \  
  --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 \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --force-delete
```

Expected output

```
{  
  "InstanceEventWindowState": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "State": "deleting"  
  }  
}
```

Tag event windows

You can tag an event window when you create it, or afterwards.

To tag an event window when you create it, see [Create event windows](#).

Use one of the following methods to tag an event window.

Console

To tag an existing 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 tag, and then choose **Actions, Manage instance event window tags**.
5. Choose **Add tag** to add a tag. Repeat for each tag.
6. Choose **Save**.

AWS CLI

To tag an existing event window

Use the [create-tags](#) command to tag existing resources. In the following example, the existing event window is tagged with Key=purpose and Value=test.

```
aws ec2 create-tags \  
  --resources iew-0abcdef1234567890 \  
  --tags Key=purpose,Value=test
```

Monitor your instances using CloudWatch

You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance. For more information, see [Enable or turn off detailed monitoring for your instances](#).

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

- [Amazon EC2 instance alarms](#)
- [Enable or turn off detailed monitoring for your instances](#)
- [List the available CloudWatch metrics for your instances](#)
- [Install and configure the CloudWatch agent using the Amazon EC2 console to add additional metrics](#)
- [Get statistics for metrics for your instances](#)
- [Graph metrics for your instances](#)
- [Create a CloudWatch alarm for an instance](#)
- [Create alarms that stop, terminate, reboot, or recover an instance](#)

Amazon EC2 instance alarms

You can view and create Amazon CloudWatch alarms for your instances from the **Instances** screen in the Amazon EC2 console.

The following screenshot indicates the console controls, numbered **1** and **2**, for viewing and creating alarms from the **Instances** screen.

<input type="checkbox"/>	Name ↗	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status
<input type="checkbox"/>	My-1-Spot-Ins...	I-01aeed690c9fb5322	✔ Running ⊗ 🔍	t3.nano	✔ 2/2 checks passed	View alarms 1 +
<input type="checkbox"/>	My-2-Spot-Ins...	I-0ba5e5bbc9d634fa6	⊖ Stopped ⊗ 🔍	t3.nano	-	View alarms 2 +

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-0123456789example*** 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](#).

Enable or turn off detailed monitoring for your instances

By default, your instance is enabled for **basic monitoring**. You can optionally enable **detailed monitoring**.

The following table highlights the differences between basic monitoring and detailed monitoring for instances.

Monitoring type	Description	Charges
Basic monitoring	Only status check metrics are available in 1-minute periods. All other metrics are available in 5-minute periods.	No charge.
Detailed monitoring	All metrics, including status check metrics, are available in 1-minute periods. To get this level of data, you	You are charged per metric that is sent to CloudWatch. You are not charged for data storage. For

Monitoring type	Description	Charges
	must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances.	more information, see Paid tier and Example 1 - EC2 Detailed Monitoring on the Amazon CloudWatch pricing page .

Topics

- [Required IAM permissions](#)
- [Enable detailed monitoring](#)
- [Turn off detailed monitoring](#)

Required IAM 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

You can enable detailed monitoring on an instance as you launch it or after the instance is running or stopped. Enabling detailed monitoring on an instance does not affect the monitoring of the EBS volumes attached to the instance. For more information, see [Amazon CloudWatch metrics for Amazon EBS](#).

Console

To enable detailed monitoring for an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Monitor and troubleshoot, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, select the **Enable** check box.

5. Choose **Save**.

To enable detailed monitoring when launching an instance

When launching an instance using the Amazon EC2 console, under **Advanced details**, select the **Detailed CloudWatch monitoring** check box.

AWS CLI

To enable detailed monitoring for an existing instance

Use the following [monitor-instances](#) command to enable detailed monitoring for the specified instances.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

To enable detailed monitoring when launching an instance

Use the [run-instances](#) command with the `--monitoring` flag to enable detailed monitoring.

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

Turn off detailed monitoring

You can turn off detailed monitoring on an instance as you launch it or after the instance is running or stopped.

Console

To turn off 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 and choose **Actions, Monitor and troubleshoot, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, clear the **Enable** check box.

5. Choose **Save**.

AWS CLI

To turn off detailed monitoring

Use the following [unmonitor-instances](#) command to turn off detailed monitoring for the specified instances.

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

List the available CloudWatch metrics for your instances

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch. By default, each data point covers the 5 minutes that follow the start time of activity for the instance. If you've enabled detailed monitoring, each data point covers the next minute of activity from the start time. Note that for the Minimum, Maximum, and Average statistics, the minimum granularity for the metrics that EC2 provides is 1 minute.

For information about getting the statistics for these metrics, see [Get statistics for metrics for your instances](#).

Contents

- [Instance metrics](#)
- [CPU credit metrics](#)
- [Dedicated Host metrics](#)
- [Amazon EBS metrics for Nitro-based instances](#)
- [Status check metrics](#)
- [Traffic mirroring metrics](#)
- [Auto Scaling group metrics](#)
- [Amazon EC2 metric dimensions](#)
- [Amazon EC2 usage metrics](#)
- [List metrics using the console](#)
- [List metrics using the AWS CLI](#)

Instance metrics

The AWS/EC2 namespace includes the following instance metrics.

Metric	Description	Unit	Meaningful statistics
CPUUtilization	<p>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.</p> <p>At a very high level, CPUUtilization is the sum of guest CPUUtilization and hypervisor CPUUtilization .</p> <p>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.</p>	Percent	<ul style="list-style-type: none"> • Average • Minimum • Maximum
DiskReadOps	<p>Completed read operations from all instance store volumes available to the instance in a specified period of time.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p>	Count	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum
DiskWriteOps	<p>Completed write operations to all instance store volumes available to the instance in a specified period of time.</p>	Count	<ul style="list-style-type: none"> • Sum • Average • Minimum

Metric	Description	Unit	Meaningful statistics
	<p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p>		<ul style="list-style-type: none"> • Maximum
DiskReadBytes	<p>Bytes read from all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (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 <code>DIFF_TIME</code> to find the bytes per second. For example, if you have graphed <code>DiskReadBytes</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in bytes/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p>	Bytes	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
DiskWrite Bytes	<p>Bytes written to all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (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 <code>DIFF_TIME</code> to find the bytes per second. For example, if you have graphed <code>DiskWrite Bytes</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in bytes/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p>	Bytes	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
MetadataNoToken	<p>The number of times the Instance Metadata Service (IMDS) was successfully accessed using a method that does not use a token.</p> <p>This metric is used to determine if there are any processes accessing instance metadata that are using Instance Metadata Service Version 1 (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.</p>	Count	<ul style="list-style-type: none"> Sum Percentiles
MetadataNoTokenRejected	<p>The number of times an IMDSv1 call was attempted after IMDSv1 was disabled.</p> <p>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.</p>	Count	<ul style="list-style-type: none"> Sum Percentiles

Metric	Description	Unit	Meaningful statistics
NetworkIn	<p>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.</p> <p>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. You can also use the CloudWatch metric math function <code>DIFF_TIME</code> to find the bytes per second. For example, if you have graphed <code>NetworkIn</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in bytes/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Bytes	<ul style="list-style-type: none">• Sum• Average• Minimum• Maximum

Metric	Description	Unit	Meaningful statistics
NetworkOut	<p>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.</p> <p>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. You can also use the CloudWatch metric math function <code>DIFF_TIME</code> to find the bytes per second. For example, if you have graphed <code>NetworkOut</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in bytes/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Bytes	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
NetworkPacketsIn	<p>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.</p> <p>This metric is available for basic monitoring 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. You can also use the CloudWatch metric math function <code>DIFF_TIME</code> to find the packets per second. For example, if you have graphed <code>NetworkPacketsIn</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in packets/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Count	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
NetworkPacketsOut	<p>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.</p> <p>This metric is available for basic monitoring 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. You can also use the CloudWatch metric math function <code>DIFF_TIME</code> to find the packets per second. For example, if you have graphed <code>NetworkPacketsOut</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in packets/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Count	<ul style="list-style-type: none"> 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	Credits (vCPU-minutes)	<ul style="list-style-type: none"> Sum Average Minimum Maximum

Metric	Description	Unit	Meaningful statistics
	<p>of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes).</p> <p>CPU credit metrics are available at a 5-minute frequency only. If you specify a period greater than five minutes, use the Sum statistic instead of the Average statistic.</p>		

Metric	Description	Unit	Meaningful statistics
<p>CPUCreditBalance</p>	<p>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the CPUCreditBalance also includes the number of launch credits that have been accrued.</p> <p>Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits do not count towards the limit.</p> <p>The credits in the CPUCreditBalance are available for the instance to spend to burst beyond its baseline CPU utilization.</p> <p>When an instance is running, credits in the CPUCreditBalance do not expire. When a T3 or T3a instance stops, the CPUCreditBalance value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the CPUCreditBalance value does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a 5-minute frequency only.</p>	<p>Credits (vCPU-minutes)</p>	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
CPUSurplusCreditBalance	<p>The number of surplus credits that have been spent by an <code>unlimited</code> instance when its <code>CPUCreditBalance</code> value is zero.</p> <p>The <code>CPUSurplusCreditBalance</code> value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge.</p> <p>CPU credit metrics are available at a 5-minute frequency only.</p>	Credits (vCPU-minutes)	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum
CPUSurplusCreditsCharged	<p>The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.</p> <p>Spent surplus credits are charged when any of the following occurs:</p> <ul style="list-style-type: none"> • The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. • The instance is stopped or terminated. • The instance is switched from <code>unlimited</code> to <code>standard</code>. <p>CPU credit metrics are available at a 5-minute frequency only.</p>	Credits (vCPU-minutes)	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> 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	<p>Completed read operations from all Amazon EBS volumes attached to the instance in a specified period of time.</p> <p>To calculate the average read I/O operations per second (Read IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (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 <code>DIFF_TIME</code> to find the operations per second. For example, if you have graphed <code>EBSReadOps</code> in CloudWatch as <code>m1</code>, the</p>	Count	<ul style="list-style-type: none"> Sum Average Minimum Maximum

Metric	Description	Unit	Meaningful statistics
	<p>metric math formula $m1/(DIFF_TIME(m1))$ returns the metric in operations/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>		
EBSWriteOps	<p>Completed write operations to all EBS volumes attached to the instance in a specified period of time.</p> <p>To calculate the average write I/O 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 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 <code>DIFF_TIME</code> to find the operations per second. For example, if you have graphed <code>EBSWriteOps</code> in CloudWatch as <code>m1</code>, the metric math formula $m1/(DIFF_TIME(m1))$ returns the metric in operations/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Count	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
EBSReadBytes	<p>Bytes read from all EBS volumes attached to the instance in a specified period of time.</p> <p>The number reported is the number of bytes read during the period. If you are using basic (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 <code>DIFF_TIME</code> to find the bytes per second. For example, if you have graphed <code>EBSReadBytes</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in bytes/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Bytes	<ul style="list-style-type: none">• Sum• Average• Minimum• Maximum

Metric	Description	Unit	Meaningful statistics
EBSWriteBytes	<p>Bytes written to all EBS volumes attached to the instance in a specified period of time.</p> <p>The number reported is the number of bytes written during the period. If you are using basic (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 <code>DIFF_TIME</code> to find the bytes per second. For example, if you have graphed <code>EBSWriteBytes</code> in CloudWatch as <code>m1</code>, the metric math formula <code>m1/(DIFF_TIME(m1))</code> returns the metric in bytes/second. For more information about <code>DIFF_TIME</code> and other metric math functions, see Use metric math in the <i>Amazon CloudWatch User Guide</i>.</p>	Bytes	<ul style="list-style-type: none"> • Sum • Average • Minimum • Maximum
EBSIOBalance%	<p>Provides information about the percentage of I/O credits remaining in the burst bucket. This metric is available for basic monitoring only.</p> <p>This metric is available only for some <code>*.4xlarge</code> instance sizes and smaller that burst to their maximum performance for only 30 minutes at least once every 24 hours.</p> <p>The Sum statistic is not applicable to this metric.</p>	Percent	<ul style="list-style-type: none"> • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
EBSByteBalance%	<p>Provides information about the percentage of throughput credits remaining in the burst bucket. This metric is available for basic monitoring only.</p> <p>This metric is available only for some <code>*.4xlarge</code> instance sizes and smaller that burst to their maximum performance for only 30 minutes at least once every 24 hours.</p> <p>The Sum statistic is not applicable to this metric.</p>	Percent	<ul style="list-style-type: none"> 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 Spot fleets, see [CloudWatch metrics for Spot Fleet](#).

Status check metrics

By default, status check metrics are available at a 1-minute frequency at no charge. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state (within a few minutes of the instance entering the `running` state). For more information about EC2 status checks, see [Status checks for your instances](#).

The AWS/EC2 namespace includes the following status check metrics.

Metric	Description	Unit	Meaningful statistics
StatusCheckFailed	<p>Reports whether the instance has passed both the instance status check and the system status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p>	Count	<ul style="list-style-type: none"> Sum Average

Metric	Description	Unit	Meaningful statistics
	By default, this metric is available at a 1-minute frequency at no charge.		
StatusCheckFailed_Instance	<p>Reports whether the instance has passed the instance status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p> <p>By default, this metric is available at a 1-minute frequency at no charge.</p>	Count	<ul style="list-style-type: none"> • Sum • Average
StatusCheckFailed_System	<p>Reports whether the instance has passed the system status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p> <p>By default, this metric is available at a 1-minute frequency at no charge.</p>	Count	<ul style="list-style-type: none"> • Sum • Average
StatusCheckFailed_AttachedEBS	<p>Reports whether the instance has passed the attached EBS status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p> <p>By default, this metric is available at a 1-minute frequency at no charge.</p>	Count	<ul style="list-style-type: none"> • Sum • Average

The AWS/EBS namespace includes the following status check metric.

Metric	Description	Unit	Meaningful statistics
VolumeStalledIOCheck	<p>Note: For Nitro instances only. Not published for volumes attached to Amazon ECS and AWS Fargate tasks.</p> <p>Reports whether a volume has passed or failed a <i>stalled IO check</i> in the last minute. This metric can be either 0 (passed) or 1 (failed).</p>	Count	<ul style="list-style-type: none"> Sum 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 <i>Auto Scaling group</i> is a collection of instances you define if you're using Auto Scaling. This dimension is available only for Amazon EC2 metrics when the instances are in such an Auto Scaling group. Available for instances with Detailed or Basic Monitoring enabled.
ImageId	This dimension filters the data you request for all instances running this Amazon EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.

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	<p>The number of the specified resources running in your account. The resources are defined by the dimensions associated with the metric.</p> <p>The most useful statistic for this metric is MAXIMUM, which represents the maximum number of resources used during the 1-minute period.</p>

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	<p>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 .</p> <p>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.</p>

List metrics using the console

Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all metrics provided by Amazon EC2, or metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling group.

To view available metrics by category (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, expand **Metrics**, and then choose **All metrics**.
3. Choose the **EC2** metric namespace.

Metrics (1,153) Info

Alarm recommendations Download alarm code Create alarm

Ireland Search iGraph

Backup	16	Directory Service	62	EBS	47
EC2	93	EC2/API	152	EC2 Capacity Reservations	8
EC2 Spot	618	EFS	36	Events	1
Logs	3	NATGateway	15	S3	12
SSM Run Command	3	Usage	87		

4. Select a metric dimension (for example, **Per-Instance Metrics**).

Metrics (93) Info

Alarm recommendations Download alarm code (14) Create alarm

Ireland All > EC2

HostId	1	Per-Instance Metrics	92
--------	---	----------------------	----

5. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric. To filter by resource, choose the resource ID and then choose **Add to search**. To filter by metric, choose the metric name and then choose **Add to search**.

The screenshot shows the AWS CloudWatch console interface. At the top, there are navigation tabs: 'Browse', 'Multi source query', 'Graphed metrics', 'Options', and 'Source'. There are also buttons for 'Add math' and 'Add query'. Below this, the 'Metrics (92) Info' section is visible, with a search bar and several action buttons like 'Download alarm code (14)', 'Create alarm', 'Graph with SQL', and 'Graph search'. The main area displays a table of metrics for EC2 instances in the 'Ireland' region. The table has columns for 'Instance name', 'Instanceid', 'Metric name', and 'Alarms'. A context menu is open over the 'fingerprint' metric for instance 'i-04747028607e63eaa', showing options such as 'Add to search', 'Exclude from search', 'Search for this only', 'Add to graph', 'Graph this metric only', 'Graph all search results', 'Graph with SQL query', 'View in Resource Health', and 'View in EC2 console'. The 'Alarms' column for all metrics shows 'No alarms'.

List metrics using the AWS CLI

Use the [list-metrics](#) command to list the CloudWatch metrics for your instances.

To list all the available metrics for Amazon EC2 (AWS CLI)

The following example specifies the AWS/EC2 namespace to view all the metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:

```
{
  "Metrics": [
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-1234567890abcdef0"
        }
      ]
    }
  ]
}
```

```

    ],
    "MetricName": "NetworkOut"
  },
  {
    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-1234567890abcdef0"
      }
    ],
    "MetricName": "CPUUtilization"
  },
  {
    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-1234567890abcdef0"
      }
    ],
    "MetricName": "NetworkIn"
  },
  ...
]
}

```

To list all the available metrics for an instance (AWS CLI)

The following example specifies the AWS/EC2 namespace and the InstanceId dimension to view the results for the specified instance only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
Name=InstanceId,Value=i-1234567890abcdef0
```

To list a metric across all instances (AWS CLI)

The following example specifies the AWS/EC2 namespace and a metric name to view the results for the specified metric only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

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](#)
- [How it works](#)
- [Costs](#)
- [Install and configure the CloudWatch agent](#)

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:

```
{
```

```

"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 [Using 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**.

Get statistics for metrics for your instances

You can get statistics for the CloudWatch metrics for your instances.

Contents

- [Statistics overview](#)
- [Get statistics for a specific instance](#)
- [Aggregate statistics across instances](#)
- [Aggregate statistics by Auto Scaling group](#)
- [Aggregate statistics by AMI](#)

Statistics overview

Statistics are metric data aggregations over specified periods of time. CloudWatch provides statistics based on the metric data points provided by your custom data or provided by other services in AWS to CloudWatch. Aggregations are made using the namespace, metric name, dimensions, and the data point unit of measure, within the time period you specify. The following table describes the available statistics.

Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.

Statistic	Description
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
Sum	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
Average	The value of $\text{Sum} / \text{SampleCount}$ during the specified period. By comparing this statistic with the Minimum and Maximum, you can determine the full scope of a metric and how close the average use is to the Minimum and Maximum. This comparison helps you to know when to increase or decrease your resources as needed.
SampleCount	The count (number) of data points used for the statistical calculation.
pNN.NN	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45).

Get statistics for a specific instance

The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

Requirements

- You must have the ID of the instance. You can get the instance ID using the AWS Management Console or the [describe-instances](#) command.
- By default, basic monitoring is enabled, but you can enable detailed monitoring. For more information, see [Enable or turn off detailed monitoring for your instances](#).

To display the CPU utilization for a specific instance (console)

- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- In the navigation pane, choose **Metrics**.
- Choose the **EC2** metric namespace.

Metrics (1,153) Info

Alarm recommendations Download alarm code Create alarm

Ireland Search iGraph

Backup • View automatic dashboard	16	Directory Service • View automatic dashboard	62	EBS • View automatic dashboard	47
EC2 • View automatic dashboard	93	EC2/API • View automatic dashboard	152	EC2 Capacity Reservations • View automatic dashboard	8
EC2 Spot • View automatic dashboard	618	EFS • View automatic dashboard	36	Events • View automatic dashboard	1
Logs • View automatic dashboard	3	NATGateway • View automatic dashboard	15	S3 • View automatic dashboard	12
SSM Run Command • View automatic dashboard	3	Usage • View automatic dashboard	87		

4. Choose the **Per-Instance Metrics** dimension.

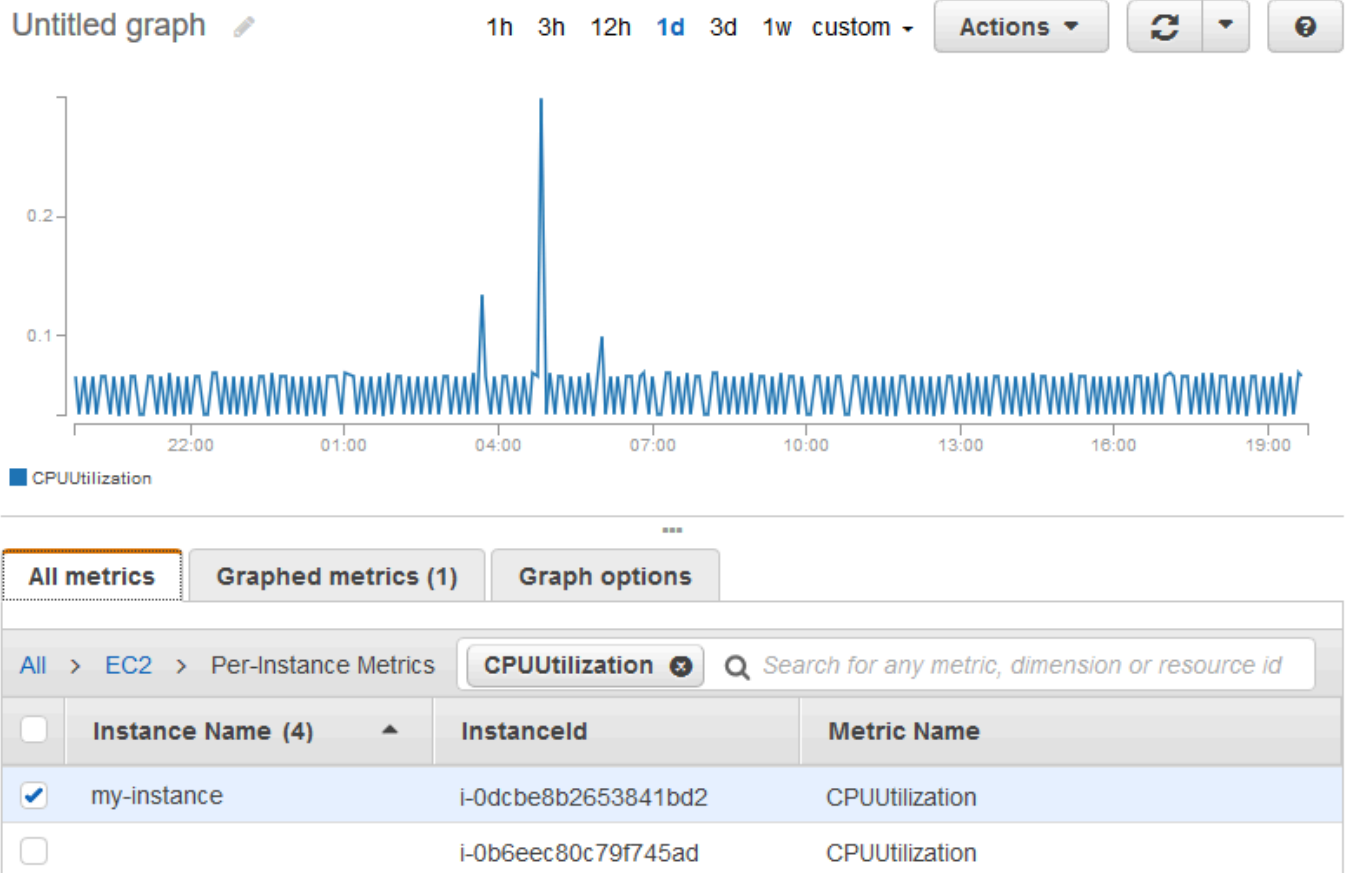
Metrics (93) Info

Alarm recommendations Download alarm code (14) Create alarm

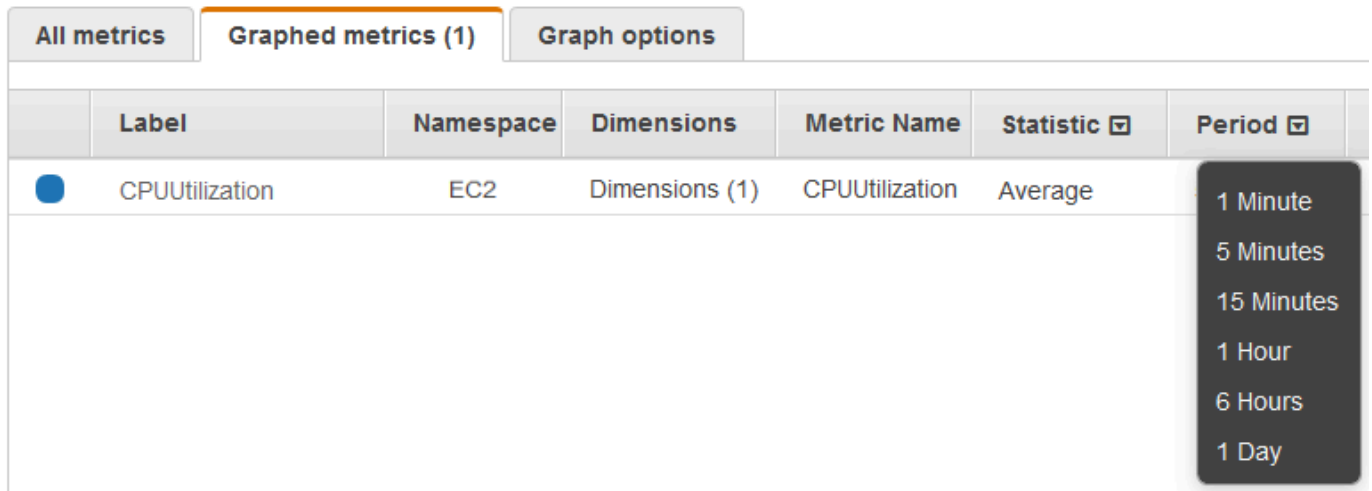
Ireland All > EC2

HostId	1	Per-Instance Metrics	92
---------------	---	-----------------------------	----

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**.



- 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

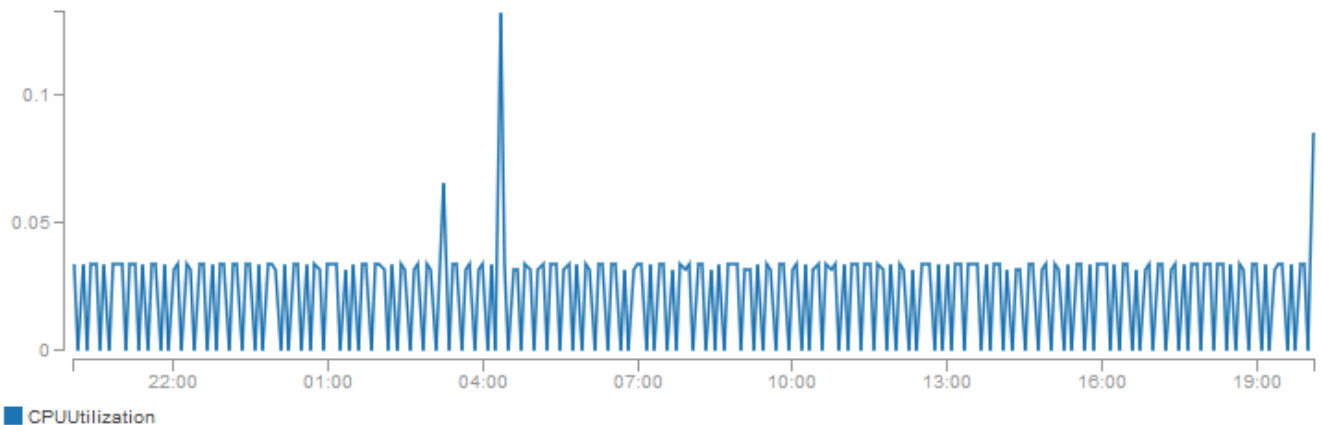
This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the 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**.

Untitled graph 1h 3h 12h **1d** 3d 1w custom ▾

Actions ▾



■ CPUUtilization

All metrics

Graphed metrics (1)

Graph options

All > EC2 > Across All Instances

Q Search for any metric, dimension or resource id

<input type="checkbox"/>	Metric Name (7)
<input checked="" type="checkbox"/>	CPUUtilization
<input type="checkbox"/>	DiskReadBytes

- 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 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"
}
```

Graph metrics for your instances

After you launch an instance, you can open the Amazon EC2 console and view the monitoring graphs for the instance on the **Monitoring** tab. Each graph is based on one of the available Amazon EC2 metrics.

The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

For more information about the metrics and the data they provide to the graphs, see [List the available CloudWatch metrics for your instances](#).

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 [Editing or deleting a CloudWatch alarm](#) in the *Amazon CloudWatch User Guide*.

Create alarms that stop, terminate, reboot, or recover an instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

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.

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 [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Contents

- [Add stop actions to Amazon CloudWatch alarms](#)
- [Add terminate actions to Amazon CloudWatch alarms](#)
- [Add reboot actions to Amazon CloudWatch alarms](#)
- [Add recover actions to Amazon CloudWatch alarms](#)
- [Use the Amazon CloudWatch console to view alarm and action history](#)
- [Amazon CloudWatch alarm action scenarios](#)

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 [Enable 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 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 **5 Minutes**.
 - f. Amazon CloudWatch automatically creates an alarm name for you. To change the name, for **Alarm name**, enter a new name. Alarm names must contain only ASCII characters.
 - g. Choose **Create**.

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 [Recover your instance](#).

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 **5 Minutes**.
- f. Amazon CloudWatch automatically creates an alarm name for you. To change the name, for **Alarm name**, enter a new name. Alarm names must contain only ASCII characters.
- g. Choose **Create**.

Use the Amazon CloudWatch console to view alarm and action history

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions (CloudWatch console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Select an alarm.
4. The **Details** tab shows the most recent state transition along with the time and metric values.
5. Choose the **History** tab to view the most recent history entries.

Amazon CloudWatch alarm action scenarios

You can use the Amazon EC2 console to create alarm actions that stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen capture of the console page where you set the alarm actions, we've numbered the settings. We've also numbered the settings in the scenarios that follow, to help you create the appropriate actions.

New console

Alarm notification [Info](#)

Configure the alarm to send notifications to an Amazon SNS topic when it is triggered.

Alarm action [Info](#)

Specify the action to take when the alarm is triggered.

Alarm thresholds

Specify the metric thresholds for the alarm.

Group samples by	Type of data to sample
<input type="text" value="2 age"/>	<input type="text" value="3"/>
Alarm When	<input type="text" value="5"/>
Consecutive Period	Period
<input type="text" value="6"/>	<input type="text" value="7 nutes"/>

Alarm name

Old console

Create Alarm ✕

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.
To edit an alarm, first choose whom to notify and then define when the notification should be sent.

1 **Send a notification to:** [create topic](#)

Take the action:

- Recover this instance (i)
- Stop this instance (i)
- Terminate this instance (i)
- Reboot this instance (i)

Whenever: **2** of **3**

Is: **4** **5** Percent

For at least: **6** consecutive period(s) of **7**

Name of alarm:

Cancel
Create Alarm

CPU Utilization Percent

Scenario 1: Stop idle development and test instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
1	Stop
2	Maximum
3	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

Setting	Value
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	-

Setting	Value
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
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 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 [EventBridge for Amazon EBS](#).

Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail

Amazon EC2 and Amazon EBS are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon EC2 and Amazon EBS. CloudTrail captures all API calls for Amazon EC2 and Amazon EBS as events, including calls from the console and from code calls to the APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon EC2 and Amazon EBS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon EC2 and Amazon EBS, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Amazon EC2 and Amazon EBS information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon EC2 and Amazon EBS, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing events with CloudTrail Event history](#).

For an ongoing record of events in your AWS account, including events for Amazon EC2 and Amazon EBS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all 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:

- [Creating a trail for your AWS account](#)
- [AWS service integrations with CloudTrail logs](#)

- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple Regions](#) and [Receiving CloudTrail log files from multiple accounts](#)

All Amazon EC2 actions, and Amazon EBS management actions, are logged by CloudTrail and are documented in the [Amazon EC2 API Reference](#). For example, calls to the [RunInstances](#), [DescribeInstances](#), or [CreateImage](#) actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity element](#).

Understand Amazon EC2 and Amazon EBS log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following log file record shows that a user terminated an instance.

```
{
  "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"
}
]
```

Use AWS CloudTrail to audit users that connect via EC2 Instance Connect

Use AWS CloudTrail to audit the users that connect to your instances via EC2 Instance Connect.

To audit SSH activity via EC2 Instance Connect using the AWS CloudTrail console

1. Open the AWS CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Verify that you are in the correct Region.
3. In the navigation pane, choose **Event history**.
4. For **Filter**, choose **Event source, ec2-instance-connect.amazonaws.com**.
5. (Optional) For **Time range**, select a time range.
6. Choose the **Refresh events** icon.
7. The page displays the events that correspond to the [SendSSHPublicKey](#) API calls. Expand an event using the arrow to view additional details, such as the user name and AWS access key that was used to make the SSH connection, and the source IP address.
8. To display the full event information in JSON format, choose **View event**. The **requestParameters** field contains the destination instance ID, OS user name, and public key that were used to make the SSH connection.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGONGNOM00CB6XYTQEXAMPLE",
    "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGUKZHNAW40SN2AEXAMPLE",
    "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 boto3/1.10.60",
  "requestParameters": {
    "instanceId": "i-0123456789EXAMPLE",
    "osUser": "ec2-user",
    "SSHKey": {
```

```
        "publicKey": "ssh-rsa ABCDEFGHIJKLMN001234567890EXAMPLE"  
    }  
  },  
  "responseElements": null,  
  "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",  
  "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "0987654321"  
}
```

If you have configured your AWS account to collect CloudTrail events in an S3 bucket, you can download and audit the information programmatically. For more information, see [Getting and viewing your CloudTrail log files](#) in the *AWS CloudTrail User Guide*.

Networking in Amazon EC2

Amazon VPC enables you to launch AWS resources, such as Amazon EC2 instances, into a virtual network dedicated to your AWS account, known as a virtual private cloud (VPC). When you launch an instance, you can select a subnet from the VPC. The instance is configured with a primary network interface, which is a logical virtual network card. The instance receives a primary private IP address from the IPv4 address of the subnet, and it is assigned to the primary network interface.

You can control whether the instance receives a public IP address from Amazon's pool of public IP addresses. The public IP address of an instance is associated with your instance only until it is stopped or terminated. If you require a persistent public IP address, you can allocate an Elastic IP address for your AWS account and associate it with an instance or a network interface. An Elastic IP address remains associated with your AWS account until you release it, and you can move it from one instance to another as needed. You can bring your own IP address range to your AWS account, where it appears as an address pool, and then allocate Elastic IP addresses from your address pool.

To increase network performance and reduce latency, you can launch instances in a placement group. You can get significantly higher packet per second (PPS) performance using enhanced networking. You can accelerate high performance computing and machine learning applications using an Elastic Fabric Adapter (EFA), which is a network device that you can attach to a supported instance type.

Features

- [Regions and Zones](#)
- [Amazon EC2 instance IP addressing](#)
- [Amazon EC2 instance hostname types](#)
- [Bring your own IP addresses \(BYOIP\) in Amazon EC2](#)
- [Elastic IP addresses](#)
- [Elastic network interfaces](#)
- [Amazon EC2 instance network bandwidth](#)
- [Enhanced networking on Linux](#)
- [Elastic Fabric Adapter](#)
- [Amazon EC2 instance topology](#)
- [Placement groups](#)

- [Network maximum transmission unit \(MTU\) for your EC2 instance](#)
- [Virtual private clouds](#)

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.

- Each Region is a separate geographic area.
- Availability Zones are multiple, isolated locations within each Region.
- Local Zones provide you the ability to place resources, such as compute and storage, in multiple locations closer to your end users.
- AWS Outposts brings native AWS services, infrastructure, and operating models to virtually any data center, co-location space, or on-premises facility.
- Wavelength Zones allow developers to build applications that deliver ultra-low latencies to 5G devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of telecommunication carriers' 5G networks.

AWS operates state-of-the-art, highly available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all of your instances in a single location that is affected by a failure, none of your instances would be available.

To help you determine which deployment is best for you, see [AWS Wavelength FAQs](#).

Contents

- [Regions](#)
- [Availability Zones](#)
- [Local Zones](#)
- [Wavelength Zones](#)
- [AWS Outposts](#)

Regions

Each Region is designed to be isolated from the other Regions. This achieves the greatest possible fault tolerance and stability.

When you view your resources, you see only the resources that are tied to the Region that you specified. This is because Regions are isolated from each other, and we don't automatically replicate resources across Regions.

When you launch an instance, you must select an AMI that's in the same Region. If the AMI is in another Region, you can copy the AMI to the Region you're using. For more information, see [Copy an AMI](#).

Note that there is a charge for data transfer between Regions. For more information, see [Amazon EC2 Pricing - Data Transfer](#).

Contents

- [Available Regions](#)
- [Regions and endpoints](#)
- [Describe your Regions](#)
- [Get the Region display name](#)
- [Specify the Region for a resource](#)

Available Regions

Your account determines the Regions that are available to you.

- An AWS account provides multiple Regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.
- An AWS GovCloud (US-West) account provides access to the AWS GovCloud (US-West) Region and the AWS GovCloud (US-East) Region. For more information, see [AWS GovCloud \(US\)](#).
- An Amazon AWS (China) account provides access to the Beijing and Ningxia Regions only. For more information, see [Amazon Web Services in China](#).

The following table lists the Regions provided by an AWS account. You can't describe or access additional Regions from an AWS account, such as the AWS GovCloud (US) Regions or the China Regions. To use a Region introduced after March 20, 2019, you must enable the Region. For more information, see [Specify which AWS Regions your account can use](#) in the *AWS Account Management Reference Guide*.

Code	Name	Opt-in status
us-east-2	US East (Ohio)	Not required
us-east-1	US East (Virginia)	Not required
us-west-1	US West (N. California)	Not required
us-west-2	US West (Oregon)	Not required
af-south-1	Africa (Cape Town)	Required
ap-east-1	Asia Pacific (Hong Kong)	Required
ap-south-2	Asia Pacific (Hyderabad)	Required
ap-southeast-3	Asia Pacific (Jakarta)	Required
ap-southeast-4	Asia Pacific (Melbourne)	Required
ap-south-1	Asia Pacific (Mumbai)	Not required
ap-northeast-3	Asia Pacific (Osaka)	Not required
ap-northeast-2	Asia Pacific (Seoul)	Not required
ap-southeast-1	Asia Pacific (Singapore)	Not required
ap-southeast-2	Asia Pacific (Sydney)	Not required
ap-northeast-1	Asia Pacific (Tokyo)	Not required
ca-central-1	Canada (Central)	Not required
ca-west-1	Canada West (Calgary)	Required
eu-central-1	Europe (Frankfurt)	Not required
eu-west-1	Europe (Ireland)	Not required
eu-west-2	Europe (London)	Not required

Code	Name	Opt-in status
eu-south-1	Europe (Milan)	Required
eu-west-3	Europe (Paris)	Not required
eu-south-2	Europe (Spain)	Required
eu-north-1	Europe (Stockholm)	Not required
eu-central-2	Europe (Zurich)	Required
il-central-1	Israel (Tel Aviv)	Required
me-south-1	Middle East (Bahrain)	Required
me-central-1	Middle East (UAE)	Required
sa-east-1	South America (São Paulo)	Not required

For more information, see [AWS Global Infrastructure](#).

The number and mapping of Availability Zones per Region may vary between AWS accounts. To list the Availability Zones that are available to your account, you can use the Amazon EC2 console or the command line interface. For more information, see [Describe your Regions](#).

Regions and endpoints

When you work with an instance using the command line interface or API actions, you must specify its Regional endpoint. For more information about the Regions and endpoints for Amazon EC2, see [Amazon EC2 endpoints and quotas](#) in the *Amazon Web Services General Reference*.

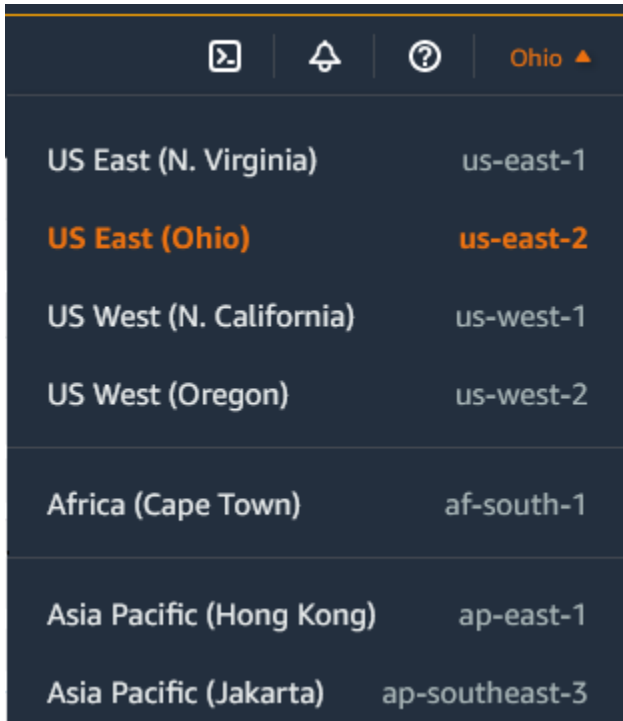
For more information about endpoints and protocols in AWS GovCloud (US-West), see [Service Endpoints](#) in the *AWS GovCloud (US) User Guide*.

Describe your Regions

You can use the Amazon EC2 console or the command line interface to determine which Regions are available for your account. For more information about these command line interfaces, see [Access Amazon EC2](#).

To find your Regions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, choose the **Regions** selector.



3. Your EC2 resources for the selected Region are displayed on the **EC2 Dashboard** in the **Resources** section.

To find your Regions using the AWS CLI

Use the [describe-regions](#) command as follows to describe the Regions that are enabled for your account.

```
aws ec2 describe-regions
```

To describe all Regions, including any Regions that are disabled for your account, add the `--all-regions` option as follows.

```
aws ec2 describe-regions --all-regions
```

Get the Region display name

You can use AWS Systems Manager Parameter Store to view the display name of a Region. Each Region has public parameters in the following path.

```
/aws/service/global-infrastructure/regions/region-code
```

The public parameters for a Region include the following:

- /aws/service/global-infrastructure/regions/*region-code*/domain
- /aws/service/global-infrastructure/regions/*region-code*/geolocationCountry
- /aws/service/global-infrastructure/regions/*region-code*/geolocationRegion
- /aws/service/global-infrastructure/regions/*region-code*/longName
- /aws/service/global-infrastructure/regions/*region-code*/partition

The longName parameter contains the Region display name. The following [get-parameters-by-path](#) command returns the display name of the af-south-1 Region. It uses the --query option to scope the output to the name of the Region. You must enclose the query string in single quotes on Linux. To run this command using the Windows Command Prompt, either omit the single quotes or change them to double quotes.

Linux

```
aws ssm get-parameters-by-path \  
  --path /aws/service/global-infrastructure/regions/af-south-1 \  
  --query 'Parameters[?Name.contains(@, `longName`)].Value' \  
  --output text
```

Windows

```
aws ssm get-parameters-by-path ^  
  --path /aws/service/global-infrastructure/regions/af-south-1 ^  
  --query "Parameters[?Name.contains(@, `longName`)].Value" ^  
  --output text
```

The following is example output.

Africa (Cape Town)

For more information, see [Working with public parameters](#) in the *AWS Systems Manager User Guide*.

Specify the Region for a resource

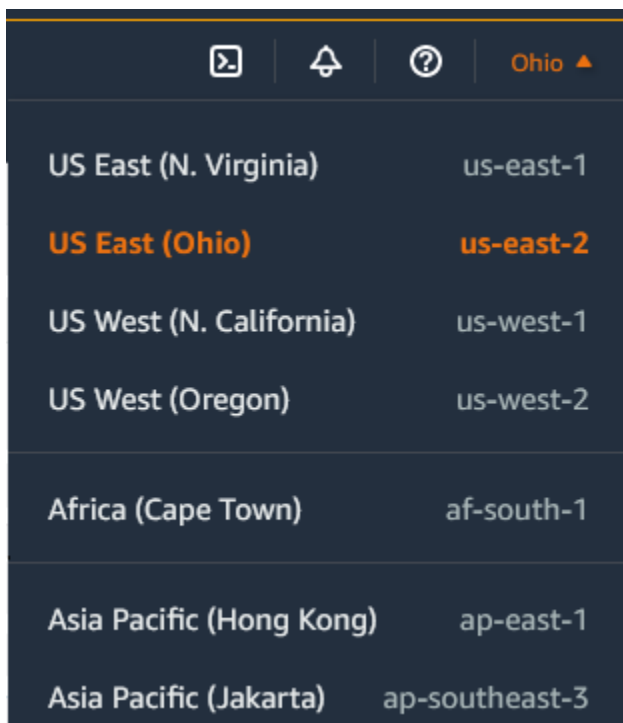
Every time you create an Amazon EC2 resource, you can specify the Region for the resource. You can specify the Region for a resource using the AWS Management Console or the command line.

Considerations

Some AWS resources might not be available in all Regions. Ensure that you can create the resources that you need in the desired Regions before you launch an instance.

To specify the Region for a resource using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, choose the **Regions** selector and then choose the Region.



To specify the default Region using the command line

You can set the value of an environment variable to the desired Regional endpoint (for example, `https://ec2.us-east-2.amazonaws.com`):

- `AWS_DEFAULT_REGION` (AWS CLI)
- `Set-AWSDefaultRegion` (AWS Tools for Windows PowerShell)

Alternatively, you can use the `--region` (AWS CLI) or `-Region` (AWS Tools for Windows PowerShell) command line option with each individual command. For example, `--region us-east-2`.

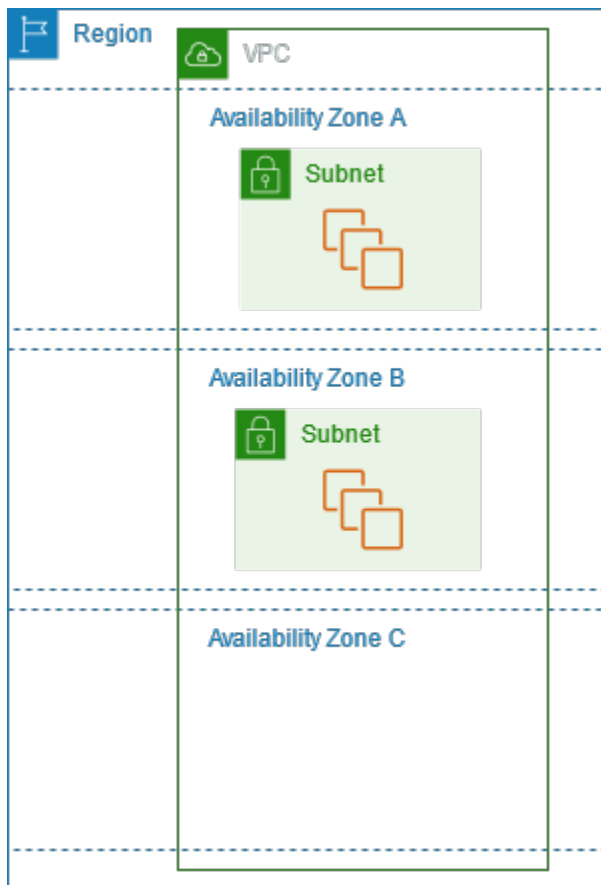
For more information about the endpoints for Amazon EC2, see [Amazon EC2 endpoints and quotas](#) in the *AWS General Reference*.

Availability Zones

Each Region has multiple, isolated locations known as *Availability Zones*. The code for Availability Zone is its Region code followed by a letter identifier. For example, `us-east-1a`.

When you launch an instance, you select a Region and a virtual private cloud (VPC), and then you can either select a subnet from one of the Availability Zones or let us choose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests. You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone.

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 instances. Availability Zone C has no subnets, therefore you can't launch instances into this Availability Zone.



As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you already have an instance in that Availability Zone. Eventually, we might also remove the constrained Availability Zone from the list of Availability Zones for new accounts. Therefore, your account might have a different number of available Availability Zones in a Region than another account.

Contents

- [AZ IDs](#)
- [Describe your Availability Zones](#)
- [Launch instances in an Availability Zone](#)
- [Migrate an instance to another Availability Zone](#)

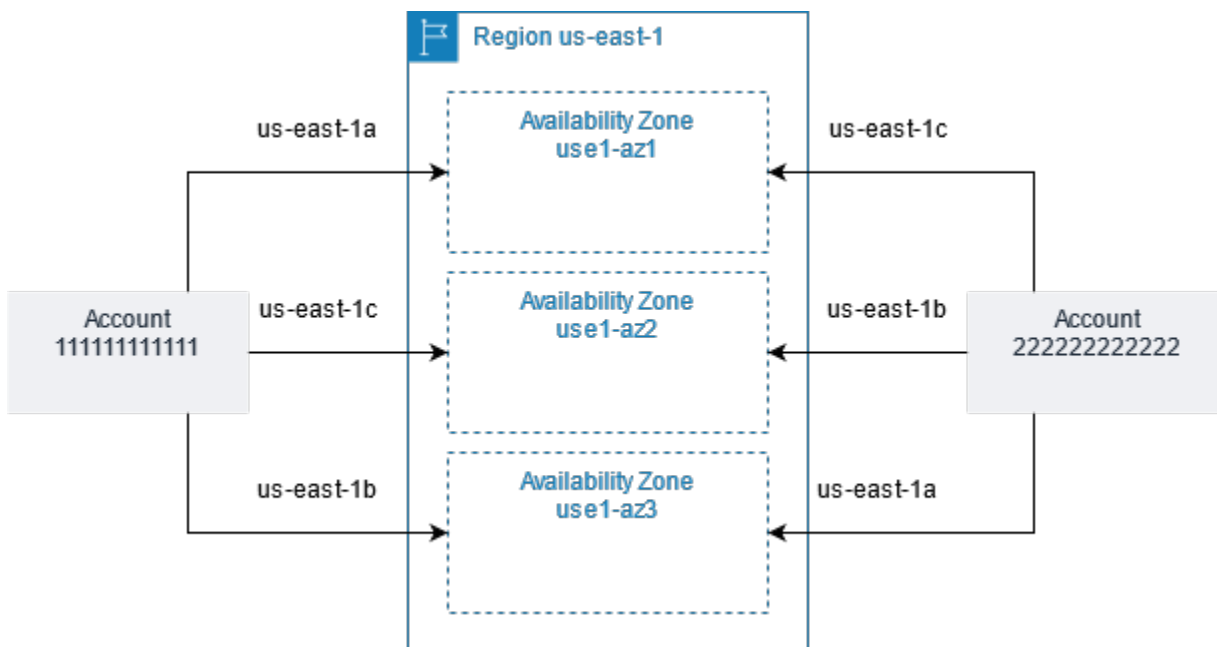
AZ IDs

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to codes for each AWS account in our oldest Regions. For

example, the `us-east-1a` for your AWS account might not be the same physical location as the `us-east-1a` for another AWS account.

To coordinate Availability Zones across accounts in all Regions even those that map Availability Zones, use the *AZ IDs*, which are unique and consistent identifiers for an Availability Zone. For example, `use1-az1` is an AZ ID for the `us-east-1` Region, and it has the same physical location in every AWS account. You can view the AZ IDs for your account to determine the physical location of your resources relative to the resources in another account. For example, if you share a subnet in the Availability Zone with the AZ ID `use1-az2` with another account, this subnet is available to that account in the Availability Zone whose AZ ID is also `use1-az2`.

The following diagram illustrates two accounts with different mappings of Availability Zone code to AZ ID.



Describe your Availability Zones

You can use the Amazon EC2 console or the command line interface to determine which Availability Zones are available for your account. For more information about these command line interfaces, see [Access Amazon EC2](#).

To find your Availability Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, choose the **Regions** selector and then choose the Region.

3. On the navigation pane, choose **EC2 Dashboard**.
4. The Availability Zones are listed in the **Service health** pane.

To find your Availability Zones using the AWS CLI

- Use the [describe-availability-zones](#) command as follows to describe the Availability Zones within the specified Region that are enabled for your account.

```
aws ec2 describe-availability-zones --region region-name
```

- Use the [describe-availability-zones](#) command as follows to describe the Availability Zones regardless of the opt-in status.

```
aws ec2 describe-availability-zones --all-availability-zones
```

Launch instances in an Availability Zone

When you launch an instance, select a Region that puts your instances closer to specific customers, or meets the legal or other requirements that you have. By launching your instances in separate Availability Zones, you can protect your applications from the failure of a single location.

When you launch an instance, you can optionally specify an Availability Zone in the Region that you are using. If you do not specify an Availability Zone, we select an Availability Zone for you. When you launch your initial instances, we recommend that you accept the default Availability Zone, because this allows us to select the best Availability Zone for you based on system health and available capacity. If you launch additional instances, specify an Availability Zone only if your new instances must be close to, or separated from, your running instances.

Migrate an instance to another Availability Zone

If necessary, you can migrate an instance from one Availability Zone to another. For example, if you try to modify the instance type of your instance and we can't launch an instance of the new instance type in the current Availability Zone, you can migrate the instance to an Availability Zone with capacity for the new instance type.

The migration process involves:

- Creating an AMI from the original instance

- Launching an instance in the new Availability Zone
- Updating the configuration of the new instance, as shown in the following procedure

To migrate an instance to another Availability Zone

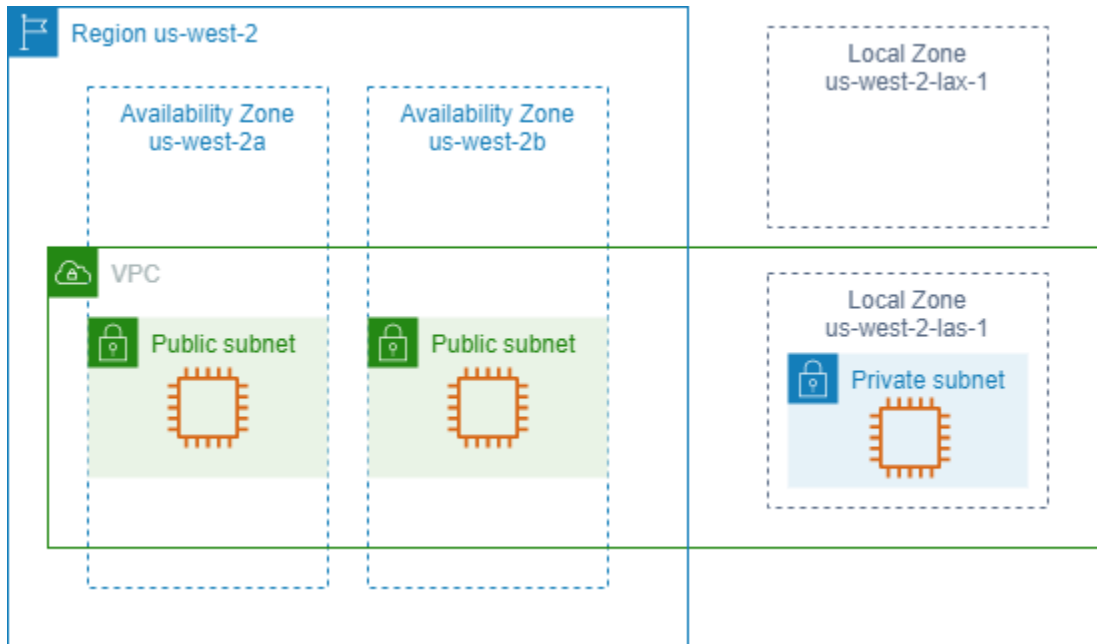
1. Create an AMI from the instance. The procedure depends on your operating system and the type of root device volume for the instance. For more information, see the documentation that corresponds to your operating system and root device volume:
 - [Create an Amazon EBS-backed Linux AMI](#)
 - [Create an instance store-backed Linux AMI](#)
 - [Create a custom Windows AMI](#)
2. If you need to preserve the private IPv4 address of the instance, you must delete the subnet in the current Availability Zone and then create a subnet in the new Availability Zone with the same IPv4 address range as the original subnet. Note that you must terminate all instances in a subnet before you can delete it. Therefore, you should create AMIs from all of the instances in your subnet so that you can move all instances from the current subnet to the new subnet.
3. Launch an instance from the AMI that you just created, specifying the new Availability Zone or subnet. You can use the same instance type as the original instance, or select a new instance type. For more information, see [Launch instances in an Availability Zone](#).
4. If the original instance has an associated Elastic IP address, associate it with the new instance. For more information, see [Disassociate an Elastic IP address](#).
5. If the original instance is a Reserved Instance, change the Availability Zone for your reservation. (If you also changed the instance type, you can also change the instance type for your reservation.) For more information, see [Submit modification requests](#).
6. (Optional) Terminate the original instance. For more information, see [Terminate an instance](#).

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.



To use a Local Zone, you must first enable it. For more information, see [the section called “Opt in to Local Zones”](#). Next, create a subnet in the Local Zone. Finally, launch resources in the Local Zone subnet, such as instances, so that your applications are close to your users.

Contents

- [Available Local Zones](#)
- [Opt in to Local Zones](#)
- [Launch instances in a Local Zone](#)

Available Local Zones

You can use the Amazon EC2 console or a command line interface to determine which Local Zones are available for your account. For a complete list, see [AWS Local Zones Locations](#).

To find your Local Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. From the navigation bar, choose the **Regions** selector and then choose the parent Region.
3. On the navigation pane, choose **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account attributes, Zones**.

To find your Local Zones using the AWS CLI

Use the [describe-availability-zones](#) command as follows to describe all Local Zones in the specified Region, even if they are not enabled. To describe only the Local Zones that you've enabled, omit the `--all-availability-zones` option.

```
aws ec2 describe-availability-zones --region region-name --filters Name=zone-type,Values=local-zone --all-availability-zones
```

Opt in to Local Zones

Before you can specify a Local Zone for a resource or service, you must opt in to Local Zones.

Consideration

Some AWS resources might not be available in all Regions. Make sure that you can create the resources that you need in the desired Regions or Local Zones before launching an instance in a specific Local Zone. For a list of services supported in each Local Zone see [AWS Local Zones Features](#).

To opt in to Local Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the upper-left corner of the page, select **New EC2 Experience**. You cannot complete this task using the old console experience.
3. From the navigation bar, choose the **Regions** selector and then choose the parent Region.
4. On the navigation pane, choose **EC2 Dashboard**.
5. In the upper-right corner of the page, choose **Account attributes, Zones**.
6. Choose a Local Zone and choose **Action > Manage Zone group**.
7. Under **Opt-in status**, choose **Enable**.
8. Choose **Update**.

To opt in to Local Zones using the AWS CLI

Use the [modify-availability-zone-group](#) command.

Launch instances in a Local Zone

When you launch an instance, you can specify a subnet that is in a Local Zone. You also allocate an IP address from a network border group. A network border group is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, `us-west-2-lax-1a`.

You can allocate the following IP addresses from a network border group:

- Amazon-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*.

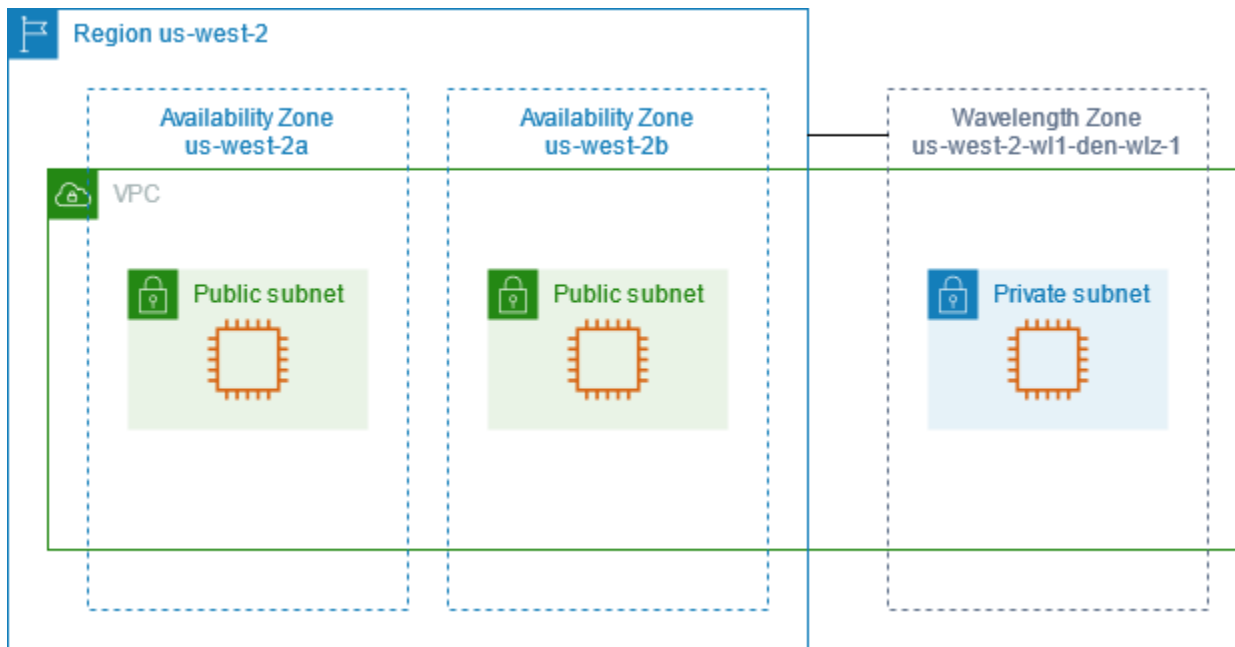
Wavelength Zones

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-w11-bos-w1z-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.



To use a Wavelength Zone, you must first opt in to the Zone. For more information, see [the section called “Enable Wavelength Zones”](#). Next, create a subnet in the Wavelength Zone. Finally, launch your resources in the Wavelength Zones subnet, so that your applications are closer to your end users.

Wavelength Zones are not available in every Region. For information about the Regions that support Wavelength Zones, see [Available Wavelength Zones](#) in the *AWS Wavelength Developer Guide*.

Contents

- [Describe your Wavelength Zones](#)
- [Enable Wavelength Zones](#)
- [Launch instances in a Wavelength Zone](#)

Describe your Wavelength Zones

You can use the Amazon EC2 console or the command line interface to determine which Wavelength Zones are available for your account. For more information about these command line interfaces, see [Access Amazon EC2](#).

To find your Wavelength Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. From the navigation bar, choose the **Regions** selector and then choose the Region.
3. On the navigation pane, choose **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account attributes, Zones**.

To find your Wavelength Zones using the AWS CLI

- Use the [describe-availability-zones](#) command as follows to describe the Wavelength Zones within the specified Region that are enabled for your account.

```
aws ec2 describe-availability-zones --region region-name
```

- Use the [describe-availability-zones](#) command as follows to describe the Wavelength Zones regardless of the opt-in status.

```
aws ec2 describe-availability-zones --all-availability-zones
```

Enable Wavelength Zones

Before you specify a Wavelength Zone for a resource or service, you must opt in to Wavelength Zones.

Considerations

- Some AWS resources are not available in all Regions. Make sure that you can create the resources that you need in the desired Region or Wavelength Zone before launching an instance in a specific Wavelength Zone.

To opt in to Wavelength Zone using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the upper-left corner of the page, select **New EC2 Experience**. You cannot complete this task using the old console experience.
3. From the navigation bar, choose the **Regions** selector and then choose the Region.
4. On the navigation pane, choose **EC2 Dashboard**.
5. In the upper-right corner of the page, choose **Account attributes, Zones**.
6. Choose a Wavelength Zone and choose **Action > Manage Zone group**.

7. Under **Opt-in status**, choose **Enable**.
8. Choose **Update**.

To enable Wavelength Zones using the AWS CLI

Use the [modify-availability-zone-group](#) command.

Launch instances in a Wavelength Zone

When you launch an instance, you can specify a subnet which is in a Wavelength Zone. You also allocate a carrier IP address from a network border group, which is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, `us-east-1-wl1-bos-wlz-1`.

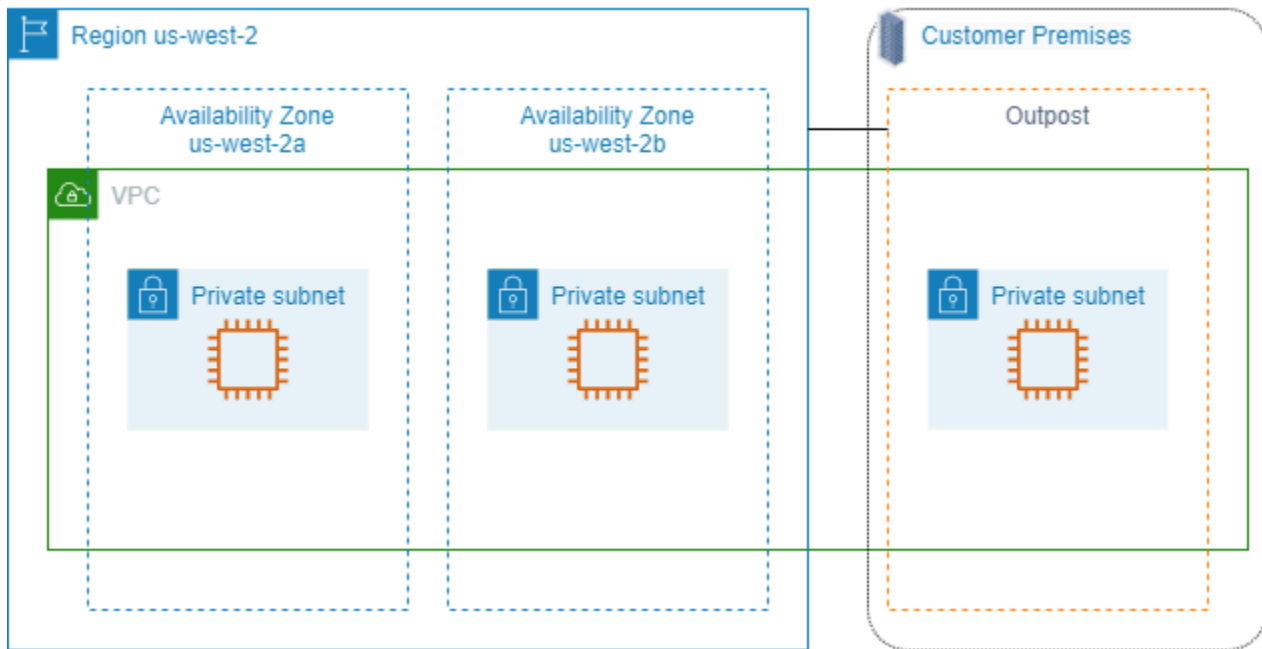
For information about how to launch an instance in a Wavelength Zone, see [Get started with AWS Wavelength](#) in the *AWS Wavelength Developer Guide*.

AWS Outposts

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources. 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.



To begin using AWS Outposts, you must create an Outpost and order Outpost capacity. For more information about Outposts configurations, see [our catalog](#). After your Outpost equipment is installed, the compute and storage capacity is available for you when you launch Amazon EC2 instances on your Outpost.

Launch instances on an Outpost

You can launch EC2 instances in the Outpost subnet that you created. Security groups control inbound and outbound traffic for instances with elastic network interfaces in an Outpost subnet, as they do for instances in an Availability Zone subnet. To connect to an EC2 instance in an Outpost subnet, you can specify a key pair when you launch the instance, as you do for instances in an Availability Zone subnet.

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](#).

For information about how to create an Outpost, see [Get started with AWS Outposts](#) in the *AWS Outposts User Guide*.

Create a volume on an Outpost rack

AWS Outposts offers rack and server form factors. If your 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](#).

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.

Contents

- [Private IPv4 addresses](#)
- [Public IPv4 addresses](#)
- [Elastic IP addresses \(IPv4\)](#)
- [IPv6 addresses](#)
- [Work with the IPv4 addresses for your instances](#)
- [Work with the IPv6 addresses for your instances](#)
- [Multiple IP addresses](#)
- [EC2 instance hostnames](#)
- [Link-local addresses](#)

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 (eth0) that is assigned the primary private IPv4 address. You can also specify additional private IPv4 addresses, known as *secondary private IPv4 addresses*. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see [Multiple IP addresses](#).

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:

- Modifying 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*.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see [Assign a public IPv4 address during instance launch](#).
- You can unassign a public IP address from your instance after launch by [managing the IP addresses associated with a network interface](#).

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.

In certain cases, we release the public IP address from your instance, or assign it a new one:

- We release your instance's public IP address when it is stopped, hibernated, or terminated. Your stopped or hibernated instance receives a new public IP address when it is started.
- We release your instance's public IP address when you associate an Elastic IP address with it. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.
- If your instance's public IP address is released while it has a secondary private IP address that is associated with an Elastic IP address, the instance does not receive a new public IP address.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see [Elastic IP addresses](#).

Note

- 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.

Elastic IP addresses (IPv4)

An Elastic IP address is a public IPv4 address that you can allocate to your account. You can associate it to and disassociate it from instances as you require. It's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see [Elastic IP addresses](#).

We do not support Elastic IP addresses for IPv6.

IPv6 addresses

You can optionally associate an IPv6 CIDR block with your VPC and associate IPv6 CIDR blocks with your subnets. The IPv6 CIDR block for your VPC is automatically assigned from Amazon's pool of IPv6 addresses; you cannot choose the range yourself. For more information, see the following topics in the *Amazon VPC User Guide*:

- [IP addressing for your VPCs and subnets](#)
- [Add an IPv6 CIDR block to your VPC](#)

- [Add an IPv6 CIDR block to your subnet](#)

IPv6 addresses are globally unique and can be configured to remain private or reachable over the Internet. Your instance receives an IPv6 address if an IPv6 CIDR block is associated with your VPC and subnet, and if one of the following is true:

- Your subnet is configured to automatically assign an IPv6 address to an instance during launch. For more information, see [Modify the IPv6 addressing attribute for your subnet](#).
- You assign an IPv6 address to your instance during launch.
- You assign an IPv6 address to the primary network interface of your instance after launch.
- You assign an IPv6 address to a network interface in the same subnet, and attach the network interface to your instance after launch.

When your instance receives an IPv6 address during launch, the address is associated with the primary network interface (eth0) of the instance. You can manage the IPv6 addresses for your instances primary network interface (eth0) in the following ways:

- 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 [IP addresses per network interface per instance type](#).
- 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](#) or [Manage IP addresses](#).

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](#).

Work with the IPv4 addresses for your instances

You can assign a public IPv4 address to your instance when you launch it. You can view the IPv4 addresses for your instance in the console through either the **Instances** page or the **Network Interfaces** page.

Contents

- [View the IPv4 addresses](#)
- [Assign a public IPv4 address during instance launch](#)

View the IPv4 addresses

You can use the Amazon EC2 console to view the public and private IPv4 addresses of your instances. You can also determine the public IPv4 and private IPv4 addresses of your instance from within your instance by using instance metadata. For more information, see [Instance metadata and user data](#).

The public IPv4 address is displayed as a property of the network interface in the console, but it's mapped to the primary private IPv4 address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig` (Linux) or `ipconfig` (Windows), the public IPv4 address is not displayed. To determine your instance's public IPv4 address from an instance, use instance metadata.

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select your instance.
3. The following information is available on the **Networking** tab:
 - **Public IPv4 address** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
 - **Private IPv4 addresses** — The private IPv4 address.
 - **Secondary private IPv4 addresses** — Any secondary private IPv4 addresses.
4. To view more detailed information, on the **Networking** tab, choose the ID of the primary network interface to open the **Network interfaces** page, and then choose the ID of the network interface to open its details page.

To view the IPv4 addresses for an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).

To determine your instance's IPv4 addresses using instance metadata

1. Connect to your instance. For more information, see [Connect to your Linux instance](#).
2. Use the following command to access the private IP address:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-ipv4
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-ipv4
```

If an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Assign a public IPv4 address during instance launch

Each subnet has an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IPv4 addressing feature is also available for you to control whether your instance is assigned a public IPv4 address; you can override the default behavior of the subnet's IP addressing attribute. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

Considerations

- You can 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 cannot auto-assign a public IP address if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IP feature if you specify an existing network interface for eth0.
- 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](#).

To assign a public IPv4 address during instance launch using the console

Follow the procedure to [launch an instance](#), and when you configure [Network Settings](#), choose the option to **Auto-assign Public IP**.

To enable or disable the public IP addressing feature using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- Use the `--associate-public-ip-address` or the `--no-associate-public-ip-address` option with the [run-instances](#) command (AWS CLI)
- Use the `-AssociatePublicIp` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell)

Work with the IPv6 addresses for your instances

You can view the IPv6 addresses assigned to your instance, assign a public IPv6 address to your instance, or unassign an IPv6 address from your instance. You can view these addresses in the console through either the **Instances** page or the **Network Interfaces** page.

Contents

- [View the IPv6 addresses](#)
- [Assign an IPv6 address to an instance](#)
- [Unassign an IPv6 address from an instance](#)

View the IPv6 addresses

You can use the Amazon EC2 console, AWS CLI, and instance metadata to view the IPv6 addresses for your instances.

To view the IPv6 addresses for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Networking** tab, locate **IPv6 addresses**.

To view the IPv6 addresses for an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).

To view the IPv6 addresses for an instance using instance metadata

1. Connect to your instance. For more information, see [Connect to your Linux instance](#).
2. Use the following command to view the IPv6 address (you can get the MAC address from `http://169.254.169.254/latest/meta-data/network/interfaces/macs/`).

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H  
"X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/  
meta-data/network/interfaces/macs/mac-address/ipv6s
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/  
macs/mac-address/ipv6s
```

Assign an IPv6 address to an instance

If your VPC and subnet have IPv6 CIDR blocks associated with them, you can assign an IPv6 address to your instance during or after launch. The IPv6 address is assigned from the IPv6 address range of the subnet, and is assigned to the network interface with the device index of eth0.

To assign an IPv6 address during instance launch

Follow the procedure to [launch an instance](#), and when you configure [Network Settings](#), choose the option to **Auto-assign IPv6 IP**.

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**. Enter an IPv6 address from the range of the subnet or leave the field blank to let Amazon choose an IPv6 address for you.
5. Choose **Save**.

To assign an IPv6 address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- Use the `--ipv6-addresses` option with the [run-instances](#) command (AWS CLI)
- Use the `Ipv6Addresses` property for `-NetworkInterface` in the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell)
- [assign-ipv6-addresses](#) (AWS CLI)
- [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Unassign an IPv6 address from an instance

You can unassign an IPv6 address from an instance at any time.

To unassign an IPv6 address from an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, and choose **Actions, Networking, Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Unassign** next to the IPv6 address.
5. Choose **Save**.

To unassign an IPv6 address from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [unassign-ipv6-addresses](#) (AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell).

Multiple IP addresses

You can specify multiple private IPv4 and IPv6 addresses for your instances. The number of network interfaces and private IPv4 and IPv6 addresses that you can specify for an instance depends on the instance type. For more information, see [IP addresses per network interface per instance type](#).

It can be useful to assign multiple IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple IP addresses for each network interface.
- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary IP address to the standby instance.

Contents

- [How multiple IP addresses work](#)
- [Work with multiple IPv4 addresses](#)
- [Work with multiple IPv6 addresses](#)

How multiple IP addresses work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IPv4 address to any network interface.
- You can assign multiple IPv6 addresses to a network interface that's in a subnet that has an associated IPv6 CIDR block.
- You must choose a secondary IPv4 address from the IPv4 CIDR block range of the subnet for the network interface.
- You must choose IPv6 addresses from the IPv6 CIDR block range of the subnet for the network interface.
- You associate security groups with network interfaces, not individual IP addresses. Therefore, each IP address you specify in a network interface is subject to the security group of its network interface.
- Multiple IP addresses can be assigned and unassigned to network interfaces attached to running or stopped instances.
- Secondary private IPv4 addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- An IPv6 address cannot be reassigned to another network interface; you must first unassign the IPv6 address from the existing network interface.
- When assigning multiple IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the IP addresses can't be assigned.

- Primary private IPv4 addresses, secondary private IPv4 addresses, Elastic IP addresses, and IPv6 addresses remain with a secondary network interface when it is detached from an instance or attached to an instance.
- Although you can't detach the primary network interface from an instance, you can reassign the secondary private IPv4 address of the primary network interface to another network interface.

The following list explains how multiple IP addresses work with Elastic IP addresses (IPv4 only):

- Each private IPv4 address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IPv4 address is reassigned to another interface, the secondary private IPv4 address retains its association with an Elastic IP address.
- When a secondary private IPv4 address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IPv4 address.

Work with multiple IPv4 addresses

You can assign a secondary private IPv4 address to an instance, associate an Elastic IPv4 address with a secondary private IPv4 address, and unassign a secondary private IPv4 address.

Tasks

- [Assign a secondary private IPv4 address](#)
- [Configure the operating system on your instance to recognize secondary private IPv4 addresses](#)
- [Associate an Elastic IP address with the secondary private IPv4 address](#)
- [View your secondary private IPv4 addresses](#)
- [Unassign a secondary private IPv4 address](#)

Assign a secondary private IPv4 address

You can assign the secondary private IPv4 address to the network interface for an instance as you launch the instance, or after the instance is running. This section includes the following procedures.

- [To assign a secondary private IPv4 address when launching an instance](#)
- [To assign a secondary IPv4 address during launch using the command line](#)
- [To assign a secondary private IPv4 address to a network interface](#)
- [To assign a secondary private IPv4 address to an existing instance using the command line](#)

New console

To assign a secondary private IPv4 address when launching an instance

1. Follow the procedure to [launch an instance](#). For [Network settings](#), choose **Edit**.
2. Select a VPC and a subnet.
3. Expand **Advanced network configuration**.
4. For **Secondary IP**, choose **Automatically assign** and enter the number of IP addresses (Amazon automatically assigns secondary IPv4 addresses) or choose **Manually assign** and enter the IPv4 addresses.
5. Complete the remaining steps to [launch the instance](#).

Old console

To assign a secondary private IPv4 address when launching an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - To add another network interface, choose **Add Device**. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP addresses per network interface per instance type](#).

Important

When you add a second network interface, the system can no longer auto-assign a public IPv4 address. You will not be able to connect to the instance over IPv4 unless you assign an Elastic IP address to the primary network interface (eth0). You can assign the Elastic IP address after you complete the Launch wizard. For more information, see [Work with Elastic IP addresses](#).

- For each network interface, under **Secondary IP addresses**, choose **Add IP**, and then enter a private IP address from the subnet range, or accept the default Auto-assign value to let Amazon select an address.
6. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Add Tags**.
 7. On the **Add Tags** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
 8. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
 9. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

Important

After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more information, see [Configure the operating system on your instance to recognize secondary private IPv4 addresses](#).

To assign a secondary IPv4 address during launch using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).
 - The `--secondary-private-ip-addresses` option with the [run-instances](#) command (AWS CLI)
 - Define `-NetworkInterface` and specify the `PrivateIpAddresses` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

To assign a secondary private IPv4 address to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Network Interfaces**, and then select the network interface for the instance.
3. Choose **Actions, Manage IP Addresses**.
4. Expand the network interface. Under **IPv4 addresses**, choose **Assign new IP address**.
5. Enter a specific IPv4 address that's within the subnet range for the instance, or leave the field blank to let Amazon select an IPv4 address for you.
6. (Optional) Select **Allow** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
7. Choose **Save**.

Alternatively, you can assign a secondary private IPv4 address to an instance. Choose **Instances** in the navigation pane, select the instance, and then choose **Actions, Networking, Manage IP addresses**. You can configure the same information as you did in the steps above. The IP address is assigned to the primary network interface (eth0) for the instance.

To assign a secondary private IPv4 address to an existing instance using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).
 - [assign-private-ip-addresses](#) (AWS CLI)
 - [Register-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Configure the operating system on your instance to recognize secondary private IPv4 addresses

After you assign a secondary private IPv4 address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

- If you are using Amazon Linux, the `ec2-net-utils` package can take care of this step for you. It configures additional network interfaces that you attach while the instance is running, refreshes secondary IPv4 addresses during DHCP lease renewal, and updates the related routing rules. You can immediately refresh the list of interfaces by using the command `sudo service network restart` and then view the up-to-date list using `ip addr li`. If you require manual control over your network configuration, you can remove the `ec2-net-utils` package. For more information, see [Configure your network interface using ec2-net-utils for Amazon Linux 2](#).

- If you are using another Linux distribution, see the documentation for your Linux distribution. Search for information about configuring additional network interfaces and secondary IPv4 addresses. If the instance has two or more interfaces on the same subnet, search for information about using routing rules to work around asymmetric routing.

For information about configuring a Windows instance, see [Configuring a secondary private IP address for your Windows instance in a VPC](#) in the *Amazon EC2 User Guide for Windows Instances*.

Associate an Elastic IP address with the secondary private IPv4 address

To associate an Elastic IP address with a secondary private IPv4 address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the check box for the Elastic IP address
4. Choose **Actions, Associate Elastic IP address**.
5. For **Resource type**, choose **Network interface**. select the network interface, and then select the secondary IP address from the **Private IP address** list.
6. For **Network interface**, select the network interface. select the secondary IP address from the **Private IP address** list.
7. For **Private IP address**, select the secondary IP address.
8. Choose **Associate**.

To associate an Elastic IP address with a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).
 - [associate-address](#) (AWS CLI)
 - [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

View your secondary private IPv4 addresses

To view the private IPv4 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the check box for the network interface.
4. On the **Details** tab, under **IP addresses**, locate **Private IPv4 address** and **Secondary private IPv4 addresses**.

To view the private IPv4 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the check box for the instance.
4. On the **Networking** tab, under **Networking details**, locate **Private IPv4 addresses** and **Secondary private IPv4 addresses**.

Unassign a secondary private IPv4 address

If you no longer require a secondary private IPv4 address, you can unassign it from the instance or the network interface. When a secondary private IPv4 address is unassigned from a network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IPv4 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions, Networking, Manage IP addresses**.
4. Expand the network interface. For **IPv4 addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Save**.

To unassign a secondary private IPv4 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface, choose **Actions, Manage IP addresses**.
4. Expand the network interface. For **IPv4 addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Save**.

To unassign a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).
 - [unassign-private-ip-addresses](#) (AWS CLI)
 - [Unregister-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Work with multiple IPv6 addresses

You can assign multiple IPv6 addresses to your instance, view the IPv6 addresses assigned to your instance, and unassign IPv6 addresses from your instance.

Contents

- [Assign multiple IPv6 addresses](#)
- [View your IPv6 addresses](#)
- [Unassign an IPv6 address](#)

Assign multiple IPv6 addresses

You can assign one or more IPv6 addresses to your instance during launch or after launch. To assign an IPv6 address to an instance, the VPC and subnet in which you launch the instance must have an associated IPv6 CIDR block.

New console

To assign multiple IPv6 addresses during launch

1. Follow the procedure to [launch an instance](#). For [Network settings](#), choose **Edit**.
2. Select a VPC and a subnet.

3. Expand **Advanced network configuration**.
4. For **IPv6 IPs**, choose **Automatically assign** and the number of IP addresses (Amazon automatically assigns the IPv6 addresses) or choose **Manually assign** and enter the IPv6 addresses.
5. Complete the remaining steps to [launch the instance](#).

Old console

To assign multiple IPv6 addresses during launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the dashboard, choose **Launch Instance**.
3. Select an AMI, choose an instance type, and choose **Next: Configure Instance Details**. Ensure that you choose an instance type that support IPv6. For more information, see [Amazon EC2 instance types](#).
4. On the **Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - To assign a single IPv6 address to the primary network interface (eth0), under **IPv6 IPs**, choose **Add IP**. To add a secondary IPv6 address, choose **Add IP** again. You can enter an IPv6 address from the range of the subnet, or leave the default **Auto-assign** value to let Amazon choose an IPv6 address from the subnet for you.
 - Choose **Add Device** to add another network interface and repeat the steps above to add one or more IPv6 addresses to the network interface. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP addresses per network interface per instance type](#).
6. Follow the next steps in the wizard to attach volumes and tag your instance.
7. On the **Configure Security Group** page, select an existing security group or create a new one. If you want your instance to be reachable over IPv6, ensure that your security group has rules that allow access from IPv6 addresses. For more information, see [Security group rules for different use cases](#). Choose **Review and Launch**.

8. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

You can use the **Instances** screen Amazon EC2 console to assign multiple IPv6 addresses to an existing instance. This assigns the IPv6 addresses to the primary network interface (eth0) for the instance. To assign a specific IPv6 address to the instance, ensure that the IPv6 address is not already assigned to another instance or network interface.

To assign multiple IPv6 addresses to an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Networking, Manage IP addresses**.
4. Expand the network interface. For **IPv6 addresses**, choose **Assign new IP address** for each IPv6 address to add. You can specify an IPv6 address from the range of the subnet, or leave the field empty to let Amazon choose an IPv6 address for you.
5. Choose **Save**.

Alternatively, you can assign multiple IPv6 addresses to an existing network interface. The network interface must have been created in a subnet that has an associated IPv6 CIDR block. To assign a specific IPv6 address to the network interface, ensure that the IPv6 address is not already assigned to another network interface.

To assign multiple IPv6 addresses to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP addresses**.
4. Expand the network interface. For **IPv6 addresses**, choose **Assign new IP address** for each IPv6 address to add. You can specify an IPv6 address from the range of the subnet, or leave the field empty to let Amazon choose an IPv6 address for you.
5. Choose **Save**.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- **Assign an IPv6 address during launch:**
 - Use the `--ipv6-addresses` or `--ipv6-address-count` options with the [run-instances](#) command (AWS CLI)
 - Define `-NetworkInterface` and specify the `Ipv6Addresses` or `Ipv6AddressCount` parameters with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).
- **Assign an IPv6 address to a network interface:**
 - [assign-ipv6-addresses](#) (AWS CLI)
 - [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

View your IPv6 addresses

You can view the IPv6 addresses for an instance or for a network interface.

To view the IPv6 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the check box for your instance.
4. On the **Networking** tab, locate the **IPv6 addresses** field.

To view the IPv6 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the check box for your network interface.
4. On the **Details** tab, under **IP addresses**, locate the **IPv6 addresses** field.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- **View the IPv6 addresses for an instance:**
 - [describe-instances](#) (AWS CLI)
 - [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).
- **View the IPv6 addresses for a network interface:**
 - [describe-network-interfaces](#) (AWS CLI)
 - [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Unassign an IPv6 address

You can unassign an IPv6 address from the primary network interface of an instance, or you can unassign an IPv6 address from a network interface.

To unassign an IPv6 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the check box for your instance, and then choose **Actions, Networking, Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Unassign** next to the IPv6 address.
5. Choose **Save**.

To unassign an IPv6 address from 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 check box for your network interface, and then choose **Actions, Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Unassign** next to the IPv6 address.
5. Choose **Save**.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [unassign-ipv6-addresses](#) (AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell).

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 [Amazon EC2 instance hostname types](#). 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](#)

Amazon EC2 instance hostname types

This section describes the Amazon EC2 instance guest OS hostname types available when you launch instances into your VPC subnets.

The hostname distinguishes the EC2 instances on your network. You may use the 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

- [Types of EC2 hostnames](#)
- [Where you see Resource name and IP name](#)
- [How to decide whether to choose Resource name or IP name](#)
- [Modify Hostname type and DNS Hostname configurations](#)

Types of EC2 hostnames

There are two 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](#)
- [Resource name](#)
- [The difference between IP name and Resource name](#)

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 [Modify Hostname type and DNS Hostname configurations](#).

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 [Modify Hostname type and DNS Hostname configurations](#).

Where you see Resource name and IP name

This section describes where you see the hostname types resource name and IP name in the EC2 console.

Contents

- [When creating an EC2 instance](#)
- [When viewing the details of an existing EC2 instance](#)

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 instance using the new launch instance wizard](#)) 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 instance using the new launch instance wizard](#)) 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.

How to decide whether to choose Resource name or IP name

When you launch an EC2 instance (see [Launch an instance using the new launch instance wizard](#)), 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.

Modify Hostname type and DNS Hostname configurations

Follow the steps in this section to modify Hostname type and DNS Hostname configurations for subnets or EC2 instances after they've been launched.

Contents

- [Subnets](#)
- [EC2 instances](#)

Subnets

Modify the configurations for a subnet by selecting a subnet in the VPC console and choosing **Actions, Edit subnet settings**.

Note

Changing the subnet settings doesn't change the configuration of EC2 instances that are already launched in the subnet.

- **Hostname type:** Determines whether you want the default setting of the guest OS hostname of the EC2 instance launched in the subnet to be the resource name or IP name.

- **Enable DNS hostname IPv4 (A record) requests:** Determines whether DNS requests/queries to your resource name resolve to the private IPv4 address (A record) of this EC2 instance.
- **Enable 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.

EC2 instances

Follow the steps in this section to modify the Hostname type and DNS Hostname configurations for an EC2 instance.

Important

- To change the **Use resource based naming as guest OS hostname** setting, you must first stop the instance. To change the **Answer DNS hostname IPv4 (A record) request** or **Answer DNS hostname IPv6 (AAAA record) requests** settings, you don't have to stop the instance.
- To modify any of the settings for non-EBS backed EC2 instance types, you cannot stop the instance. You must terminate the instance and launch a new instance with the desired Hostname type and DNS Hostname configurations.

To modify the Hostname type and DNS Hostname configurations for an EC2 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.

Bring your own IP addresses (BYOIP) in Amazon EC2

You can bring part or all of your publicly routable IPv4 or IPv6 address range from your on-premises network to your AWS account. You continue to control the address range and you can advertise the address range on the internet through AWS. After you bring the address range to AWS, it appears in your AWS account as an address pool.

For a list of Regions where BYOIP is available, see [Regional availability](#).

To see BYOIP information for Windows instances, switch to this page in the *Amazon EC2 User Guide for Windows Instances* guide: [Bring your own IP addresses \(BYOIP\) in Amazon EC2](#).

Note

- The steps on this page describe 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*.

Contents

- [BYOIP definitions](#)
- [Requirements and quotas](#)
- [Onboarding prerequisites for your BYOIP address range](#)
- [Onboard your BYOIP](#)
- [Work with your address range](#)
- [Validate your BYOIP](#)

- [Regional availability](#)
- [Local Zone availability](#)
- [Learn more](#)

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 /56 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).

Onboarding prerequisites for your BYOIP address range

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.

Preparation phase

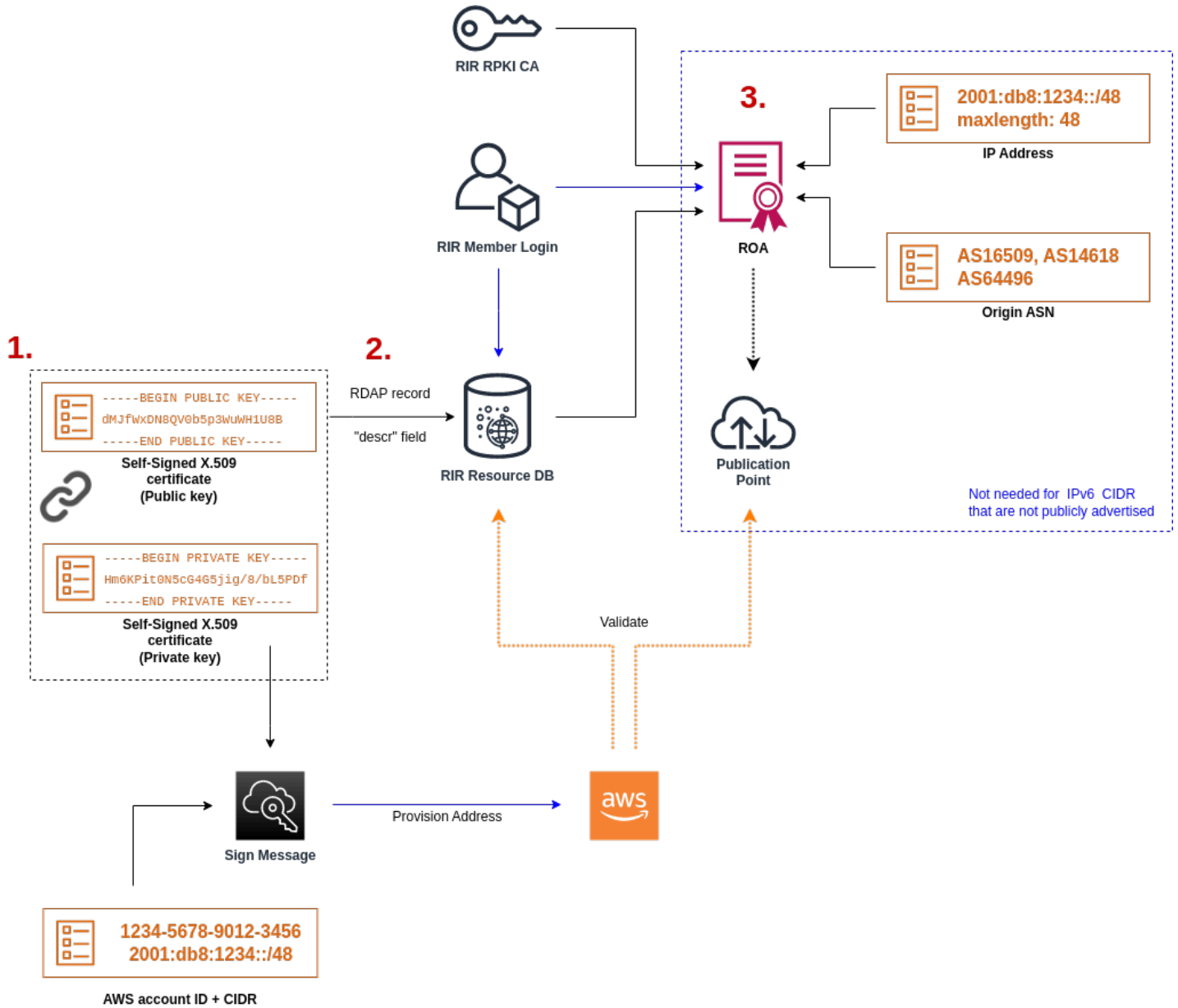
1. [Create a private key](#) and use it to generate a self-signed X.509 certificate for authentication purposes. This certificate is only used during the provisioning phase.

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.

⚠ Important

Before onboarding your address range, complete the following prerequisites. The tasks in this section require a Linux terminal and may be performed using Linux, the [AWS CloudShell](#), or the [Windows Subsystem for Linux](#).

1. 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-----
MIIEvGIBADANBqkqhkiG9w0BAQEFAASCbGwggSkAgEAAoIBAQDFBXHRI4HVKAhH
3seiciooizCRTbJe1+YsXNTja4XyKypVGIFWDGhZs44FCH1P00SVJ+NqP74w96oM
7DPS3xo9kaQyZBFn2YEp2EBq5vf307KHNRmZZUmkn0zH0SEpNmY2fMxISBxewLxR
FAniwmSd/8TDvHJMY9FvAIvWuTsv510tJKk+a91K4+t03UdDR7Sno5WEXefsBrW3
g1ydo3TBsx8i5/YiV0cNApy7ge2/FiwY3aCXJB6r6nuF6H8mRgI4r4vkMRs01AhJ
DnZPNNeweboo+K3Q31wbgbm0KD/z9svk8N/+hUTBtIX0fRtbG+PLIw3xWRHGfMSn2
BzsPVuDLAgMBAAECggEACiJUj2hfJkKv47Dc3es3Zex67A5uDVjXmxf0x2Xhdupn
fAcNqAptV6fXt0SPUNbhUxbBKNbshoJGufFwXPLi1SxnpzvkdU4Hyco4zgbhXfSE
RNYjYf0GzTPwdBLpNMB6k3Tp4RHse6dNrLH0jDhpioL8cQEBdBjyVF5X0wymEbmV
mC0jgH/MxsBAPWW6ZKicg9ULM1WiAZ3MRAZPjHHgpYkAAsUWKAbCBwVQcVjG059W
jfZjzTX5pQtVvH68rucih88DTZCwjCkjbHxg+0IkJBLE5wkh82jIHSivZ63flwLw
z+E0+HhELSZJrn2MY6Jxmik3qNNUOF/Z+3msdj2luQKBgQDjw1C/3jxp8zJy6P8o
JQKv7TdvMwUj4VSW0HZBHLv4evJaaia0uQjIo1UDa8AYitqhX1NmCCehGH8yuXj/
v6V3CzMKDkmRr1Nr0NnSz5QsndQ04Z6ihAQ1PmJ96g4wKtgoC7AYpyP0g1a+4/sj
b1+o3YQI4pD/F71c+qaztH7PRwKBgQDdc23yNmT3+Jyptf0fKjEv0NK+xwUKzi9c
L/0zBq5y0IC1Pz2T85g0e1i8kwZws+xlpG6uBT61mIJELd0k59FyupNu4dPvX5SD
6GGqd4xjk9KvI74usGe0BohmF0phTHkrWKBxXiyT0oS8zjnJlEn8ysIpGg028jJr
LpaHNZ/MXQKBgQDFLncnS0LzpsS2aK0tzyZU8SMYqVH0GMxj7quhneBq2T6FbiLD
T9TV1YaGNZ0j71vQaLI19q0ubWymbautH00p5KV8owdf4+bf1/NJaPI0zhDUSIjD
Qo01WW31Z9XDSRhKFTnWzmCjBdeIcajyzf10YKsycAW91Itu8aBrMndnQKBgQDb
nNp/JyRwqj0rNljK7DHEs+SD39kHQzzCfkd+dnTPv2sc06+cpym3yu1QcbokULpy
fmRo3bin/pvJQ3aZX/Bdh9woTXqhXDdrrSwWInVYMQPyPk8f/D9mIOJp5FUWmWHD
U+whIZSxsEeE+jtixlWtheKRYkQmzQZXBWdIhYyI3QKBgD+F/6wcZ85QW8nAUyKA
3WrSIx/3cwDgdm4NRGct8Z0ZjTHjiy9ojMOD1L7iMhRQ/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: xxxxxxxx
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-----.

2. 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, email the certificate to helpdesk@apnic.net to manually add it to the "remarks" field for your address range. Send the email using the APNIC authorized contact for the IP addresses.

You can remove the certificate from your RIR's record after the provisioning stage below has been completed.

3. 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 /56 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 BYOIP

The onboarding process for BYOIP has the following tasks depending on your needs:

Topics

- [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](#)

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 [AWS CLI configuration](#) 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 from other locations before you advertise it through AWS. If you keep advertising your IP address range from other locations, we can't reliably support it or troubleshoot issues. Specifically, we can't guarantee that traffic to the address range will enter our network.

To minimize down time, you can configure your AWS resources to use an address from your address pool before it is advertised, and then simultaneously stop advertising it from the current location and start advertising it through AWS. For more information about allocating an Elastic IP address from your address pool, see [Allocate an Elastic IP address](#).

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.

Work with your address range

You can view and use the IPv4 and IPv6 address ranges that you've provisioned in your account.

IPv4 address ranges

You can create an Elastic IP address from your IPv4 address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers.

To view information about the IPv4 address pools that you've provisioned in your account, use the following [describe-public-ipv4-pools](#) command.

```
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.

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
wwKQ11PSVAgRGVtbzETMBEGA1UEAwwKQ11PSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjE5MDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvaWwvYXNjaWwv
Yydm1jZXMxEzARBgNVBAcMCKJZT01QIERlbW8xEzARBgNVBAMMCKJZT01QIERlb
W8wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqur9wXkfNanAeskgAseyFypwEEQr4CJijI/5hp9
prh+jswHwWkFRoBRR9FBtwcU/45XDXLga7D3stsI5QeshVRw0aXUdprAnndaTug
mDPkD0vr1475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wX1AqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnViC7NqnhdEiW48QaYjhM1UEf
xdaqYUinz8KpjfADZ4Hvqj9jWZ/eXo/9b2rG1HWkJsbnr0VEUyAGu1bwkgcdww
```

```
3A7Nj0xQbAgMBAAGjUzBRMB0GA1UdDgQWBBSstFyujN6SYBr2g1HpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2g1HpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIB3DQEBCwJAA4IBAQBx6nn6YLhz5211fyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0c1r00aFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVz0yt6fPzj0vJLcsqc1hcF9wySL507XQz76Uk5cFypB0zbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRD1/qd0/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIIsoNPYqrJRzqFU9F3FBjiPJF
-----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
d1YiBTZXJ2aWw1czETMBEGA1UECwwKQ11PSVAgRGVtbzETMBEGA1UEAwwKQ11PS
VAgRGVtbzAeFw0yMTEyMDcyMDI0NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNV
BAYTAk5aMREwDwYDVQQIDAhBdWNrbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDA
aBgNVBAoME0FtYXpvaWw1bXZlZG90YXN0ZXQvZD1/cj8lTHoar17Uo9y/Q5qJIIso
NPYqrJRzqFU9F3FBjiPJFgEKAoIBAQCfmacvDp0wZ0ceiXXcR/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanA
EskgAseyFypwEEQr4CJijI/5hp9prh+jswHwWkFRoBRR9FBtwcU/45XDxLga7D3
stsI5QesHVRw0aXUdprAnndaTugmDPkD0vr1475JWDSIm+PUxGWLy+60aBqiaZq
35wU/x+wX1AqBXg4MZK2KoUu27kYt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp
1ZnVIc7NqnhdEiW48QaYjhm1UEfxdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2r
G1HWkJsbnr0VEUyAGu1bwkgcdww3A7Nj0xQbAgMBAAGjUzBRMB0GA1UdDgQWBBS
tFyujN6SYBr2g1HpGt0XGF7GbGTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIB3DQE
BCwJAA4IBAQBx6nn6YLhz5211fyVfxY0t6o3410bQAeAF08ud+ICtmQ4IO4A4B7z
V3zIVYr0c1r00aFyLxngwMYN0XY5tVhDQqk4/gmDNEKSZy2QkX4Eg0YUWVz0yt6f
Pzj0vJLcsqc1hcF9wySL507XQz76Uk5cFypB0zbnk35UkWrzA9KK97cXckfIESgK/
k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4ydjgH/LBfdTsVarmmy2vtWBx
wrqkFvpdhSGC
```

```
vRD1/qd0/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
LBQAwEzELMAkGA1UEBhMCTloXETAPBgNVBAGMCEY1Y2tsYW5kMREwDwYDVQQHDA
hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlcjETMBEGA1UEC
wwKQ1lPSVAgRGVtbzETMBEGA1UEAwwKQ1lPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjE0MDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvaW5kZXZlZG
VydmljZXNlZARBgNVBAsMCKJZT01QIERlbW8xEzARBgNVBAMMCKJZT01QIERlb
W8wgGElMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqur9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRw0aXUdprAnndaTug
mDPkD0vr1475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRj9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhM1UEf
xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rG1HWkJsbhr0VEUyAGu1bwkgcdww
3A7Nj0xQbAgMBAAGjUzBRMB0GA1UdDgQWBBSstFyujN6SYBr2g1HpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2g1HpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBcQUAA4IBAQBx6nn6YLhZ5211fyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0c1r00aFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVz0yt6fPzj0vJLcsqc1hcF9wySL507XQz76Uk5cFypB0zbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvphdSGCvRD1/qd0/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.

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:

- us-east-1-dfw-2
- us-west-2-lax-1
- us-west-2-phx-2

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.

Learn more

For more information, see the AWS Online Tech talk [Deep Dive on Bring Your Own IP](#).

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, or [Set up dynamic DNS on your Amazon Linux instance](#).

An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet. For example, this allows you to connect to your instance from your local computer.

Contents

- [Elastic IP address pricing](#)
- [Elastic IP address basics](#)
- [Work with Elastic IP addresses](#)
- [Elastic IP address quota](#)

Elastic IP address pricing

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.
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance, it is also associated with the instance's primary network interface. When you associate an Elastic IP address with a network interface that is attached to an instance, it is also associated with the instance.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see [Public IPv4 addresses](#).
- 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.

Work with Elastic IP addresses

The following sections describe how you can work with Elastic IP addresses.

Tasks

- [Allocate an Elastic IP address](#)
- [Describe your Elastic IP addresses](#)
- [Tag an Elastic IP address](#)
- [Associate an Elastic IP address with an instance or network interface](#)
- [Disassociate an Elastic IP address](#)
- [Transfer Elastic IP addresses](#)
- [Release an Elastic IP address](#)
- [Recover an Elastic IP address](#)
- [Use reverse DNS for email applications](#)

Allocate an Elastic IP address

You can allocate an Elastic IP address from Amazon's pool of public IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account. For more information about bringing your own IP address range to your AWS account, see [Bring your own IP addresses \(BYOIP\) in Amazon EC2](#).

You can allocate an Elastic IP address using one of the following methods.

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 (see [Local Zones](#)). 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 an IPv4 address from an IP address pool that you have brought to your AWS account. This option is disabled if you do not have any IP address pools.
- **Customer owned pool of IPv4 addresses**—If you want to allocate an IPv4 address from a pool created from your on-premises network for use with an AWS Outpost. This option is disabled if you do not have an AWS Outpost.

6. (Optional) Add or remove a tag.

[Add a tag] Choose **Add new tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Choose **Remove** to the right of the tag's Key and Value.

7. Choose **Allocate**.

AWS CLI

To allocate an Elastic IP address

Use the [allocate-address](#) AWS CLI command.

PowerShell

To allocate an Elastic IP address

Use the [New-EC2Address](#) AWS Tools for Windows PowerShell command.

Describe your Elastic IP addresses

You can describe an Elastic IP address using one of the following methods.

Console

To describe your Elastic IP addresses

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to view and choose **Actions, View details**.

AWS CLI

To describe your Elastic IP addresses

Use the [describe-addresses](#) AWS CLI command.

PowerShell

To describe your Elastic IP addresses

Use the [Get-EC2Address](#) AWS Tools for Windows PowerShell command.

Tag an Elastic IP address

You can assign custom tags to your Elastic IP addresses to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Elastic IP address based on the custom tags that you assigned to it.

Cost allocation tracking using Elastic IP address tags is not supported.

You can tag an Elastic IP address using one of the following methods.

Console

To tag an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to tag and choose **Actions, View details**.
4. In the **Tags** section, choose **Manage tags**.
5. Specify a tag key and value pair.
6. (Optional) Choose **Add tag** to add additional tags.

7. Choose **Save**.

AWS CLI

To tag an Elastic IP address

Use the [create-tags](#) AWS CLI command.

```
aws ec2 create-tags --resources eipalloc-12345678 --tags Key=Owner,Value=TeamA
```

PowerShell

To tag an Elastic IP address

Use the [New-EC2Tag](#) AWS Tools for Windows PowerShell command.

The `New-EC2Tag` command needs a `Tag` parameter, which specifies the key and value pair to be used for the Elastic IP address tag. The following commands create the `Tag` parameter.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource eipalloc-12345678 -Tag $tag
```

Associate an Elastic IP address with an instance or network interface

If you're associating an Elastic IP address with your instance to enable communication with the internet, you must also ensure that your instance is in a public subnet. For more information, see [Internet gateways](#) in the *Amazon VPC User Guide*.

You can associate an Elastic IP address with an instance or network interface using one of the following methods.

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.

PowerShell

To associate an Elastic IP address

Use the [Register-EC2Address](#) AWS Tools for Windows PowerShell command.

Disassociate an Elastic IP address

You can disassociate an Elastic IP address from an instance or network interface at any time. After you disassociate the Elastic IP address, you can reassociate it with another resource.

You can disassociate an Elastic IP address using one of the following methods.

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.

PowerShell

To disassociate an Elastic IP address

Use the [Unregister-EC2Address](#) AWS Tools for Windows PowerShell command.

Transfer Elastic IP addresses

This section describes how to transfer Elastic IP addresses from one AWS account to another. Transferring Elastic IP addresses can be helpful in the following situations:

- **Organizational restructuring** – Use Elastic IP address transfers to quickly move workloads from one AWS account to another. You don't have to wait for new Elastic IP addresses to be allowlisted in your security groups and NACLs.
- **Centralized security administration** – Use a centralized AWS security account to track and transfer Elastic IP addresses that have been vetted for security compliance.
- **Disaster recovery** – Use Elastic IP address transfers to quickly remap IPs for public-facing internet workloads during emergency events.

There is no charge for transferring Elastic IP addresses.

Tasks

- [Enable Elastic IP address transfer](#)
- [Disable Elastic IP address transfer](#)
- [Accept a transferred Elastic IP address](#)

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).
- 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](#) AWS CLI 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](#) AWS CLI command) for three 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.
- 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.

PowerShell

To enable Elastic IP address transfer

Use the [Enable-EC2AddressTransfer](#) command.

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.

PowerShell

To disable Elastic IP address transfer

Use the [Disable-EC2AddressTransfer](#) command.

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](#) AWS CLI 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 [Use reverse DNS for email applications](#).
- **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 AWS 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.

PowerShell

To accept an Elastic IP address transfer

Use the [Approve-EC2AddressTransfer](#) command.

Release an Elastic IP address

If you no longer need an Elastic IP address, we recommend that you release it using one of the following methods. The address to release must not be currently associated with an AWS resource, such as an EC2 instance, NAT gateway, or Network Load Balancer.

Note

If you contacted AWS support to set up reverse DNS for an Elastic IP (EIP) address, you can remove the reverse DNS, but you can't release the Elastic IP address because it's been locked by AWS support. To unlock the Elastic IP address, contact [AWS Support](#). Once the Elastic IP address is unlocked, you can release the Elastic IP address.

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.

PowerShell

To release an Elastic IP address

Use the [Remove-EC2Address](#) AWS Tools for Windows PowerShell command.

Recover an Elastic IP address

If you have released your Elastic IP address, you might be able to recover it. The following rules apply:

- You cannot recover an Elastic IP address if it has been allocated to another AWS account, or if it will result in your exceeding your Elastic IP address limit.
- You cannot recover tags associated with an Elastic IP address.
- You can recover an Elastic IP address using the Amazon EC2 API or a command line tool only.

AWS CLI

To recover an Elastic IP address

Use the [allocate-address](#) AWS CLI command and specify the IP address using the `--address` parameter as follows.

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

PowerShell

To recover an Elastic IP address

Use the [New-EC2Address](#) AWS Tools for Windows PowerShell command and specify the IP address using the `-Address` parameter as follows.

```
PS C:\> New-EC2Address -Address 203.0.113.3 -Domain vpc -Region us-east-1
```

Use reverse DNS for email applications

If you intend to send email to third parties from an instance, we recommend that you provision one or more Elastic IP addresses and assign static reverse DNS records to the Elastic IP addresses that you use to send email. This can help you avoid having your email flagged as spam by some anti-spam organizations. 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.
- **AWS GovCloud (US) Region**

You can't create a reverse DNS record using the console or AWS CLI. AWS must assign the static reverse DNS records for you. Open [Request to remove reverse DNS and email sending limitations](#) and provide us with your Elastic IP addresses and reverse DNS records.

Create a reverse DNS record

To create a reverse DNS record, choose the tab that matches your preferred method.

Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address and choose **Actions, Update reverse DNS**.

4. For **Reverse DNS domain name**, enter the domain name.
5. Enter **update** to confirm.
6. Choose **Update**.

AWS CLI

Use the [modify-address-attribute](#) command in the AWS CLI, as shown in the following example:

```
aws ec2 modify-address-attribute --allocation-id eipalloc-abcdef01234567890 --  
domain-name example.com  
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.net."  
      "PtrRecordUpdate": {  
        "Value": "example.com.",  
        "Status": "PENDING"  
      }  
    }  
  ]  
}
```

Remove a reverse DNS record

To remove a reverse DNS record, choose the tab that matches your preferred method.

Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address and choose **Actions, Update reverse DNS**.
4. For **Reverse DNS domain name**, clear the domain name.
5. Enter **update** to confirm.
6. Choose **Update**.

AWS CLI

Use the [reset-address-attribute](#) command in the AWS CLI, as shown in the following example:

```
aws ec2 reset-address-attribute --allocation-id eipalloc-abcdef01234567890 --
attribute domain-name
{
  "Addresses": [
    {
      "PublicIp": "192.0.2.0",
      "AllocationId": "eipalloc-abcdef01234567890",
      "PtrRecord": "example.com."
      "PtrRecordUpdate": {
        "Value": "example.net.",
        "Status": "PENDING"
      }
    }
  ]
}
```

Note

If you receive the following error when you run the command, you can submit a [Request to remove email sending limitations](#) to AWS Support for assistance.

The address with allocation id cannot be released because it is locked to your account.

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 encourage you to use an Elastic IP address primarily for the ability to remap the address to another instance in the case of instance failure, and to use [DNS hostnames](#) for all other inter-node communication.

To verify how many Elastic IP addresses are in use

Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> and choose **Elastic IPs** from the navigation pane.

To verify your current account quota for Elastic IP addresses

1. Open the Service Quotas console at <https://console.aws.amazon.com/servicequotas/>.

2. From the navigation bar (at the top of the screen), select a Region.
3. On the Dashboard, choose **Amazon Elastic Compute Cloud (Amazon EC2)**.

If Amazon Elastic Compute Cloud (Amazon EC2) is not listed on the Dashboard, choose **AWS services**, enter **EC2** in the search field, and then choose **Amazon Elastic Compute Cloud (Amazon EC2)**.

4. On the Amazon EC2 service quotas page, enter **IP** in the search field. The limit is **EC2-VPC Elastic IPs**. For more information, choose the limit.

If you think your architecture warrants additional Elastic IP addresses, you can request a quota increase directly from the Service Quotas console. To request a quota increase, choose **Request increase at account-level**. For more information, see [Amazon EC2 service quotas](#).

Elastic network interfaces

An *elastic network interface* is a logical networking component in a VPC that represents a virtual network card. It can include the following attributes:

- A primary private IPv4 address from the IPv4 address range of your VPC
- A primary IPv6 address from the IPv6 address range of your VPC
- One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
- One Elastic IP address (IPv4) per private IPv4 address
- One public IPv4 address
- One or more IPv6 addresses
- One or more security groups
- A MAC address
- A source/destination check flag
- A description

You can create and configure network interfaces and attach them to instances in the same Availability Zone. Your account might also have *requester-managed* network interfaces, which are created and managed by AWS services to enable you to use other resources and services. You cannot manage these network interfaces yourself. For more information, see [Requester-managed network interfaces](#).

This AWS resource is referred to as a *network interface* in the AWS Management Console and the Amazon EC2 API. Therefore, we use "network interface" in this documentation instead of "elastic network interface". The term "network interface" in this documentation always means "elastic network interface".

Contents

- [Network interface basics](#)
- [Network cards](#)
- [IP addresses per network interface per instance type](#)
- [Work with network interfaces](#)
- [Best practices for configuring network interfaces](#)
- [Scenarios for network interfaces](#)
- [Requester-managed network interfaces](#)
- [Assign prefixes to Amazon EC2 network interfaces](#)

Network interface basics

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Primary network interface

Each instance has a default network interface, called the *primary network interface*. You cannot detach a primary network interface from an instance. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see [IP addresses per network interface per instance type](#).

Public IPv4 addresses for network interfaces

In a VPC, all subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are assigned a public IPv4 address. For more information, see [Subnet settings](#) in the *Amazon VPC User Guide*. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses. When you launch an instance, the IP address is assigned to the primary network interface that's created.

When you create a network interface, it inherits the public IPv4 addressing attribute from the subnet. If you later modify the public IPv4 addressing attribute of the subnet, the network interface keeps the setting that was in effect when it was created. If you launch an instance and specify an existing network interface as the primary network interface, the public IPv4 address attribute is determined by this network interface.

For more information, see [Public IPv4 addresses](#).

Elastic IP addresses for network interface

If you have an Elastic IP address, you can associate it with one of the private IPv4 addresses for the network interface. You can associate one Elastic IP address with each private IPv4 address.

If you disassociate an Elastic IP address from a network interface, you can release it back to the address pool. This is the only way to associate an Elastic IP address with an instance in a different subnet or VPC, as network interfaces are specific to subnets.

IPv6 addresses for network interfaces

If you associate IPv6 CIDR blocks with your VPC and subnet, you can assign one or more IPv6 addresses from the subnet range to a network interface. Each IPv6 address can be assigned to one network interface.

All subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are automatically assigned an IPv6 address from the range of the subnet. For more information, see [Subnet settings](#) in the *Amazon VPC User Guide*. When you launch an instance, the IPv6 address is assigned to the primary network interface that's created.

For more information, see [IPv6 addresses](#).

Prefix Delegation

A Prefix Delegation prefix is a reserved private IPv4 or IPv6 CIDR range that you allocate for automatic or manual assignment to network interfaces that are associated with an instance. By using Delegated Prefixes, you can launch services faster by assigning a range of IP addresses as a single prefix.

Termination behavior

You can set the termination behavior for a network interface that's attached to an instance. You can specify whether the network interface should be automatically deleted when you terminate the instance to which it's attached.

Source/destination checking

You can enable or disable source/destination checks, which ensure that the instance is either the source or the destination of any traffic that it receives. Source/destination checks are enabled by default. You must disable source/destination checks if the instance runs services such as network address translation, routing, or firewalls.

Monitoring IP traffic

You can enable a VPC flow log on your network interface to capture information about the IP traffic going to and from a network interface. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs. For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

Automatic public IPv4 address assignment

You can enable and disable automatic assignment of a public IPv4 address to a network interface. This option can be enabled for any network interface but will only apply to the primary network interface (eth0). For more information, see [Manage IP addresses](#).

Network cards

Instances with multiple network cards provide higher network performance, including bandwidth capabilities above 100 Gbps and improved packet rate performance. Each network interface is attached to a network card. The primary network interface must be assigned to network card index 0.

If you enable Elastic Fabric Adapter (EFA) when you launch an instance that supports multiple network cards, all network cards are available. You can assign up to one EFA per network card. An EFA counts as a network interface.

The following instances support multiple network cards. All other instance types support one network card.

Instance type	Number of network cards
c6in.32xlarge	2

Instance type	Number of network cards
c6in.metal	2
d11.24xlarge	4
hpc6id.32xlarge	2
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
r6idn.32xlarge	2
r6idn.metal	2
r6in.32xlarge	2
r6in.metal	2
trn1.32xlarge	8
trn1n.32xlarge	16

IP addresses per network interface per instance type

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. Not all instance types support IPv6 addressing.

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](#)

To retrieve network interface information using the AWS CLI

You can use the [describe-instance-types](#) AWS CLI command to display information about an instance type, such as its 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
```

```
-----
|           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	
+-----+	+-----+	+-----+	+-----+

Work with network interfaces

You can work with network interfaces using the Amazon EC2 console or the command line.

Contents

- [Create a network interface](#)
- [View details about a network interface](#)
- [Attach a network interface to an instance](#)
- [Detach a network interface from an instance](#)
- [Manage IP addresses](#)
- [Modify network interface attributes](#)
- [Add or edit tags](#)
- [Delete a network interface](#)

Create a network interface

You can create a network interface in a subnet. You can't move the network interface to another subnet after it's created. You must attach a network interface to an instance in the same Availability Zone.

To create a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create network interface**.
4. (Optional) For **Description**, enter a descriptive name.
5. For **Subnet**, select a subnet. 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 **Private IPv4 address**, do one of the following:

- Choose **Auto-assign** to allow Amazon EC2 to select an IPv4 address from the subnet.
 - Choose **Custom** and enter an IPv4 address that you select from the subnet.
7. (Subnets with IPv6 addresses only) For **IPv6 address**, do one of the following:
 - Choose **None** if you do not want to assign an IPv6 address to the network interface.
 - Choose **Auto-assign** to allow Amazon EC2 to select an IPv6 address from the subnet.
 - Choose **Custom** and enter an IPv6 address that you select from the subnet.
 8. (Optional) 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 cannot 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.
 9. (Optional) To create an Elastic Fabric Adapter, choose **Elastic Fabric Adapter, Enable**.
 10. (Optional) Under **Advanced settings**, for **Idle connection tracking timeout**, modify the default idle connection timeouts. For more information about these options, 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 (5 days). Default: 432000 seconds. 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 (3 minutes). Default: 180 seconds.
 11. For **Security groups**, select one or more security groups.
 12. (Optional) For each tag, choose **Add new tag** and enter a tag key and an optional tag value.

13. Choose **Create network interface**.

To create a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [create-network-interface](#) (AWS CLI)
- [New-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

View details about a network interface

You can view all the network interfaces in your account.

To describe a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. To view the details page for a network interface, select the ID of the network interface. Alternatively, to view information without leaving the network interfaces page, select the checkbox for the network interface.

To describe a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [describe-network-interfaces](#) (AWS CLI)
- [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

To describe a network interface attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [describe-network-interface-attribute](#) (AWS CLI)
- [Get-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Attach a network interface to an instance

You can attach a network interface to any instance in the same Availability Zone as the network interface, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console. Alternatively, you can specify existing network interfaces when you [launch instances](#).

Important

For EC2 instances in an IPv6-only subnet, if you attach a secondary network interface to the instance, the private DNS hostname of the second network interface will resolve to the first IPv6 address on the instance's first network interface. For more information about EC2 instance private DNS hostnames, see [Amazon EC2 instance hostname types](#).

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](#).

Instances page

To attach a network interface to an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the checkbox for the instance.
4. Choose **Actions, Networking, Attach network interface**.
5. Choose a VPC. If you are attaching a secondary network interface to the instance, 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 a subnet that is in the same Availability Zone as your 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**.

Network Interfaces page

To attach a network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. Choose **Actions, Attach**.
5. Choose an instance. If the instance supports multiple network cards, you can choose a network card.
6. Choose **Attach**.

To attach a network interface to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

Note

You can attach a network interface that's in another VPC (but in the same Availability Zone) to an instance using the [attach-network-interface](#) AWS CLI command. You cannot do this using the AWS Management Console.

- [attach-network-interface](#) (AWS CLI)
- [Add-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Detach a network interface from an instance

You can detach a secondary network interface that is attached to an EC2 instance at any time, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console.

If you try to detach a network interface that is attached to a resource from another service, such as an Elastic Load Balancing load balancer, a Lambda function, a WorkSpace, or a NAT gateway, you get an error that you do not have permission to access the resource. To find which service created the resource attached to a network interface, check the description of the network interface. If you delete the resource, then its network interface is deleted.

Instances page

To detach a network interface from an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the checkbox for the instance. Check the **Network interfaces** section of the **Networking** tab to verify that the network interface is attached to an instance as a secondary network interface.
4. Choose **Actions, Networking, Detach network interface**.
5. Select the network interface and choose **Detach**.

Network Interfaces page

To detach a network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface. Check the **Instance details** section of the **Details** tab to verify that the network interface is attached to an instance as a secondary network interface.
4. Choose **Actions, Detach**.
5. When prompted for confirmation, choose **Detach**.
6. If the network interface fails to detach from the instance, choose **Force detachment, Enable** and then try again. We recommend that force detachment only as a last resort. Forcing a detachment can prevent you from attaching a different network interface on the same index until you restart the instance. It can also prevent the instance metadata from reflecting that the network interface was detached until you restart the instance.

To detach a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [detach-network-interface](#) (AWS CLI)
- [Dismount-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Manage IP addresses

You can manage the following IP addresses for your network interfaces:

- Elastic IP addresses (one per private IPv4 address)
- IPv4 addresses
- IPv6 addresses
- Primary IPv6 address

To manage the Elastic IP addresses of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. To associate an Elastic IP address, do the following:
 - a. Choose **Actions, Associate address**.
 - b. For **Elastic IP address**, select the Elastic IP address.
 - c. For **Private IPv4 address**, select the private IPv4 address to associate with the Elastic IP address.
 - d. (Optional) Choose **Allow the Elastic IP address to be reassociated** if the network interface is currently associated with another instance or network interface.
 - e. Choose **Associate**.
5. To disassociate an Elastic IP address, do the following:
 - a. Choose **Actions, Disassociate address**.
 - b. For **Public IP address**, select the Elastic IP address.
 - c. Choose **Disassociate**.

To manage the IPv4 and IPv6 addresses of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.

4. Choose **Actions, Manage IP addresses**.
5. Expand the network interface.
6. For **IPv4 addresses**, modify the IP addresses as needed. To assign an IPv4 address, choose **Assign new IP address** and then specify an IPv4 address from the subnet range or let AWS choose one for you. To unassign an IPv4 address, choose **Unassign** next to the address.
7. To assign a public IPv4 address to a network interface, choose **Auto-assign public IP**. This option can be enabled for any network interface but will only apply to the primary network interface (eth0).
8. For **IPv6 addresses**, modify the IP addresses as needed. To assign an IPv6 address, choose **Assign new IP address** and then specify an IPv6 address from the subnet range or let AWS choose one for you. To unassign an IPv6 address, choose **Unassign** next to the address.
9. (Optional) If your modifying a network interface in a dual-stack or IPv6-only subnet, you have the option to **Assign Primary IPv6 IP**. 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 cannot 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. Choose **Save**.

To manage the IP addresses of a network interface using the AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [assign-ipv6-addresses](#)
- [associate-address](#)
- [disassociate-address](#)
- [unassign-ipv6-addresses](#)

To manage the IP addresses of a network interface using the Tools for Windows PowerShell

You can use one of the following commands.

- [Register-EC2Address](#)
- [Register-EC2Ipv6AddressList](#)
- [Unregister-EC2Address](#)
- [Unregister-EC2Ipv6AddressList](#)

Modify network interface attributes

You can change the following network interface attributes:

- [Description](#)
- [Security groups](#)
- [Delete on termination](#)
- [Source/destination check](#)

To change the description of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. Choose **Actions, Change description**.
5. For **Description**, enter a description for the network interface.
6. Choose **Save**.

To change the security groups of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.

4. Choose **Actions, Change security groups**.
5. For **Associated security groups**, select the security groups to use, and then choose **Save**.

The security group and network interface must be created for the same VPC. To change the security group for interfaces owned by other services, such as Elastic Load Balancing, do so through that service.

To change the termination behavior of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. Choose **Actions, Change termination behavior**.
5. Select or clear **Delete on termination, Enable** as needed, and then choose **Save**.

To change source/destination checking for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. Choose **Actions, Change source/dest check**.
5. Select or clear **Source/destination check, Enable** as needed, and then choose **Save**.

To change idle connection tracking timeouts:

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, Modify connection timeout**.
5. Modify the idle connection tracking timeouts. For more information about these options, 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 (5 days). Default: 432000 seconds. 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 (3 minutes). Default: 180 seconds.

6. Choose **Save**.

To modify network interface attributes using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Add or edit tags

Tags are metadata that you can add to a network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see [Tag your Amazon EC2 resources](#).

To add or edit tags for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. In **Tags** tab, choose **Manage tags**.
5. For each tag to create, choose **Add new tag** and enter a key and optional value. When you're done, choose **Save**.

To add or edit tags for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [create-tags](#) (AWS CLI)
- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

Delete a network interface

Deleting a network interface releases all attributes associated with the interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

You cannot delete a network interface that is in use. First, you must [detach the network interface](#).

To delete a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface, and then choose **Actions, Delete**.
4. When prompted for confirmation, choose **Delete**.

To delete a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [delete-network-interface](#) (AWS CLI)
- [Remove-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Best practices for configuring network interfaces

- You can attach a network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary network interfaces when the instance is running or stopped. However, you can't detach the primary network interface.

- You can move a secondary network interface from one instance to another, if the instances are in the same Availability Zone and VPC but in different subnets.
- When launching an instance using the CLI, API, or an SDK, you can specify the primary network interface and additional network interfaces.
- Launching an Amazon Linux or Windows Server instance with multiple network interfaces automatically configures interfaces, private IPv4 addresses, and route tables on the operating system of the instance.
- A warm or hot attach of an additional network interface might require you to manually bring up the second interface, configure the private IPv4 address, and modify the route table accordingly. Instances running Amazon Linux or Windows Server automatically recognize the warm or hot attach and configure themselves.
- You cannot attach another network interface to an instance (for example, a NIC teaming configuration) to increase or double the network bandwidth to or from the dual-homed instance.
- If you attach two or more network interfaces from the same subnet to an instance, you might encounter networking issues such as asymmetric routing. If possible, use a secondary private IPv4 address on the primary network interface instead.

Configure your network interface using `ec2-net-utils` for Amazon Linux 2

Note

For AL2023, the `amazon-ec2-net-utils` package generates interface-specific configurations in the `/run/systemd/network` directory. For more information, see [Networking service](#) in the *Amazon Linux 2023 User Guide*.

Amazon Linux 2 AMIs may contain additional scripts installed by AWS, known as `ec2-net-utils`. These scripts optionally automate the configuration of your network interfaces. These scripts are available for Amazon Linux 2 only.

Use the following command to install the package on Amazon Linux 2 if it's not already installed, or update it if it's installed and additional updates are available:

```
$ yum install ec2-net-utils
```

The following components are part of `ec2-net-utils`:

udev rules (/etc/udev/rules.d)

Identifies network interfaces when they are attached, detached, or reattached to a running instance, and ensures that the hotplug script runs (53-ec2-network-interfaces.rules). Maps the MAC address to a device name (75-persistent-net-generator.rules, which generates 70-persistent-net.rules).

hotplug script

Generates an interface configuration file suitable for use with DHCP (/etc/sysconfig/network-scripts/ifcfg-ethN). Also generates a route configuration file (/etc/sysconfig/network-scripts/route-ethN).

DHCP script

Whenever the network interface receives a new DHCP lease, this script queries the instance metadata for Elastic IP addresses. For each Elastic IP address, it adds a rule to the routing policy database to ensure that outbound traffic from that address uses the correct network interface. It also adds each private IP address to the network interface as a secondary address.

ec2ifup ethN (/usr/sbin/)

Extends the functionality of the standard **ifup**. After this script rewrites the configuration files `ifcfg-ethN` and `route-ethN`, it runs **ifup**.

ec2ifdown ethN (/usr/sbin/)

Extends the functionality of the standard **ifdown**. After this script removes any rules for the network interface from the routing policy database, it runs **ifdown**.

ec2ifscan (/usr/sbin/)

Checks for network interfaces that have not been configured and configures them.

This script isn't available in the initial release of `ec2-net-utils`.

To list any configuration files that were generated by `ec2-net-utils`, use the following command:

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

To disable the automation, you can add `EC2SYNC=no` to the corresponding `ifcfg-ethN` file. For example, use the following command to disable the automation for the `eth1` interface:

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

To disable the automation completely, you can remove the package using the following command:

```
$ yum remove ec2-net-utils
```

Scenarios for network interfaces

Attaching multiple network interfaces to an instance is useful when you want to:

- Create a management network.
- Use network and security appliances in your Virtual Private Cloud (VPC).
- Create dual-homed instances with workloads/roles on distinct subnets.
- Create a low-budget, high-availability solution.

Create a management network

This scenario describes how you can create a management network with network interfaces, given the following criteria and settings (image follows).

Criteria

- The primary network interface on the instance (*eth0*) handles public traffic.
- The secondary network interface on the instance (*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 (AZ) as the primary network interface.

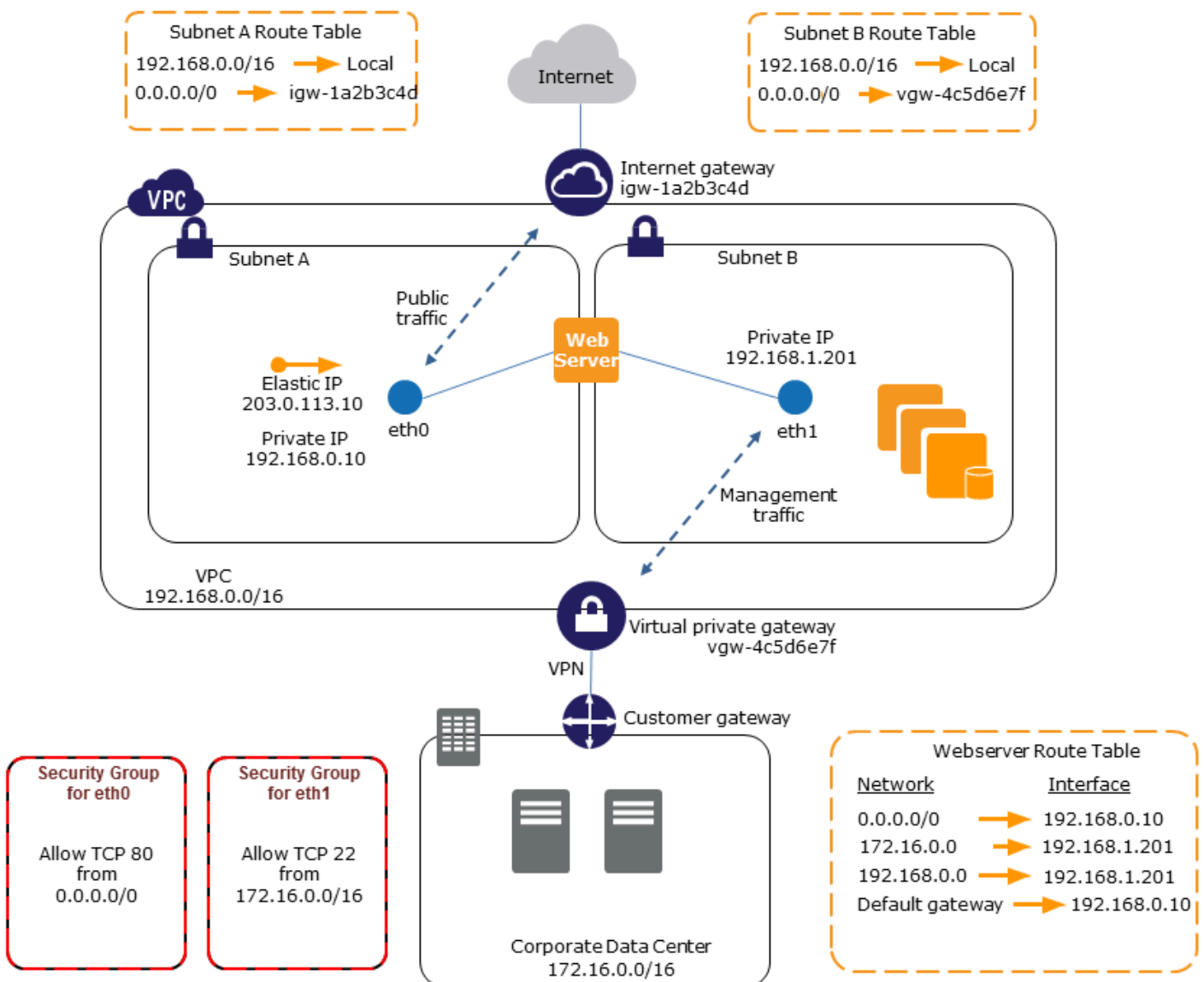
Settings

- 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 AZ 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.



Use network and security appliances in your VPC

Some network and security appliances, such as load balancers, network address translation (NAT) servers, and proxy servers prefer to be configured with multiple network interfaces. You can create and attach secondary network interfaces to instances that are running these types of applications and configure the additional interfaces with their own public and private IP addresses, security groups, and source/destination checking.

Creating dual-homed instances with workloads/roles on distinct subnets

You can place a network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a backend network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the backend, and then sends requests to the servers on the backend network.

Creating dual-homed instances with workloads/roles on distinct VPCs within the same account

You can launch an EC2 instance in one VPC and attach a secondary ENI from another VPC (but in the same Availability Zone) to the instance. This enables you to create multi-homed instances across VPCs with different networking and security configurations. You cannot create multi-homed instances across VPCs across 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.

Create a low budget high availability solution

If one of your instances serving a particular function fails, its network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use a network interface as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you

(or more likely, the code running on your behalf) can attach the network interface to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic begins flowing to the standby instance as soon as you attach the network interface to the replacement instance. Users experience a brief loss of connectivity between the time the instance fails and the time that the network interface is attached to the standby instance, but no changes to the route table or your DNS server are required.

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.

To view requester-managed network interfaces using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network & 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.

To view requester-managed network interfaces using the AWS CLI

Use the [describe-network-interfaces](#) command as follows.

```
aws ec2 describe-network-interfaces --filters Name=requester-managed,Values=true
```

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`.

```
{
  ...
  "Description": "VPC Endpoint Interface vpce-089f2123488812123",
  ...
  "InterfaceType": "vpc_endpoint",
  ...
  "NetworkInterfaceId": "eni-0d11e3ccd2c0e6c57",
  ...
  "RequesterId": "727180483921",
  "RequesterManaged": true,
  ...
}
```

To view requester-managed network interfaces using the Tools for Windows PowerShell

Use the [Get-EC2NetworkInterface](#) cmdlet as follows.

```
Get-EC2NetworkInterface -Filter @{ Name="requester-managed"; Values="true" }
```

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`.

```
Description      : VPC Endpoint Interface vpce-089f2123488812123
...
InterfaceType    : vpc_endpoint
...
NetworkInterfaceId : eni-0d11e3ccd2c0e6c57
...
RequesterId      : 727180483921
RequesterManaged : True
...
```

Assign prefixes to Amazon EC2 network interfaces

You can assign a private IPv4 or IPv6 CIDR range, either automatically or manually, to your network interfaces. By assigning prefixes, you scale and simplify the management of applications, including container and networking applications that require multiple IP addresses on an instance. 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 from your VPC subnet's IPv4 or IPv6 CIDR block and assigns it to your network interface.
- **Manual Assignment** — You specify the prefix from your VPC subnet's IPv4 or IPv6 CIDR block, and AWS verifies that the prefix is not already assigned to other resources before assigning it to your network interface.

Assigning prefixes has the following benefits:

- **Increased IP addresses on a network interface** — When you use a prefix, you assign a block of IP addresses as opposed to individual IP addresses. This increases the number of IP addresses 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 for assigning prefixes](#)
- [Considerations and limits for prefixes](#)
- [Work with prefixes](#)

Basics for assigning prefixes

- 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 and limits for prefixes

Take the following into consideration when you use prefixes:

- Network interfaces with prefixes are supported with [instances built on the AWS Nitro System](#).
- Prefixes for network interfaces are limited to IPv6 addresses and 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 [IP addresses per network interface per instance type](#).
- Prefixes are included in source/destination checks.

Work with prefixes

You can use prefixes with your network interfaces as follows.

Tasks

- [Assign prefixes during network interface creation](#)
- [Assign prefixes to existing network interfaces](#)
- [Configure your operating system for network interfaces with prefixes](#)
- [View the prefixes assigned to your network interfaces](#)
- [Remove prefixes from your network interfaces](#)

Assign prefixes during network interface creation

If you use the automatic assignment option, you can reserve a block of IP addresses in your subnet. AWS chooses the prefixes from this block. For more information, see [Subnet CIDR reservations](#) in the *Amazon VPC User Guide*.

After you have created the network interface, use the [attach-network-interface](#) AWS CLI command to attach the network interface to your instance. You must configure your operating system to work with network interfaces with prefixes. For more information, see [Configure your operating system for network interfaces with prefixes](#).

Tasks

- [Assign automatic prefixes during network interface creation](#)
- [Assign specific prefixes during network interface creation](#)

Assign automatic prefixes during network interface creation

You can assign automatic prefixes during network interface creation using one of the following methods.

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** and then choose **Create network interface**.

3. Specify a description for the network interface, select the subnet in which to create the network interface, and configure the private IPv4 and IPv6 addresses.
4. Expand **Advanced settings** and do the following:
 - a. To automatically assign an IPv4 prefix, for **IPv4 prefix delegation**, choose **Auto-assign**. Then for **Number of IPv4 prefixes**, specify the number of prefixes to assign.
 - b. To automatically assign an IPv6 prefix, for **IPv6 prefix delegation**, choose **Auto-assign**. Then for **Number of IPv6 prefixes**, specify the number of prefixes to assign.

 **Note**

IPv6 prefix delegation appears only if the selected subnet is enabled for IPv6.

5. Select the security groups to associate with the network interface and assign resource tags if needed.
6. 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 prefixes that you want AWS to assign. In the following example, AWS assigns 1 prefix.

```
$ aws ec2 create-network-interface \  
--subnet-id subnet-047cfed18eEXAMPLE \  
--description "IPv4 automatic example" \  
--ipv4-prefix-count 1
```

Example output

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "IPv4 automatic example",  
    "Groups": [  
      {  
        "GroupName": "default",  
        "GroupId": "sg-044c2de2c4EXAMPLE"      }  
    ]  
  }  
}
```

```

    }
  ],
  "InterfaceType": "interface",
  "Ipv6Addresses": [],
  "MacAddress": "02:98:65:dd:18:47",
  "NetworkInterfaceId": "eni-02b80b4668EXAMPLE",
  "OwnerId": "123456789012",
  "PrivateIpAddress": "10.0.0.62",
  "PrivateIpAddresses": [
    {
      "Primary": true,
      "PrivateIpAddress": "10.0.0.62"
    }
  ],
  "Ipv4Prefixes": [
    {
      "Ipv4Prefix": "10.0.0.208/28"
    }
  ],
  "RequesterId": "AIDAIV5AJI5LXF5XXDPC0",
  "RequesterManaged": false,
  "SourceDestCheck": true,
  "Status": "pending",
  "SubnetId": "subnet-047cfed18eEXAMPLE",
  "TagSet": [],
  "VpcId": "vpc-0e12f52b21EXAMPLE"
}
}

```

To assign automatic IPv6 prefixes during network interface creation

Use the [create-network-interface](#) command and set `--ipv6-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 prefix.

```

$ aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv6 automatic example" \
--ipv6-prefix-count 1

```

Example output

```
{
```



```
"NetworkInterface": {
  "AvailabilityZone": "us-west-2a",
  "Description": "IPv6 automatic example",
  "Groups": [
    {
      "GroupName": "default",
      "GroupId": "sg-044c2de2c4EXAMPLE"
    }
  ],
  "InterfaceType": "interface",
  "Ipv6Addresses": [],
  "MacAddress": "02:bb:e4:31:fe:09",
  "NetworkInterfaceId": "eni-006edbcfa4EXAMPLE",
  "OwnerId": "123456789012",
  "PrivateIpAddress": "10.0.0.73",
  "PrivateIpAddresses": [
    {
      "Primary": true,
      "PrivateIpAddress": "10.0.0.73"
    }
  ],
  "Ipv6Prefixes": [
    {
      "Ipv6Prefix": "2600:1f13:fc2:a700:1768::/80"
    }
  ],
  "RequesterId": "AIDAIV5AJI5LXF5XXDPC0",
  "RequesterManaged": false,
  "SourceDestCheck": true,
  "Status": "pending",
  "SubnetId": "subnet-047cfed18eEXAMPLE",
  "TagSet": [],
  "VpcId": "vpc-0e12f52b21EXAMPLE"
}
```


Assign specific prefixes during network interface creation

You can assign specific prefixes during network interface creation using one of the following methods.

Console

To assign specific 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** and then choose **Create network interface**.
3. Specify a description for the network interface, select the subnet in which to create the network interface, and configure the private IPv4 and IPv6 addresses.
4. Expand **Advanced settings** and do the following:
 - a. To assign a specific IPv4 prefix, for **IPv4 prefix delegation**, choose **Custom**. Then choose **Add new prefix** and enter the prefix to use.
 - b. To assign a specific IPv6 prefix, for **IPv6 prefix delegation**, choose **Custom**. Then choose **Add new prefix** and enter the prefix to use.

 **Note**

IPv6 prefix delegation appears only if the selected subnet is enabled for IPv6.

5. Select the security groups to associate with the network interface and assign resource tags if needed.
6. Choose **Create network interface**.

AWS CLI

To assign specific IPv4 prefixes during network interface creation

Use the [create-network-interface](#) command and set `--ipv4-prefixes` to the prefixes. AWS selects IP addresses from this range. In the following example, the prefix CIDR is `10.0.0.208/28`.

```
$ aws ec2 create-network-interface \  
  --subnet-id subnet-047cfed18eEXAMPLE \  
  --description "IPv4 manual example" \  
  --ipv4-prefixes Ipv4Prefix=10.0.0.208/28
```

Example output

```
{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "Description": "IPv4 manual example",
    "Groups": [
      {
        "GroupName": "default",
        "GroupId": "sg-044c2de2c4EXAMPLE"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "02:98:65:dd:18:47",
    "NetworkInterfaceId": "eni-02b80b4668EXAMPLE",
    "OwnerId": "123456789012",
    "PrivateIpAddress": "10.0.0.62",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateIpAddress": "10.0.0.62"
      }
    ],
    "Ipv4Prefixes": [
      {
        "Ipv4Prefix": "10.0.0.208/28"
      }
    ],
    "RequesterId": "AIDAIV5AJI5LXF5XXDPC0",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-047cfed18eEXAMPLE",
    "TagSet": [],
    "VpcId": "vpc-0e12f52b21EXAMPLE"
  }
}
```

To assign specific IPv6 prefixes during network interface creation

Use the [create-network-interface](#) command and set `--ipv6-prefixes` to the prefixes. AWS selects IP addresses from this range. In the following example, the prefix CIDR is `2600:1f13:fc2:a700:1768::/80`.

```
$ aws ec2 create-network-interface \  
  --subnet-id subnet-047cfed18eEXAMPLE \  
  --description "IPv6 manual example" \  
  --ipv6-prefixes Ipv6Prefix=2600:1f13:fc2:a700:1768::/80
```

Example output

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "IPv6 automatic example",  
    "Groups": [  
      {  
        "GroupName": "default",  
        "GroupId": "sg-044c2de2c4EXAMPLE"  
      }  
    ],  
    "InterfaceType": "interface",  
    "Ipv6Addresses": [],  
    "MacAddress": "02:bb:e4:31:fe:09",  
    "NetworkInterfaceId": "eni-006edbcfa4EXAMPLE",  
    "OwnerId": "123456789012",  
    "PrivateIpAddress": "10.0.0.73",  
    "PrivateIpAddresses": [  
      {  
        "Primary": true,  
        "PrivateIpAddress": "10.0.0.73"  
      }  
    ],  
    "Ipv6Prefixes": [  
      {  
        "Ipv6Prefix": "2600:1f13:fc2:a700:1768::/80"  
      }  
    ],  
    "RequesterId": "AIDAIV5AJI5LXF5XXDPC0",  
    "RequesterManaged": false,  
    "SourceDestCheck": true,  
    "Status": "pending",  
    "SubnetId": "subnet-047cfed18eEXAMPLE",  
    "TagSet": [],  
    "VpcId": "vpc-0e12f52b21EXAMPLE"  
  }  
}
```

```
}
```

Assign prefixes to existing network interfaces

After you have assigned the prefixes, use the [attach-network-interface](#) AWS CLI command to attach the network interface to your instance. You must configure your operating system to work with network interfaces with prefixes. For more information, see [Configure your operating system for network interfaces with prefixes](#).

Tasks

- [Assign automatic prefixes to an existing network interface](#)
- [Assign specific prefixes to an existing network interface](#)

Assign automatic prefixes to an existing network interface

You can assign automatic prefixes to an existing network interface using one of the following methods.

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. To automatically assign an IPv4 prefix, for **IPv4 prefix delegation**, choose **Auto-assign**. Then for **Number of IPv4 prefixes**, specify the number of prefixes to assign.
5. To automatically assign an IPv6 prefix, for **IPv6 prefix delegation**, choose **Auto-assign**. Then for **Number of IPv6 prefixes**, specify the number of prefixes to assign.

Note

IPv6 prefix delegation appears only if the selected subnet is enabled for IPv6.

6. Choose **Save**.

AWS CLI

You can use the [assign-ipv6-addresses](#) command to assign IPv6 prefixes and the [assign-private-ip-addresses](#) command to assign IPv4 prefixes to existing network interfaces.

To assign automatic IPv4 prefixes to an existing network interface

Use the [assign-private-ip-addresses](#) command and set `--ipv4-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 IPv4 prefix.

```
$ aws ec2 assign-private-ip-addresses \  
--network-interface-id eni-081fbb4095EXAMPLE \  
--ipv4-prefix-count 1
```

Example output

```
{  
  "NetworkInterfaceId": "eni-081fbb4095EXAMPLE",  
  "AssignedIpv4Prefixes": [  
    {  
      "Ipv4Prefix": "10.0.0.176/28"  
    }  
  ]  
}
```

To assign automatic IPv6 prefixes to an existing network interface

Use the [assign-ipv6-addresses](#) command and set `--ipv6-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 IPv6 prefix.

```
$ aws ec2 assign-ipv6-addresses \  
--network-interface-id eni-00d577338cEXAMPLE \  
--ipv6-prefix-count 1
```

Example output

```
{  
  "AssignedIpv6Prefixes": [  
    "2600:1f13:fc2:a700:18bb::/80"  
  ],  
  "NetworkInterfaceId": "eni-00d577338cEXAMPLE"  
}
```

Assign specific prefixes to an existing network interface

You can assign specific prefixes to an existing network interface using one of the following methods.

Console

To assign specific 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. To assign a specific IPv4 prefix, for **IPv4 prefix delegation**, choose **Custom**. Then choose **Add new prefix** and enter the prefix to use.
5. To assign a specific IPv6 prefix, for **IPv6 prefix delegation**, choose **Custom**. Then choose **Add new prefix** and enter the prefix to use.

Note

IPv6 prefix delegation appears only if the selected subnet is enabled for IPv6.

6. Choose **Save**.

AWS CLI

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
```

Example output

```
{
```

```

    "NetworkInterfaceId": "eni-081fbb4095EXAMPLE",
    "AssignedIpv4Prefixes": [
      {
        "Ipv4Prefix": "10.0.0.208/28"
      }
    ]
  }

```

Assign specific IPv6 prefixes to an existing network interface

Use the [assign-ipv6-addresses](#) command and set `--ipv6-prefixes` to the prefix. AWS selects IPv6 addresses from this range. In the following example, the prefix CIDR is `2600:1f13:fc2:a700:18bb::/80`.

```

$ aws ec2 assign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80

```

Example output

```

{
  "NetworkInterfaceId": "eni-00d577338cEXAMPLE",
  "AssignedIpv6Prefixes": [
    {
      "Ipv6Prefix": "2600:1f13:fc2:a700:18bb::/80"
    }
  ]
}

```

Configure your operating system for network interfaces with prefixes

Amazon Linux AMIs might contain additional scripts installed by AWS, known as `ec2-net-utils`. These scripts optionally automate the configuration of your network interfaces. They are available for Amazon Linux only.

If you are not using Amazon Linux, you can use a Container Network Interface (CNI) for Kubernetes plug-in, or `dockerd` if you use Docker to manage your containers.

View the prefixes assigned to your network interfaces

You can view the prefixes assigned to your network interfaces using one of the following methods.

Console

To view the automatic prefixes assigned 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 for which to view the prefixes and choose the **Details** tab.
4. The **IPv4 Prefix Delegation** field lists the assigned IPv4 prefixes, and the **IPv6 Prefix Delegation** field lists the assigned IPv6 prefixes.

AWS CLI

You can use the [describe-network-interfaces](#) AWS CLI command to view the prefixes assigned to your network interfaces.

```
$ aws ec2 describe-network-interfaces
```

Example output

```
{
  "NetworkInterfaces": [
    {
      "AvailabilityZone": "us-west-2a",
      "Description": "IPv4 automatic example",
      "Groups": [
        {
          "GroupName": "default",
          "GroupId": "sg-044c2de2c4EXAMPLE"
        }
      ],
      "InterfaceType": "interface",
      "Ipv6Addresses": [],
      "MacAddress": "02:98:65:dd:18:47",
      "NetworkInterfaceId": "eni-02b80b4668EXAMPLE",
      "OwnerId": "123456789012",
      "PrivateIpAddress": "10.0.0.62",
      "PrivateIpAddresses": [
        {
          "Primary": true,
          "PrivateIpAddress": "10.0.0.62"
        }
      ]
    }
  ]
}
```

```

    ],
    "Ipv4Prefixes": [
      {
        "Ipv4Prefix": "10.0.0.208/28"
      }
    ],
    "Ipv6Prefixes": [],
    "RequesterId": "AIDAIV5AJI5LXF5XXDPC0",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "available",
    "SubnetId": "subnet-05eef9fb78EXAMPLE",
    "TagSet": [],
    "VpcId": "vpc-0e12f52b2146bf252"
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Description": "IPv6 automatic example",
    "Groups": [
      {
        "GroupName": "default",
        "GroupId": "sg-044c2de2c411c91b5"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "02:bb:e4:31:fe:09",
    "NetworkInterfaceId": "eni-006edbcfa4EXAMPLE",
    "OwnerId": "123456789012",
    "PrivateIpAddress": "10.0.0.73",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateIpAddress": "10.0.0.73"
      }
    ],
    "Ipv4Prefixes": [],
    "Ipv6Prefixes": [
      {
        "Ipv6Prefix": "2600:1f13:fc2:a700:1768::/80"
      }
    ],
    "RequesterId": "AIDAIV5AJI5LXF5XXDPC0",
    "RequesterManaged": false,

```

```
        "SourceDestCheck": true,  
        "Status": "available",  
        "SubnetId": "subnet-05eef9fb78EXAMPLE",  
        "TagSet": [],  
        "VpcId": "vpc-0e12f52b21EXAMPLE"  
    }  
]  
}
```

Remove prefixes from your network interfaces

You can remove prefixes from your network interfaces using one of the following methods.

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 from which to remove the prefixes and choose **Actions**, **Manage prefixes**.
4. Do one of the following:
 - To remove all assigned prefixes, for **IPv4 prefix delegation** and **IPv6 prefix delegation**, choose **Do not assign**.
 - To remove specific assigned prefixes, for **IPv4 prefix delegation** or **IPv6 prefix delegation**, choose **Custom** and then choose **Unassign** next to the prefixes to remove.

Note

IPv6 prefix delegation appears only if the selected subnet is enabled for IPv6.

5. 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 address that you want 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 address that you want to remove.

```
$ aws ec2 unassign-ipv6-addresses \  
--network-interface-id eni-00d577338cEXAMPLE \  
--ipv6-prefix 2600:1f13:fc2:a700:18bb::/80
```

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 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 aggregate multi-flow traffic available to an instance depends on the destination of the traffic.

- Within the Region – Traffic can utilize the full network bandwidth available to the instance.
- To other Regions, an internet gateway, Direct Connect, or local gateways (LGW) – Traffic can utilize up to 50% of the network bandwidth available to a current generation instance with a minimum of 32 vCPUs. Bandwidth for a current generation instance with less than 32 vCPUs is limited to 5 Gbps.

Single-flow traffic

Baseline 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 subnet to achieve up to 25 Gbps between those instances.

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 to another instance in a different Region 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](#)

To view network performance using the AWS CLI

You can use the [describe-instance-types](#) AWS CLI command to display information about an instance type. The following example displays network performance information for all C5 instances.

```
aws ec2 describe-instance-types --filters "Name=instance-type,Values=c5.*"
--query "InstanceTypes[][InstanceType, NetworkInfo.NetworkPerformance,
NetworkInfo.NetworkCards[0].BaselineBandwidthInGbps]" --output table
```

```
-----
|          DescribeInstanceTypes          |
+-----+-----+-----+
| c5.4xlarge | Up to 10 Gigabit | 5.0 |
| c5.xlarge  | Up to 10 Gigabit | 1.25 |
| c5.12xlarge | 12 Gigabit       | 12.0 |
| c5.24xlarge | 25 Gigabit       | 25.0 |
| c5.metal   | 25 Gigabit       | 25.0 |
| c5.9xlarge | 12 Gigabit       | 12.0 |
| c5.2xlarge | Up to 10 Gigabit | 2.5  |
| c5.large   | Up to 10 Gigabit | 0.75 |
| c5.18xlarge | 25 Gigabit       | 25.0 |
+-----+-----+-----+
```

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](#)
- [Network performance metrics](#)

Enhanced networking on Linux

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 inter-instance latencies. There is no additional charge for using enhanced networking.

For information about the supported network speed for each instance type, see [Amazon EC2 Instance Types](#).

Contents

- [Enhanced networking support](#)
- [Enable enhanced networking on your instance](#)
- [Enable enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances](#)
- [Improve network performance with ENA Express on Linux instances](#)

- [Enable enhanced networking with the Intel 82599 VF interface on Linux instances](#)
- [Operating system optimizations](#)
- [Monitor network performance for your EC2 instance](#)
- [Troubleshoot the Elastic Network Adapter \(ENA\)](#)
- [Improve network latency for Linux based Amazon EC2 instances](#)
- [Nitro system considerations for performance tuning](#)

Enhanced networking support

All current generation instance types support enhanced networking, except for T2 instances.

You can enable enhanced networking using one of the following mechanisms:

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 100 Gbps for supported instance types.

All [instances built on the AWS Nitro System](#) use ENA for enhanced networking. In addition, the following Xen instance types support ENA: H1, G3, m4.16xlarge, P2, P3, P3dn, and R4.

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.

Enable enhanced networking on your instance

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in [Enable enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances](#).

If your instance type supports the Intel 82599 VF interface for enhanced networking, follow the procedures in [Enable enhanced networking with the Intel 82599 VF interface on Linux instances](#).

Enable enhanced networking with the Elastic Network Adapter (ENA) on Linux instances

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA). To use enhanced networking, you must install the required ENA module and enable ENA support.

Contents

- [Requirements](#)
- [Enhanced networking performance](#)
- [Test whether enhanced networking is enabled](#)
- [Enable enhanced networking on the Amazon Linux AMI](#)
- [Enable enhanced networking on Ubuntu](#)
- [Enable enhanced networking on Linux](#)
- [Enable enhanced networking on Ubuntu with DKMS](#)
- [Driver release notes](#)
- [Troubleshoot](#)

Requirements

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Launch an [instance built on the AWS Nitro System](#).
- Launch the instance using a supported version of the Linux kernel and a supported distribution, so that ENA enhanced networking is enabled for your instance automatically. For more information, see [ENA Linux Kernel Driver Release Notes](#).
- Ensure that the instance has internet connectivity.
- 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.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as

enabling the `enaSupport` attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Enhanced networking performance

The following documentation provides a summary of the network performance for the instance types that support ENA enhanced networking:

- [Network specifications for accelerated computing instances](#)
- [Network specifications for compute optimized instances](#)
- [Network specifications for general purpose instances](#)
- [Network specifications for high-performance computing instances](#)
- [Network specifications for memory optimized instances](#)
- [Network specifications for storage optimized instances](#)

Test whether enhanced networking is enabled

The following AMIs include the required ENA module and have ENA support enabled:

- 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

To test whether enhanced networking is already enabled, verify that the `ena` module is installed on your instance and that the `enaSupport` attribute is set. If your instance satisfies these two conditions, then the `ethtool -i ethn` command should show that the module is in use on the network interface.

Kernel module (ena)

To verify that the `ena` module is installed, use the `modinfo` command as shown in the following example.

```
[ec2-user ~]$ modinfo ena
filename:      /lib/modules/4.14.33-59.37.amzn2.x86_64/kernel/drivers/amazon/net/ena/
ena.ko
version:      1.5.0g
license:      GPL
description:   Elastic Network Adapter (ENA)
author:       Amazon.com, Inc. or its affiliates
srcversion:   692C7C68B8A9001CB3F31D0
alias:        pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:        pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:        pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:        pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
retpoline:    Y
intree:       Y
name:         ena
...
```

In the above Amazon Linux case, the `ena` module is installed.

```
ubuntu:~$ modinfo ena
ERROR: modinfo: could not find module ena
```

In the above Ubuntu instance, the module is not installed, so you must first install it. For more information, see [Enable enhanced networking on Ubuntu](#).

Instance attribute (enaSupport)

To check whether an instance has the enhanced networking `enaSupport` attribute set, use one of the following commands. If the attribute is set, the response is `true`.

- [describe-instances](#) (AWS CLI/AWS CloudShell)

```
aws ec2 describe-instances --instance-ids instance_id --query  
"Reservations[].Instances[].EnaSupport"
```

- [Get-EC2Instance](#) (Tools for Windows PowerShell)

```
(Get-EC2Instance -InstanceId instance-id).Instances.EnaSupport
```

Image attribute (enaSupport)

To check whether an AMI has the enhanced networking `enaSupport` attribute set, use one of the following commands. If the attribute is set, the response is true.

- [describe-images](#) (AWS CLI/AWS CloudShell)

```
aws ec2 describe-images --image-id ami_id --query "Images[].EnaSupport"
```

- [Get-EC2Image](#) (Tools for Windows PowerShell)

```
(Get-EC2Image -ImageId ami_id).EnaSupport
```

Network interface driver

Use the following command to verify that the `ena` module is being used on a particular interface, substituting the interface name that you want to check. If you are using a single interface (default), it this is `eth0`. If the operating system supports [predictable network names](#), this could be a name like `ens5`.

In the following example, the `ena` module is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0  
driver: vif  
version:  
firmware-version:  
bus-info: vif-0  
supports-statistics: yes  
supports-test: no  
supports-eprom-access: no  
supports-register-dump: no
```

```
supports-priv-flags: no
```

In this example, the ena module is loaded and at the minimum recommended version. This instance has enhanced networking properly configured.

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 1.5.0g
firmware-version:
expansion-rom-version:
bus-info: 0000:00:05.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

Enable enhanced networking on the Amazon Linux AMI

Amazon Linux 2 and the latest versions of the Amazon Linux AMI include the module required for enhanced networking with ENA installed and have ENA support enabled. Therefore, if you launch an instance with an HVM version of Amazon Linux on a supported instance type, enhanced networking is already enabled for your instance. For more information, see [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 on Amazon Linux AMI

1. Connect to your instance.
2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including ena:

```
[ec2-user ~]$ sudo yum update
```

3. From your local computer, reboot your instance using the Amazon EC2 console or one of the following commands: [reboot-instances](#) (AWS CLI), [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell).

4. Connect to your instance again and verify that the `ena` module is installed and at the minimum recommended version using the `modinfo ena` command from [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), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking on Amazon Linux AMI \(instance store-backed instances\)](#).

6. From your local computer, enable the enhanced networking attribute using one of the following commands:

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance-id -EnaSupport $true
```

7. (Optional) Create an AMI from the instance, as described in [Create an Amazon EBS-backed Linux 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.
8. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
9. Connect to your instance and verify that the `ena` module is installed and loaded on your network interface using the `ethtool -i ethn` command from [Test whether enhanced networking is enabled](#).

If you are unable to connect to your instance after enabling enhanced networking, see [Troubleshoot the Elastic Network Adapter \(ENA\)](#).

To enable enhanced networking on Amazon Linux AMI (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Create an instance store-backed Linux 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 ...
```

Enable enhanced networking on Ubuntu

The latest Ubuntu HVM AMIs include the module required for enhanced networking with ENA installed and have ENA support enabled. Therefore, if you launch an instance with the latest Ubuntu HVM AMI on a supported instance type, enhanced networking is already enabled for your instance. For more information, see [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 [AWS Support](#).

To install the `linux-aws` kernel package (Ubuntu Trusty 14.04)

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

⚠ Important

If during the update process you are prompted to install `grub`, use `/dev/xvda` to install `grub` onto, and then choose to keep the current version of `/boot/grub/menu.lst`.

3. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking on Ubuntu \(instance store-backed instances\)](#).

4. From your local computer, enable the enhanced networking attribute using one of the following commands:

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance-id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance-id -EnaSupport $true
```

5. (Optional) Create an AMI from the instance, as described in [Create an Amazon EBS-backed Linux 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), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.

To enable enhanced networking on Ubuntu (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Create an instance store-backed Linux 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 ...
```

Enable enhanced networking on Linux

The latest AMIs for Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and CentOS include the module required for enhanced networking with ENA and have ENA support enabled. Therefore, if you launch an instance with the latest AMI on a supported instance type, enhanced networking is already enabled for your instance. For more information, see [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 module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>. (SUSE Linux Enterprise Server 12 SP2 and later include ENA 2.02 by default, so you are not required to download and compile the ENA driver. For SUSE Linux Enterprise Server 12 SP2 and later, you should file a request to add the driver version you want to the stock kernel).

```
git clone https://github.com/amzn/amzn-drivers
```

3. Compile and install the `ena` module on your instance. These steps depend on the Linux distribution. For more information about compiling the module on Red Hat Enterprise Linux, see the [AWS Knowledge Center article](#).
4. Run the `sudo depmod` command to update module dependencies.
5. Update `initramfs` on your instance to ensure that the new module loads at boot time. For example, if your distribution supports `dracut`, you can use the following command.

```
dracut -f -v
```

6. Determine if your system uses predictable network interface names by default. Systems that use `systemd` or `udev` versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see [Predictable Network Interface Names](#) on the freedesktop.org website.
 - a. You can check the `systemd` or `udev` versions on RPM-based systems with the following command.

```
rpm -qa | grep -e '^systemd-[0-9]\+|\^udev-[0-9]\+'  
systemd-208-11.e17_0.2.x86_64
```

In the above Red Hat Enterprise Linux 7 example, the `systemd` version is 208, so predictable network interface names must be disabled.

- b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

```
sudo sed -i '/^GRUB_CMDLINE_LINUX/s/\ "$\ net.ifnames=0\ "' /etc/default/  
grub
```

- c. Rebuild the grub configuration file.

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking on Linux \(instance store-backed instances\)](#).

8. From your local computer, enable the enhanced networking `enaSupport` attribute using one of the following commands:

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance-id -EnaSupport $true
```

9. (Optional) Create an AMI from the instance, as described in [Create an Amazon EBS-backed Linux 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.

Important

If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` might fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
11. (Optional) Connect to your instance and verify that the module is installed.

If you are unable to connect to your instance after enabling enhanced networking, see [Troubleshoot the Elastic Network Adapter \(ENA\)](#).

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 Linux 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 ...
```

Enable enhanced networking on Ubuntu with DKMS

This method is for testing and feedback purposes only. It is not intended for use with production deployments. For production deployments, see [Enable enhanced networking on 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 [Enable enhanced networking on Ubuntu](#).
2. Install the `build-essential` packages to compile the kernel module and the `dkms` package so that your `ena` module is rebuilt every time your kernel is updated.

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

3. Clone the source for the `ena` module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

4. Move the `amzn-drivers` package to the `/usr/src/` directory so DKMS can find it and build it for each kernel update. Append the version number (you can find the current version number in the release notes) of the source code to the directory name. For example, version `1.0.0` is shown in the following example.

```
ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
```

5. Create the DKMS configuration file with the following values, substituting your version of `ena`.
Create the file.

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

Edit the file and add the following values.

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
BUILT_MODULE_NAME[0]="ena"
BUILT_MODULE_LOCATION="kernel/linux/ena"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ena"
AUTOINSTALL="yes"
```

6. Add, build, and install the `ena` module on your instance using DKMS.

Add the module to DKMS.

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

Build the module using the `dkms` command.

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

Install the module using `dkms`.

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

7. Rebuild `initramfs` so the correct module is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. Verify that the `ena` module is installed using the `modinfo ena` command from [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 [Enable enhanced networking on Ubuntu](#).

Driver release notes

For information about the versions of the Linux ENA driver, see the [ENA Linux kernel driver release notes](#).

Troubleshoot

For troubleshooting information, see [Troubleshoot the Elastic Network Adapter \(ENA\)](#).

Improve network performance with ENA Express on Linux instances

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 to 25 Gbps within the subnet, 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.

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 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.

Prerequisites for Linux instances

To ensure that ENA Express can operate effectively, update the following settings for your instance.

- 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.

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.
- To utilize full bandwidth potential, use driver version 2.2.9 or higher.
- 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.

Supported instance types for ENA Express

The following tabs show instance types that support ENA Express.

General purpose

Instance type	Architecture
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
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-24x1	x86_64
m7i.metal-48x1	x86_64

Compute optimized

Instance type	Architecture
c6a.48xlarge	x86_64
c6a.metal	x86_64
c6gn.16xlarge	arm64

Instance type	Architecture
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
c7g.12xlarge	arm64
c7g.16xlarge	arm64
c7g.metal	arm64
c7gd.12xlarge	arm64
c7gd.16xlarge	arm64
c7gd.metal	arm64
c7i.12xlarge	x86_64
c7i.16xlarge	x86_64

Instance type	Architecture
c7i.24xlarge	x86_64
c7i.48xlarge	x86_64
c7i.metal-24x1	x86_64
c7i.metal-48x1	x86_64

Memory optimized

Instance type	Architecture
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
r6id.8xlarge	x86_64
r6id.12xlarge	x86_64
r6id.16xlarge	x86_64
r6id.24xlarge	x86_64
r6id.32xlarge	x86_64

Instance type	Architecture
r6id.metal	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-24x1	x86_64
r7i.metal-48x1	x86_64
x2idn.16xlarge	x86_64
x2idn.24xlarge	x86_64
x2idn.32xlarge	x86_64
x2idn.metal	x86_64
x2iedn.8xlarge	x86_64
x2iedn.16xlarge	x86_64
x2iedn.24xlarge	x86_64

Instance type	Architecture
x2iedn.32xlarge	x86_64
x2iedn.metal	x86_64

Accelerated computing

Instance type	Architecture
g6.48xlarge	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
im4gn.4xlarge	arm64
im4gn.8xlarge	arm64
im4gn.16xlarge	arm64

List and view ENA Express settings

This section covers how to list and view ENA Express information from the AWS Management Console or from the AWS CLI. For more information, choose the tab that matches the method you'll use.

Console

This tab covers how to find information about your current ENA Express settings and to view instance type support in the AWS Management Console.

View instance type support

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instance types**.
3. Select an instance type to see the details for that instance. You can choose the **Instance type** 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 **Networking** tab or that section on the detail page, **ENA Express support** shows a true or false value to indicate if the instance type supports this feature.

View settings from the Network interface list

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**.

View settings from instances

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

This tab covers how to find information about your current ENA Express settings and to view instance type support in the AWS CLI.

Describe instance types

For information about instance type settings for a specific instance type, run the [describe-instance-types](#) command in the AWS CLI, and substitute the instance type as follows:

```
[ec2-user ~]$ aws ec2 describe-instance-types --instance-types m6i.metal
{
  "InstanceTypes": [
    {
      "InstanceType": "m6i.metal",
      "CurrentGeneration": true,
      ...
    },
    "NetworkInfo": {
      ...
      "EnaSrdSupported": true
    },
    ...
  ]
}
```

Describe instances

For information about ENA Express configuration for specified instances, run the [describe-instances](#) command in the AWS CLI, as follows. 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.

```
[ec2-user ~]$ aws ec2 describe-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7 --query 'Reservations[*].Instances[*].[InstanceId, NetworkInterfaces[*].Attachment.EnaSrdSpecification]'
```

```
[
  [
    "i-1234567890abcdef0",
    [
      {
        "EnaSrdEnabled": true,
        "EnaSrdUdpSpecification": {
          "EnaSrdUdpEnabled": false
        }
      }
    ]
  ],
  [
    [
      "i-0598c7d356eba48d7",
      [
        {
          "EnaSrdEnabled": true,
          "EnaSrdUdpSpecification": {
            "EnaSrdUdpEnabled": false
          }
        }
      ]
    ]
  ]
]
```

Describe network interfaces

For information about ENA Express settings for a networking interface, run the [describe-network-interfaces](#) command in the AWS CLI as follows:

```
[ec2-user ~]$ aws ec2 describe-network-interfaces
```

```
{
  "NetworkInterfaces": [
    {
      "Association": {
        ....IPs, DNS...
      },
```

```

"Attachment": {
  "AttachTime": "2022-11-17T09:04:28+00:00",
  "AttachmentId": "eni-attach-0ab1c23456d78e9f0",
  "DeleteOnTermination": true,
  "DeviceIndex": 0,
  "NetworkCardIndex": 0,
  "InstanceId": "i-1234567890abcdef0",
  "InstanceOwnerId": "111122223333",
  "Status": "attached",
  "EnaSrdSpecification": {
    "EnaSrdEnabled": true,
    "EnaSrdUdpSpecification": {
      "EnaSrdUdpEnabled": true
    }
  }
},
...
"NetworkInterfaceId": "eni-1234567890abcdef0",
"OwnerId": "111122223333",
...
}
]
}

```

PowerShell

This tab covers how to find information about your current ENA Express settings and to view instance type support using PowerShell.

Describe instance types

For information on instance type settings for a specific instance type, run the [Get-EC2InstanceType Cmdlet](#) with the Tools for PowerShell, and substitute the instance type as follows:

```

PS C:\> Get-EC2InstanceType -InstanceType m6i.metal | `
Select-Object `
  InstanceType,
  CurrentGeneration,
  @{Name = 'EnaSrdSupported'; Expression = { $_.NetworkInfo.EnaSrdSupported } } |
Format-List

```

```
InstanceType      : m6i.metal
CurrentGeneration : True
EnaSrdSupported   : True
```

If ENA Express is enabled, a value of `True` is returned.

Describe network interfaces

For information on ENA Express settings for a networking interface, run the [Get-EC2NetworkInterface Cmdlet](#) with the Tools for PowerShell as follows:

```
PS C:\> Get-EC2NetworkInterface -NetworkInterfaceId eni-0d1234e5f6a78901b | `
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 } }

Association      :
NetworkInterfaceId : eni-0d1234e5f6a78901b
OwnerId          : 111122223333
AttachTime       : 6/11/2022 1:13:11 AM
AttachmentId     : eni-attach-0d1234e5f6a78901b
DeleteOnTermination : True
NetworkCardIndex : 0
InstanceId       : i-0d1234e5f6a78901b
InstanceOwnerId  : 111122223333
Status           : attached
EnaSrdEnabled    : True
EnaSrdUdpEnabled : False
```

Configure ENA Express settings

You can configure ENA Express for supported EC2 instance types without needing to install any additional software.

This section covers how to configure ENA Express from the AWS Management Console or from the AWS CLI. For more information, choose the tab that matches the method you'll use.

Console

This tab covers how to manage ENA Express settings for network interfaces that are attached to an instance.

Manage ENA Express from the Network interface list

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.

Note

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** check box.
6. When ENA Express is enabled, you can configure UDP settings. To use **ENA Express UDP**, select the **Enable** check box.
7. To save your settings, choose **Save**.

Manage ENA Express from the Instance list

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** check box.
8. When ENA Express is enabled, you can configure UDP settings. To use **ENA Express UDP**, select the **Enable** check box.
9. To save your settings, choose **Save**.

Configure ENA Express when you attach a network interface to an EC2 instance

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** check box.
6. When ENA Express is enabled, you can configure UDP settings. To use **ENA Express UDP**, select the **Enable** check box.
7. To attach the network interface to the instance and save your ENA Express settings, choose **Attach**.

AWS CLI

This tab covers how to configure ENA Express settings in the AWS CLI.

Configure ENA Express when you attach a network interface

To configure ENA Express when you attach a network interface to an instance, run the [attach-network-interface](#) command in the AWS CLI, as shown in the following examples:

Example 1: Use ENA Express for TCP traffic, but not for UDP traffic

In this example, we configure `EnaSrdEnabled` as `true`, and we allow `EnaSrdUdpEnabled` to default to `false`.

```
[ec2-user ~]$ aws ec2 attach-network-interface --network-interface-id eni-0123f4567890a1b23 --instance-id i-0f1a234b5cd67e890 --device-index 1 --ena-srd-specification 'EnaSrdEnabled=true'
{
  "AttachmentId": "eni-attach-012c3d45e678f9012"
}
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

In this example, we configure both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
[ec2-user ~]$ aws ec2 attach-network-interface --network-interface-id eni-0123f4567890a1b23 --instance-id i-0f1a234b5cd67e890 --device-index 1 --ena-srd-specification 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
{
  "AttachmentId": "eni-attach-012c3d45e678f9012"
}
```

Update ENA Express settings for your network interface attachment

To update ENA Express settings for a network interface that's attached to an instance, run the [modify-network-interface-attribute](#) command in the AWS CLI, as shown in the following examples:

Example 1: Use ENA Express for TCP traffic, but not for UDP traffic

In this example, we configure `EnaSrdEnabled` as `true`, and we allow `EnaSrdUdpEnabled` to default to `false` if it has never been set previously.

```
[ec2-user ~]$ aws ec2 modify-network-interface-attribute --network-interface-id eni-0123f4567890a1b23 --ena-srd-specification 'EnaSrdEnabled=true'
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

In this example, we configure both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
[ec2-user ~]$ aws ec2 modify-network-interface-attribute --  
network-interface-id eni-0123f4567890a1b23 --ena-srd-specification  
'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

Example 3: Stop using ENA Express for UDP traffic

In this example, we configure `EnaSrdUdpEnabled` as *false*.

```
[ec2-user ~]$ aws ec2 modify-network-interface-attribute --  
network-interface-id eni-0123f4567890a1b23 --ena-srd-specification  
'EnaSrdUdpSpecification={EnaSrdUdpEnabled=false}'
```

PowerShell

This tab covers how to configure ENA Express settings using PowerShell.

Configure ENA Express when you attach a network interface

To configure ENA Express settings for a networking interface, run the [Add-EC2NetworkInterface Cmdlet](#) with the Tools for PowerShell as shown in the following examples:

Example 1: Use ENA Express for TCP traffic, but not for UDP traffic

In this example, we configure `EnaSrdEnabled` as *true*, and we allow `EnaSrdUdpEnabled` to default to *false*.

```
PS C:\> Add-EC2NetworkInterface `   
-NetworkInterfaceId eni-0123f4567890a1b23 `   
-InstanceId i-0f1a234b5cd67e890 `   
-DeviceIndex 1 `   
-EnaSrdSpecification_EnaSrdEnabled $true   
  
eni-attach-012c3d45e678f9012
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

In this example, we configure both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as *true*.

```
PS C:\> Add-EC2NetworkInterface `   
-NetworkInterfaceId eni-0123f4567890a1b23 `   
-InstanceId i-0f1a234b5cd67e890 `   
-DeviceIndex 1 `
```



```
-EnaSrdSpecification_EnaSrdEnabled $true `
-EnaSrdUdpSpecification_EnaSrdUdpEnabled $true

eni-attach-012c3d45e678f9012
```

Update ENA Express settings for your network interface attachment

To update ENA Express settings for a network interface that's attached to an instance, run the [Add-EC2NetworkInterface Cmdlet](#) in the Tools for PowerShell, as shown in the following examples:

Example 1: Use ENA Express for TCP traffic, but not for UDP traffic

In this example, we configure `EnaSrdEnabled` as `true`, and we allow `EnaSrdUdpEnabled` to default to `false` if it has never been set previously.

```
PS C:\> Edit-EC2NetworkInterfaceAttribute `
-NetworkInterfaceId eni-0123f4567890a1b23 `
-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

NetworkInterfaceId : eni-0123f4567890a1b23
EnaSrdEnabled      : True
EnaSrdUdpEnabled   : False
```

Example 2: Use ENA Express for both TCP traffic and UDP traffic

In this example, we configure both `EnaSrdEnabled` and `EnaSrdUdpEnabled` as `true`.

```
PS C:\> Edit-EC2NetworkInterfaceAttribute `
-NetworkInterfaceId eni-0123f4567890a1b23 `
-EnaSrdSpecification_EnaSrdEnabled $true `
-EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
```

```

    @{Name = 'EnaSrdEnabled'; Expression =
  { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
  { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List

NetworkInterfaceId : eni-0123f4567890a1b23
EnaSrdEnabled      : True
EnaSrdUdpEnabled   : True

```

Example 3: Stop using ENA Express for UDP traffic

In this example, we configure `EnaSrdUdpEnabled` as *false*.

```

PS C:\> 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

NetworkInterfaceId : eni-0123f4567890a1b23
EnaSrdEnabled      : True
EnaSrdUdpEnabled   : False

```

Configure ENA Express when you launch an EC2 instance

You can use one of the following methods to configure ENA Express for an AMI when you launch an instance from the AWS Management Console.

- You can configure ENA Express for your AMI when you launch an instance with the launch instance wizard. For configuration details, see **Advanced network configuration** in the [Network settings](#) for the launch instance wizard.
- You can configure ENA Express for your AMI when you use a launch template. For more information about launch template configuration, see **Advanced network configuration** in the [Network settings](#) for launch templates.

Monitor ENA Express performance for your Linux instance

After 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.

To see a list of metrics that are filtered for ENA Express, run the following **ethtool** command for your network interface (shown here as `eth0`):

```
[ec2-user ~]$ ethtool -S eth0 | grep ena_srd
NIC statistics:
  ena_srd_mode: 0
  ena_srd_tx_pkts: 0
  ena_srd_eligible_tx_pkts: 0
  ena_srd_rx_pkts: 0
  ena_srd_resource_utilization: 0
```

Verify ENA Express settings for an instance

To verify the current ENA Express settings for the network interface attachment on your instance, run the `ethtool` command to list ENA Express metrics, and take note of the value of the `ena_srd_mode` metric. 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

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. 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.

You can monitor your SRD resource utilization using the `ena_srd_resource_utilization` metric. If your instance is close to exhausting its SRD resources, you'll know it's time to scale out the instance.

For more information about ENA Express metrics, see [Metrics for ENA Express](#).

Tune performance for ENA Express settings on your Linux instance

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.

Enable enhanced networking with the Intel 82599 VF interface on Linux instances

Amazon EC2 provides enhanced networking capabilities through the Intel 82599 VF interface, which uses the Intel `ixgbevf` driver.

Contents

- [Requirements](#)
- [Test whether enhanced networking is enabled](#)
- [Enable enhanced networking on Amazon Linux](#)
- [Enable enhanced networking on Ubuntu](#)
- [Enable enhanced networking on other Linux distributions](#)
- [Troubleshoot connectivity issues](#)

Requirements

To prepare for enhanced networking using the Intel 82599 VF interface, set up your instance as follows:

- Select from the following supported instance types: C3, C4, D2, I2, M4 (excluding `m4.16xlarge`), and R3.
- Launch the instance from an HVM AMI using Linux kernel version of 2.6.32 or later. The latest Amazon Linux HVM AMIs have the modules required for enhanced networking installed and have the required attributes set. Therefore, if you launch an Amazon EBS-backed, enhanced networking-supported instance using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- Ensure that the instance has internet connectivity.
- 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.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `sriovNetSupport` attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Test whether enhanced networking is enabled

Enhanced networking with the Intel 82599 VF interface is enabled if the `ixgbevf` module is installed on your instance and the `sriovNetSupport` attribute is set.

Instance attribute (`sriovNetSupport`)

To check whether an instance has the enhanced networking `sriovNetSupport` attribute set, use one of the following commands:

AWS CLI

[describe-instance-attribute](#) (AWS CLI/AWS CloudShell)

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

PowerShell

[Get-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Get-EC2InstanceAttribute -InstanceId instance-id -Attribute sriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty. If the attribute is set, the value is simple, as shown in the following example output.

```
"SriovNetSupport": {  
  "Value": "simple"  
},
```

Image attribute (sriovNetSupport)

To check whether an AMI already has the enhanced networking `sriovNetSupport` attribute set, use one of the following commands:

AWS CLI

[describe-images](#) (AWS CLI/AWS CloudShell)

```
aws ec2 describe-images --image-id ami_id --query "Images[].SriovNetSupport"
```

PowerShell

[Get-EC2Image](#) (AWS Tools for Windows PowerShell)

```
(Get-EC2Image -ImageId ami-id).SriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty. If the attribute is set, the value is simple.

Network interface driver

Use the following command to verify that the module is being used on a particular interface, substituting the interface name that you want to check. If you are using a single interface (default),

this is `eth0`. If the operating system supports [predictable network names](#), 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
```

Enable enhanced networking on Amazon Linux

The latest Amazon Linux HVM AMIs have the `ixgbevf` module required for enhanced networking installed and have the required `sriovNetSupport` attribute set. Therefore, if you launch an instance type using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance. For more information, see [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.

⚠ Warning

There is no way to disable the enhanced networking attribute after you've enabled it.

To enable enhanced networking

1. Connect to your instance.
2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including `ixgbevf`:

```
[ec2-user ~]$ sudo yum update
```

3. From your local computer, reboot your instance using the Amazon EC2 console or one of the following commands: [reboot-instances](#) (AWS CLI), [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell).
4. Connect to your instance again and verify that the `ixgbevf` module is installed and at the minimum recommended version using the `modinfo ixgbevf` command from [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), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking \(instance store-backed instances\)](#).
6. From your local computer, enable the enhanced networking attribute using one of the following commands:

AWS CLI

[modify-instance-attribute](#) (AWS CLI/AWS CloudShell)

```
aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```


PowerShell

[Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

7. (Optional) Create an AMI from the instance, as described in [Create an Amazon EBS-backed Linux 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), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
9. Connect to your instance and verify that the `ixgbevf` module is installed and loaded on your network interface using the `ethtool -i ethn` command from [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 Linux AMI](#), making sure to enable the enhanced networking attribute when you register the AMI.

AWS CLI

[register-image](#) (AWS CLI/AWS CloudShell)

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

[Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Enable enhanced networking on 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`.

Enable enhanced networking on 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/AWS CloudShell), [Stop-](#)

[EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking \(instance store-backed instances\)](#).

8. From your local computer, enable the enhanced networking attribute using one of the following commands:

AWS CLI

[modify-instance-attribute](#) (AWS CLI/AWS CloudShell)

```
aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support simple
```

PowerShell

[Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

9. (Optional) Create an AMI from the instance, as described in [Create an Amazon EBS-backed Linux 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.

Important

If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` might fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.

11. (Optional) Connect to your instance and verify that the module is installed.

To enable enhanced networking (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Create an instance store-backed Linux AMI](#), making sure to enable the enhanced networking attribute when you register the AMI.

AWS CLI

[register-image](#) (AWS CLI/AWS CloudShell)

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

[Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

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 enable and configure enhanced networking on my EC2 instances?](#).

Operating system optimizations

To achieve the maximum network performance on instances with enhanced networking, you might need to modify the default operating system configuration. For more information, see [ENA Linux Driver Best Practices and Performance Optimization Guide](#) on GitHub.

Monitor network performance for your EC2 instance

The Elastic Network Adapter (ENA) driver publishes network performance metrics from the instances where they are enabled. You can use these metrics to troubleshoot instance performance issues, choose the right instance size for a workload, plan scaling activities proactively, and benchmark applications to determine whether they maximize the performance available on an instance.

Amazon EC2 defines network maximums at the instance level to ensure a high-quality networking experience, including consistent network performance across instance sizes. AWS provides maximums for the following for each instance:

- **Bandwidth capability** – Each EC2 instance has a maximum bandwidth for aggregate inbound and outbound traffic, based on instance type and size. Some instances use a network I/O credit mechanism to allocate network bandwidth based on average bandwidth utilization. Amazon EC2 also has maximum bandwidth for traffic to AWS Direct Connect and the internet. 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 [Security group connection tracking](#)
- **Link-local service access** – Amazon EC2 provides a maximum PPS per network interface for traffic to services such as the DNS service, the Instance Metadata Service, and the Amazon Time Sync Service.

When the network traffic for an instance exceeds a maximum, AWS shapes the traffic that exceeds the maximum by queueing and then dropping network packets. You can monitor when traffic exceeds a maximum using the network performance metrics. These metrics inform you, in real time, of impact to network traffic and possible network performance issues.

Contents

- [Requirements](#)
- [Metrics for the ENA driver](#)
- [View the network performance metrics for your Linux instance](#)
- [Metrics for ENA Express](#)

- [Network performance metrics with the DPDK driver for ENA](#)
- [Metrics on instances running FreeBSD](#)

Requirements

The following requirements apply to 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.
- In order to support `contrack_allowance_available` metric, install ENA driver version 2.8.1.

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.

The following metrics are available on Linux instances, FreeBSD instances, and DPDK environments.

Metric	Description	Supported on
<code>bw_in_allowance_exceeded</code>	The number of packets queued or dropped because the inbound aggregate bandwidth exceeded the maximum for the instance.	All instance types
<code>bw_out_allowance_exceeded</code>	The number of packets queued or dropped because the outbound	All instance types

Metric	Description	Supported on
	aggregate bandwidth exceeded the maximum for the instance.	
contrack_allowance_exceeded	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.	All instance types
contrack_allowance_available	The number of tracked connections that can be established by the instance before hitting the Connections Tracked allowance of that instance type.	instances built on the AWS Nitro System only. Not supported with FreeBSD instances or DPDK environments.
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 DNS service, the Instance Metadata Service, and the Amazon Time Sync Service.	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

View the network performance metrics for your Linux instance

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
```

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. 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.


The following ENA Express metrics are available using the **ethtool** command for Linux based instances.

- `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 packets sent within a given time period that meet SRD requirements for eligibility, 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 see a list of metrics that are filtered for ENA Express, run the following **ethtool** command for your network interface (shown here as `eth0`):

```
[ec2-user ~]$ ethtool -S eth0 | grep ena_srd
NIC statistics:
  ena_srd_mode: 0
  ena_srd_tx_pkts: 0
  ena_srd_eligible_tx_pkts: 0
  ena_srd_rx_pkts: 0
  ena_srd_resource_utilization: 0
```

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.

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
contrack_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*.

Once you are collecting FreeBSD metrics, you can retrieve the latest set of collected metrics by running the following command.

```
sysctl dev.ena.network_interface.eni_metrics
```

Troubleshoot the Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) is designed to improve operating system health and reduce the chances of long-term disruption because of unexpected hardware behavior and or failures. The ENA architecture keeps device or driver failures as transparent to the system as possible. This topic provides troubleshooting information for ENA.

If you are unable to connect to your instance, start with the [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 migrating my EC2 instance to a sixth generation instance to make sure that I get maximum network performance?](#) in the AWS Knowledge Center.

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

- [Troubleshoot connectivity issues](#)
- [Keep-alive mechanism](#)
- [Register read timeout](#)
- [Statistics](#)
- [Driver error logs in syslog](#)
- [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 using the Amazon EC2 console or one of the following commands: [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance. Instead, proceed to [To disable enhanced networking with ENA \(instance store-backed instances\)](#).

2. From your local computer, disable the enhanced networking attribute using the following command.
 - [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

3. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
4. (Optional) Connect to your instance and try reinstalling the `ena` module with your current kernel version by following the steps in [Enable enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances](#).

To disable enhanced networking with ENA (instance store-backed instances)

If your instance is an instance store-backed instance, create a new AMI as described in [Create an instance store-backed Linux AMI](#). Be sure to disable the enhanced networking `enaSupport` attribute when you register the AMI.

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --no-ena-support ...
```

- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
C:\> Register-EC2Image -EnaSupport $false ...
```

Keep-alive mechanism

The ENA device posts keep-alive events at a fixed rate (usually once every second). The ENA driver implements a watchdog mechanism, which checks for the presence of these keep-alive messages. If a message or messages are present, the watchdog is rearmed, otherwise the driver concludes that the device experienced a failure and then does the following:

- Dumps its current statistics to syslog
- Resets the ENA device
- Resets the ENA driver state

The above reset procedure may result in some traffic loss for a short period of time (TCP connections should be able to recover), but should not otherwise affect the user.

The ENA device may also indirectly request a device reset procedure, by not sending a keep-alive notification, for example, if the ENA device reaches an unknown state after loading an irrecoverable configuration.

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
contrack_allowance_available: 450878
contrack_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.

`contrack_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 DNS service, the Instance Metadata Service, and the Amazon Time Sync Service.

`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 with ENA Express on Linux instances](#).

- **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 with ENA Express on Linux instances](#).

- **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
```

Improve network latency for Linux based Amazon 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 network hops

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.

Linux kernel configuration

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](#) in the *Linux kernel user's and administrator's guide*.

```
$ 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.

```
$ 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](#).

ENA driver configuration

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.

```
$ 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.

```
$ 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 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.

Considerations

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 workload configuration

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. If connection tracking is enabled for your security groups, we recommend that you avoid using an asymmetric routing topology. For more information about security group connection tracking, untracked connections, and automatically tracked connections, see [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.

Note

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 or upgrade Elastic Network Adapter \(ENA\) driver](#).

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.

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. 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 processed at the standard rate (not accelerated). This can cause variations in your performance. Amazon EC2 supports jumbo frames of 9001 bytes, however not all services support it. 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.

Number of Elastic Network Adapter (ENA) queues

By default, the maximum number of ENA queues are allocated to a network interface based on your instance size and type. Reducing the queue count can reduce the maximum PPS rate achievable. We recommend using the default queue allocation for best performance.

For Linux, a network interface is configured with the maximum by default. For applications based on the Data Plane Development Kit (DPDK), we recommend that you configure the maximum number of queues available.

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 handled as exceptions, and can't take advantage of hardware acceleration.

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.

Maximize network performance on your Nitro system

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 [Considerations](#).

Note

For more considerations and best practices for your instance configuration, see:

Linux instances: [ENA Linux Driver Best Practices and Performance Optimization Guide](#) on the GitHub website.

Windows instances: [Best practices for configuring network interfaces](#).

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.

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.

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 your EC2 instance](#).

Elastic Fabric Adapter

An Elastic Fabric Adapter (EFA) is a network device that you can attach to your Amazon EC2 instance to accelerate High Performance Computing (HPC) and machine learning applications.

EFA enables you to achieve the application performance of an on-premises HPC cluster, with the scalability, flexibility, and elasticity provided by the AWS Cloud.

EFAs provide lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems. It enhances the performance of inter-instance communication that is critical for scaling HPC and machine learning applications. It is optimized to work on the existing AWS network infrastructure and it can scale depending on application requirements.

EFAs integrate with Libfabric 1.7.0 and later and it supports Open MPI 5 and later and Intel MPI 2019 Update 5 and later for HPC applications, and Nvidia Collective Communications Library (NCCL) for machine learning applications.

Note

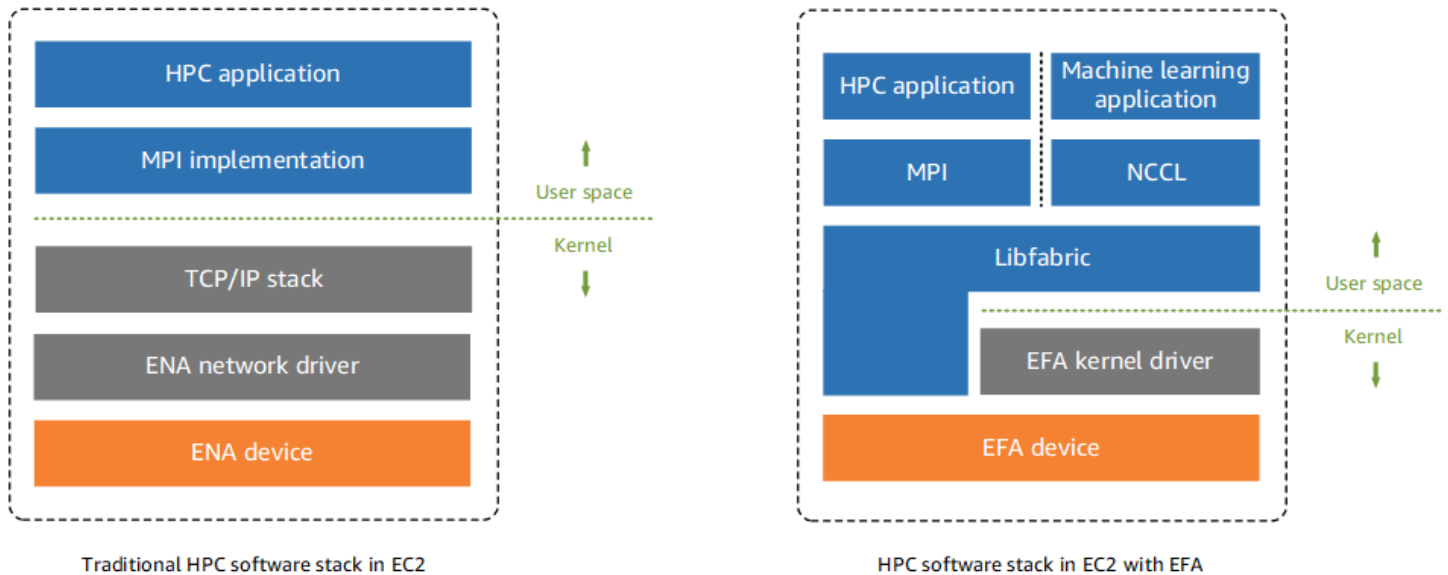
The OS-bypass capabilities of EFAs are not supported on Windows instances. If you attach an EFA to a Windows instance, the instance functions as an Elastic Network Adapter, without the added EFA capabilities.

Contents

- [EFA basics](#)
- [Supported interfaces and libraries](#)
- [Supported instance types](#)
- [Supported operating systems](#)
- [EFA limitations](#)
- [EFA pricing](#)
- [Get started with P5 instances and EFA](#)
- [Get started with EFA and MPI](#)
- [Get started with EFA and NCCL](#)
- [Work with EFA](#)
- [Monitor an EFA](#)
- [Verify the EFA installer using a checksum](#)

EFA basics

An EFA is an Elastic Network Adapter (ENA) with added capabilities. It provides all of the functionality of an ENA, with an additional OS-bypass functionality. OS-bypass is an access model that allows HPC and machine learning applications to communicate directly with the network interface hardware to provide low-latency, reliable transport functionality.



Traditionally, HPC applications use the Message Passing Interface (MPI) to interface with the system's network transport. In the AWS Cloud, this has meant that applications interface with MPI, which then uses the operating system's TCP/IP stack and the ENA device driver to enable network communication between instances.

With an EFA, HPC applications use MPI or NCCL to interface with the *Libfabric* API. The Libfabric API bypasses the operating system kernel and communicates directly with the EFA device to put packets on the network. This reduces overhead and enables the HPC application to run more efficiently.

Note

Libfabric is a core component of the OpenFabrics Interfaces (OFI) framework, which defines and exports the user-space API of OFI. For more information, see the [Libfabric OpenFabrics website](#).

Differences between EFAs and ENAs

Elastic Network Adapters (ENAs) provide traditional IP networking features that are required to support VPC networking. EFAs provide all of the same traditional IP networking features as ENAs, and they also support OS-bypass capabilities. OS-bypass enables HPC and machine learning applications to bypass the operating system kernel and to communicate directly with the EFA device.

Supported interfaces and libraries

EFAs support the following interfaces and libraries:

- Open MPI 5 and later
- Open MPI 4.0 or newer is preferred for Graviton
- Intel MPI 2019 Update 5 and later
- NVIDIA Collective Communications Library (NCCL) 2.4.2 and later

Supported instance types

The following instance types support EFAs:

- General purpose: m5dn.24xlarge | m5dn.metal | m5n.24xlarge | m5n.metal | m5zn.12xlarge | m5zn.metal | m6a.48xlarge | m6a.metal | m6i.32xlarge | m6i.metal | m6id.32xlarge | m6id.metal | m6idn.32xlarge | m6idn.metal | m6in.32xlarge | m6in.metal | m7a.48xlarge | m7a.metal-48x1 | m7g.16xlarge | m7g.metal | m7gd.16xlarge | m7gd.metal | m7i.48xlarge | m7i.metal-48x1
- Compute optimized: c5n.9xlarge | c5n.18xlarge | c5n.metal | c6a.48xlarge | c6a.metal | c6gn.16xlarge | c6i.32xlarge | c6i.metal | c6id.32xlarge | c6id.metal | c6in.32xlarge | c6in.metal | c7a.48xlarge | c7a.metal-48x1 | c7g.16xlarge | c7g.metal | c7gd.16xlarge | c7gd.metal | c7gn.16xlarge | c7gn.metal | c7i.48xlarge | c7i.metal-48x1
- Memory optimized: r5dn.24xlarge | r5dn.metal | r5n.24xlarge | r5n.metal | r6a.48xlarge | r6a.metal | r6i.32xlarge | r6i.metal | r6idn.32xlarge | r6idn.metal | r6in.32xlarge | r6in.metal | r6id.32xlarge | r6id.metal | r7a.48xlarge | r7a.metal-48x1 | r7g.16xlarge | r7g.metal | r7gd.16xlarge | r7gd.metal | r7i.48xlarge | r7i.metal-48x1 | r7iz.32xlarge | r7iz.metal-32x1 |

x2idn.32xlarge | x2idn.metal | x2iedn.32xlarge | x2iedn.metal | x2iezn.12xlarge | x2iezn.metal

- Storage optimized: i3en.12xlarge | i3en.24xlarge | i3en.metal | i4g.16xlarge | i4i.32xlarge | i4i.metal | im4gn.16xlarge
- Accelerated computing: dl1.24xlarge | dl2q.24xlarge | g4dn.8xlarge | g4dn.12xlarge | g4dn.16xlarge | g4dn.metal | g5.8xlarge | g5.12xlarge | g5.16xlarge | g5.24xlarge | g5.48xlarge | g6.8xlarge | g6.12xlarge | g6.16xlarge | g6.24xlarge | g6.48xlarge | gr6.8xlarge | inf1.24xlarge | p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | p5.48xlarge | trn1.32xlarge | trn1n.32xlarge | vt1.24xlarge
- High-performance computing: hpc6a.48xlarge | hpc6id.32xlarge | hpc7a.12xlarge | hpc7a.24xlarge | hpc7a.48xlarge | hpc7a.96xlarge | hpc7g.4xlarge | hpc7g.8xlarge | hpc7g.16xlarge

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

The following operating systems support EFAs with Intel/AMD x86-based instance types:

- Amazon Linux 2023
- Amazon Linux 2
- CentOS 7
- RHEL 7, 8, and 9
- Debian 10 and 11
- Rocky Linux 8 and 9
- Ubuntu 20.04 and 22.04
- SUSE Linux Enterprise 15 SP2 and later

- OpenSUSE Leap 15.4 and later

Note

Ubuntu 20.04 supports peer direct support when used with d11.24xlarge instances.

The following operating systems support EFAs with Arm-based (Graviton) instance types:

- Amazon Linux 2023
- Amazon Linux 2
- RHEL 8/9 and Rocky Linux 8/9
- Debian 10 and 11
- Ubuntu 20.04 and 22.04
- SUSE Linux Enterprise 15 SP2 and later

EFA limitations

EFAs have the following limitations:

- All P4d and P5 instance types support NVIDIA GPUDirect Remote Direct Memory Access (RDMA).
- EFA traffic 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.
- For c7g.16xlarge, m7g.16xlarge and r7g.16xlarge Dedicated Instances and Dedicated Hosts are not supported when an EFA is attached.
- EFA OS-bypass traffic is limited to a single subnet. In other words, EFA traffic cannot be sent from one subnet to another. Normal IP traffic from the EFA can be sent from one subnet to another.
- EFA OS-bypass traffic is not routable. Normal IP traffic from the EFA remains routable.
- The EFA must be a member of a security group that allows all inbound and outbound traffic to and from the security group itself.
- EFA is not supported on AWS [Outposts](#).


```
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  
"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
\  
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
\  

```

```
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=17,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=18,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=19,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=21,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=22,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=23,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=25,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=26,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=27,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\  

```

```

"NetworkCardIndex=29,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\

"NetworkCardIndex=30,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
\

"NetworkCardIndex=31,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=e
...

```

If you are using a launch template, specify the required number of network interfaces in the launch template. For each network interface, for `InterfaceType`, specify `efa`. For the primary network interface, for `NetworkCardIndex` and `DeviceIndex` specify `0`. For the remaining network interfaces, for `NetworkCardIndex` specify a unique value from 1 to 31, and for `DeviceIndex` specify 1. 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",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 2,
  "DeviceIndex": 1,
  "InterfaceType": "efa",

```



```
"AssociatePublicIpAddress":false,
"Groups":[
  "security_group_id"
],
"DeleteOnTermination":true
}
...
```

When launching a P5 instance with more than one network interface, you cannot auto-assign public IP addresses. However, you can attach an Elastic IP address to the primary network interface (*NetworkCardIndex=0, DeviceIndex=0*) after launch for internet connectivity. Both Ubuntu 20.04 and later and Amazon Linux 2 and later are configured to use the primary network interface for internet traffic when the instance is launched as recommended above.

Get started with EFA and MPI

This tutorial helps you to launch an EFA and MPI-enabled instance cluster for HPC workloads. In this tutorial, you will perform the following steps:

Contents

- [Step 1: Prepare an EFA-enabled security group](#)
- [Step 2: Launch a temporary instance](#)
- [Step 3: Install the EFA software](#)
- [Step 4: \(Optional\) Enable Open MPI 5](#)
- [Step 5: \(Optional\) Install Intel MPI](#)
- [Step 6: Disable ptrace protection](#)
- [Step 7. Confirm installation](#)
- [Step 8: Install your HPC application](#)
- [Step 9: Create an EFA-enabled AMI](#)
- [Step 10: Launch EFA-enabled instances into a cluster placement group](#)
- [Step 11: Terminate the temporary instance](#)
- [Step 12: Enable passwordless SSH](#)

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, and for **Elastic Fabric Adapter**, select **Enable**.
8. In the **Storage** section, configure the volumes as needed.
9. In the **Summary** panel on the right, choose **Launch instance**.

Step 3: Install the EFA software

Install the EFA-enabled kernel, EFA drivers, Libfabric, and Open MPI stack that is required to support EFA on your temporary instance.

The steps differ depending on whether you intend to use EFA with Open MPI, with Intel MPI, or with Open MPI and Intel MPI.

To install the EFA software

1. Connect to the instance you launched. For more information, see [Connect to your Linux instance](#).
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 7/8/9, CentOS 7, Rocky Linux 8/9

```
$ sudo yum update -y
```

- Ubuntu 20.04/22.04 and Debian 10/11

```
$ 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.

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.32.0.tar.gz
```

You can also get the latest version by replacing the version number with `latest` in the preceding command.

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 AWS Support.

- c. Download the signature file and verify the signature of the EFA tarball file.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.32.0.tar.gz.sig && gpg --verify ./aws-efa-installer-1.32.0.tar.gz.sig
```

The following shows example output.


```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC
```

If the result includes `Good signature`, and the fingerprint matches the fingerprint returned in the previous step, proceed to the next step. If not, don't run the EFA installation script, and contact AWS Support.

6. Extract the files from the compressed `.tar.gz` file and navigate into the extracted directory.

```
$ tar -xf aws-efa-installer-1.32.0.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 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, 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 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 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, 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 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.

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 and Open MPI 5 are installed by default. Alternatively, you can choose to install only Open MPI 4 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 Amazon Linux instance](#).

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 that 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 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 [Capacity Reservations in cluster placement groups](#).

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 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, and for **Elastic Fabric Adapter**, select **Enable**.

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 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: 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

The NVIDIA Collective Communications Library (NCCL) is a library of standard collective communication routines for multiple GPUs across a single node or multiple nodes. NCCL can be used together with EFA, Libfabric, and MPI to support various machine learning workloads. For more information, see the [NCCL](#) website.

Note

- NCCL with EFA is supported with `p3dn.24xlarge`, `p4d.24xlarge`, and `p5.48xlarge` only.
- Only NCCL 2.4.2 and later is supported with EFA.

The following tutorials help you to launch an EFA and NCCL-enabled instance cluster for machine learning workloads.

- [Use a base AMI](#)
- [Use an AWS Deep Learning AMI](#)

Use a base AMI

The following steps help you to get started with Elastic Fabric Adapter using an AMI for one of the [supported base operating systems](#).

Note

- Only the p3dn.24xlarge, p4d.24xlarge, p5.48xlarge instance types are supported.
- Only Amazon Linux 2, RHEL 7/8/9, CentOS 7, Rocky Linux 8/9, and Ubuntu 20.04/22.04 base AMIs are supported.

Contents

- [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 aws-ofi-nccl plugin](#)
- [Step 8: Install the NCCL tests](#)
- [Step 9: Test your EFA and NCCL configuration](#)
- [Step 10: Install your machine learning applications](#)
- [Step 11: Create an EFA and NCCL-enabled AMI](#)
- [Step 12: Terminate the temporary instance](#)
- [Step 13: Launch EFA and NCCL-enabled instances into a cluster placement group](#)
- [Step 14: Enable passwordless SSH](#)

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 either p3dn.24xlarge, p4d.24xlarge, or p5.48xlarge.
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, and for **Elastic Fabric Adapter**, select **Enable**.
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 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. Install the EPEL repository for DKMS and enable any optional repos for your Linux distribution.

```
$ sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- b. Install the CUDA repository public GPG key.

```
$ distribution='rhel7'
```

- c. 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
```

- d. (*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.

- *p3dn.24xlarge*

```
$ sudo yum clean all \  
&& sudo yum -y install kmod-nvidia-latest-dkms nvidia-driver-latest-dkms \  
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcuda8-devel
```

- *p4d.24xlarge and p5.48xlarge*

```
$ sudo yum clean all \  
&& sudo yum -y install kmod-nvidia-open-dkms nvidia-driver-latest-dkms \  
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcuda8-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.

CentOS 7

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 \
&& sudo yum install -y tar bzip2 make automake pciutils elfutils-libelf-devel
libglvnd-devel iptables firewalld vim bind-utils
```

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 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. Open `/etc/default/grub` using your preferred text editor and add the following.

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- d. Rebuild the Grub configuration.

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. Reboot the instance and reconnect to it.
5. Install the Nvidia GPU drivers, NVIDIA CUDA toolkit, and cuDNN.

- a. Install the EPEL repository for DKMS and enable any optional repos for your Linux distribution.

```
$ sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- b. Install the CUDA repository public GPG key.

```
$ distribution='rhel7'
```

- c. 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
```

d. Install the NVIDIA, CUDA drivers and cuDNN.

```
$ sudo yum clean all \  
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcuda8-devel
```

6. Reboot the instance and reconnect to it.

7. (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 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.

RHEL 7/8/9 and Rocky Linux 8/9

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. 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 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. Open `/etc/default/grub` using your preferred text editor and add the following.

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- d. Rebuild the Grub configuration.

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. Reboot the instance and reconnect to it.
5. Install the Nvidia GPU drivers, NVIDIA CUDA toolkit, and cuDNN.
 - a. Install the EPEL repository for DKMS and enable any optional repos for your Linux distribution.

- RHEL 7

```
$ sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- RHEL 8 and Rocky Linux 8/9

```
$ sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

- RHEL 9

```
$ sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

- b. Install the CUDA repository public GPG key.

```
$ distribution=$(. /etc/os-release;echo $ID`rpm -E "%{?rhel}%{?fedora}"`)
```

- c. 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
```

- d. Install the NVIDIA, CUDA drivers and cuDNN.

```
$ sudo yum clean all \  
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcuda8-devel
```

6. Reboot the instance and reconnect to it.
7. (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 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.

Ubuntu 20.04/22.04

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 2, CentOS 7, RHEL 7/8/9, and Rocky Linux 8/9

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 20.04/22.04

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, and Open MPI stack that is required to support EFA on your temporary instance.

To install the EFA software

1. Connect to the instance you launched. For more information, see [Connect to your Linux instance](#).
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.

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.32.0.tar.gz
```

You can also get the latest version by replacing the version number with `latest` in the preceding command.

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 AWS Support.

- c. Download the signature file and verify the signature of the EFA tarball file.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.32.0.tar.gz.sig && gpg --verify ./aws-efa-installer-1.32.0.tar.gz.sig
```

The following shows example output.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
```

```
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC
```

If the result includes `Good` signature, and the fingerprint matches the fingerprint returned in the previous step, proceed to the next step. If not, don't run the EFA installation script, and contact AWS Support.

4. Extract the files from the compressed `.tar.gz` file and navigate into the extracted directory.

```
$ tar -xf aws-efa-installer-1.32.0.tar.gz && cd aws-efa-installer
```

5. Run the EFA software installation script.

Note

From EFA 1.30.0, both Open MPI 4 and Open MPI 5 are installed by default. Unless you need Open MPI 5, we recommend that you install only Open MPI 4. The following command installs Open MPI 4 only. If you want to install Open MPI 4 and Open MPI 5, remove `--mpi=openmpi4`.

```
$ sudo ./efa_installer.sh -y --mpi=openmpi4
```

Libfabric is installed in the `/opt/amazon/efa` directory, while 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/nvcc.git && cd nvcc
```

3. Build and install NCCL and specify the CUDA installation directory.

```
$ sudo make -j src.build CUDA_HOME=/usr/local/cuda
```

Step 7: Install the aws-ofi-nccl plugin

The aws-ofi-nccl plugin maps NCCL's connection-oriented transport APIs to Libfabric's connection-less reliable interface. This enables you to use Libfabric as a network provider while running NCCL-based applications. For more information about the aws-ofi-nccl plugin, see the [aws-ofi-nccl repository](#).

To install the aws-ofi-nccl plugin

1. Navigate to your home directory.

```
$ cd $HOME
```

2. (*Amazon Linux 2 and Ubuntu only*) Install the required utilities.

- Amazon Linux 2

```
$ sudo yum install hwloc-devel
```

- Ubuntu 20.04

```
$ sudo apt-get install libhwloc-dev
```

3. Download the aws-ofi-nccl plugin files. The files are packaged into a compressed tarball (.tar.gz).

```
$ wget https://github.com/aws/aws-ofi-nccl/releases/download/v1.9.1-aws/aws-ofi-nccl-1.9.1-aws.tar.gz
```

4. Extract the files from the compressed .tar.gz file and navigate into the extracted directory.

```
$ tar -xf aws-ofi-nccl-1.9.1-aws.tar.gz && cd aws-ofi-nccl-1.9.1-aws
```

5. To generate the *make* files, run the configure script and specify the MPI, Libfabric, NCCL, and CUDA installation directories.

```
$ ./configure --prefix=/opt/aws-ofi-nccl --with-mpi=/opt/amazon/openmpi \  
--with-libfabric=/opt/amazon/efa \  
--with-cuda=/usr/local/cuda \  
--enable-platform-aws
```

6. Add the Open MPI directory to the PATH variable.

```
$ export PATH=/opt/amazon/openmpi/bin/:$PATH
```

7. Install the aws-ofi-nccl plugin.

```
$ make && sudo make install
```

Step 8: Install the NCCL tests

Install the NCCL tests. The NCCL tests enable you to confirm that NCCL is properly installed and that it is operating as expected. For more information about the NCCL tests, see the [nccl-tests repository](#).

To install the NCCL tests

1. Navigate to your home directory.

```
$ cd $HOME
```

2. Clone the official nccl-tests repository to the instance and navigate into the local cloned repository.

```
$ git clone https://github.com/NVIDIA/nccl-tests.git && cd nccl-tests
```

3. Add the Libfabric directory to the LD_LIBRARY_PATH variable.

- Amazon Linux, Amazon Linux 2, RHEL, Rocky Linux 8/9, and CentOS

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

- Ubuntu

```
$ 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 9: Test your EFA and NCCL configuration

Run a test to ensure that your temporary instance is properly configured for EFA and NCCL.

To test your EFA and NCCL configuration

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/aws-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/aws-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-24x1-topo.xml
```

Step 10: Install your machine learning applications

Install the machine learning applications on the temporary instance. The installation procedure varies depending on the specific machine learning application. For more information about installing software on your Linux instance, see [Managing Software on Your Linux Instance](#).

Note

Refer to your machine learning application's documentation for installation instructions.

Step 11: Create an EFA and NCCL-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions, Image, Create image**.
4. For **Create image**, do the following:
 - a. For **Image name**, enter a descriptive name for the AMI.
 - b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.
 - c. Choose **Create image**.
5. In the navigation pane, choose **AMIs**.
6. Locate the AMI that you created in the list. Wait for the status to change from pending to available before continuing to the next step.

Step 12: Terminate the temporary instance

At this point, you no longer need the temporary instance that you launched. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions, Instance state, Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

Step 13: Launch EFA and NCCL-enabled instances into a cluster placement group

Launch your EFA and NCCL-enabled instances into a cluster placement group using the EFA-enabled AMI and the EFA-enabled security group that you created earlier.

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 [Capacity Reservations in 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, and for **Elastic Fabric Adapter**, select **Enable**.
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 14: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

The key pair is created in the `$HOME/.ssh/` directory.

4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.
6. For each member node in the cluster, do the following:
 - a. Connect to the instance.
 - b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.
7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Use an AWS Deep Learning AMI

The following steps help you to get started with one of the following AWS Deep Learning AMIs:

- Deep Learning AMI (Amazon Linux 2)
- Deep Learning AMI (Ubuntu 20.04)

For more information, see the [AWS Deep Learning AMI User Guide](#).

Note

Only the `p3dn.24xlarge` and `p4d.24xlarge` instance types are supported.

Contents

- [Step 1: Prepare an EFA-enabled security group](#)
- [Step 2: Launch a temporary instance](#)
- [Step 3: Test your EFA and NCCL configuration](#)
- [Step 4: Install your machine learning applications](#)
- [Step 5: Create an EFA and NCCL-enabled AMI](#)
- [Step 6: Terminate the temporary instance](#)
- [Step 7: Launch EFA and NCCL-enabled instances into a cluster placement group](#)
- [Step 8: Enable passwordless SSH](#)

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 a supported **AWS Deep Learning AMI Version 25.0 or later**.
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, and for **Elastic Fabric Adapter**, select **Enable**.
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: 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/aws-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

```
$ /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/aws-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-24x1-topo.xml
```

Step 4: Install your machine learning applications

Install the machine learning applications on the temporary instance. The installation procedure varies depending on the specific machine learning application. For more information about installing software on your Linux instance, see [Managing Software on Your Linux Instance](#).

Note

Refer to your machine learning application's documentation for installation instructions.

Step 5: Create an EFA and NCCL-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions, Image, Create image**.
4. For **Create image**, do the following:
 - a. For **Image name**, enter a descriptive name for the AMI.
 - b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.
 - c. Choose **Create image**.

5. In the navigation pane, choose **AMIs**.
6. Locate the AMI that you created in the list. Wait for the status to change from pending to available before continuing to the next step.

Step 6: Terminate the temporary instance

At this point, you no longer need the temporary instance that you launched. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions, Instance state, Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

Step 7: Launch EFA and NCCL-enabled instances into a cluster placement group

Launch your EFA and NCCL-enabled instances into a cluster placement group using the EFA-enabled AMI and the EFA-enabled security group that you created earlier.

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 [Capacity Reservations in 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, and for **Elastic Fabric Adapter**, select **Enable**.
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 8: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

The key pair is created in the `$HOME/.ssh/` directory.

4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.
6. For each member node in the cluster, do the following:
 - a. Connect to the instance.
 - b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.
7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Work with EFA

You can create, use, and manage an EFA much like any other elastic network interface in Amazon EC2. However, unlike elastic network interfaces, EFAs cannot be attached to or detached from an instance in a running state.

EFA requirements

To use an EFA, you must do the following:

- Choose one of the [supported instance types](#).
- Use an AMI for one of the [supported operating systems](#).
- Install the EFA software components. For more information, see [Step 3: Install the EFA software](#) and [Step 5: \(Optional\) Install Intel MPI](#).
- Use a security group that allows all inbound and outbound traffic to and from the security group itself. For more information, see [Step 1: Prepare an EFA-enabled security group](#).

Contents

- [Create an EFA](#)
- [Attach an EFA to a stopped instance](#)
- [Attach an EFA when launching an instance](#)
- [Add an EFA to a launch template](#)
- [Manage IP addresses for an EFA](#)
- [Change the security group for an EFA](#)
- [Detach an EFA](#)
- [View EFAs](#)
- [Delete an EFA](#)

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.

To create a new EFA using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.
4. For **Description**, enter a descriptive name for the EFA.
5. For **Subnet**, select the subnet in which to create the EFA.
6. For **Private IP**, enter the primary private IPv4 address. If you don't specify an IPv4 address, we select an available private IPv4 address from the selected subnet.
7. (IPv6 only) If you selected a subnet that has an associated IPv6 CIDR block, you can optionally specify an IPv6 address in the **IPv6 IP** field.
8. For **Security groups**, select one or more security groups.
9. For **EFA**, choose **Enabled**.
10. Choose **Yes, Create**.

To create a new EFA using the AWS CLI

Use the [create-network-interface](#) command and for `interface-type`, specify `efa`, as shown in the following example.

```
aws ec2 create-network-interface --subnet-id subnet-01234567890 --  
description example_efa --interface-type efa
```

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 to an instance](#).

Attach an EFA when launching an instance

To attach an existing EFA when launching an instance (AWS CLI)

Use the [run-instances](#) command and for `NetworkInterfaceId`, specify the ID of the EFA, as shown in the following example.

```
aws ec2 run-instances --image-id ami_id --count 1 --instance-  
type c5n.18xlarge --key-name my_key_pair --network-interfaces  
DeviceIndex=0,NetworkInterfaceId=efa_id,Groups=sg_id,SubnetId=subnet_id
```

To attach a new EFA when launching an instance (AWS CLI)

Use the [run-instances](#) command and for **InterfaceType**, specify `efa`, as shown in the following example.

```
aws ec2 run-instances --image-id ami_id --count 1 --instance-  
type c5n.18xlarge --key-name my_key_pair --network-interfaces  
DeviceIndex=0,InterfaceType=efa,Groups=sg_id,SubnetId=subnet_id
```

Add an EFA to a launch template

You can create a launch template that contains the configuration information needed to launch EFA-enabled instances. To create an EFA-enabled launch template, create a new launch template and specify a supported instance type, your EFA-enabled AMI, and an EFA-enabled security group. For more information, see [Get started with EFA and MPI](#).

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 a launch template](#).

Manage IP addresses for an EFA

You can change the IP addresses associated with an EFA. If you have an Elastic IP address, you can associate it with an EFA. If your EFA is provisioned in a subnet that has an associated IPv6 CIDR block, you can assign one or more IPv6 addresses to the EFA.

You assign an Elastic IP (IPv4) and IPv6 address to an EFA in the same way that you assign an IP address to an elastic network interface. For more information, see [Managing IP addresses](#).

Change the security group for an EFA

You can change the security group that is associated with an EFA. To enable OS-bypass functionality, the EFA must be a member of a security group that allows all inbound and outbound traffic to and from the security group itself.

You change the security group that is associated with an EFA in the same way that you change the security group that is associated with an elastic network interface. For more information, see [Changing the security group](#).

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 from an instance](#).

View EFAs

You can view all of the EFAs in your account.

You view EFAs in the same way that you view elastic network interfaces. For more information, see [View details about 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 EFA

You can use the following features to monitor the performance of your Elastic Fabric Adapters.

Amazon VPC flow logs

You can create an Amazon VPC Flow Log to capture information about the traffic going to and from an EFA. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you create a flow log, you can retrieve and view its data in the chosen destination. For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

You create a flow log for an EFA in the same way that you create a flow log for an elastic network interface. For more information, see [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

Amazon CloudWatch provides metrics that enable you to monitor your EFAs in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For more information, see [Monitor your instances using CloudWatch](#).

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 AWS Support.

For example, the following command verifies the EFA 1.9.4 tarball using the SHA256 checksum.

```
$ sha256sum aws-efa-installer-1.9.4.tar.gz
```

```
1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14 aws-efa-
installer-1.9.4.tar.gz
```

The following table lists the checksums for recent versions of EFA.

Version	Download URL	Checksums
EFA 1.32.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.32.0.tar.gz	<p>MD5: db8d65cc028d8d08b5a9f2d88881c1b1</p> <p>SHA256: 5f7233760be57f6fee6de8c09acbfbf59238de848e06048dc54d156ef578fc66</p>
EFA 1.31.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.31.0.tar.gz	<p>MD5: 856352f12bef2ccbadcd75e35aa52aaf</p> <p>SHA256: 943325bd37902a4300ac9e5715163537d56ecb4e7b87b37827c3e547aa1897bf</p>
EFA 1.30.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.30.0.tar.gz	<p>MD5: 31f48e1a47fe93ede8ebd273fb747358</p> <p>SHA256: 876ab9403e07a0c3c91a1a34685a52eced890ae052df94857f6081c5f6c78a0a</p>

Version	Download URL	Checksums
EFA 1.29.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.29.1.tar.gz	MD5: e1872ca815d752c1d7c2b5c175e52a16 SHA256: 178b263b8c25845b63dc93b25bcdff5870df5204ec509af26f43e8d283488744
EFA 1.29.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.29.0.tar.gz	MD5: 39d06a002154d94cd982ed348133f385 SHA256: 836655f87015547e733e7d9f7c760e4e24697f8bbc261bb5f3560abd4206bc36
EFA 1.28.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.28.0.tar.gz	MD5: 9dc13b7446665822605e66febe074035 SHA256: 2e625d2d6d3e073b5178e8e861891273d896b66d03cb1a32244fd56789f1c435
EFA 1.27.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.27.0.tar.gz	MD5: 98bfb515ea3e8d93f554020f3837fa15 SHA256: 1d49a97b0bf8d964d91652a79ac851f2550e33a5bf9d0cf86ec9357ff6579aa3

Version	Download URL	Checksums
EFA 1.26.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.26.1.tar.gz	MD5: 884e74671fdef4725501f7cd2d451d0c SHA256: c616994c924f54ebfabfab32b7fe8ac56947fae00a0ff453d975e298d174fc96
EFA 1.26.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.26.0.tar.gz	MD5: f8839f12ff2e3b9ba09ae8a82b30e663 SHA256: bc1abc1f76e97d204d3755d2a9ca307fc423e51c63141f798c2f15be3715aa11
EFA 1.25.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.25.1.tar.gz	MD5: 6d876b894547847a45bb8854d4431f18 SHA256: d2abc553d22b89a4ce92882052c1fa6de450d3a801fe005da718b7d4b9602b06
EFA 1.25.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.25.0.tar.gz	MD5: 1993836ca749596051da04694ea0d00c SHA256: 98b7b26ce031a2d6a93de2297cc71b03af647194866369ca53b60d82d45ad342

Version	Download URL	Checksums
EFA 1.24.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.24.1.tar.gz	MD5: 211b249f39d53086f3cb0c07665f4e6f SHA256: 120cfeec233af0955623ac7133b674143329f9561a9a8193e473060f596aec62
EFA 1.24.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.24.0.tar.gz	MD5: 7afe0187951e2dd2c9cc4b572e62f924 SHA256: 878623f819a0d9099d76ecd41cf4f569d4c3aac0c9bb7ba9536347c50b6bf88e
EFA 1.23.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.23.1.tar.gz	MD5: 22491e114b6ee7160a8290145dca0c28 SHA256: 5ca848d8e0ff4d1571cd443c36f8d27c8cdf2a0c97e9068ebf000c303fc40797
EFA 1.23.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.23.0.tar.gz	MD5: 38a6d7c1861f5038dba4e441ca7683ca SHA256: 555d497a60f22e3857fdeb3dfc53aa86d05926023c68c916d15d2dc3df6525bd

Version	Download URL	Checksums
EFA 1.22.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.22.1.tar.gz	MD5: 600c0ad7cdbc06e8e846cb763f92901b SHA256: f90f3d5f59c031b9a964466b5401e86fd0429272408f6c207c3f9048254e9665
EFA 1.22.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.22.0.tar.gz	MD5: 8f100c93dc8ab519c2aeb5dab89e98f8 SHA256: f329e7d54a86a03ea51da6ea9a5b68fb354fbae4a57a02f9592e21fce431dc3a
EFA 1.21.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.21.0.tar.gz	MD5: 959ccc3a4347461909ec02ed3ba7c372 SHA256: c64e6ca34ccfc3ebe8e82d08899ae8442b3ef552541cf5429c43d11a04333050
EFA 1.20.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.20.0.tar.gz	MD5: 7ebfbb8e85f1b94709df4ab3db47913b SHA256: aeefd2681ffd5c4c631d1502867db5b831621d6eb85b61fe3ec80df983d1dcf0

Version	Download URL	Checksums
EFA 1.19.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.19.0.tar.gz	MD5: 2fd45324953347ec55 18da7e3fefaf0ec SHA256: 99b77821b9e72c8dea 015cc92c96193e8db3 07deee05b91a58094c c331f16709
EFA 1.18.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.18.0.tar.gz	MD5: fc2571a72f5d3c7b7b 576ce2de38d91e SHA256: acb18a0808aedb9a5e 485f1469225b9ac97f 21db9af78e4cd69397 00debe1cb6
EFA 1.17.3	https://efa-installer.amazonaws.com/aws-efa-installer-1.17.3.tar.gz	MD5: 0517df4a190356ab55 9235147174cafd SHA256: 5130998b0d2883bbae 189b21ab215ecbc1b0 1ae0231659a9b4a17b 0a33ebc6ca
EFA 1.17.2	https://efa-installer.amazonaws.com/aws-efa-installer-1.17.2.tar.gz	MD5: a329dedab53c4832df 218a24449f4c9a SHA256: bca1fdde8b32b00346 e175e597ffab32a09a 08ee9ab136875fb382 83cc4cd099

Version	Download URL	Checksums
EFA 1.17.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.17.1.tar.gz	MD5: 733ae2cfc9d14b52017eaf0a2ab6b0ff SHA256: f29322640a88ae9279805993cb836276ea240623820848463ca686c8ce02136f
EFA 1.17.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.17.0.tar.gz	MD5: d430fc841563c11c3805c5f82a4746b1 SHA256: 75ab0cee4fb6bd38889dce313183f5d3a83bd233e0a6ef6205d8352821ea901d
EFA 1.16.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.16.0.tar.gz	MD5: 399548d3b0d2e812d74dd67937b696b4 SHA256: cecec36495a1bc6fdc82f97761a541e4fb6c9a3cbf3cfcb145acf25ea5dbd45b
EFA 1.15.2	https://efa-installer.amazonaws.com/aws-efa-installer-1.15.2.tar.gz	MD5: 955fea580d5170b05823d51acde7ca21 SHA256: 84df4fbc1b3741b6c073176287789a601a589313accc8e6653434e8d4c20bd49

Version	Download URL	Checksums
EFA 1.15.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.15.1.tar.gz	MD5: c4610267039f72bbe4e35d7bf53519bc SHA256: be871781a1b9a15fca342a9d169219260069942a8bda7a8ad06d4baeb5e2efd7
EFA 1.15.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.15.0.tar.gz	MD5: 9861694e1cc00d884fadac07d22898be SHA256: b329862dd5729d2d098d0507fb486bf859d7c70ce18b61c302982234a3a5c88f
EFA 1.14.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.14.1.tar.gz	MD5: 50ba56397d359e57872fde1f74d4168a SHA256: c7b1b48e86fe4b3eaa4299d3600930919c4fe6d88cc6e2c7e4a408a3f16452c7
EFA 1.14.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.14.0.tar.gz	MD5: 40805e7fd842c36ececb9fd7f921b1ae SHA256: 662d62c12de85116df33780d40e0533ef7dad92709f4f613907475a7a1b60a97

Version	Download URL	Checksums
EFA 1.13.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.13.0.tar.gz	MD5: c91d16556f4fd53bec adbb345828221e SHA256: ad6705eb23a3fce44a f3afc0f76430915956 53a723ad0374084f4f 2b715192e1
EFA 1.12.3	https://efa-installer.amazonaws.com/aws-efa-installer-1.12.3.tar.gz	MD5: 818aee81f097918cfa ebd724eddea678 SHA256: 2c225321824788b8ca 3fbc118207b944cdb0 96b847e1e0d1d853ef 2f0d727172
EFA 1.12.2	https://efa-installer.amazonaws.com/aws-efa-installer-1.12.2.tar.gz	MD5: 956bb1fc5ae0d6f0f8 7d2e481d49fccf SHA256: 083a868a2c212a5a4f cf3e4d732b685ce39c ceb3ca7e5d50d0b74e 7788d06259
EFA 1.12.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.12.1.tar.gz	MD5: f5bfe52779df435188 b0a2874d0633ea SHA256: 5665795c2b4f09d5f3 f767506d4d4c429695 b36d4a17e5758b27f0 33aee58900

Version	Download URL	Checksums
EFA 1.12.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.12.0.tar.gz	MD5: d6c6b49fafb39b770297e1cc44fe68a6 SHA256: 28256c57e9ecc0b0778b41c1f777a9982b4e8eae782343dfe1246079933dca59
EFA 1.11.2	https://efa-installer.amazonaws.com/aws-efa-installer-1.11.2.tar.gz	MD5: 2376cf18d1353a4551e35c33d269c404 SHA256: a25786f98a3628f7f54f7f74ee2b39bc6734ea9374720507d37d3e8bf8ee1371
EFA 1.11.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.11.1.tar.gz	MD5: 026b0d9a0a48780cc7406bd51997b1c0 SHA256: 6cb04baf5ffc58ddf319e956b5461289199c8dd805fe216f8f9ab8d102f6d02a
EFA 1.11.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.11.0.tar.gz	MD5: 7d9058e010ad65bf2e14259214a36949 SHA256: 7891f6d45ae33e822189511c4ea1d14c9d54d000f6696f97be54e915ce2c9dfa

Version	Download URL	Checksums
EFA 1.10.1	https://efa-installer.amazonaws.com/aws-efa-installer-1.10.1.tar.gz	MD5: 78521d3d668be22976 f46c6fecc7b730 SHA256: 61564582de7320b21d e319f532c3a677d26c c46785378eb3b95c63 6506b9bcb4
EFA 1.10.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.10.0.tar.gz	MD5: 46f73f5a7afe41b4bb 918c81888fef9 SHA256: 136612f96f2a085a7d 98296da0afb6fa807b 38142e2fc0c548fa98 6c41186282
EFA 1.9.5	https://efa-installer.amazonaws.com/aws-efa-installer-1.9.5.tar.gz	MD5: 95edb8a209c18ba8d2 50409846eb6ef4 SHA256: a4343308d7ea4dc943 ccc21bcebed913e886 8e59bfb2ac93599c61 a7c87d7d25
EFA 1.9.4	https://efa-installer.amazonaws.com/aws-efa-installer-1.9.4.tar.gz	MD5: f26dd5c350422c1a98 5e35947fa5aa28 SHA256: 1009b5182693490d90 8ef0ed2c1dd4f813cc 310a5d2062ce9619c4 c12b5a7f14

Version	Download URL	Checksums
EFA 1.9.3	https://efa-installer.amazonaws.com/aws-efa-installer-1.9.3.tar.gz	MD5: 95755765a097802d3e6d5018d1a5d3d6 SHA256: 46ce732d6f3fcc9edf6a6e9f9df0ad136054328e24675567f7029edab90c68f1
EFA 1.8.4	https://efa-installer.amazonaws.com/aws-efa-installer-1.8.4.tar.gz	MD5: 85d594c41e831afc6c9305263140457e SHA256: 0d974655a09b213d7859e658965e56dc4f23a0eee2dc44bb41b6d039cc5bab45

Amazon EC2 instance topology

Describing your instance topology provides a hierarchical view of the relative proximity between 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 use [Capacity Reservations in cluster placement groups](#).

Pricing

There is no additional cost to describe your instance topology.

Contents

- [How instance topology works](#)
- [Prerequisites for instance topology](#)

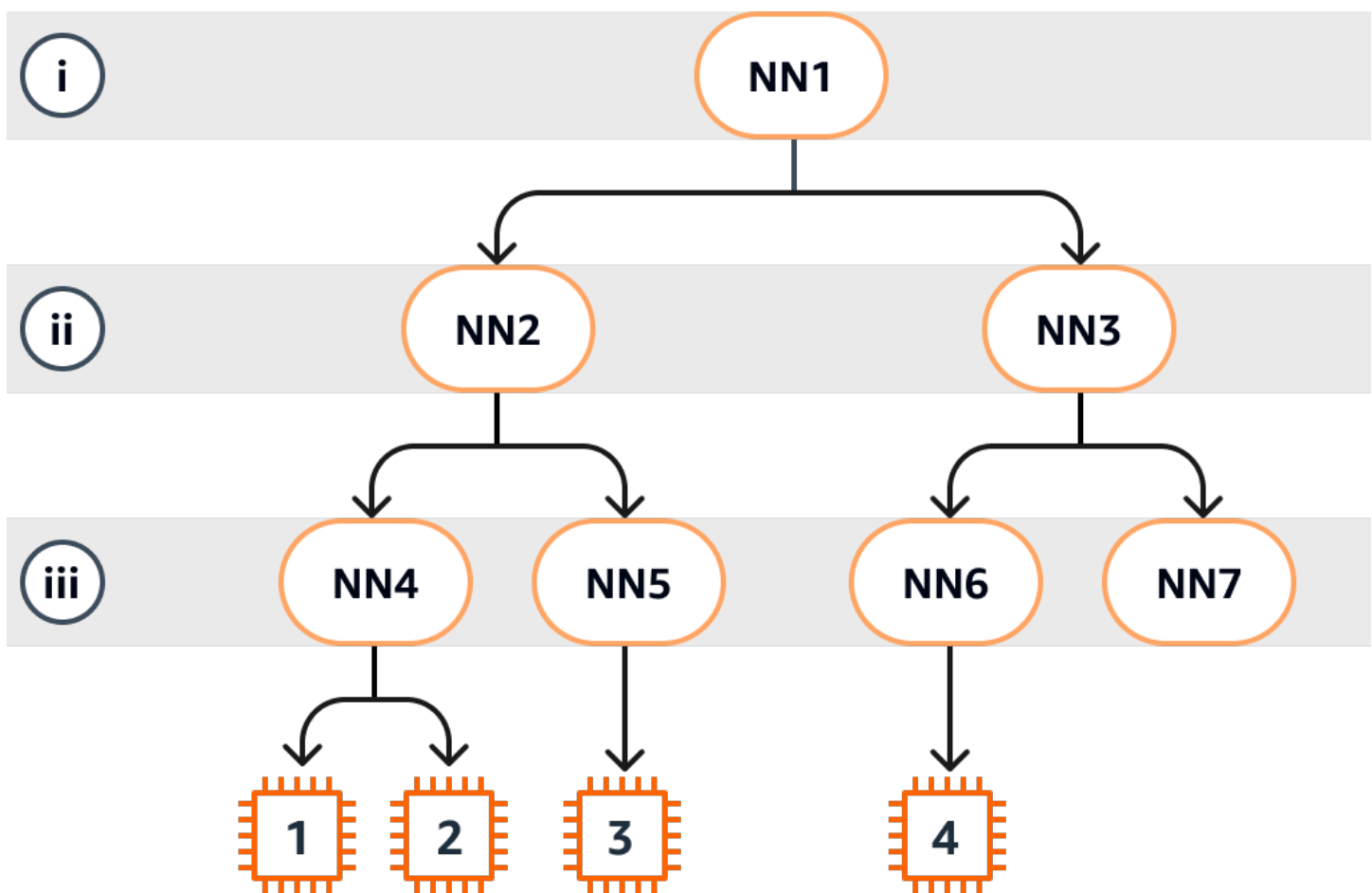
- [Amazon EC2 instance topology examples](#)

How instance topology works

Every EC2 instance connects to a node set. A node set comprises three network nodes, with each node representing a different layer in the AWS network. The network layers are arranged in a hierarchy of 3 or more layers. The node set provides the top-down view of this hierarchy, with the bottom layer connected closest to an instance.

The information about the node set is called the *instance topology*.

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**. 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.

```
{
  "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
      ],
      "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
      ],
      "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
    ],
    "ZoneId": "usw2-az2",
    "AvailabilityZone": "us-west-2a"
  },
  {
    "InstanceId": "i-4444444444example", //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
    ],
    "ZoneId": "usw2-az2",
    "AvailabilityZone": "us-west-2a"
  }
],
"NextToken": "SomeEncryptedToken"
}

```

Limitations

The following limitations apply:

- The instances must be in the running state.
- Each instance topology view is unique per account.
- The AWS Management Console does not support viewing the instance topology.

Prerequisites for 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](#)

AWS Regions

Supported AWS Regions:

- US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon)
- Asia Pacific (Seoul), Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt), Europe (Ireland), Europe (Stockholm)

Instance types

Supported instance types:

- 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
- trn1.2xlarge | trn1.32xlarge | trn1n.32xlarge

To see the available instance types in a specific Region

The available instance types vary by Region. To see if an instance type is available in a Region, use the [describe-instance-types-offerings](#) command with the `--region` parameter. Include the `--filters` parameter to scope the results to the instance family or instance type that you're interested in and the `--query` parameter to scope the output to the value of `InstanceType`.

```
aws ec2 describe-instance-type-offerings \  
  --region us-east-2 \  
  --filters 'Name=instance-type, Values=trn1*' \  
  --query 'InstanceTypeOfferings[].InstanceType'
```

Expected output

```
[
```

```
"trn1.2xlarge",  
"trn1.32xlarge",  
"trn1n.32xlarge"  
]
```

Instance state

Instances must be in the `running` state. You can't get instance topology information for instances that are in another state.

Amazon EC2 instance topology examples

You can use the [describe-instance-topology](#) CLI 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 [Specify the Region for a resource](#).

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](#) in the *AWS CLI Command Reference*.

Require permissions

The following permission is required to describe instance topology:

- `ec2:DescribeInstanceTopology`

Examples

- [Example 1 - No parameters or filters](#)
- [Example 2 - instance-type filter](#)

- [Example 2a – Exact match filter for a specified instance type](#)
- [Example 2b – Wild card filter for an instance family](#)
- [Example 2c – Combined instance family and exact match filters](#)
- [Example 3 – zone-id filter](#)
 - [Example 3a – Availability Zone filter](#)
 - [Example 3b – Local Zone filter](#)
 - [Example 3c – Combined Availability Zone and Local Zone filters](#)
- [Example 4 – Combined instance-type and zone-id filters](#)
- [Example 5 – Placement group name parameter](#)
- [Example 6 – Instance IDs](#)

Example 1 - No parameters or filters

To describe the instance topology of all your instances

Use the [describe-instance-topology](#) CLI command without specifying any parameters or filters.

```
aws ec2 describe-instance-topology --region us-west-2
```

The response returns only the instances that match the supported instance types for this API. The instances can be in different Availability Zones, Local Zones (ZoneId), and placement groups (GroupName). If an instance is not in a placement group, the GroupName field does not appear in the output. In the following example output, only one instance is in a placement group.

Example output

```
{
  "Instances": [
    {
      "InstanceId": "i-1111111111example",
      "InstanceType": "p4d.24xlarge",
      "GroupName": "my-ml-cpg",
      "NetworkNodes": [
        "nn-1111111111example",
        "nn-2222222222example",
        "nn-3333333333example"
      ],
      "ZoneId": "usw2-az2",
```

```
    "AvailabilityZone": "us-west-2a"
  },
  {
    "InstanceId": "i-2222222222example",
    "InstanceType": "p4d.24xlarge",
    "NetworkNodes": [
      "nn-1111111111example",
      "nn-2222222222example",
      "nn-3333333333example"
    ],
    "ZoneId": "usw2-az2",
    "AvailabilityZone": "us-west-2a"
  },
  {
    "InstanceId": "i-3333333333example",
    "InstanceType": "trn1.32xlarge",
    "NetworkNodes": [
      "nn-1212121212example",
      "nn-1211122211example",
      "nn-1311133311example"
    ],
    "ZoneId": "usw2-az4",
    "AvailabilityZone": "us-west-2d"
  },
  {
    "InstanceId": "i-4444444444example",
    "InstanceType": "trn1.2xlarge",
    "NetworkNodes": [
      "nn-1111111111example",
      "nn-5434334334example",
      "nn-1235301234example"
    ],
    "ZoneId": "usw2-az2",
    "AvailabilityZone": "us-west-2a"
  }
],
"NextToken": "SomeEncryptedToken"
}
```

Example 2 – 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 2a – Exact match filter for a specified instance type

To describe the instance topology of all your instances that match a specified instance type

Use the [describe-instance-topology](#) CLI command with the `instance-type` filter. In this example, the output is filtered for `trn1n.32xlarge` instances. The response will return only the instances that match the specified instance type.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --filters Name=instance-type,Values=trn1n.32xlarge
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "trn1n.32xlarge",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"  
}
```

Example 2b – Wild card filter for an instance family

To describe the instance topology of all your instances that match an instance family

Use the [describe-instance-topology](#) CLI command with the `instance-type` filter. In this example, the output is filtered for `trn1*` instances. The response will return only the instances that match the specified instance family.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --filters Name=instance-type,Values=trn1*
```

```
--filters Name=instance-type,Values="trn1*"
```

Example output

```
{
  "Instances": [
    {
      "InstanceId": "i-2222222222example",
      "InstanceType": "trn1n.32xlarge",
      "NetworkNodes": [
        "nn-1111111111example",
        "nn-2222222222example",
        "nn-3333333333example"
      ],
      "ZoneId": "usw2-az2",
      "AvailabilityZone": "us-west-2a"
    },
    {
      "InstanceId": "i-3333333333example",
      "InstanceType": "trn1.32xlarge",
      "NetworkNodes": [
        "nn-1212121212example",
        "nn-1211122211example",
        "nn-1311133311example"
      ],
      "ZoneId": "usw2-az4",
      "AvailabilityZone": "us-west-2d"
    },
    {
      "InstanceId": "i-4444444444example",
      "InstanceType": "trn1.2xlarge",
      "NetworkNodes": [
        "nn-1111111111example",
        "nn-5434334334example",
        "nn-1235301234example"
      ],
      "ZoneId": "usw2-az2",
      "AvailabilityZone": "us-west-2a"
    }
  ],
  "NextToken": "SomeEncryptedToken"
}
```

Example 2c – Combined instance family and exact match filters

To describe the instance topology of all your instances that match an instance family or specified instance type

Use the [describe-instance-topology](#) CLI command with the `instance-type` filter. In this example, the output is filtered for `pd4d*` or `trn1n.32xlarge` instances. The response will return the instances that match any of the specified filters.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge"
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "ML-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "trn1n.32xlarge",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-4343434343example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"
```

```
}
```

Example 3 – 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 3a – Availability Zone filter

To describe the instance topology of all your instances that match a specified Availability Zone

Use the [describe-instance-topology](#) CLI command with the `zone-id` filter. In this example, the output is filtered for the `us-west-2a` Availability Zone. The response will return only the instances that match the specified Availability Zone.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --filters Name=zone-id,Values="us-west-2a"
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "trn1n.32xlarge",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3214313214example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"  
}
```

Example 3b – Local Zone filter

To describe the instance topology of all your instances that match a specified Local Zone

Use the [describe-instance-topology](#) CLI command with the `zone-id` filter. In this example, the output is filtered for the `usw2-az2` Local Zone. The response will return only the instances that match the specified Local Zone.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --filters Name=zone-id,Values=usw2-az2
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "ML-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"  
}
```

Example 3c – Combined Availability Zone and Local Zone filters

To describe the instance topology of all your instances that match a specified Availability Zone or Local Zone

Use the [describe-instance-topology](#) CLI command with the `zone-id` filter. In this example, the output is filtered for the `us-west-2a` Availability Zone and `usw2-az2` Local Zone. The response will return the instances that match any of the specified filters.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --filters "Name=zone-id,Values=us-west-2a,usw2-az2"
```

Example output

```
{
  "Instances": [
    {
      "InstanceId": "i-1111111111example",
      "InstanceType": "p4d.24xlarge",
      "GroupName": "ML-group",
      "NetworkNodes": [
        "nn-1111111111example",
        "nn-2222222222example",
        "nn-3333333333example"
      ],
      "ZoneId": "usw2-az2",
      "AvailabilityZone": "us-west-2a"
    },
    {
      "InstanceId": "i-2222222222example",
      "InstanceType": "trn1n.32xlarge",
      "NetworkNodes": [
        "nn-1111111111example",
        "nn-2222222222example",
        "nn-3214313214example"
      ],
      "ZoneId": "usw2-az2",
      "AvailabilityZone": "us-west-2a"
    }
  ],
  "NextToken": "SomeEncryptedToken"
}
```

Example 4 – Combined instance-type and zone-id filters

You can combine all the filters in a single command.

To describe the instance topology of all your instances that match a specified instance type, instance family, Availability Zone, or Local Zone

Use the [describe-instance-topology](#) CLI command with the `instance-type` and `zone-id` filters. In this example, the output is filtered for the `p4d*` instance family, `trn1n.32xlarge` instance type, `us-west-2a` Availability Zone, and `usw2-az2` Local Zone. The response will return the instances that match `p4d*` or `trn1n.32xlarge` instances in the `us-west-2a` or `usw2-az2` zones.

```
aws ec2 describe-instance-topology \
```

```
--region us-west-2 \  
--filters "Name=instance-type,Values=p4d*,trn1n.32xlarge" "Name=zone-id,Values=us-west-2a,usw2-az2"
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "ML-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "trn1n.32xlarge",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3214313214example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"  
}
```

Example 5 – Placement group name parameter

To describe the instance topology of all your instances in a specified placement group

Use the [describe-instance-topology](#) CLI command with the group-names parameter. In the following example, the instances can be in the ML-group or HPC-group placement group. The response will return instances that are in either of the placement groups.

```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --group-names ML-group HPC-group
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "ML-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "trn1n.32xlarge",  
      "GroupName": "HPC-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3214313214example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"  
}
```

Example 6 – Instance IDs

To describe the instance topology of specified instances

Use the [describe-instance-topology](#) CLI command with the `--instance-ids` parameter. The response will return the instances that match the specified instance IDs.


```
aws ec2 describe-instance-topology \  
  --region us-west-2 \  
  --instance-ids i-1111111111example i-2222222222example
```

Example output

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "ML-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "trn1n.32xlarge",  
      "GroupName": "HPC-group",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3214313214example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    }  
  ],  
  "NextToken": "SomeEncryptedToken"  
}
```

Placement groups

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.

There is no charge for creating a placement group.

Placement strategies

You can create a placement group using one of the following placement strategies.

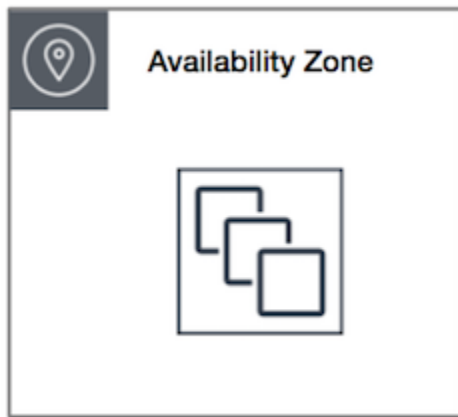
Placement strategies:

- [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. 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.

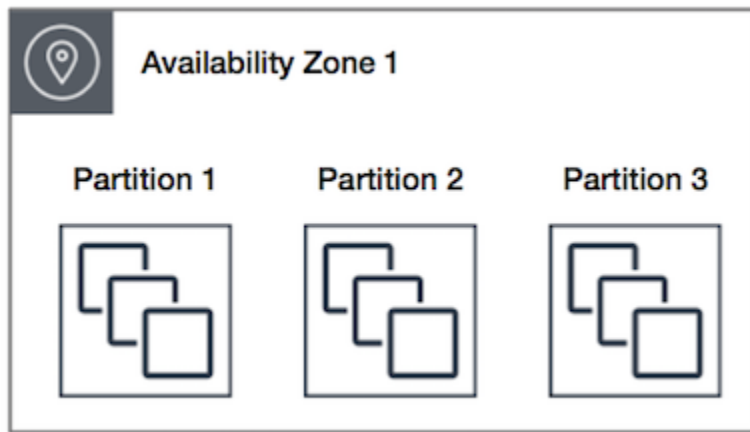
If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Starting the instances may migrate them to hardware that has capacity for all of the requested instances.

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical

segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—**Partition 1**, **Partition 2**, and **Partition 3**. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.



Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.

A partition placement group can have partitions in multiple Availability Zones in the same Region. A partition placement group can have a maximum of seven partitions per Availability Zone. The number of instances that can be launched into a partition placement group is limited only by the limits of your account.

In addition, partition placement groups offer visibility into the partitions — you can see which instances are in which partitions. You can share this information with topology-aware applications, such as HDFS, HBase, and Cassandra. These applications use this information to make intelligent data replication decisions for increasing data availability and durability.

If you start or launch an instance in a partition placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

Spread placement groups

A spread placement group is a group of instances that are each placed on distinct 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”](#).

Placement group rules and limitations

Topics

- [General rules and limitations](#)
- [Cluster placement group rules and limitations](#)
- [Partition placement group rules and limitations](#)
- [Spread placement group rules and limitations](#)

General rules and limitations

Before you use placement groups, be aware of the following rules:

- You can create a maximum of 500 placement groups per account in each Region.
- The name that you specify for a placement group must be unique within your AWS account for the Region.
- You can't merge placement groups.
- An instance can be launched in one placement group at a time; it cannot span multiple placement groups.
- [On-Demand Capacity 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.

If you plan to launch instances into a *cluster* placement group, we recommend that you reserve capacity explicitly in the cluster placement group. You can do this by creating an [On-Demand Capacity Reservation in a specified cluster placement group](#). Note that while you can reserve capacity in this way using On-Demand Capacity Reservations, the same can't be done with zonal Reserved Instances as they can't reserve capacity explicitly in 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.

Cluster placement group 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.
- 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 group rules and limitations

The following rules apply to partition placement groups:

- A partition placement group supports a maximum of seven partitions per Availability Zone. The number of instances that you can launch in a partition placement group is limited only by your account limits.

- When instances are launched into a partition placement group, Amazon EC2 tries to evenly distribute the instances across all partitions. Amazon EC2 doesn't guarantee an even distribution of instances across all partitions.
- A partition placement group with Dedicated Instances can have a maximum of two partitions.
- Capacity Reservations do not reserve capacity in a partition placement group.

Spread placement group 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.

Work with placement groups

Contents

- [Create a placement group](#)
- [View placement group information](#)
- [Tag a placement group](#)
- [Launch instances in a placement group](#)

- [Describe instances in a placement group](#)
- [Change the placement group for an instance](#)
- [Remove an instance from a placement group](#)
- [Delete a placement group](#)

Create a placement group

You can create a placement group using one of the following methods.

Console

To create a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**.
3. Choose **Create placement group**.
4. Specify a name for the group.
5. Choose the placement strategy for the group.
 - If you choose **Spread**, choose the spread level.
 - **Rack** - no restrictions
 - **Host** - only for Outposts
 - If you choose **Partition**, choose the number of partitions within the group.
6. To tag the placement group, choose **Add tag**, and then enter a key and value. Choose **Add tag** for each tag that you want to add.
7. Choose **Create group**.

AWS CLI

To create a placement group using the AWS CLI

Use the [create-placement-group](#) command. The following example creates a placement group named `my-cluster` that uses the `cluster` placement strategy, and it applies a tag with a key of `purpose` and a value of `production`.

```
aws ec2 create-placement-group \
```

```
--group-name my-cluster \  
--strategy cluster \  
--tag-specifications 'ResourceType=placement-  
group,Tags={Key=purpose,Value=production}'
```

To create a partition placement group using the AWS CLI

Use the [create-placement-group](#) command. Specify the `--strategy` parameter with the value `partition`, and specify the `--partition-count` parameter with the desired number of partitions. In this example, the partition placement group is named `HDFS-Group-A` and is created with five partitions.

```
aws ec2 create-placement-group \  
  --group-name HDFS-Group-A \  
  --strategy partition \  
  --partition-count 5
```

PowerShell

To create a placement group using the AWS Tools for Windows PowerShell

Use the [New-EC2PlacementGroup](#) command.

View placement group information

You can view all your placement groups and the information about them using one of the following methods.

Console

To view information about one or more placement groups

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Network & Security**, choose **Placement Groups**.
3. In the **Placement groups** table, for each placement group, you can view the following information:
 - **Group name** – The name that you gave to the placement group.
 - **Group Id** – The ID of the placement group.
 - **Strategy** – The placement strategy for the placement group.

- **State** – The state of the placement group.
- **Partition** – The number of partitions. Valid only if the strategy is **partition**.
- **Group ARN** – The Amazon Resource Name (ARN) of the placement group.

AWS CLI

To describe all your placement groups

Use the [describe-placement-groups](#) AWS CLI command.

```
aws ec2 describe-placement-groups
```

Example response

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster-pg",
      "State": "available",
      "Strategy": "cluster",
      "GroupId": "pg-0123456789example",
      "GroupArn": "arn:aws:ec2:eu-west-1:111111111111:placement-group/my-
cluster-pg"
    },
    ...
  ]
}
```

To describe a specified placement group

Use the [describe-placement-groups](#) AWS CLI command. You can specify either the `--group-id` or the `--group-name` parameter.

Specify placement group ID:

```
aws ec2 describe-placement-groups --group-id pg-0123456789example
```

Specify placement group name:

```
aws ec2 describe-placement-groups --group-name my-cluster-pg
```

Example response

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster-pg",
      "State": "available",
      "Strategy": "cluster",
      "GroupId": "pg-0123456789example",
      "GroupArn": "arn:aws:ec2:eu-west-1:111111111111:placement-group/my-
cluster-pg"
    }
  ]
}
```

Tag a placement group

To help categorize and manage your existing placement groups, you can tag them with custom metadata. For more information about how tags work, see [Tag your Amazon EC2 resources](#).

When you tag a placement group, the instances that are launched into the placement group are not automatically tagged. You need to explicitly tag the instances that are launched into the placement group. For more information, see [Add a tag when you launch an instance](#).

You can view, add, and delete tags using one of the following methods.

Console

To view, add, or delete a tag for an existing placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**.
3. Select a placement group, and then choose **Actions, Manage tags**.
4. The **Manage tags** screen displays any tags that are assigned to the placement group.
 - To add a tag, choose **Add tag**, and then enter the tag key and value. You can add up to 50 tags per placement group. For more information, see [Tag restrictions](#).
 - To delete a tag, choose **Remove** next to the tag that you want to delete.
5. Choose **Save**.

AWS CLI

To view placement group tags

Use the [describe-tags](#) command to view the tags for the specified resource. In the following example, you describe the tags for all of your placement groups.

```
aws ec2 describe-tags \  
  --filters Name=resource-type,Values=placement-group
```

```
{  
  "Tags": [  
    {  
      "Key": "Environment",  
      "ResourceId": "pg-0123456789EXAMPLE",  
      "ResourceType": "placement-group",  
      "Value": "Production"  
    },  
    {  
      "Key": "Environment",  
      "ResourceId": "pg-9876543210EXAMPLE",  
      "ResourceType": "placement-group",  
      "Value": "Production"  
    }  
  ]  
}
```

You can also use the [describe-tags](#) command to view the tags for a placement group by specifying its ID. In the following example, you describe the tags for `pg-0123456789EXAMPLE`.

```
aws ec2 describe-tags \  
  --filters Name=resource-id,Values=pg-0123456789EXAMPLE
```

```
{  
  "Tags": [  
    {  
      "Key": "Environment",  
      "ResourceId": "pg-0123456789EXAMPLE",  
      "ResourceType": "placement-group",  
      "Value": "Production"  
    }  
  ]  
}
```

```
]
}
```

You can also view the tags of a placement group by describing the placement group.

Use the [describe-placement-groups](#) command to view the configuration of the specified placement group, which includes any tags that were specified for the placement group.

```
aws ec2 describe-placement-groups \
  --group-name my-cluster
```

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster",
      "State": "available",
      "Strategy": "cluster",
      "GroupId": "pg-0123456789EXAMPLE",
      "Tags": [
        {
          "Key": "Environment",
          "Value": "Production"
        }
      ]
    }
  ]
}
```

To tag an existing placement group using the AWS CLI

You can use the [create-tags](#) command to tag existing resources. In the following example, the existing placement group is tagged with Key=Cost-Center and Value=CC-123.

```
aws ec2 create-tags \
  --resources pg-0123456789EXAMPLE \
  --tags Key=Cost-Center,Value=CC-123
```

To delete a tag from a placement group using the AWS CLI

You can use the [delete-tags](#) command to delete tags from existing resources. For examples, see [Examples](#) in the *AWS CLI Command Reference*.

PowerShell

To view placement group tags

Use the [Get-EC2Tag](#) command.

To describe the tags for a specific placement group

Use the [Get-EC2PlacementGroup](#) command.

To tag an existing placement group

Use the [New-EC2Tag](#) command.

To delete a tag from a placement group

Use the [Remove-EC2Tag](#) command.

Launch instances in a placement group

You can launch an instance into a placement group if the [placement group rules and limitations are met](#) using one of the following methods.

Console

To launch instances into a placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**. Complete the form as directed, taking care to do the following:
 - Under **Instance type**, select an instance type that can be launched into a placement group.
 - In the **Summary** box, under **Number of instances**, enter the total number of instances that you need in this placement group, because you might not be able to add instances to the placement group later.
 - Under **Advanced details**, for **Placement group name**, you can choose to add the instances to a new or existing placement group. If you choose a placement group with a partition strategy, for **Target partition**, choose the partition in which to launch the instances.

AWS CLI

To launch instances into a placement group

Use the [run-instances](#) command and specify the placement group name using the `--placement "GroupName = my-cluster"` parameter. In this example, the placement group is named `my-cluster`.

```
aws ec2 run-instances --placement "GroupName = my-cluster"
```

To launch instances into a specific partition of a partition placement group using the AWS CLI

Use the [run-instances](#) command and specify the placement group name and partition using the `--placement "GroupName = HDFS-Group-A, PartitionNumber = 3"` parameter. In this example, the placement group is named `HDFS-Group-A` and the partition number is `3`.

```
aws ec2 run-instances --placement "GroupName = HDFS-Group-A, PartitionNumber = 3"
```

PowerShell

To launch instances into a placement group using AWS Tools for Windows PowerShell

Use the [New-EC2Instance](#) command and specify the placement group name using the `-Placement_GroupName` parameter.

Describe instances in a placement group

You can view the placement information of your instances using one of the following methods. You can also filter partition placement groups by the partition number using the AWS CLI.

Console

To view the placement group and partition number 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 instance.

4. On the **Details** tab, under **Host and placement group**, find **Placement group**. If the instance is not in a placement group, the field is empty. Otherwise, it contains the name of the placement group name. If the placement group is a partition placement group, **Partition number** contains the partition number for the instance.

AWS CLI

To view the partition number for an instance in a partition placement group

Use the [describe-instances](#) command and specify the `--instance-id` parameter.

```
aws ec2 describe-instances --instance-id i-0123a456700123456
```

The response contains the placement information, which includes the placement group name and the partition number for the instance.

```
"Placement": {  
  "AvailabilityZone": "us-east-1c",  
  "GroupName": "HDFS-Group-A",  
  "PartitionNumber": 3,  
  "Tenancy": "default"  
}
```

To filter instances for a specific partition placement group and partition number

Use the [describe-instances](#) command and specify the `--filters` parameter with the `placement-group-name` and `placement-partition-number` filters. In this example, the placement group is named `HDFS-Group-A` and the partition number is 7.

```
aws ec2 describe-instances --filters "Name = placement-group-name, Values = HDFS-Group-A" "Name = placement-partition-number, Values = 7"
```

The response lists all the instances that are in the specified partition within the specified placement group. The following is example output showing only the instance ID, instance type, and placement information for the returned instances.

```
"Instances": [  
  {  
    "InstanceId": "i-0a1bc23d4567e8f90",
```

```
    "InstanceType": "r4.large",
  },

  "Placement": {
    "AvailabilityZone": "us-east-1c",
    "GroupName": "HDFS-Group-A",
    "PartitionNumber": 7,
    "Tenancy": "default"
  }

{
  "InstanceId": "i-0a9b876cd5d4ef321",
  "InstanceType": "r4.large",
},

  "Placement": {
    "AvailabilityZone": "us-east-1c",
    "GroupName": "HDFS-Group-A",
    "PartitionNumber": 7,
    "Tenancy": "default"
  }
],
```

Change the placement group for an instance

You can change the placement group for an instance as follows:

- Move an existing instance to a placement group
- Move an instance from one placement group to another

Before you can move the instance, the instance must be in the stopped state.

Console

To move an instance to a placement group

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 **Instance state, Stop instance**.
4. With the instance selected, choose **Actions, Instance settings, Modify instance placement**.

5. For **Placement group**, choose the placement group to move the instance to.
6. Choose **Save**.

AWS CLI

To move an instance to a placement group

1. Stop the instance using the [stop-instances](#) command.
2. Use the [modify-instance-placement](#) command and specify the name of the placement group to move the instance to.

```
aws ec2 modify-instance-placement \  
  --instance-id i-0123a456700123456 \  
  --group-name MySpreadGroup
```

3. Start the instance using the [start-instances](#) command.

PowerShell

To move an instance to a placement group using the AWS Tools for Windows PowerShell

1. Stop the instance using the [Stop-EC2Instance](#) command.
2. Use the [Edit-EC2InstancePlacement](#) command and specify the name of the placement group to which to move the instance.
3. Start the instance using the [Start-EC2Instance](#) command.

Remove an instance from a placement group

You can remove an instance from a placement group using one of the following methods.

Before you can remove an instance from a placement group, the instance must be in the stopped state.

Console

To remove an instance from a placement group

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 **Instance state, Stop instance**.
4. With the instance selected, choose **Actions, Instance settings, Modify instance placement**.
5. For **Placement group**, choose **None**.
6. Choose **Save**.

AWS CLI

To remove an instance from a placement group

1. Stop the instance using the [stop-instances](#) command.
2. Use the [modify-instance-placement](#) command and specify an empty string for the placement group name.

```
aws ec2 modify-instance-placement \  
  --instance-id i-0123a456700123456 \  
  --group-name ""
```

3. Start the instance using the [start-instances](#) command.

PowerShell

To remove an instance from a placement group using the AWS Tools for Windows PowerShell

1. Stop the instance using the [Stop-EC2Instance](#) command.
2. Use the [Edit-EC2InstancePlacement](#) command and specify an empty string for the placement group name.
3. Start the instance using the [Start-EC2Instance](#) command.

Delete a placement group

If you need to replace a placement group or no longer need one, you can delete it. You can delete a placement group using one of the following methods.

Prerequisite

Before you can delete a placement group, it must contain no instances. You can [terminate](#) all instances that you launched in the placement group, [move](#) instances to another placement group, or [remove](#) instances 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 **De1ete** and then choose **Delete**.

AWS CLI

To delete a placement group

Use the [delete-placement-group](#) command and specify the placement group name to delete the placement group. In this example, the placement group name is `my-cluster`.

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

To delete a placement group using the AWS Tools for Windows PowerShell

Use the [Remove-EC2PlacementGroup](#) command to delete the placement group.

Share a placement group

Placement group sharing allows you to influence the placement of interdependent instances that are owned by separate AWS accounts. You can share a placement group across multiple AWS accounts or within your organizations. You can launch instances in a shared placement group.

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

Note

The AWS account from which you want to share a placement group must have the following permissions in the IAM policy.

- `ec2:PutResourcePolicy`
- `ec2>DeleteResourcePolicy`

Topics

- [Rules and limitations](#)
- [Share across Availability Zones](#)
- [Share a placement group](#)
- [Identify a shared placement group](#)
- [Launch an instance in a shared placement group](#)
- [Unshare a shared placement group](#)

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 cannot 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](#).
- You are responsible for managing the instances owned by you in a shared placement group.
- You cannot view or modify instances and capacity reservations that are associated with a shared placement group but not owned by you.

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>.
2. The Availability Zone IDs for the current Region are displayed under **Your AZ ID** in the right panel.

Share a placement group

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 <https://console.aws.amazon.com/ram> or AWS CLI.

AWS RAM console

To **share a placement group** you own using <https://console.aws.amazon.com/ram>, see [Creating a resource share](#).

AWS CLI

To **share a placement group** you own, use the [create-resource-share](#) command.

Identify a shared placement group

The Amazon Resource Name (ARN) of a placement group contains the 12-digit account ID of the account that owns the placement group. You can use the account ID to identify the owner of a placement group that is shared with you.

You can find the placement group ARN using one of the following methods. For more information, see [View placement group information](#).

Amazon EC2 console

To identify a shared placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Network & Security**, choose **Placement Groups**.
3. The **Placement groups** table lists all the placement groups that are owned by you and shared with you. The **Group ARN** column displays the placement group ARN.

If the **Group ARN** column isn't visible, choose settings (



) in the top-right corner, turn on **Group ARN**, and choose **Confirm**.

AWS CLI

To identify a shared placement group

Use [describe-placement-groups](#) command to list all the placement groups that are owned by you and shared with you. In the response, the `GroupId` parameter displays the ARN of a placement group.

Launch an instance in a shared placement group

Important

When using the AWS CLI to launch an instance in a shared placement group, you must specify the placement group ID by using the `GroupId` parameter.

You can use the placement group name only if you're the owner of the placement group being shared. We recommend using the placement group ID to avoid potential placement group name collisions between AWS accounts.

You can find the ID of a placement group in the Amazon EC2 console on the **Placement Groups** screen or by using the [describe-placement-groups](#) AWS CLI command. For more information, see [View placement group information](#).

Console

To launch instances into a shared placement group

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.
2. Under **Instance type**, select a supported instance type. For more information, see [Placement group rules and limitations](#).
3. Expand **Advanced details**, and configure the placement group settings as follows:
 - a. For **Placement group**, select the placement group that was shared with you.

Note

If there are placement groups with the same name, check the placement group ID to make sure that you select the correct placement group.

- b. If you choose a placement group with a partition strategy, for **Target partition**, choose the partition in which to launch the instance.
4. In the **Summary** panel, do the following:

- a. For **Number of instances**, enter the total number of instances that you need in this placement group, because you might not be able to add instances to the placement group later.
- b. Review your instance configuration, and then choose **Launch instance**.

For more information, see [Launch an instance using the new launch instance wizard](#).

AWS CLI

To launch instances in a shared placement group

Use the [run-instances](#) command and specify the placement group ID of the shared placement group.

```
aws ec2 run-instances --placement "GroupId = pg-0123456789example"
```

To launch instances into a specific partition of a shared partition placement group

Use the [run-instances](#) command and specify the placement group ID and the partition number of the shared placement group.

```
aws ec2 run-instances --placement "GroupId = pg-0123456789example, PartitionNumber = 3"
```

Tip

Use VPC peering to connect instances owned by separate AWS accounts and get the full latency benefits offered by shared cluster placement groups. For more information, see [What is VPC peering?](#)

Unshare a shared placement group

The placement group owner can unshare a shared placement group at any time.

When you unshare a shared placement group, the following changes will take effect.

- The AWS accounts with which a placement group was shared will no longer be able to launch instances or reserve capacity.

- If your instances were running in a shared placement group, they will be disassociated from the placement group but continue to run normally in your AWS account.
- If you had capacity reservations in a shared placement group, they will be disassociated from the placement group but you will continue to have access to them in your AWS account.

You can unshare a shared placement group using one of the following methods.

AWS RAM console

To unshare a shared placement group using <https://console.aws.amazon.com/ram>, see [Deleting a resource share](#).

AWS CLI

To unshare a shared placement group using AWS Command Line Interface, use the [disassociate-resource-share](#) command.

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.

The following rules apply to instances that are in Wavelength Zones:

- Traffic that goes from one instance to another within a VPC in the same Wavelength Zone has an MTU of 1300.
- Traffic that goes from one instance to another that uses the carrier IP within a Wavelength Zone has an MTU of 1500.
- Traffic that goes from one instance to another between a Wavelength Zone and the Region that uses a public IP address has an MTU of 1500.
- Traffic that goes from one instance to another between a Wavelength Zone and the Region that uses a private IP address has an MTU of 1300.

The following rules apply to instances that are in Outposts:

- Traffic that goes from an instance in Outposts to an instance in the Region has an MTU of 1300.

To see Network MTU information for Windows instances, switch to this page in the *Amazon EC2 User Guide for Windows Instances* guide: [Network maximum transmission unit \(MTU\) for your EC2 instance](#).

Contents

- [Jumbo frames \(9001 MTU\)](#)
- [Path MTU Discovery](#)
- [Check the path MTU between two hosts](#)
- [Check and set the MTU on your Linux instance](#)
- [Troubleshoot](#)

Jumbo frames (9001 MTU)

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, and thus increasing the percentage of the packet that is not packet overhead. Fewer packets are needed to send the same amount of usable data. However, traffic is limited to a maximum MTU of 1500 in the following cases:

- Traffic over an internet gateway
- Traffic over an inter-region VPC peering connection
- Traffic over VPN connections

- Traffic outside of a given AWS Region for EC2-Classic

If packets are over 1500 bytes, they are fragmented, or they are dropped if the Don't Fragment flag is set in the IP header.

Jumbo frames should be used with caution for internet-bound traffic or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside a VPC and not slow traffic that's bound for outside the VPC, you can configure the MTU size by route, or use multiple elastic network interfaces with different MTU sizes and different routes.

For instances that are collocated inside a cluster placement group, jumbo frames help to achieve the maximum network throughput possible, and they are recommended in this case. For more information, see [Placement groups](#).

You can use jumbo frames for traffic between your VPCs and your on-premises networks over AWS Direct Connect. For more information, and for how to verify Jumbo Frame capability, see [Setting Network MTU](#) in the *AWS Direct Connect User Guide*.

All Amazon EC2 instance types support 1500 MTU and all current generation instance types support jumbo frames. The following previous generation instance types support jumbo frames: A1, C3, I2, M3, and R3.

For more information about supported MTU sizes:

- For NAT gateways, see [NAT gateway basics](#) in the *Amazon VPC User Guide*.
- For transit gateways, see [MTU](#) in the *Amazon VPC Transit Gateways User Guide*.
- For Local Zones, see [Considerations](#) in the *AWS Local Zones 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.

Check the path MTU between two hosts

You can check the path MTU between two hosts using the `tracert` command, which is part of the `iputils` package that is available by default on many Linux distributions, including Amazon Linux.

To check path MTU using `tracert`

Use the following command to check the path MTU between your EC2 instance and another host. You can use a DNS name or an IP address as the destination. If the destination is another EC2 instance, verify that the security group allows inbound UDP traffic. This example checks the path MTU between an EC2 instance and `amazon.com`.

```
[ec2-user ~]$ tracert amazon.com
1?: [LOCALHOST]      pmtu 9001
1:  ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)    0.187ms pmtu 1500
1:  no reply
2:  no reply
3:  no reply
4:  100.64.16.241 (100.64.16.241)                               0.574ms
5:  72.21.222.221 (72.21.222.221)                               84.447ms asymm 21
6:  205.251.229.97 (205.251.229.97)                            79.970ms asymm 19
7:  72.21.222.194 (72.21.222.194)                              96.546ms asymm 16
8:  72.21.222.239 (72.21.222.239)                              79.244ms asymm 15
9:  205.251.225.73 (205.251.225.73)                            91.867ms asymm 16
...
31: no reply
    Too many hops: pmtu 1500
    Resume: pmtu 1500
```

In this example, the path MTU is 1500.

Check and set the MTU on your Linux instance

Some instances are configured to use jumbo frames, and others are configured to use standard frame sizes. You may want to use jumbo frames for network traffic within your VPC or you may want to use standard frames for internet traffic. Whatever your use case, we recommend verifying that your instance will behave the way you expect it to. You can use the procedures in this section to check your network interface's MTU setting and modify it if needed.

To check the MTU setting on a Linux instance

You can check the current MTU value using the following **ip** command. If your primary network interface is not `eth0`, replace `eth0` with your network interface.

```
[ec2-user ~]$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode
   DEFAULT group default qlen 1000
    link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

Note

In the example output, **mtu 9001** indicates that this instance uses jumbo frames.

To set the MTU value on a Linux instance

1. You can set the MTU value using the `ip` command. The following command sets the desired MTU value to 1500, but you could use 9001 instead.

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
```

2. (Optional) To persist your network MTU setting after a reboot, modify the following configuration files, based on your operating system type.
 - For Amazon Linux 2, add the following line to the `/etc/sysconfig/network-scripts/ifcfg-eth0` file:

```
MTU=1500
```

Add the following line to the `/etc/dhcp/dhclient.conf` file:

```
request subnet-mask, broadcast-address, time-offset, routers, domain-name,  
domain-search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-  
servers;
```

- For Amazon Linux AMI, add the following lines to your `/etc/dhcp/dhclient-eth0.conf` file.

```
interface "eth0" {  
supersede interface-mtu 1500;  
}
```

- For other Linux distributions, consult their specific documentation.

3. (Optional) Reboot your instance and verify that the MTU setting is correct.

Troubleshoot

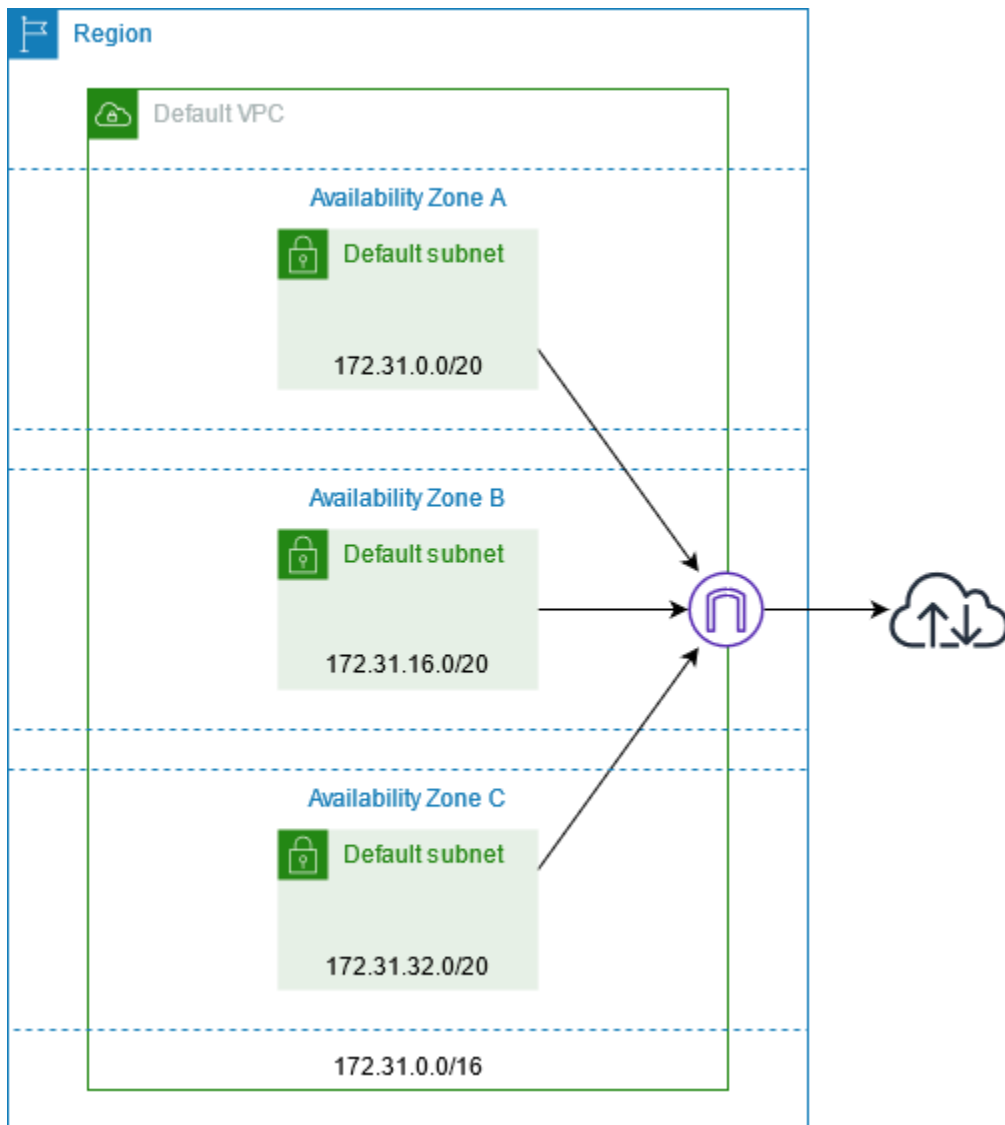
If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see [Queries Appear to Hang](#) in the *Amazon Redshift Management Guide*

Virtual private clouds

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud* 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.

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. Alternatively, you can create your own VPC and configure it to meet your needs.



Create additional VPCs

Use the following procedure to create a VPC with the subnets, gateways, and routing configuration that you need.

To create a VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Choose **Create VPC**.
3. Under **Resources to create**, choose **VPC and more**.
4. For **Name tag auto-generation**, enter a name for the VPC.

5. For **IPv4 CIDR block**, either keep the default suggestion, enter the CIDR block required by your application or network.
6. For **Number of Availability Zones**, choose **2**, so that you can launch instances in multiple Availability Zones to ensure high availability.
7. If your instances must be accessible from the internet, do one of the following:
 - If your instances can be in a public subnet, select a nonzero value for **Number of public subnets**. Keep both options under **DNS options** selected. You can optionally add private subnets now or later on.
 - If your instances must be in a private subnet, select **0** for **Number of public subnets**. For **Number of private subnets**, select a number depending on your needs (the possible values correspond to 1 or 2 private subnets per Availability Zone). For **NAT gateways**, if your instances in both Availability Zones send or receive a significant volume of traffic across Availability Zones, select **1 per AZ**. Otherwise, select **In 1 AZ** and launch instances that send or receive cross-zone traffic in the same Availability Zone as the NAT gateway.
8. Expand **Customize subnet CIDR blocks**. Either keep the default suggestions, or enter a CIDR block for each subnet. For more information, see [Subnet CIDR blocks](#) in the *Amazon VPC User Guide*.
9. Review the **Preview** pane, which shows the VPC resources that will be created based on your selections.
10. Choose **Create VPC**.

Access the internet from your instances

Instances launched into a default subnet have access to the internet, as the VPC is 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 the subnets that you create in your VPCs, do one of the following to ensure that 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*.

SSH access to your instances

To connect to an instance, you must authorize SSH traffic to the instance from your network. You must also specify a key pair when you launch the instance and specify the `.pem` file when you connect to the instance. For more information, see [Locate the private key and set permissions](#).

Shared subnets

When launching EC2 instances into shared VPC subnets, note the following:

- Participants can run instances in a shared VPC subnet by passing in the shared subnet ID. If participants want to pass in a security group ID or network interface ID when they run an instance, the participant must own the security group or network interface.
- Participants can start, stop, terminate, and describe instances that they've created in a shared VPC subnet. Participants cannot start, stop, terminate, or describe instances created by the VPC owner in a shared VPC subnet.
- VPC owners cannot start, stop, terminate, or describe instances created by participants in a shared VPC subnet.

For more information 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 [instances built on the AWS Nitro System](#).

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 in Amazon EC2](#).
 - 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

- [Infrastructure security in Amazon EC2](#)
- [Resilience in Amazon EC2](#)
- [Data protection in Amazon EC2](#)
- [Identity and access management for Amazon EC2](#)

- [Amazon EC2 key pairs and Amazon EC2 instances](#)
- [Amazon EC2 security groups for Linux instances](#)
- [Access Amazon EC2 using an interface VPC endpoint](#)
- [Update management in Amazon EC2](#)
- [Compliance validation for Amazon EC2](#)
- [NitroTPM](#)

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](#). For example, you can allow traffic only from the address ranges for your corporate network.
- Use private subnets for your instances if they should not be accessed directly from the internet. Use a bastion host or NAT gateway for internet access from an instance in a private subnet.
- Use AWS Virtual Private Network or AWS Direct Connect to establish private connections from your remote networks to your VPCs. For more information, see [Network-to-Amazon VPC Connectivity Options](#).
- Use [VPC Flow Logs](#) to monitor the traffic that reaches your instances.

- Use [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 ports and managing SSH keys.
- Use [AWS Systems Manager Run Command](#) to automate common administrative tasks instead of opening inbound SSH ports and managing SSH keys.

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

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.
- 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 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form 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.

Topics

- [Amazon EBS data security](#)
- [Encryption at rest](#)
- [Encryption in transit](#)

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*.

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 Graviton processors. AWS Graviton2, AWS Graviton3, and AWS Graviton3E 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). Instances with 3rd generation AMD EPYC processors (Milan) also support AMD Secure Encrypted Virtualization-Secure Nested Paging (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
 - **Compute optimized:** C5a, C5ad, C5n, C6a, C6gn, C6i, C6id, C6in, C7a, C7g, C7gd, C7gn, C7i
 - **Memory optimized:** R5dn, R5n, R6a, R6i, R6idn, R6in, R6id, R7a, R7g, R7gd, R7i, R7iz, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, X2idn, X2iedn, X2iezn
 - **Storage optimized:** D3, D3en, I3en, I4g, I4i, Im4gn, Is4gen
 - **Accelerated computing:** DL1, DL2q, G4ad, G4dn, G5, G6, Gr6, Inf1, Inf2, P3dn, P4d, P4de, P5, Trn1, Trn1n, 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

SSH provides a secure communications channel for remote access to your Linux 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.

Identity and access management for Amazon EC2

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2 resources without sharing your security credentials. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

For best practices for securing your AWS resources using IAM, see [Security best practices in IAM](#).

Contents

- [Network access to your instance](#)
- [Amazon EC2 permission attributes](#)
- [IAM and Amazon EC2](#)
- [IAM policies for Amazon EC2](#)
- [AWS managed policies for Amazon Elastic Compute Cloud](#)
- [IAM roles for Amazon EC2](#)

Network access to your instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

For more information, see [Security group rules](#).

Amazon EC2 permission attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These permissions work at the AWS account level only; you can't restrict permissions for specific users within the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see [Make an AMI public](#).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see [Share an Amazon EBS snapshot](#) in the *Amazon EBS User Guide*.

IAM and Amazon EC2

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account

- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

This topic helps you answer the following questions:

- How do I create groups and users in IAM?
- How do I create a policy?
- What IAM policies do I need to carry out tasks in Amazon EC2?
- How do I grant permissions to perform actions in Amazon EC2?
- How do I grant permissions to perform actions on specific resources in Amazon EC2?

Create users, groups, and roles

You can create users and groups for your AWS account and then assign them the permissions they require. As a best practice, users should acquire the permissions by assuming IAM roles. For more information on how to set up users and groups for your AWS account, see [Set up to use Amazon EC2](#).

An [IAM role](#) is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. For more information on how to create IAM roles and grant permissions with them, see [IAM roles for Amazon EC2](#).

Related topics

For more information about IAM, see the following:

- [IAM policies for Amazon EC2](#)
- [IAM roles for Amazon EC2](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [IAM User Guide](#)

IAM 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 custom IAM policies, see [Managing IAM policies](#).

Getting Started

An IAM policy must grant or deny permissions to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.

Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action. Instead, you have to allow users to work with all resources for that action.

Task	Topic
Understand the basic structure of a policy	Policy syntax
Define actions in your policy	Actions for Amazon EC2
Define specific resources in your policy	Amazon Resource Names (ARNs) for Amazon EC2
Apply conditions to the use of the resources	Condition keys for Amazon EC2

Task	Topic
Work with the available resource-level permissions for Amazon EC2	Actions, resources, and condition keys for Amazon EC2
Test your policy	Check that users have the required permissions
Generate an IAM policy	Generate policies based on access activity
Example policies for a CLI or SDK	Example policies for working with the AWS CLI or an AWS SDK
Example policies for the Amazon EC2 console	Example policies for working in the Amazon EC2 console

Grant permissions to users, groups, and roles

The following are examples of some AWS managed policies that are available to utilize if they meet your needs:

- `PowerUserAccess`
- `ReadOnlyAccess`
- `AmazonEC2FullAccess`
- `AmazonEC2ReadOnlyAccess`

For more information on the AWS managed policies available to work with Amazon EC2, see [AWS managed policies for Amazon Elastic Compute Cloud](#).

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 [Creating 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 [Creating 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*.

Policy structure

The following topics explain the structure of an IAM policy.

Contents

- [Policy syntax](#)
- [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](#)
- [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 for working with the AWS CLI or an AWS SDK](#).

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 [IAM 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 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*.

To use a condition key in your IAM policy, use the `Condition` statement. For example, the following policy grants users permission to add and remove inbound and outbound rules for any security group. It uses the `ec2:Vpc` condition key to specify that these actions can only be performed on security groups in a specific VPC.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress",
      "ec2:RevokeSecurityGroupIngress",
      "ec2:RevokeSecurityGroupEgress"],
    "Resource": "arn:aws:ec2:region:account:security-group/*",
    "Condition": {
      "StringEquals": {
        "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-11223344556677889"
      }
    }
  ]
}
```

```
}  
]  
}
```

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permissions to be granted, all conditions must be met.

You can also use placeholders when you specify conditions. For more information, see [IAM policy elements: Variables and tags](#) in the *IAM User Guide*.

Important

Many condition keys are specific to a resource, and some API actions use multiple resources. If you write a policy with a condition key, use the `Resource` element of the statement to specify the resource to which the condition key applies. If not, the policy may prevent users from performing the action at all, because the condition check fails for the resources to which the condition key does not apply. If you do not want to specify a resource, or if you've written the `Action` element of your policy to include multiple API actions, then you must use the `...IfExists` condition type to ensure that the condition key is ignored for resources that do not use it. For more information, see [...IfExists Conditions](#) in the *IAM User Guide*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see [Example: Restrict access to a specific Region](#).

ec2:SourceInstanceARN condition key

The `ec2:SourceInstanceARN` condition key can be used for conditions that specify the ARN of the instance from which a request is made. This is an AWS global condition key and is not service-specific. For policy examples, see [Amazon EC2: Attach or detach volumes to an EC2 instance](#) and [Example: Allow a specific instance to view resources in other AWS services](#). The `ec2:SourceInstanceARN` key cannot be used as a variable to populate the ARN for the `Resource` element in a statement.

For example policy statements for Amazon EC2, see [Example policies for working with the AWS CLI or an AWS SDK](#).

ec2:Attribute condition key

The `ec2:Attribute` condition key can be used for conditions that filter access by an attribute of a resource. The condition key supports only properties that are of a primitive data type (such as a string or integer), or complex [AttributeValue](#) objects that have only a **Value** property (such as the **Description** or **ImdsSupport** objects of the [ModifyImageAttribute](#) API action).

Important

The condition key can't be used with complex objects that have multiple properties, such as the **LaunchPermission** object of the [ModifyImageAttribute](#) API action.

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**.

```
{
  "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.

```
{
  "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:*ResourceID* 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:*ResourceID* 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:*ResourceID* 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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:ImageID": "*"
        }
      }
    }
  ]
}
```

Condition key	API action			
ec2:DhcpOptionsID	<ul style="list-style-type: none"> • CreateDhcpOptions 			
ec2:ImageID	<ul style="list-style-type: none"> • CopyImage • CreateImage • ImportImage • RegisterImage 			
ec2:InstanceID	<ul style="list-style-type: none"> • RunInstances • ImportInstance 			
ec2:InternetGatewayID	<ul style="list-style-type: none"> • CreateInternetGateway 			
ec2:NetworkACLID	<ul style="list-style-type: none"> • CreateNetworkAcl 			
ec2:NetworkInterfaceID	<ul style="list-style-type: none"> • CreateNetworkInterface 			

Condition key	API action			
ec2:PlacementGroupName	<ul style="list-style-type: none"> • CreatePlacementGroup 			
ec2:RouteTableID	<ul style="list-style-type: none"> • CreateRouteTable 			
ec2:SecurityGroupID	<ul style="list-style-type: none"> • CreateSecurityGroup 			
ec2:SnapshotID	<ul style="list-style-type: none"> • CopySnapshot • CreateSnapshot • CreateSnapshots • ImportSnapshots 			
ec2:SubnetID	<ul style="list-style-type: none"> • CreateSubnet 			
ec2:VolumeID	<ul style="list-style-type: none"> • CreateVolume • ImportVolume 			

Condition key	API action			
ec2:VpcID	<ul style="list-style-type: none"> CreateVpc 			
ec2:VpcPeeringConnectionID	<ul style="list-style-type: none"> CreateVpcPeeringConnection 			

We recommend that you avoid using `ec2:ResourceID` condition keys with these API actions. Instead, if you need to filter access based on specific resource IDs, we recommend that you do so using the Resource policy element, as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-01234567890abcdef"
    }
  ]
}
```

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](#) in the *AWS CLI Command Reference*.

Grant permission to tag resources during creation

Some resource-creating Amazon EC2 API actions enable you to specify tags when you create the resource. You can use resource tags to implement attribute-based control (ABAC). For more information, see [Tag your resources](#) and [Control access to EC2 resources using resource tags](#).

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `ec2:RunInstances` or `ec2:CreateVolume`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `ec2:CreateTags` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `ec2:CreateTags` action.

In the IAM policy definition for the `ec2:CreateTags` action, use the `Condition` element with the `ec2:CreateAction` condition key to give tagging permissions to the action that creates the resource.

The following example demonstrates a policy that allows users to launch instances and apply any tags to instances and volumes during launch. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:*/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction" : "RunInstances"
        }
      }
    }
  ]
}
```

Similarly, the following policy allows users to create volumes and apply any tags to the volumes during volume creation. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVolume"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],

```

```
"Resource": "arn:aws:ec2:region:account:*/**",
"Condition": {
  "StringEquals": {
    "ec2:CreateAction" : "CreateVolume"
  }
}
]
```

The `ec2:CreateTags` action is only evaluated if tags are applied during the resource-creating action. Therefore, a user that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `ec2:CreateTags` action if no tags are specified in the request. However, if the user attempts to create a resource with tags, the request fails if the user does not have permissions to use the `ec2:CreateTags` action.

The `ec2:CreateTags` action is also evaluated if tags are provided in a launch template. For an example policy, see [Tags in a launch template](#).

Control access to specific tags

You can use additional conditions in the `Condition` element of your IAM policies to control the tag keys and values that can be applied to resources.

The following condition keys can be used with the examples in the preceding section:

- `aws:RequestTag`: To indicate that a particular tag key or tag key and value must be present in a request. Other tags can also be specified in the request.
- Use with the `StringEquals` condition operator to enforce a specific tag key and value combination, for example, to enforce the tag `cost-center=cc123`:

```
"StringEquals": { "aws:RequestTag/cost-center": "cc123" }
```

- Use with the `StringLike` condition operator to enforce a specific tag key in the request; for example, to enforce the tag key `purpose`:

```
"StringLike": { "aws:RequestTag/purpose": "*" }
```

- `aws:TagKeys`: To enforce the tag keys that are used in the request.

- Use with the `ForAllValues` modifier to enforce specific tag keys if they are provided in the request (if tags are specified in the request, only specific tag keys are allowed; no other tags are allowed). For example, the tag keys `environment` or `cost-center` are allowed:

```
"ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }
```

- Use with the `ForAnyValue` modifier to enforce the presence of at least one of the specified tag keys in the request. For example, at least one of the tag keys `environment` or `webserver` must be present in the request:

```
"ForAnyValue:StringEquals": { "aws:TagKeys": ["environment","webserver"] }
```

These condition keys can be applied to resource-creating actions that support tagging, as well as the `ec2:CreateTags` and `ec2>DeleteTags` actions. To learn whether an Amazon EC2 API action supports tagging, see [Actions, resources, and condition keys for Amazon EC2](#).

To force users to specify tags when they create a resource, you must use the `aws:RequestTag` condition key or the `aws:TagKeys` condition key with the `ForAnyValue` modifier on the resource-creating action. The `ec2:CreateTags` action is not evaluated if a user does not specify tags for the resource-creating action.

For conditions, the condition key is not case-sensitive and the condition value is case-sensitive. Therefore, to enforce the case-sensitivity of a tag key, use the `aws:TagKeys` condition key, where the tag key is specified as a value in the condition.

For example IAM policies, see [Example policies for working with the AWS CLI or an AWS SDK](#). For more information about multi-value conditions, see [Creating a Condition That Tests Multiple Key Values](#) in the *IAM User Guide*.

Control access to EC2 resources using resource tags

When you create an IAM policy that grants 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 for working with the AWS CLI or an AWS SDK](#).

If you allow or deny users access to resources based on tags, you must consider explicitly denying users the ability to add those tags to or remove them from the same resources. Otherwise, it's possible for a user to circumvent your restrictions and gain access to a resource by modifying its tags.

Example policies for working with the AWS CLI or an AWS SDK

You must grant users the permissions they require for Amazon EC2 using IAM policies. The following examples show policy statements that you could use to control the permissions that users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI or an AWS SDK. For more information, see [Creating IAM policies](#) in the IAM User Guide. For example policies for working in the Amazon EC2 console, see [Example policies for working in the Amazon EC2 console](#). For examples of IAM policies specific to Amazon VPC, see [Identity and Access Management for Amazon VPC](#).

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](#)

Example: Read-only access

The following policy grants users permissions to use all Amazon EC2 API actions whose names begin with Describe. The Resource element uses a wildcard to indicate that users can specify all resources with these API actions. The * wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Actions, resources, and condition keys for Amazon EC2](#).

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
  ]
}
```

Example: Restrict access to a specific Region

The following policy denies users permission to use all Amazon EC2 API actions unless the Region is Europe (Frankfurt). It uses the global condition key `aws:RequestedRegion`, which is supported by all Amazon EC2 API actions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
```

```
    "Action": "ec2:*",
    "Resource": "*",
    "Condition": {
      "StringNotEquals": {
        "aws:RequestedRegion": "eu-central-1"
      }
    }
  }
]
```

Alternatively, you can use the condition key `ec2:Region`, which is specific to Amazon EC2 and is supported by all Amazon EC2 API actions.

```
{
  "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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeImages",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

Example: Describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances `i-1234567890abcdef0` and `i-0598c7d356eba48d7`, and to terminate only instances in the US East (N. Virginia) Region (`us-east-1`) with the resource tag `"purpose=test"`.

The first statement uses a `*` wildcard for the `Resource` element to indicate that users can specify all resources with the action; in this case, they can list all instances. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, `ec2:DescribeInstances`). For more information about which ARNs you can use with which Amazon EC2 API actions, see [Actions, resources, and condition keys for Amazon EC2](#).

The second statement uses resource-level permissions for the `StopInstances` and `StartInstances` actions. The specific instances are indicated by their ARNs in the `Resource` element.

The third statement allows users to terminate all instances in the US East (N. Virginia) Region (us-east-1) that belong to the specified AWS account, but only where the instance has the tag "purpose=test". The Condition element qualifies when the policy statement is in effect.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:account-id:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:account-id:instance/i-0598c7d356eba48d7"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:account-id: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).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-9e1670f7",
        "arn:aws:ec2:region::image/ami-45cf5c3c",
      ]
    }
  ]
}
```

```

    "arn:aws:ec2:region:account-id:instance/*",
    "arn:aws:ec2:region:account-id:volume/*",
    "arn:aws:ec2:region:account-id:key-pair/*",
    "arn:aws:ec2:region:account-id:security-group/*",
    "arn:aws:ec2:region:account-id:subnet/*",
    "arn:aws:ec2:region:account-id: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).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:Owner": "amazon"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account-id:instance/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:key-pair/*",
        "arn:aws:ec2:region:account-id:security-group/*"
      ]
    }
  ]
}

```



```
]
}
```

Instance types

The following policy allows users to launch instances using only the `t2.micro` or `t2.small` instance type, which you might do to control costs. The users can't launch larger instances because the `Condition` element of the first statement tests whether `ec2:InstanceType` is either `t2.micro` or `t2.small`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account-id:instance/*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:InstanceType": ["t2.micro", "t2.small"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:key-pair/*",
        "arn:aws:ec2:region:account-id:security-group*"
      ]
    }
  ]
}
```

Alternatively, you can create a policy that denies users permissions to launch any instances except `t2.micro` and `t2.small` instance types.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account-id:instance/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "ec2:InstanceType": ["t2.micro", "t2.small"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:instance/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:key-pair/*",
        "arn:aws:ec2:region:account-id:security-group/*"
      ]
    }
  ]
}
```

Subnets

The following policy allows users to launch instances using only the specified subnet, subnet-*12345678*. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
```

```

    "Resource": [
      "arn:aws:ec2:region:account-id:subnet/subnet-12345678",
      "arn:aws:ec2:region:account-id:network-interface/*",
      "arn:aws:ec2:region:account-id:instance/*",
      "arn:aws:ec2:region:account-id:volume/*",
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account-id:key-pair/*",
      "arn:aws:ec2:region:account-id:security-group/*"
    ]
  }
]
}

```

Alternatively, you could create a policy that denies users permissions to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet `subnet-12345678` is specified. This denial overrides any other policies that are created to allow launching instances into other subnets.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account-id:network-interface/*"
      ],
      "Condition": {
        "ArnNotEquals": {
          "ec2:Subnet": "arn:aws:ec2:region:account-id:subnet/subnet-12345678"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:instance/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:key-pair/*",

```

```

        "arn:aws:ec2:region:account-id:security-group/*"
    ]
}
]
}

```

EBS volumes

The following policy allows users to launch instances only if the EBS volumes for the instance are encrypted. The user must launch an instance from an AMI that was created with encrypted snapshots, to ensure that the root volume is encrypted. Any additional volume that the user attaches to the instance during launch must also be encrypted.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Condition": {
        "Bool": {
          "ec2:Encrypted": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:image/ami-*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:key-pair/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    }
  ]
}

```

Tags

Tag instances on creation

The following policy allows users to launch instances and tag the instances during creation. For resource-creating actions that apply tags, users must have permissions to use the `CreateTags` action. The second statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `RunInstances`, and only for instances. Users cannot tag existing resources, and users cannot tag volumes using the `RunInstances` request.

For more information, see [Grant permission to tag resources during creation](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:account-id: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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region::image/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:security-group/*",
        "arn:aws:ec2:region:account-id:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:instance/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "production" ,
          "aws:RequestTag/purpose": "webserver"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "Condition": {
        "StringEquals": {
```

```

        "ec2:CreateAction" : "RunInstances"
    }
}
]
}

```

Tag instances and volumes on creation with at least one specific tag

The following policy uses the `ForAnyValue` modifier on the `aws:TagKeys` condition to indicate that at least one tag must be specified in the request, and it must contain the key `environment` or `webserver`. The tag must be applied to both instances and volumes. Any tag values can be specified in the request.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region::image/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:security-group/*",
        "arn:aws:ec2:region:account-id:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:instance/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": ["environment", "webserver"]
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:region:account-id:*/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction" : "RunInstances"
      }
    }
  }
]
}

```

If instances are tagged on creation, they must be tagged with a specific tag

In the following policy, users do not have to specify tags in the request, but if they do, the tag must be purpose=test. No other tags are allowed. Users can apply the tags to any taggable resource in the RunInstances request.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/purpose": "test",
          "ec2:CreateAction" : "RunInstances"
        }
      }
    }
  ]
}

```



```

    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": "purpose"
    }
  }
}
]
}

```

To disallow anyone called tag on create for RunInstances

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",
        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
        "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
      ]
    },
    {
      "Sid": "VisualEditor0",
      "Effect": "Deny",
      "Action": "ec2:CreateTags",
      "Resource": "*"
    }
  ]
}

```

Only allow specific tags for spot-instances-request. Surprise inconsistency number 2 comes into play here. Under normal circumstances, specifying no tags will result in Unauthenticated. In the

case of `spot-instances-request`, this policy will not be evaluated if there are no `spot-instances-request` tags, so a non-tag Spot on Run request will succeed.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",
        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
      ]
    },
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "production"
        }
      }
    }
  ]
}
```

Tags in a launch template

In the following example, users can launch instances, but only if they use a specific launch template (`lt-09477bcd97b0d310e`). The `ec2:IsLaunchTemplateResource` condition key prevents users from overriding any of the resources specified in the launch template. The second

part of the statement allows users to tag instances on creation—this part of the statement is necessary if tags are specified for the instance in the launch template.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/lt-09477bcd97b0d310e"
        },
        "Bool": {
          "ec2:IsLaunchTemplateResource": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id: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.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:*:account-id: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:*:account-id:network-interface/*",
        "arn:aws:ec2:*:account-id:instance/*",
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:volume/*",
        "arn:aws:ec2:*:account-id:key-pair/*",
        "arn:aws:ec2:*:account-id:security-group*"
      ]
    }
  ]
}

```

Launch templates

In the following example, users can launch instances, but only if they use a specific launch template (lt-09477bcd97b0d310e). Users can override any parameters in the launch template by specifying the parameters in the RunInstances action.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/lt-09477bcd97b0d310e"
        }
      }
    }
  ]
}
```

In this example, users can launch instances only if they use a launch template. The policy uses the `ec2:IsLaunchTemplateResource` condition key to prevent users from overriding any pre-existing ARNs in the launch template.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/*"
        },
        "Bool": {
          "ec2:IsLaunchTemplateResource": "true"
        }
      }
    }
  ]
}
```

The following example policy allows user to launch instances, but only if they use a launch template. Users cannot override the subnet and network interface parameters in the request; these

parameters can only be specified in the launch template. The first part of the statement uses the [NotResource](#) element to allow all other resources except subnets and network interfaces. The second part of the statement allows the subnet and network interface resources, but only if they are sourced from the launch template.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": ["arn:aws:ec2:region:account-id:subnet/*",
                     "arn:aws:ec2:region:account-id:network-interface/*" ],
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": ["arn:aws:ec2:region:account-id:subnet/*",
                   "arn:aws:ec2:region:account-id:network-interface/*" ],
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id: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=Webserver`s. Users cannot override any of the launch template parameters in the `RunInstances` action.

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "NotResource": "arn:aws:ec2:region:account-id:launch-template/*",
    "Condition": {
      "ArnLike": {
        "ec2:LaunchTemplate": "arn:aws:ec2:region:account-id:launch-template/*"
      },
      "Bool": {
        "ec2:IsLaunchTemplateResource": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:region:account-id: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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1::image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",

```



```

        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
        "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
    ]
}
]
}

```

Warning

NOT SUPPORTED – Example: Deny users permission to request Spot Instances using RunInstances

The following policy is not supported for the `spot-instances-request` resource. The following policy is meant to give users the permission to launch On-Demand Instances, but deny users the permission to request Spot Instances. The `spot-instances-request` resource, which is created by `RunInstances`, is the resource that requests Spot Instances. The second statement is meant to deny the `RunInstances` action for the `spot-instances-request` resource. However, this condition is not supported because Amazon EC2 does not evaluate the `spot-instances-request` resource in the `RunInstances` statement if the Spot Instance request is not tagged on create.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1::image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",
        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*"
      ]
    }
  ]
}

```

```

    ]
  },
  {
    "Sid": "DenySpotInstancesRequests - NOT SUPPORTED - DO NOT USE!",
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
  }
]
}

```

Example: Tag Spot Instance requests on create

The following policy allows users to tag all resources that are created during instance launch. The first statement allows RunInstances to create the listed resources. The `spot-instances-request` resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement provides a `*` wildcard to allow all resources to be tagged when they are created at instance launch.

Note

If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the RunInstances statement. Therefore, you must explicitly allow the `spot-instances-request` resource for the RunInstances action, otherwise the call will fail.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1::image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",

```

```

        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
        "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
    ]
},
{
    "Sid": "TagResources",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*"
}
]
}

```

Example: Deny tag on create for Spot Instance requests

The following policy denies users the permission to tag the resources that are created during instance launch.

The first statement allows RunInstances to create the listed resources. The `spot-instances-request` resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement provides a `*` wildcard to deny all resources being tagged when they are created at instance launch. If `spot-instances-request` or any other resource is tagged on create, the RunInstances call will fail.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
            ]
        }
    ]
}

```

```

        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
        "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
    ]
},
{
    "Sid": "DenyTagResources",
    "Effect": "Deny",
    "Action": "ec2:CreateTags",
    "Resource": "*"
}
]
}

```

Warning

NOT SUPPORTED – Example: Allow creating a Spot Instance request only if it is assigned a specific tag

The following policy is not supported for the `spot-instances-request` resource. The following policy is meant to grant RunInstances the permission to create a Spot Instance request only if the request is tagged with a specific tag.

The first statement allows RunInstances to create the listed resources.

The second statement is meant to grant users the permission to create a Spot Instance request only if the request has the tag `environment=production`. If this condition is applied to other resources created by RunInstances, specifying no tags results in an Unauthenticated error. However, if no tags are specified for the Spot Instance request, Amazon EC2 does not evaluate the `spot-instances-request` resource in the RunInstances statement, which results in non-tagged Spot Instance requests being created by RunInstances.

Note that specifying another tag other than `environment=production` results in an Unauthenticated error, because if a user tags a Spot Instance request, Amazon EC2 evaluates the `spot-instances-request` resource in the RunInstances statement.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [

```

```

        "ec2:RunInstances"
    ],
    "Resource": [
        "arn:aws:ec2:us-east-1::image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",
        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*"
    ]
},
{
    "Sid": "RequestSpotInstancesOnlyIfTagIs_environment=production - NOT
SUPPORTED - DO NOT USE!",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/environment": "production"
        }
    }
},
{
    "Sid": "TagResources",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*"
}
]
}

```

Example: Deny creating a Spot Instance request if it is assigned a specific tag

The following policy denies RunInstances the permission to create a Spot Instance request if the request is tagged with `environment=production`.

The first statement allows RunInstances to create the listed resources.

The second statement denies users the permission to create a Spot Instance request if the request has the tag `environment=production`. Specifying `environment=production` as a tag results in an Unauthenticated error. Specifying other tags or specifying no tags will result in the creation of a Spot Instance request.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:image/*",
        "arn:aws:ec2:us-east-1:*:subnet/*",
        "arn:aws:ec2:us-east-1:*:network-interface/*",
        "arn:aws:ec2:us-east-1:*:security-group/*",
        "arn:aws:ec2:us-east-1:*:key-pair/*",
        "arn:aws:ec2:us-east-1:*:volume/*",
        "arn:aws:ec2:us-east-1:*:instance/*",
        "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
      ]
    },
    {
      "Sid": "DenySpotInstancesRequests",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "production"
        }
      }
    },
    {
      "Sid": "TagResources",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "*"
    }
  ]
}
```

```
}
```

Example: 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
        "ec2:ModifyReservedInstances",
        "ec2:PurchaseReservedInstancesOffering",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
    }
  ]
}
```

To allow users to view and modify the Reserved Instances in your account, but not purchase new Reserved Instances.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```

    "ec2:DescribeReservedInstances",
    "ec2:ModifyReservedInstances",
    "ec2:DescribeAvailabilityZones"
  ],
  "Resource": "*"
}
]
}
```

Example: 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.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id: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.

```

{
  "Version": "2012-10-17",
  "Statement": [
```



```

    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "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.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DeleteTags",
      "Resource": "arn:aws:ec2:us-east-1:account-id: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.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteTags"
      ],
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "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.

```

{
  "Version": "2012-10-17",

```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AssociateIamInstanceProfile",
      "ec2:ReplaceIamInstanceProfileAssociation",
      "ec2:DisassociateIamInstanceProfile"
    ],
    "Resource": "arn:aws:ec2:us-east-1:account-id:instance/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/department": "test"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DescribeIamInstanceProfileAssociations",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::account-id: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](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation"
      ],
      "Resource": "*"
    }
  ]
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeIamInstanceProfileAssociations",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id: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.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteRoute",
        "ec2:CreateRoute",
        "ec2:ReplaceRoute"
      ],
      "Resource": [
        "arn:aws:ec2:region:account-id:route-table/*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": "arn:aws:ec2:region:account-id:vpc/vpc-ec43eb89"
        }
      }
    }
  ]
}

```

Example: Allow a specific instance to view resources in other AWS services

The following is an example of a policy that you might attach to an IAM role. The policy allows an instance to view resources in various AWS services. It uses the `ec2:SourceInstanceARN` condition key to specify that the instance from which the request is made must be instance `i-093452212644b0dd6`. If the same IAM role is associated with another instance, the other instance cannot perform any of these actions.

The `ec2:SourceInstanceARN` key is an AWS global condition key, therefore it can be used for other service actions, not just Amazon EC2.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "s3:ListAllMyBuckets",
        "dynamodb:ListTables",
        "rds:DescribeDBInstances"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "ArnEquals": {
          "ec2:SourceInstanceARN": "arn:aws:ec2:region:account-id: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.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Action": [
      "ec2:CreateLaunchTemplateVersion",
      "ec2:ModifyLaunchTemplate"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:region:account-id:launch-template/lt-09477bcd97b0d3abc"
  }
]
}

```

The following policy allows users to delete any launch template and launch template version, provided that the launch template has the tag Purpose=Testing.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:region:account-id: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 [Replacing a Launch Configuration with a Launch Template](#) in the *Amazon EC2 Auto Scaling User Guide*.
5. For an Auto Scaling group that uses a launch template, make sure that it uses a new launch template with the instance metadata options configured, or uses a new version of the current launch template with the instance metadata options configured. For more information, see [update-auto-scaling-group](#) in the *AWS CLI Command Reference*.

Examples

- [Require the use of IMDSv2](#)
- [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.

```
{
  "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`.

```

{
  "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.

```
{
  "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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOnlyImdsAdminsToModifySettings",
      "Effect": "Deny",
      "Action": "ec2:ModifyInstanceMetadataOptions",
      "Resource": "*",
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": "arn:aws:iam:*:*:role/ec2-imds-admins"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

Require role credentials to be retrieved from IMDSv2

The following policy specifies that if this policy is applied to a role, and the role is assumed by the EC2 service and the resulting credentials are used to sign a request, then the request must be signed by EC2 role credentials retrieved from IMDSv2. Otherwise, all of its API calls will get an `UnauthorizedOperation` error. This statement/policy can be applied generally because, if the request is not signed by EC2 role credentials, it has no effect.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireAllEc2RolesToUseV2",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NumericLessThan": {
          "ec2:RoleDelivery": "2.0"
        }
      }
    }
  ]
}

```

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 for working in the Amazon EC2 console

You must grant users the permissions they require for Amazon EC2 using IAM policies. You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI or an AWS SDK. For more information, see [Example](#)

[policies for working with the AWS CLI or an AWS SDK](#) and [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](#).

Tip

To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#). If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the [DecodeAuthorizationMessage](#) API action for AWS STS, or the [decode-authorization-message](#) command in the AWS CLI.

Examples

- [Example: Read-only access](#)
- [Example: Use the EC2 launch instance wizard](#)
- [Example: Work with security groups](#)
- [Example: Work with Elastic IP addresses](#)
- [Example: Work with Reserved Instances](#)

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.

```
{
  "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.

```
{
  "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.

```
{
```

```
"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 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 to EC2 resources using resource tags](#).

- 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 `sa-east-1` 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.

```
{
  "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:sa-east-1:111122223333:network-interface/*",
      "arn:aws:ec2:sa-east-1:111122223333:volume/*",
      "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
      "arn:aws:ec2:sa-east-1:111122223333:security-group/*",
      "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:sa-east-1:111122223333:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:InstanceType": "t2.micro"
      }
    }
  }
}
```

```

},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1::image/ami-*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:Owner": "amazon"
    }
  }
}
]
}

```

Restrict access to specific Systems Manager parameters

The following policy grants access to use Systems Manager parameters with a specific name.

The first statement grants users the permission to view Systems Manager parameters when selecting an AMI in the launch instance wizard. The second statement grants users the permission to only use parameters that are named `prod-*`.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ssm:DescribeParameters"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:GetParameters"
    ],
    "Resource": "arn:aws:ssm:us-east-2:123456123:parameter/prod-*"
  }
]
}

```

Example: 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.

```
{
  "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:region:111122223333:security-group/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Department": "Test"
      }
    }
  }
],
{
```

```
    "Effect": "Allow",
    "Action": [
      "ec2:ModifySecurityGroupRules"
    ],
    "Resource": [
      "arn:aws:ec2:region:111122223333:security-group-rule/*"
    ]
  }
}]}
```

Work with the Create Security Group dialog box

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at 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`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeSecurityGroups",
      "ec2:CreateSecurityGroup",
      "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2>DeleteSecurityGroup",
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:security-group/*",
    "Condition": {
      "ArnEquals": {
```

```
        "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"
    }
}
]
}
```

Example: 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.

```
{
  "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.

```
{
  "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 Elastic Compute Cloud

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

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: **AWSEC2CapacityReservationFleetRolePolicy**

This policy is attached to the service-linked role named **AWSServiceRoleForEC2CapacityReservationFleet** to allow Capacity Reservations to create, modify, and cancel Capacity Reservations on your behalf. For more information, see [Service-linked role 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: **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*.

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
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.

Note

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 the

AccessDenied error. For more information, see [Modifying a role trust policy](#) in the *IAM User Guide*.

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 [Using IAM roles](#) 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.

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](#)
- [Retrieve security credentials from instance metadata](#)
- [Grant a user permission to pass an IAM role to an instance](#)
- [Work with IAM roles](#)

Instance profiles

Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the IAM console, the console creates an instance profile automatically and gives it the same name as the role to which it corresponds. If you use the Amazon EC2 console to launch an instance with an IAM role or to attach an IAM role to an instance, you choose the role based on a list of instance profile names.

If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, with potentially different names. If you then use the AWS CLI, API, or an AWS SDK to launch an instance with an IAM role or to attach an IAM role to an instance, specify the instance profile name.

An instance profile can contain only one IAM role. This limit cannot be increased.

For more information, see [Instance Profiles](#) in the *IAM User Guide*.

Retrieve security credentials from instance metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/role-name`. The application is granted the permissions for the actions and resources that you've defined for the role through the security credentials associated with the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes before the expiration of the old credentials.

Warning

If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

IMDSv2

```
[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/iam/security-credentials/s3access
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

The following is example output.

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

For applications, AWS CLI, and Tools for Windows PowerShell commands that run on the instance, you do not have to explicitly get the temporary security credentials—the AWS SDKs, AWS CLI, and Tools for Windows PowerShell automatically get the credentials from the EC2 instance metadata service and use them. To make a call outside of the instance using temporary security credentials (for example, to test IAM policies), you must provide the access key, secret key, and the session token. For more information, see [Using Temporary Security Credentials to Request Access to AWS Resources](#) in the *IAM User Guide*.

For more information about instance metadata, see [Instance metadata and user data](#). For information about the instance metadata IP address, see [Retrieve instance metadata](#).

Grant a user permission to pass an IAM role to an instance

To enable a user to launch an instance with an IAM role or to attach or replace an IAM role for an existing instance, you must grant the user permission to use the following API actions:

- `iam:PassRole`
- `ec2:AssociateIamInstanceProfile`
- `ec2:ReplaceIamInstanceProfileAssociation`

For example, the following IAM policy grants users permission to launch instances with an IAM role, or to attach or replace an IAM role for an existing instance using the AWS CLI.

Note

If you want the policy to grant users access to all of your roles, specify the resource as `*` in the policy. However, please consider the principle of [least privilege](#) as a best-practice .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/DevTeam*"
    }
  ]
}
```

To grant users permission to launch instances with an IAM role, or to attach or replace an IAM role for an existing instance using the Amazon EC2 console, you must grant them permission to use

`iam:ListInstanceProfiles`, `iam:PassRole`, `ec2:AssociateIamInstanceProfile`, and `ec2:ReplaceIamInstanceProfileAssociation` in addition to any other permissions they might need. For example policies, see [Example policies for working in the Amazon EC2 console](#).

Work with IAM roles

You can create an IAM role and attach it to an instance during or after launch. You can also replace or detach an IAM role for an instance.

Contents

- [Create an IAM role](#)
- [Launch an instance with an IAM role](#)
- [Attach an IAM role to an instance](#)
- [Replace an IAM role](#)
- [Detach an IAM role](#)
- [Generate a policy for your IAM role based on access activity](#)

Create an IAM role

You must create an IAM role before you can launch an instance with that role or attach it to an instance.

Console

To create an IAM role using the IAM console

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles** then choose **Create role**.
3. On the **Select trusted entity** page, choose **AWS service**, and then select the **EC2** use case. Choose **Next**.
4. On the **Add permissions** page, select the policies that grants your instances access to the resources that they need. Choose **Next**.
5. On the **Name, review, and create** page, enter a name and description for the role. Optionally, add tags to the role. Choose **Create role**.

Command line

The following example creates an IAM role with a policy that allows the role to use an Amazon S3 bucket.

To create an IAM role and instance profile (AWS CLI)

1. Create the following trust policy and save it in a text file named `ec2-role-trust-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Create the `s3access` role and specify the trust policy that you created using the [create-role](#) command.

```
aws iam create-role \
  --role-name s3access \
  --assume-role-policy-document file://ec2-role-trust-policy.json
```

Example response

```
{
  "Role": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          }
        }
      ]
    }
  }
}
```



```

    ]
  },
  "RoleId": "AROAIIZKPBKS2LEXAMPLE",
  "CreateDate": "2013-12-12T23:46:37.247Z",
  "RoleName": "s3access",
  "Path": "/",
  "Arn": "arn:aws:iam::123456789012:role/s3access"
}
}

```

3. Create an access policy and save it in a text file named `ec2-role-access-policy.json`. For example, this policy grants administrative permissions for Amazon S3 to applications running on the instance.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:*"],
      "Resource": ["*"]
    }
  ]
}

```

4. Attach the access policy to the role using the [put-role-policy](#) command.

```

aws iam put-role-policy \
  --role-name s3access \
  --policy-name S3-Permissions \
  --policy-document file://ec2-role-access-policy.json

```

5. Create an instance profile named `s3access-profile` using the [create-instance-profile](#) command.

```

aws iam create-instance-profile --instance-profile-name s3access-profile

```

Example response

```

{
  "InstanceProfile": {
    "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",

```

```
    "Roles": [],  
    "CreateDate": "2013-12-12T23:53:34.093Z",  
    "InstanceProfileName": "s3access-profile",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:instance-profile/s3access-profile"  
  }  
}
```

6. Add the s3access role to the s3access-profile instance profile.

```
aws iam add-role-to-instance-profile \  
  --instance-profile-name s3access-profile \  
  --role-name s3access
```

Alternatively, you can use the following AWS Tools for Windows PowerShell commands:

- [New-IAMRole](#)
- [Register-IAMRolePolicy](#)
- [New-IAMInstanceProfile](#)

Launch an instance with an IAM role

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important

After you create an IAM role, it might take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting IAM roles](#) in the *IAM User Guide*.

New console

To launch an instance with an IAM role (console)

1. Follow the procedure to [launch an instance](#).

2. Expand **Advanced details**, and for **IAM instance profile**, select the IAM role that you created.

Note

The **IAM instance profile** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

3. Configure any other details that you require for your instance or accept the defaults, and select a key pair. For information about the fields in the launch instance wizard, see [Launch an instance using defined parameters](#).
4. In the **Summary** panel, review your instance configuration, and then choose **Launch instance**.
5. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. The AWS SDK does this for you.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

IMDSv1


```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Old console

To launch an instance with an IAM role (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch instance**.

3. Select an AMI and instance type and then choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **IAM role**, select the IAM role that you created.

 **Note**

The **IAM role** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

5. Configure any other details, then follow the instructions through the rest of the wizard, or choose **Review and Launch** to accept default settings and go directly to the **Review Instance Launch** page.
6. Review your settings, then choose **Launch** to choose a key pair and launch your instance.
7. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. The AWS SDK does this for you.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
  "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/iam/security-credentials/role_name
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-
credentials/role_name
```

Command line

You can use the AWS CLI to associate a role with an instance during launch. You must specify the instance profile in the command.

To launch an instance with an IAM role (AWS CLI)

1. Use the [run-instances](#) command to launch an instance using the instance profile. The following example shows how to launch an instance with the instance profile.

```
aws ec2 run-instances \  
  --image-id ami-11aa22bb \  
  --iam-instance-profile Name="s3access-profile" \  
  --key-name my-key-pair \  
  --security-groups my-security-group \  
  --subnet-id subnet-1a2b3c4d
```

Alternatively, use the [New-EC2Instance](#) Tools for Windows PowerShell command.

2. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. The AWS SDK does this for you.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Attach an IAM role to an instance

To attach an IAM role to an instance that has no role, the instance can be in the stopped or running state.

Console

To attach an IAM role to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, Security, Modify IAM role**.
4. Select the IAM role to attach to your instance, and choose **Save**.

Command line

To attach an IAM role to an instance (AWS CLI)

1. If required, describe your instances to get the ID of the instance to which to attach the role.

```
aws ec2 describe-instances
```

2. Use the [associate-iam-instance-profile](#) command to attach the IAM role to the instance by specifying the instance profile. You can use the Amazon Resource Name (ARN) of the instance profile, or you can use its name.

```
aws ec2 associate-iam-instance-profile \  
  --instance-id i-1234567890abcdef0 \  
  --iam-instance-profile Name="TestRole-1"
```

Example response

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-1234567890abcdef0",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0dbd8529a48294120",  
    "IamInstanceProfile": {  
      "Id": "AIPAJLNLDX3AMYZWNYYAY",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-1"  
    }  
  }  
}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2Instance](#)
- [Register-EC2IamInstanceProfile](#)

Replace an IAM role

To replace the IAM role on an instance that already has an attached IAM role, the instance must be in the running state. You can do this if you want to change the IAM role for an instance without detaching the existing one first. For example, you can do this to ensure that API actions performed by applications running on the instance are not interrupted.

Console

To replace an IAM role for an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, Security, Modify IAM role**.
4. Select the IAM role to attach to your instance, and choose **Save**.

Command line

To replace an IAM role for an instance (AWS CLI)

1. If required, describe your IAM instance profile associations to get the association ID for the IAM instance profile to replace.

```
aws ec2 describe-iam-instance-profile-associations
```

2. Use the [replace-iam-instance-profile-association](#) command to replace the IAM instance profile by specifying the association ID for the existing instance profile and the ARN or name of the instance profile that should replace it.

```
aws ec2 replace-iam-instance-profile-association \  
  --association-id iip-assoc-0044d817db6c0a4ba \  
  --iam-instance-profile Name="TestRole-2"
```

Example response

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-087711ddaf98f9489",  
    "State": "associating",  
    "AssociationId": "iip-assoc-09654be48e33b91e0",  
    "IamInstanceProfile": {  
      "Id": "AIPAJCJEDKX7QYHWYK7GS",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
    }  
  }  
}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2IamInstanceProfileAssociation](#)
- [Set-EC2IamInstanceProfileAssociation](#)

Detach an IAM role

You can detach an IAM role from a running or stopped instance.

Console

To detach an IAM role from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, Security, Modify IAM role**.
4. For **IAM role**, choose **No IAM Role**. Choose **Save**.
5. In the confirmation dialog box, enter **Detach**, and then choose **Detach**.

Command line

To detach an IAM role from an instance (AWS CLI)

1. If required, use [describe-iam-instance-profile-associations](#) to describe your IAM instance profile associations and get the association ID for the IAM instance profile to detach.

```
aws ec2 describe-iam-instance-profile-associations
```

Example response

```
{
  "IamInstanceProfileAssociations": [
    {
      "InstanceId": "i-088ce778fbfeb4361",
      "State": "associated",
      "AssociationId": "iip-assoc-0044d817db6c0a4ba",
      "IamInstanceProfile": {
        "Id": "AIPAJEDNCAA64SSD265D6",
        "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"
      }
    }
  ]
}
```



```
    }  
  }  
]  
}
```

2. Use the [disassociate-iam-instance-profile](#) command to detach the IAM instance profile using its association ID.

```
aws ec2 disassociate-iam-instance-profile --association-id iip-  
assoc-0044d817db6c0a4ba
```

Example response

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-087711ddaf98f9489",  
    "State": "disassociating",  
    "AssociationId": "iip-assoc-0044d817db6c0a4ba",  
    "IamInstanceProfile": {  
      "Id": "AIPAJEDNCAA64SSD265D6",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
    }  
  }  
}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2IamInstanceProfileAssociation](#)
- [Unregister-EC2IamInstanceProfile](#)

Generate a policy for your IAM role based on access activity

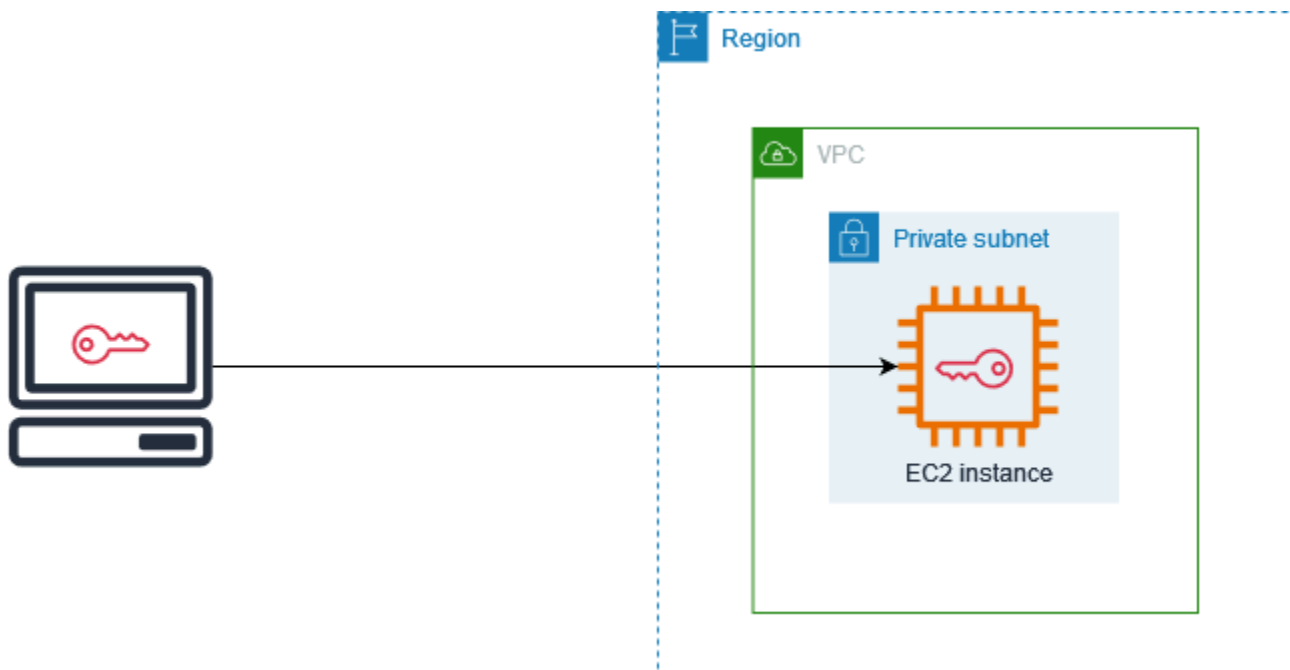
When you first create an IAM role for your applications, you might sometimes grant permissions beyond what is required. Before launching your application in your production environment, you can generate an IAM policy that is based on the access activity for an IAM role. IAM Access Analyzer reviews your AWS CloudTrail logs and generates a policy template that contains the permissions that have been used by the role in your specified date range. You can use the template to create a managed policy with fine-grained permissions and then attach it to the IAM role. That way, you grant only the permissions that the role needs to interact with AWS resources for your specific

use case. This helps you adhere to the best practice of [granting least privilege](#). To learn more, see [Generate policies based on access activity](#) in the *IAM User Guide*.

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](#). If you plan to connect to the instance using SSH, you must specify 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. 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 instance, see [Connect to your Linux 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 Linux 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](#)
- [Tag a key pair](#)
- [Describe your key pairs](#)
- [Delete your key pair](#)
- [Add or remove a public key on your instance](#)
- [Verify the fingerprint of your key pair](#)

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 ED25519 and 2048-bit SSH-2 RSA keys for Linux instances. Amazon EC2 supports 2048-bit SSH-2 RSA keys for Windows instances. ED25519 keys are not supported for Windows instances.

For steps to connect to your Linux instance using SSH after you have created a key pair, see [Connect to your Linux instance](#).

Contents

- [Create a key pair using Amazon EC2](#)
- [Create a key pair using AWS CloudFormation](#)
- [Create a key pair using a third-party tool and import the public key to Amazon EC2](#)

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 *AWS 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. For **Key pair type**, choose either **RSA** or **ED25519**.
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.

For `--key-name`, specify a name for the public key. The name can be up to 255 ASCII characters.

For `--key-type`, specify either `rsa` or `ed25519`. If you do not include the `--key-type` parameter, an `rsa` key is created by default. Note that ED25519 keys are not supported for Windows instances.

For `--key-format`, specify either `pem` or `ppk`. If you do not include the `--key-format` parameter, a `pem` file is created by default.

`--query "KeyMaterial"` prints the private key material to the output.

`--output text > my-key-pair.pem` saves the private key material in a file with the specified extension. The extension can be either `.pem` or `.ppk`. The private key can have a name that's different from the public key name, but for ease of use, use the same name.

```
aws ec2 create-key-pair \  
  --key-name my-key-pair \  
  --key-type rsa \  
  --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](#) AWS Tools for Windows PowerShell command as follows to generate the key and save it to a .pem or .ppk file.

For `-KeyName`, specify a name for the public key. The name can be up to 255 ASCII characters.

For `-KeyType`, specify either `rsa` or `ed25519`. If you do not include the `-KeyType` parameter, an `rsa` key is created by default. Note that ED25519 keys are not supported for Windows instances.

For `-KeyFormat`, specify either `pem` or `ppk`. If you do not include the `-KeyFormat` parameter, a `pem` file is created by default.

`KeyMaterial` prints the private key material to the output.

`Out-File -Encoding ascii -FilePath C:\path\my-key-pair.pem` saves the private key material in a file with the the specified extension. The extension can be `.pem` or `.ppk`. The private key can have a name that's different from the public key name, but for ease of use, use the same name.

```
PS C:\> (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: RSA and ED25519. Amazon EC2 does not accept DSA keys.

Note

ED25519 keys are not supported for Windows instances.

- Supported formats:
 - OpenSSH public key format (the format in `~/.ssh/authorized_keys`). If you connect using SSH while using the EC2 Instance Connect API, the SSH2 format is also supported.
 - 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. 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 an RSA or ED25519 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`. 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`.

⚠ 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](#) AWS CLI command.

To verify that the key pair was imported successfully

Use the [describe-key-pairs](#) AWS CLI command.

PowerShell

To import the public key to Amazon EC2

Use the [Import-EC2KeyPair](#) AWS Tools for Windows PowerShell command.

To verify that the key pair was imported successfully

Use the [Get-EC2KeyPair](#) AWS Tools for Windows PowerShell command.

Tag a key pair

To help categorize and manage the key pairs that you've either created using Amazon EC2 or imported to Amazon EC2, you can tag them with custom metadata. For more information about how tags work, see [Tag your Amazon EC2 resources](#).

Console

To view, add, or delete a tag for a key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Select a public key, and then choose **Actions, Manage tags**.
4. The **Manage tags** page displays any tags that are assigned to the public key.
 - To add a tag, choose **Add tag**, and then enter the tag key and value. You can add up to 50 tags per key. For more information, see [Tag restrictions](#).
 - To delete a tag, choose **Remove** next to the tag to delete.

5. Choose **Save**.

AWS CLI

To view the tags for your key pairs

Use the [describe-tags](#) AWS CLI command. In the following example, you describe the tags for all of your public keys.

```
$ aws ec2 describe-tags --filters "Name=resource-type,Values=key-pair"
```

```
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "key-0123456789EXAMPLE",
      "ResourceType": "key-pair",
      "Value": "Production"
    },
    {
      "Key": "Environment",
      "ResourceId": "key-9876543210EXAMPLE",
      "ResourceType": "key-pair",
      "Value": "Production"
    }
  ]
}
```

To describe the tags for a key pair

Use the [describe-key-pairs](#) AWS CLI command.

```
$ aws ec2 describe-key-pairs --key-pair-ids key-0123456789EXAMPLE
```

```
{
  "KeyPairs": [
    {
      "KeyName": "MyKeyPair",
      "KeyFingerprint":
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
      "KeyPairId": "key-0123456789EXAMPLE",
      "Tags": [
```

```
{
  {
    "Key": "Environment",
    "Value": "Production"
  }
}
```

To tag a key pair

Use the [create-tags](#) AWS CLI command. In the following example, the public key is tagged with `Key=Cost-Center` and `Value=CC-123`.

```
$ aws ec2 create-tags --resources key-0123456789EXAMPLE --tags Key=Cost-Center,Value=CC-123
```

To delete a tag from a key pair

Use the [delete-tags](#) AWS CLI command. For examples, see [Examples](#) in the *AWS CLI Command Reference*.

PowerShell

To view tags for your key pairs

Use the [Get-EC2Tag](#) command.

To describe the tags for a key pair

Use the [Get-EC2KeyPair](#) command.

To tag a key pair

Use the [New-EC2Tag](#) command.

To delete a tag from a key pair

Use the [Remove-EC2Tag](#) command.

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.

Topics

- [Describe your key pairs](#)
- [Retrieve the public key material](#)
- [Identify the public key specified at launch](#)

Describe your key pairs

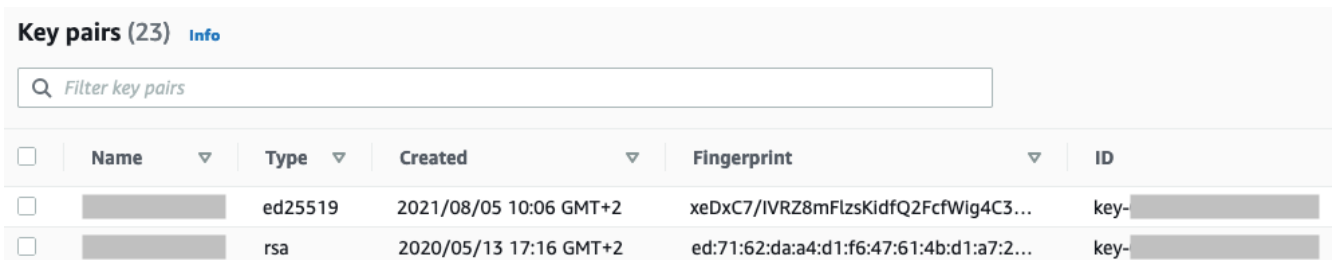
You can view the following information about your public keys that are stored in Amazon EC2: public key name, ID, key type, fingerprint, public key material, the date and time (in the UTC time zone) the key was created by Amazon EC2 (if the key was created by a third-party tool, then it's the date and time the key was imported to Amazon EC2), and any tags that are associated with the public key.

You can use the Amazon EC2 console or AWS CLI to view information about your public keys.

Console

To view information about your public keys

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.



The screenshot shows the 'Key pairs (23)' section in the Amazon EC2 console. It includes a search bar labeled 'Filter key pairs' and a table with columns: Name, Type, Created, Fingerprint, and ID. Two key pairs are visible in the table.

<input type="checkbox"/>	Name	Type	Created	Fingerprint	ID
<input type="checkbox"/>	[REDACTED]	ed25519	2021/08/05 10:06 GMT+2	xeDxC7/IVRZ8mFlzsKidfQ2FcfWig4C3...	key-[REDACTED]
<input type="checkbox"/>	[REDACTED]	rsa	2020/05/13 17:16 GMT+2	ed:71:62:da:a4:d1:f6:47:61:4b:d1:a7:2...	key-[REDACTED]

4. To view a public key's tags, select the check box next to the key, and then choose **Actions, Manage tags**.

AWS CLI

To describe a public key

Use the [describe-key-pairs](#) command and specify the `--key-names` parameter.

```
aws ec2 describe-key-pairs --key-names key-pair-name
```

Example output

```
{
  "KeyPairs": [
    {
      "KeyPairId": "key-0123456789example",
      "KeyFingerprint":
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
      "KeyName": "key-pair-name",
      "KeyType": "rsa",
      "Tags": [],
      "CreateTime": "2022-04-28T11:37:26.000Z"
    }
  ]
}
```

Alternatively, instead of `--key-names`, you can specify the `--key-pair-ids` parameter to identify the public key.

```
aws ec2 describe-key-pairs --key-pair-ids key-0123456789example
```

To view the public key material in the output, you must specify the `--include-public-key` parameter.

```
aws ec2 describe-key-pairs --key-names key-pair-name --include-public-key
```

Example output – In the output, the `PublicKey` field contains the public key material.

```
{
  "KeyPairs": [
    {
      "KeyPairId": "key-0123456789example",
      "KeyFingerprint":
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
      "KeyName": "key-pair-name",
      "KeyType": "rsa",
      "Tags": [],
      "PublicKey": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIIj7az1DjVHAsSxgcpCRZ3oWnTm0nAFM64y9jd22ioI/ my-key-pair",
      "CreateTime": "2022-04-28T11:37:26.000Z"
    }
  ]
}
```

```
]
}
```

Retrieve the public key material

You can use various methods to get access to the public key material. You can retrieve the public key material from the matching private key on your local computer, or from the instance metadata or the `authorized_keys` file on the instance that was launched with the public key, or by using the `describe-key-pairs` AWS CLI command.

Use one of the following methods to retrieve the public key material.

From the private key

To retrieve the public key material from the private key

On your local Linux or macOS computer, you can use the `ssh-keygen` command to retrieve the public key for your key pair. Specify the path where you downloaded your private key (the `.pem` file).

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

The command returns the public key, as shown in the following example.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzo0WbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJ0I0iBXr
lsLnBItnctkiJ7FbtXJMXLvwwJryDUi1BMTjYtwB+QhYXUM0zce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WriUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb70z1PnW0yN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

If the command fails, run the following command to ensure that you've changed the permissions on your private key pair file so that only you can view it.

```
chmod 400 key-pair-name.pem
```

From the instance metadata

You can use Instance Metadata Service Version 2 or Instance Metadata Service Version 1 to retrieve the public key from the instance metadata.

Note

If you change the key pair that you use to connect to the instance, Amazon EC2 does not update the instance metadata to show the new public key. The instance metadata continues to show the public key for the key pair that you specified when you launched the instance.

To retrieve the public key material from the instance metadata

Use one of the following commands from your instance.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Example output

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V  
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzo0WbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr  
lsLnBItnctkiJ7FbtXJMXLvVwJryDUi1BMTjYtwB+QhYXUM0zce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ  
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb  
BQoQzd8v7yeb70z1PnW0yN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE key-pair-name
```

For more information about instance metadata, see [Retrieve instance metadata](#).

From the instance

If you specify a key pair when launching a Linux instance, when the instance boots for the first time, the content of the public key is placed on the instance in an entry within `~/ .ssh/authorized_keys`.

To retrieve the public key material from an instance

1. [Connect to your instance](#).
2. In the terminal window, open the `authorized_keys` file using your favorite text editor (such as **vim** or **nano**).

```
[ec2-user ~]$ nano ~/.ssh/authorized_keys
```

The `authorized_keys` file opens, displaying the public key followed by the name of the key pair. The following is an example entry for the key pair named *key-pair-name*.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzo0WbkM4yxyb/wB96xbiFveSFJu0p/d6RJhJ0I0iBXr
lsLnBItnctkiJ7FbtXJMXLvwwJryDUi1BMTjYtwB+QhYXUM0zce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJR6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb70z1PnW0yN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE key-pair-name
```

From `describe-key-pairs`

To retrieve the public key material from the `describe-key-pairs` AWS CLI command

Use the [describe-key-pairs](#) command and specify the `--key-names` parameter to identify the public key. To include the public key material in the output, specify the `--include-public-key` parameter.

```
aws ec2 describe-key-pairs --key-names key-pair-name --include-public-key
```

Example output – In the output, the `PublicKey` field contains the public key material.

```
{
  "KeyPairs": [
    {
      "KeyPairId": "key-0123456789example",
      "KeyFingerprint":
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
      "KeyName": "key-pair-name",
      "KeyType": "rsa",
      "Tags": [],
      "PublicKey": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIij7az1DjVHAsSxgcpCRZ3oWnTm0nAFM64y9jd22ioI/ my-key-pair",
```

```
        "CreateTime": "2022-04-28T11:37:26.000Z"
      }
    ]
  }
```

Alternatively, instead of `--key-names`, you can specify the `--key-pair-ids` parameter to identify the public key.

```
aws ec2 describe-key-pairs --key-pair-ids key-0123456789example --include-public-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.

To identify the public key that was specified at launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Details** tab, under **Instance details**, the **Key pair assigned at launch** field displays the name of the public key that you specified when you launched the instance.

Note

The value of the **Key pair assigned at launch** field does not change even if you change the public key on the instance, or add public keys.

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](#) AWS CLI command.

PowerShell

To delete your public key on Amazon EC2

Use the [Remove-EC2KeyPair](#) AWS Tools for Windows PowerShell command.

Add or remove a public key on your 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 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.

Hash functions used to calculate fingerprints

Key pair source	RSA key pairs	ED25519 key pairs
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 [Windows 10 Installation Guide](#). The example in the instructions installs the Ubuntu distribution of Linux, but you can install any distribution. You are prompted to restart your computer for the changes to take effect.

- **If you created 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
```

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
```

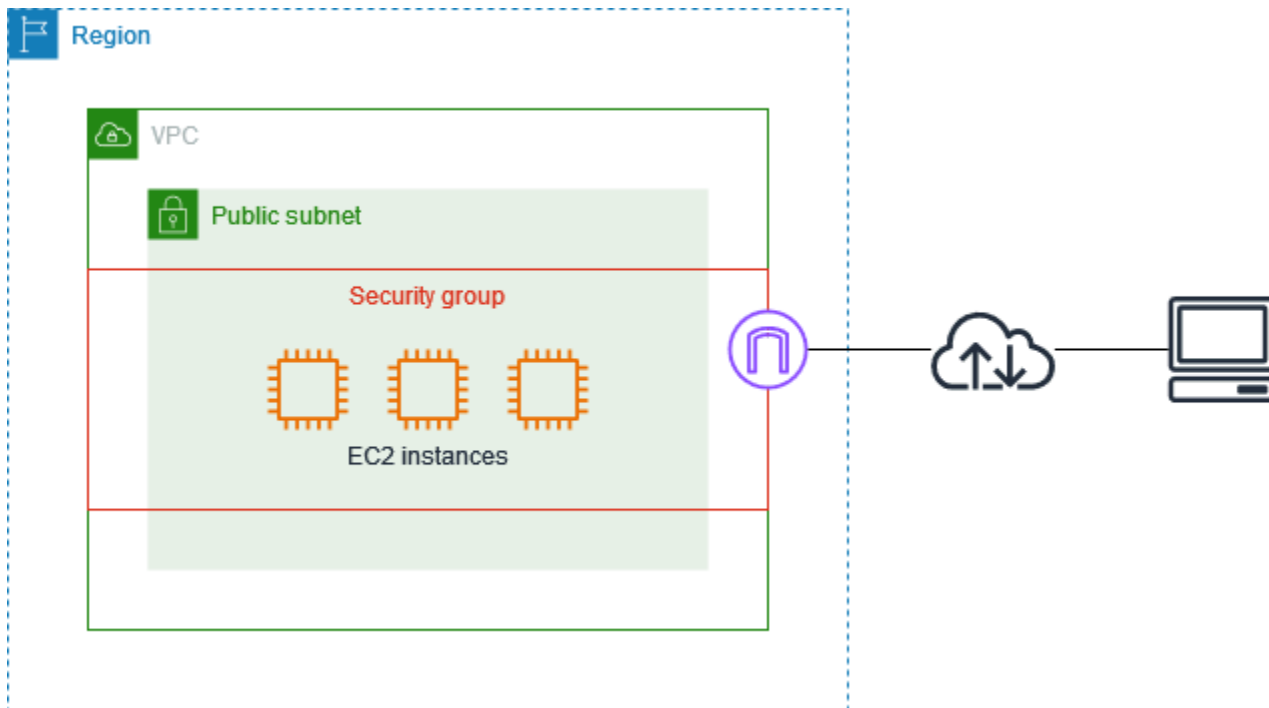
For ED25519 key pairs:

```
ssh-keygen -l -f path_to_private_key
```

Amazon EC2 security groups for Linux instances

A *security group* acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance. When you launch an instance, you can specify one or more security groups. If you don't specify a security group, Amazon EC2 uses the default security group for the VPC. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. New and modified rules are automatically applied to all instances that are associated with the security group. When Amazon EC2 decides whether to allow traffic to reach an instance, it evaluates all of the rules from all of the security groups that are associated with the instance.

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 assigned to 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 assigned to it, then each instance can receive any traffic sent from the other instances.



After you launch an instance, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (eth0). For more information, see [Change an instance's security group](#). You can also change the security groups associated with any other network interface. For more information, see [Modify network interface attributes](#).

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.

To allow traffic to a Windows instance, see [Amazon EC2 security groups for Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

There is no additional charge for using security groups.

Contents

- [Security group rules](#)
- [Security group connection tracking](#)
- [Default and custom security groups](#)
- [Work with security groups](#)

- [Security group rules for different use cases](#)

Security group rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group. The rules also control the outbound traffic that's allowed to leave them.

The following are the characteristics of security group rules:

- By default, security groups contain outbound rules that allow all outbound traffic. You can delete these rules. Note that Amazon EC2 blocks traffic on port 25 by default. For more information, see [Restriction on email sent using port 25](#).
- Security group rules are always permissive; you can't create rules that deny access.
- Security group rules enable you to filter traffic based on protocols and port numbers.
- Security groups are stateful—if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. For more information, see [Security group connection tracking](#).
- You can add and remove rules at any time. Your changes are automatically applied to the instances that are associated with the security group.

The effect of some rule changes can depend on how the traffic is tracked. For more information, see [Security group connection tracking](#).

- When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. Amazon EC2 uses this set of rules to determine whether to allow access.

You can assign multiple security groups to an instance. Therefore, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. We recommend that you condense your rules as much as possible.

Note

Security groups cannot block DNS requests 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*).

For each rule, you specify the following:

- **Name:** The name for the security group (for example, "my-security-group").

A name can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=;{}!$*`. When the name contains trailing spaces, we trim the spaces when we save the name. For example, if you enter "Test Security Group " for the name, we store it as "Test Security Group".

- **Protocol:** The protocol to allow. The most common protocols are 6 (TCP), 17 (UDP), and 1 (ICMP).
- **Port range:** For TCP, UDP, or a custom protocol, the range of ports to allow. You can specify a single port number (for example, 22), or range of port numbers (for example, 7000-8000).
- **ICMP type and code:** For ICMP, the ICMP type and code. For example, use type 8 for ICMP Echo Request or type 128 for ICMPv6 Echo Request.
- **Source or destination:** The source (inbound rules) or destination (outbound rules) for the traffic to allow. Specify one of the following:
 - A single IPv4 address. You must use the /32 prefix length. For example, 203.0.113.1/32.
 - A single IPv6 address. You must use the /128 prefix length. For example, 2001:db8:1234:1a00::123/128.
 - A range of IPv4 addresses, in CIDR block notation. For example, 203.0.113.0/24.
 - A range of IPv6 addresses, in CIDR block notation. For example, 2001:db8:1234:1a00::/64.
 - The ID of a prefix list. For example, p1-1234abc1234abc123. For more information, see [Prefix lists](#) in the *Amazon VPC User Guide*.
 - The ID of a security group (referred to here as the specified security group). For example, the current security group, a security group from the same VPC, or a security group for a peered VPC. This allows traffic based on the private IP addresses of the resources associated with the specified security group. This does not add rules from the specified security group to the current security group.

- **(Optional) Description:** You can add a description for the rule, which can help you identify it later. A description can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=;{}!$*`.

When you create a security group rule, AWS assigns a unique ID to the rule. You can use the ID of a rule when you use the API or CLI to modify or delete the rule.

When you specify a security group as the source or destination for a rule, the rule affects all instances that are associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses). For more information about IP addresses, see [Amazon EC2 instance IP addressing](#). If your security group rule references a deleted security group in the same VPC or in a peer VPC, or if it references a security group in a peer VPC for which the VPC peering connection has been deleted, the rule is marked as stale. For more information, see [Working with Stale Security Group Rules](#) in the *Amazon VPC Peering Guide*.

If there is more than one rule for a specific port, Amazon EC2 applies the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address 203.0.113.1, and another rule that allows access to TCP port 22 from everyone, everyone has access to TCP port 22.

When you add, update, or remove rules, the changes are automatically applied to all instances associated with the security group.

Security group connection tracking

Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied. With this approach, security groups are stateful. This means that responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa.

As an example, suppose that you initiate 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. These connections must be tracked to ensure symmetric routing, as there could be multiple valid reply paths.

- Egress-only internet gateways
- Gateway Load Balancers
- Global Accelerator accelerators
- NAT gateways
- Network Firewall firewall endpoints
- Network Load Balancers
- AWS PrivateLink (interface VPC endpoints)
- Transit gateway attachments
- AWS Lambda (Hyperplane elastic network interfaces)

Throttling

Amazon EC2 defines the maximum number of connections that can be tracked per instance. After the maximum is reached, any packets that are sent or received are dropped because a new connection cannot be established. When this happens, applications that send and receive packets cannot communicate properly. Use the `contrack_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 `contrack_allowance_exceeded` network performance metric. For more information, see [Monitor network performance for 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.

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 only available for [instances built on the AWS Nitro System](#).

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 (5 days). Default: 432000 seconds. 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 (3 minutes). 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 `contrack_allowance_exceeded` and `contrack_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 `contrack_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 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.

Default and custom security groups

Your AWS account automatically has a default security group for the default VPC in each Region. If you don't specify a security group when you launch an instance, the instance is automatically associated with the default security group for the VPC. If you don't want your instances to use the default security group, you can create your own custom security groups and specify them when you launch your instances.

Contents

- [Default security groups](#)
- [Custom security groups](#)

Default security groups

Each VPC comes with a default security group. We recommend that you create security groups for specific instances or groups of instances instead of using the default security group. However, if

you don't specify a security group when you launch an instance, we associate the instance with the default security group for the VPC.

The name of a default security group is "default". The following are the default rules for a default security group.

Inbound

Source	Protocol	Port range	Description
<code>sg-1234567890abcde</code> <code>f0</code>	All	All	Allows inbound traffic from all resources that are assigned to this security group. The source is the ID of this security group.

Outbound

Destination	Protocol	Port range	Description
<code>0.0.0.0/0</code>	All	All	Allows all outbound IPv4 traffic.
<code>::/0</code>	All	All	Allows all outbound IPv6 traffic. This rule is added only if your VPC has an associated IPv6 CIDR block.

Default security group basics

- You can change the rules for a default security group.
- You can't delete a default security group. If you try to delete a default security group, we return the following error code: `Client.CannotDelete`.

Custom security groups

You can create multiple security groups to reflect the different roles that your instances play; for example, web servers or database servers.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=&:{}!$*`

A security group name cannot start with the following: `sg-`. A security group name must be unique for the VPC.

The following are the default rules for a security group that you create:

- Allows no inbound traffic
- Allows all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. You can also change its outbound rules.

For more information about the rules you can add to a security group, see [Security group rules for different use cases](#).

Work with security groups

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group. For more information, see [Assign a security group to an instance](#).

After you launch an instance, you can change its security groups. For more information, see [Change an instance's security group](#).

You can create, view, update, and delete security groups and security group rules using the Amazon EC2 console and the command line tools.

Tasks

- [Create a security group](#)
- [Copy a security group](#)
- [View your security groups](#)
- [Add rules to a security group](#)

- [Update security group rules](#)
- [Delete rules from a security group](#)
- [Delete a security group](#)
- [Assign a security group to an instance](#)
- [Change an instance's security group](#)

Create a security group

Although you can use the default security group for your instances, you might want to create your own groups to reflect the different roles that instances play in your system.

By default, new security groups start with only an outbound rule that allows all traffic to leave the instances. You must add rules to enable any inbound traffic or to restrict the outbound traffic.

A security group can be used only in the VPC for which it is created.

Console

To create a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create security group**.
4. In the **Basic details** section, do the following.
 - a. Enter a descriptive name and brief description for the security group. They can't be edited after the security group is created. The name and description can be up to 255 characters long. The valid characters are a-z, A-Z, 0-9, spaces, and `._-:/()#,@[]+=&:{}!$*`.
 - b. For **VPC**, choose the VPC.
5. You can add security group rules now, or you can add them later. For more information, see [Add rules to a security group](#).
6. You can add tags now, or you can add them later. To add a tag, choose **Add new tag** and enter the tag key and value.
7. Choose **Create security group**.

Command line

To create a security group

Use one of the following commands:

- [create-security-group](#) (AWS CLI)
- [New-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Copy a security group

You can create a new security group by creating a copy of an existing one. When you copy a security group, the copy is created with the same inbound and outbound rules as the original security group. If the original security group is in a VPC, the copy is created in the same VPC unless you specify a different one.

The copy receives a new unique security group ID and you must give it a name. You can also add a description.

You can't copy a security group from one Region to another Region.

You can create a copy of a security group using the Amazon EC2 console.

To copy a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to copy and choose **Actions, Copy to new security group**.
4. Specify a name and optional description, and change the VPC and security group rules if needed.
5. Choose **Create**.

View your security groups

You can view information about your security groups using one of the following methods.

Console

To view your security groups

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Your security groups are listed. To view the details for a specific security group, including its inbound and outbound rules, choose its ID in the **Security group ID** column.

Command line

To view your security groups

Use one of the following commands.

- [describe-security-groups](#) (AWS CLI)
- [describe-security-group-rules](#) (AWS CLI)
- [Get-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Amazon EC2 Global View

You can use Amazon EC2 Global View to view your security groups across all Regions for which your AWS account is enabled. For more information, see [Amazon EC2 Global View](#).


Add rules to a security group

When you add a rule to a security group, the new rule is automatically applied to any instances that are associated with the security group. There might be a short delay before the rule is applied. For more information, see [Security group rules for different use cases](#) and [Security group rules](#).

Console

To add an inbound rule to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group, and choose **Actions, Edit inbound rules**.
4. For each rule, choose **Add rule** and do the following.

- a. For **Type**, choose the type of protocol to allow.
 - For **Custom TCP** or **Custom UDP**, you must enter the port range to allow. For example, 0-99.
 - For **Custom ICMP**, you must choose the ICMP type from **Protocol**. The port range is configured for you. For example, to allow **ping** commands, choose **Echo Request** from **Protocol**.
 - For any other type, the protocol and port range are configured for you.
 - b. For **Source**, do one of the following to allow traffic.
 - Choose **Custom** and then enter an IP address in CIDR notation, a CIDR block, another security group, or a prefix list.
 - Choose **Anywhere** to allow all traffic for the specified protocol to reach your instance. This option automatically adds the 0.0.0.0/0 IPv4 CIDR block as the source. If your security group is in a VPC that's enabled for IPv6, this option automatically adds a rule for the ::/0 IPv6 CIDR block.
-  **Warning**

If you choose **Anywhere**, you enable all IPv4 and IPv6 addresses to access your instance the specified protocol. If you are adding rules for ports 22 (SSH) or 3389 (RDP), you should authorize only a specific IP address or range of addresses to access your instance.
- Choose **My IP** to allow inbound traffic from only your local computer's public IPv4 address.
 - c. For **Description**, optionally specify a brief description for the rule.
5. Choose **Preview changes**, **Save rules**.

To add an outbound rule to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group, and choose **Actions**, **Edit outbound rules**.
4. For each rule, choose **Add rule** and do the following.

- a. For **Type**, choose the type of protocol to allow.
 - For **Custom TCP** or **Custom UDP**, you must enter the port range to allow. For example, 0-99.
 - For **Custom ICMP**, you must choose the ICMP type from **Protocol**. The port range is configured for you.
 - For any other type, the protocol and port range are configured automatically.
 - b. For **Destination**, do one of the following.
 - Choose **Custom** and then enter an IP address in CIDR notation, a CIDR block, another security group, or a prefix list for which to allow outbound traffic.
 - Choose **Anywhere** to allow outbound traffic to all IP addresses. This option automatically adds the 0.0.0.0/0 IPv4 CIDR block as the destination.

If your security group is in a VPC that's enabled for IPv6, this option automatically adds a rule for the ::/0 IPv6 CIDR block.
 - Choose **My IP** to allow outbound traffic only to your local computer's public IPv4 address.
 - c. (Optional) For **Description**, specify a brief description for the rule.
5. Choose **Preview changes, Confirm**.

Command line

To add rules to a security group

Use one of the following commands.

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

To add one or more egress rules to a security group

Use one of the following commands.

- [authorize-security-group-egress](#) (AWS CLI)
- [Grant-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

Update security group rules

You can update a security group rule using one of the following methods. The updated rule is automatically applied to any instances that are associated with the security group.

Console

When you modify the protocol, port range, or source or destination of an existing security group rule using the console, the console deletes the existing rule and adds a new one for you.

To update a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group.
4. Choose **Actions, Edit inbound rules** to update a rule for inbound traffic or **Actions, Edit outbound rules** to update a rule for outbound traffic.
5. Update the rule as required.
6. Choose **Preview changes, Confirm**.

To tag a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group.
4. On the **Inbound rules** or **Outbound rules** tab, select the check box for the rule and then choose **Manage tags**.
5. The **Manage tags** page displays any tags that are assigned to the rule. To add a tag, choose **Add tag** and enter the tag key and value. To delete a tag, choose **Remove** next to the tag that you want to delete.
6. Choose **Save changes**.

Command line

You cannot modify the protocol, port range, or source or destination of an existing rule using the Amazon EC2 API or a command line tools. Instead, you must delete the existing rule and add a new rule. You can, however, update the description of an existing rule.

To update a rule

Use one the following command.

- [modify-security-group-rules](#) (AWS CLI)

To update the description for an existing inbound rule

Use one of the following commands.

- [update-security-group-rule-descriptions-ingress](#) (AWS CLI)
- [Update-EC2SecurityGroupRuleIngressDescription](#) (AWS Tools for Windows PowerShell)

To update the description for an existing outbound rule

Use one of the following commands.

- [update-security-group-rule-descriptions-egress](#) (AWS CLI)
- [Update-EC2SecurityGroupRuleEgressDescription](#) (AWS Tools for Windows PowerShell)

To tag a security group rule

Use one of the following commands.

- [create-tags](#) (AWS CLI)
- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

Delete rules from a security group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

You can delete rules from a security group using one of the following methods.

Console

To delete a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update, choose **Actions**, and then choose **Edit inbound rules** to remove an inbound rule or **Edit outbound rules** to remove an outbound rule.
4. Choose the **Delete** button to the right of the rule to delete.
5. Choose **Save rules**. Alternatively, choose **Preview changes**, review your changes, and choose **Confirm**.

Command line

To remove one or more ingress rules from a security group

Use one of the following commands.

- [revoke-security-group-ingress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

To remove one or more egress rules from a security group

Use one of the following commands.

- [revoke-security-group-egress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

Delete a security group

You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group in the same VPC. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

Console

To delete a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group and choose **Actions, Delete security groups**.
4. When prompted for confirmation, choose **Delete**.

Command line

To delete a security group

Use one of the following commands.

- [delete-security-group](#) (AWS CLI)
- [Remove-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Assign a security group to an instance

You can assign one or more security groups to an instance when you launch the instance. You can also specify one or more security groups in a launch template. The security groups are assigned to all instances that are launched using the launch template.

- To assign a security group to an instance when you launch the instance, see [Network settings](#) of [Launch an instance using defined parameters](#) (new console) or [Step 6: Configure Security Group](#) (old console).
- To specify a security group in a launch template, see [Network settings](#) of [Create a launch template from parameters](#).

Change an instance's security group

After you launch an instance, you can change its security groups by adding or removing security groups.

Requirements

- The instance must be in the `running` or `stopped` state.

- A security group is specific to a VPC. You can assign a security group to one or more instances launched in the VPC for which you created the security group.

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**.

Command line

To change the security groups for an instance

Use one of the following commands.

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Security group rules for different use cases

You can create a security group and add rules that reflect the role of the instance that's associated with the security group. For example, an instance that's configured as a web server needs security group rules that allow inbound HTTP and HTTPS access. Likewise, a database instance needs rules that allow access for the type of database, such as access over port 3306 for MySQL.

The following are examples of the kinds of rules that you can add to security groups for specific kinds of access.

Examples

- [Web server rules](#)

- [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](#)
- [VPC peering 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
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
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

Protocol type	Protocol number	Port	Source IP
			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 target) associated with this security group

To mount an Amazon EFS file system on your Amazon EC2 instance, you must connect to your instance. Therefore, the security group associated with your instance must have rules that allow inbound SSH from your local computer or local network.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	22 (SSH)	The IP address range of your local computer, or the range of IP addresses (in CIDR block notation) for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing rules

If you're using a load balancer, the security group associated with your load balancer must have rules that allow communication with your instances or targets. For more information, see [Configure security groups for your Classic Load Balancer](#) in the *User Guide for Classic Load Balancers*, and [Security groups for your Application Load Balancer](#) in the *User Guide for Application Load Balancers*.

VPC peering rules

You can update the inbound or outbound rules for your VPC security groups to reference security groups in the peered VPC. Doing so allows traffic to flow to and from instances that are associated with the referenced security group in the peered VPC. For more information about how to configure security groups for VPC peering, see [Updating your security groups to reference peer VPC groups](#).

Access Amazon EC2 using an interface VPC endpoint

You can improve the security posture of your VPC by creating a private connection between your VPC and Amazon EC2. You can access Amazon EC2 as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to access Amazon EC2.

For more information, see [Access AWS services through AWS PrivateLink](#) in the *AWS PrivateLink Guide*.

Contents

- [Create an interface VPC endpoint](#)

- [Create an endpoint policy](#)

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.

```
{
  "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"
        }
      }
    }
  ]
}
```

Update management in Amazon EC2

We recommend that you regularly patch, update, and secure the operating system and applications on your EC2 instances. You can use [AWS Systems Manager Patch Manager](#) to automate the process of installing security-related updates for both the operating system and applications.

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.

To view the latest updates for the AWS Windows AMIs, see [Windows AMI version history](#) in the *AWS Windows AMI Reference*.

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 and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

Note

Not all AWS services are HIPAA eligible. For more information, see the [HIPAA Eligible Services Reference](#).

- [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](#).
- [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.

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](#). 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 [instances built on the AWS Nitro System](#). You can select one of Amazon's prebuilt AMIs or create one yourself.

Costs

There is no additional cost for using NitroTPM. You pay only for the underlying resources that you use.

Topics

- [Considerations](#)
- [Prerequisites for launching Linux instances](#)
- [Create a Linux AMI for NitroTPM support](#)
- [Verify whether an AMI is enabled for NitroTPM](#)
- [Enable or stop using NitroTPM on an instance](#)

Considerations

The following considerations apply when using NitroTPM:

- BitLocker volumes that are encrypted with NitroTPM-based keys can only be used on the original instance.
- The NitroTPM state is not included in [Amazon EBS snapshots](#).
- The NitroTPM state is not included in [VM Import/Export](#) images.
- NitroTPM support is enabled by specifying a value of `v2.0` for the `tpm-support` parameter when creating an AMI. After you launch an instance with the AMI, you can't modify the attributes on the instance. Instances with NitroTPM do not support the [ModifyInstanceAttribute](#) API.
- You can only create an AMI with NitroTPM configured by using the [RegisterImage](#) API by using the AWS CLI and not with the Amazon EC2 console.
- NitroTPM is not supported on Outposts.
- NitroTPM is not supported in Local Zones or Wavelength Zones.

Prerequisites for launching Linux instances

To launch a Linux instance with NitroTPM enabled, the following prerequisites must be in place. For the prerequisites for launching a Windows instance, see [Prerequisites for launching Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

AMI

Requires an AMI with NitroTPM enabled.

Currently, there are no NitroTPM-enabled Amazon Linux AMIs. 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 for NitroTPM support](#).

Operating system

The AMI must include an operating system with a TPM 2.0 Command Response Buffer (CRB) driver. Most current operating systems, such as Amazon Linux 2, contain a TPM 2.0 CRB driver.

Instance type

Supported virtualized instance types:

- **General purpose:** M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6a, M6i, M6id, M6idn, M6in, M7a, M7i, M7i-flex, T3, T3a
- **Compute optimized:** C5, C5a, C5ad, C5d, C5n, C6a, C6i, C6id, C6in, C7a, C7i
- **Memory optimized:** R5, R5a, R5ad, R5b, R5d, R5dn, R5n, R6a, R6i, R6idn, R6in, R6id, R7a, R7i, R7iz, z1d
- **Storage optimized:** D3, D3en, I3en, I4i
- **Accelerated computing:** G4dn, G5, G6, Gr6, Inf1, Inf2
- **High-performance computing:** Hpc6a, Hpc6id

Not supported: Graviton-based instances, Xen instances, Mac instances, and bare metal instances

UEFI boot mode

NitroTPM requires that an instance runs in UEFI boot mode, which requires that the AMI must be configured for UEFI boot mode. For more information, see [UEFI Secure Boot](#).

Create a Linux AMI for NitroTPM support

You configure your Linux AMI for NitroTPM support when you register the AMI. You can't configure NitroTPM support later.

For the list of Windows AMIs that are preconfigured for NitroTPM support, see [Prerequisites for launching Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

To register a Linux AMI for NitroTPM support

1. Launch a temporary instance with your required Linux AMI.

2. After the instance reaches the running state, create a snapshot of the instance's root volume.
3. Register the new AMI. Use the [register-image](#) command. For `--tpm-support`, specify `v2.0`. For `--boot-mode`, specify `uefi`. And specify a block device mapping for the root volume using the snapshot you created in the previous step.

```
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=snapshot_id} \  
  --tpm-support v2.0
```

Expected output

```
{  
  "ImageId": "ami-0123456789example"  
}
```

4. Terminate the temporary instance you launched in step 1, if it is no longer needed.

Verify whether an AMI is enabled for NitroTPM

You can use either `describe-images` or `describe-image-attributes` to verify whether an AMI is enabled for NitroTPM.

To verify whether an AMI is enabled for NitroTPM using `describe-images`

Use the [describe-images](#) command and specify the AMI ID.

```
aws ec2 describe-images --image-ids ami-0123456789example
```

If NitroTPM is enabled for the AMI, `"TpmSupport": "v2.0"` appears in the output.

```
{  
  "Images": [  
    {  
      ...  
      "BootMode": "uefi",  
      ...  
    }  
  ]  
}
```

```

    "TpmSupport": "v2.0"
  }
]
}

```

To verify whether an AMI is enabled for NitroTPM using `describe-image-attribute`

Use the [describe-image-attribute](#) command and specify the attribute parameter with the `tpmSupport` value.

Note

You must be the AMI owner to call `describe-image-attribute`.

```

aws ec2 describe-image-attribute \
  --region us-east-1 \
  --image-id ami-0123456789example \
  --attribute tpmSupport

```

If NitroTPM is enabled for the AMI, the value for `TpmSupport` is `"v2.0"`. Note that `describe-image-attribute` only returns the attributes that are specified in the request.

```

{
  "ImageId": "ami-0123456789example",
  "TpmSupport": {
    "Value": "v2.0"
  }
}

```

Enable or stop using NitroTPM on an instance

When you launch an instance from an AMI that has NitroTPM support enabled, the instance launches with NitroTPM enabled. You can configure the instance to stop using NitroTPM. You can verify whether an instance is enabled for NitroTPM.

Topics

- [Launch an instance with NitroTPM enabled](#)
- [Stop using NitroTPM on an instance](#)

- [Verify whether NitroTPM is accessible inside the instance](#)

Launch an instance with NitroTPM enabled

When you launch an instance with the [prerequisites](#), NitroTPM is automatically enabled on the instance. You can only enable NitroTPM on an instance at launch. For information about launching an instance, see [Launch your 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, use `tpm-tools`.

For more information about disabling the device driver, see the documentation for your operating system.

Verify whether NitroTPM is accessible inside the instance

To verify whether an instance is enabled for NitroTPM support using the AWS CLI

Use the [describe-instances](#) AWS CLI command and specify the instance ID. Currently, the Amazon EC2 console does not display the `TpmSupport` field.

```
aws ec2 describe-instances --instance-ids i-0123456789example
```

If NitroTPM support is enabled on the instance, `"TpmSupport": "v2.0"` appears in the output.

```
"Instances": {  
  "InstanceId": "0123456789example",  
  "InstanceType": "c5.large",  
  ...  
  "BootMode": "uefi",  
  "TpmSupport": "v2.0"  
  ...  
}
```

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.

[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](#)

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.

[Amazon EFS](#)

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances.

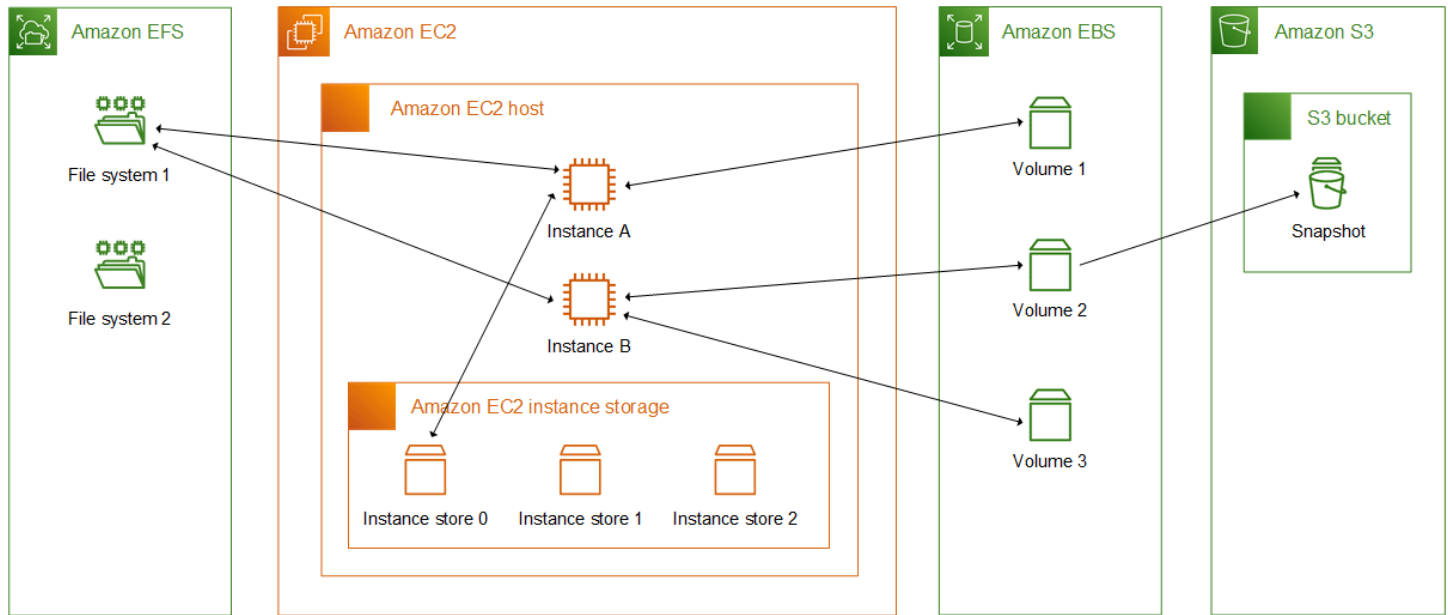
[Amazon S3](#)

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.

[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.

The following figure shows the relationship between these storage options and your instance.



Storage pricing

Open [AWS Pricing](#), scroll to **Pricing for AWS products** and select **Storage**. Choose the storage product to open its pricing page.

Use Amazon EBS with Amazon EC2

Amazon Elastic Block Store (Amazon EBS) provides scalable, high-performance block storage resources that can be used with Amazon Elastic Compute Cloud (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 Amazon EBS volumes to an instance during launch, and you can create and attach EBS volumes to an instance at any time after launch. And you can create snapshots from a volume at any time after creation.

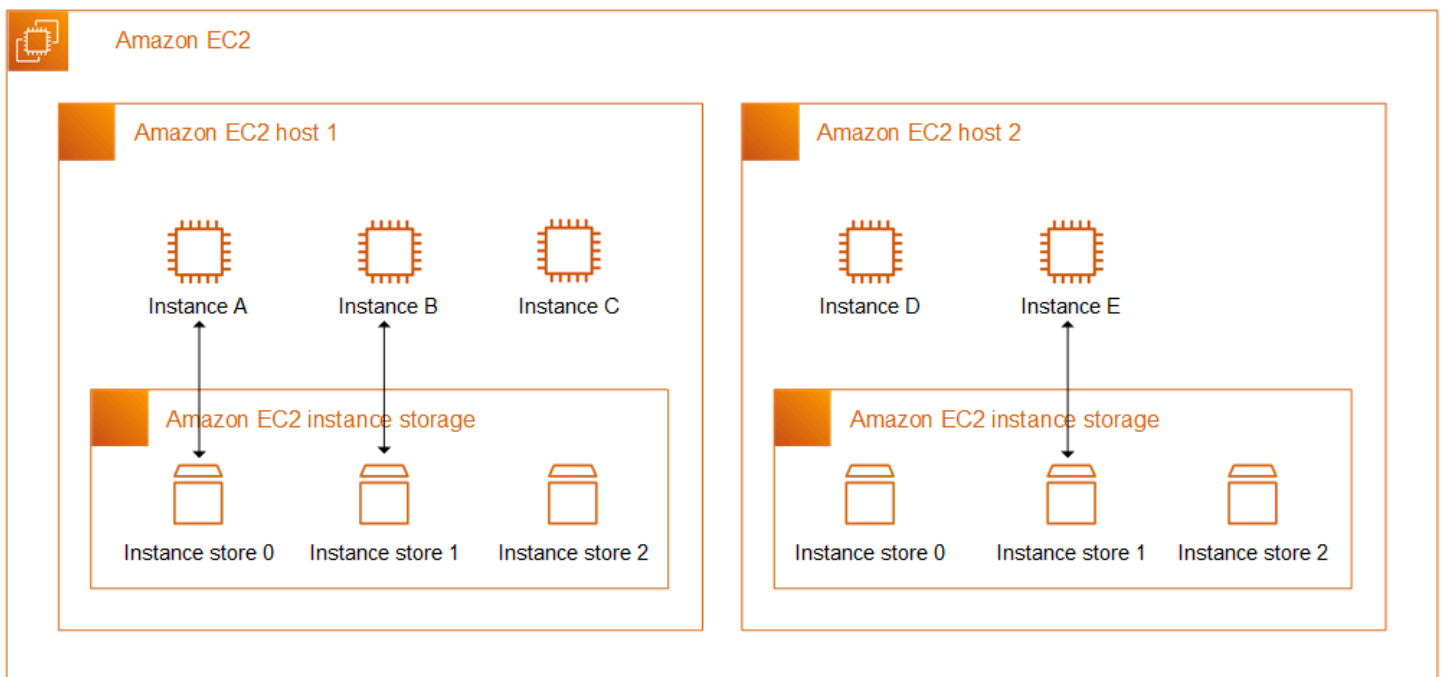
For more information about working with volumes and snapshots, see the [Amazon EBS User Guide](#).

Amazon EC2 instance store

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content. 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 more information, see [Instance store volumes](#).

The virtual devices for instance store volumes are `ephemeral[0-23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two or more instance store volumes have `ephemeral0`, `ephemeral1`, and so on.



Instance store pricing

Instance store volumes are included as part of the instance's usage cost.

Contents

- [Instance store volume and data lifetime](#)
- [Instance store volumes](#)

- [Add instance store volumes to your EC2 instance](#)
- [SSD instance store volumes](#)
- [Instance store swap volumes](#)
- [Optimize disk performance for instance store volumes](#)

Instance store volume and data lifetime

The number, size, and type of instance store volumes are determined by the instance type and instance size. For more information, see [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

Event[The instance is hibernated](#)[The instance is terminated](#)[The instance type is changed](#)[A Windows AMI is created from the instance](#)[An EBS-backed AMI is created from the instance](#)[An instance store-backed AMI is created from the instance](#)**User-initiated OS events**

A shutdown is initiated

A restart is initiated

AWS scheduled events[Instance stop](#)[Instance reboot](#)[System reboot](#)[Instance retirement](#)**Unplanned events**[Simplified automatic recovery](#)[CloudWatch action based recovery](#)

The underlying disk fails

Power failure

What happens to your data?

The data does not persist

The data does not persist

The data does not persist *

The data does not persist in the created AMI **

The data does not persist in the created AMI **

The data persists in the AMI bundle uploaded to Amazon S3 ***

The data does not persist †

The data persists

The data does not persist

The data persists

The data persists

The data does not persist

The data does not persist

The data does not persist

The data on the failed disk does not persist

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 volumes

The number, size, and type of instance store volumes are determined by the instance type and instance size. 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](#).

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](#)

To retrieve instance store volume information using the AWS CLI

You can use the [describe-instance-types](#) AWS CLI command to display information about an instance type, such as its instance store volumes. The following example displays the total size of instance storage for all R5 instances with instance store volumes.

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=r5*" "Name=instance-storage-
supported,Values=true" \
  --query "InstanceTypes[][InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
  --output table
```

Example output

```
-----
| DescribeInstanceTypes |
+-----+-----+
| r5ad.24xlarge | 3600 |
| r5ad.12xlarge | 1800 |
| r5dn.8xlarge  | 1200 |
| r5ad.8xlarge  | 1200 |
| r5ad.large    | 75   |
| r5d.4xlarge   | 600  |
| . . .        |      |
```

```
| r5dn.2xlarge | 300 |
| r5d.12xlarge | 1800 |
+-----+-----+
```

The following example displays the complete instance storage details for the specified instance type.

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=r5d.4xlarge" \
  --query "InstanceTypes[].InstanceStorageInfo"
```

The example output shows that this instance type has two 300 GB NVMe SSD volumes, for a total of 600 GB of instance storage.

```
[
  {
    "TotalSizeInGB": 600,
    "Disks": [
      {
        "SizeInGB": 300,
        "Count": 2,
        "Type": "ssd"
      }
    ],
    "NvmeSupport": "required"
  }
]
```

Add instance store volumes to your 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](#)

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 either an Amazon EBS volume or an instance store volume. The root volume is mounted automatically. For instances with an instance store volume for the root volume, the size of this volume varies by AMI, but the maximum size is 10 GB. For more information, see [Storage for the root device](#).

Contents

- [Add instance store volumes to an AMI](#)
- [Add non-NVME instance store volumes to an instance](#)
- [Make instance store volumes available on your instance](#)

Add instance store volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes.

If you launch an instance 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.

Considerations

- For M3 instances, specify instance store volumes in the block device mapping of the instance, not the AMI. Amazon EC2 might ignore instance store volume block device mappings in the AMI.
- When you launch an instance, you can omit non-NVMe instance store volumes specified in the AMI block device mapping or add instance store volumes.

New console

To add instance store volumes to an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Actions, Image and templates, Create image**.
4. On the **Create image** page, enter a meaningful name and description for your image.
5. For each instance store volume to add, choose **Add volume**, from **Volume type** select an instance store volume, and from **Device** select a device name. (For more information, see [Device names on Linux instances](#).) The number of available instance store volumes depends on the instance type. For instances with NVMe instance store volumes, the device mapping of these volumes depends on the order in which the operating system enumerates the volumes.
6. Choose **Create image**.

AWS CLI

To add instance store volumes to an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [create-image](#) or [register-image](#) (AWS CLI)
- [New-EC2Image](#) and [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

Add non-NVME instance store volumes to an instance

When you launch an instance that supports **non-NVMe instance store volumes** you must specify block device mappings for the instance store volumes to attach. 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.

Considerations

- For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.

To specify block device mappings in the instance launch request, use one of the following methods.

Amazon EC2 console

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. The number of available instance store volumes depends on the instance type.
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**.

Command line

You can use one of the following options commands with the corresponding option.

- `--block-device-mappings` with [run-instances](#) (AWS CLI)
- `-BlockDeviceMapping` with [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Make instance store volumes available on your instance

After you launch an instance with attached instance store volumes, you must mount the volumes before you can access them.

For Linux instances, many instance store volumes are pre-formatted with the ext3 file system. SSD-based instance store volumes that support TRIM instruction are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance. For more information, see [Instance store volume TRIM support](#). For Windows instances, we reformat the instance store volumes with the NTFS file system.

You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see [View the instance block device mapping for instance store volumes](#).

For Linux instances, you can view and mount the instance store volumes as described in the following procedure.

To make an instance store volume available on Linux

1. Connect to the instance using an SSH client. For more information, see [Connect to your Linux instance](#).
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
```

For instructions on how to mount an attached volume automatically after reboot, see [Automatically mount an attached volume after reboot](#) in the *Amazon EBS User Guide*.

SSD instance store volumes

To ensure the best IOPS performance from your SSD instance store volumes on Linux, we recommend that you use the most recent version of Amazon Linux, or another Linux AMI with a kernel version of 3.8 or later. If you do not use a Linux AMI with a kernel version of 3.8 or later, your instance won't achieve the maximum IOPS performance available for these instance types.

Like other instance store volumes, you must map the SSD instance store volumes for your instance when you launch it. The data on an SSD instance volume persists only for the life of its associated instance. For more information, see [Add instance store volumes to your EC2 instance](#).

NVMe SSD volumes

Some instances offer non-volatile memory express (NVMe) solid state drives (SSD) instance store volumes. For more information about the type of instance store volume supported by each instance type, see [Instance store volumes](#).

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.xlarge` 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/nvme/nvme_core.ko
```

The NVMe volumes are compliant with the NVMe 1.0e specification. You can use the NVMe commands with your NVMe volumes. With Amazon Linux, you can install the `nvme-cli` package from the repo using the **yum install** command. With other supported versions of Linux, you can download the `nvme-cli` package if it's not available in the image.

The data on NVMe instance storage is encrypted using an XTS-AES-256 block cipher implemented in a hardware module on the instance. The encryption keys are generated using the hardware module and are unique to each NVMe instance storage device. All encryption keys are destroyed when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

Non-NVMe SSD volumes

The following instances support instance store volumes that use non-NVMe SSDs to deliver high random I/O performance: C3, I2, M3, R3, and X1. For more information about the instance store volumes supported by each instance type, see [Instance store volumes](#).

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 volumes](#).

Instance store volumes that support TRIM are fully trimmed before they are allocated to your instance. These volumes are not formatted with a file system when an instance launches, so you must format them before they can be mounted and used. For faster access to these volumes, you should skip the TRIM operation when you format them.

With instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller when you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. On Linux, use the `fstrim` command to enable periodic TRIM.

Instance store swap volumes

Swap space in Linux can be used when a system requires more memory than it has been physically allocated. When swap space is enabled, Linux systems can swap infrequently used memory pages from physical memory to swap space (either a dedicated partition or a swap file in an existing file system) and free up that space for memory pages that require high-speed access.

Note

Using swap space for memory paging is not as fast or efficient as using RAM. If your workload is regularly paging memory into swap space, you should consider migrating to a larger instance type with more RAM. For more information, see [Change the instance type](#).

The `c1.medium` and `m1.small` instance types have a limited amount of physical memory to work with, and they are given a 900 MiB swap volume at launch time to act as virtual memory for Linux AMIs. Although the Linux kernel sees this swap space as a partition on the root device, it is actually a separate instance store volume, regardless of your root device type.

Amazon Linux automatically enables and uses this swap space, but your AMI may require some additional steps to recognize and use this swap space. To see if your instance is using swap space, you can use the **swapon -s** command.

```
[ec2-user ~]$ swapon -s
```

Filename	Type	Size	Used	Priority
/dev/xvda3	partition	917500	0	-1

The above instance has a 900 MiB swap volume attached and enabled. If you don't see a swap volume listed with this command, you may need to enable swap space for the device. Check your available disks using the **lsblk** command.

```
[ec2-user ~]$ lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
xvda1	202:1	0	8G	0	disk	/
xvda3	202:3	0	896M	0	disk	

Here, the swap volume `xvda3` is available to the instance, but it is not enabled (notice that the `MOUNTPOINT` field is empty). You can enable the swap volume with the **swapon** command.

Note

You must prepend `/dev/` to the device name listed by **lsblk**. Your device may be named differently, such as `sda3`, `sde3`, or `xvde3`. Use the device name for your system in the command below.

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

Now the swap space should show up in **lsblk** and **swapon -s** output.

```
[ec2-user ~]$ lsblk
NAME MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
xvda1 202:1   0    8G  0  disk /
xvda3 202:3   0   896M 0  disk [SWAP]
[ec2-user ~]$ swapon -s
Filename                                Type              Size      Used      Priority
/dev/xvda3                              partition        917500    0         -1
```

You also need to edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

Append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvda3    none    swap    sw    0    0
```

To use an instance store volume as swap space

Any instance store volume can be used as swap space. For example, the `m3.medium` instance type includes a 4 GB SSD instance store volume that is appropriate for swap space. If your instance store volume is much larger (for example, 350 GB), you may consider partitioning the volume with a smaller swap partition of 4-8 GB and the rest for a data volume.

Note

This procedure applies only to instance types that support instance storage. For a list of supported instance types, see [Instance store volumes](#).

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.

- (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
```

- 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
```

- Enable the new swap space.

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

- Verify that the new swap space is being used.

```
[ec2-user ~]$ swapon -s
Filename      Type  Size Used Priority
/dev/xvdb                                partition 4188668 0 -1
```

- 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.

Optimize disk performance for instance store volumes

Because of the way that Amazon EC2 virtualizes disks, the first write to any location on some instance store volumes performs more slowly than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you initialize your drives by writing once to every drive location before production use.

ℹ Note

Some instance types with direct-attached solid state drives (SSD) and TRIM support provide maximum performance at launch time, without initialization. For information about the instance store for each instance type, see [Instance store volumes](#).

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`).

ℹ Note

Make sure to unmount the drive before performing this command.
Initialization can take a long time (about 8 hours for an extra large instance).

To initialize the instance store volumes, use the following commands on the `m1.large`, `m1.xlarge`, `c1.xlarge`, `m2.xlarge`, `m2.2xlarge`, and `m2.4xlarge` instance types:

```
dd if=/dev/zero of=/dev/sdb bs=1M
```

```
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

To perform initialization on all instance store volumes at the same time, use the following command:

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

Configuring drives for RAID initializes them by writing to every drive location. When configuring software-based RAID, make sure to change the minimum reconstruction speed:

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

File storage

Cloud file storage is a method for storing data in the cloud that provides servers and applications access to data through shared file systems. This compatibility makes cloud file storage ideal for workloads that rely on shared file systems and provides simple integration without code changes.

There are many file storage solutions that exist, ranging from a single node file server on a compute instance using block storage as the underpinnings with no scalability or few redundancies to protect the data, to a do-it-yourself clustered solution, to a fully-managed solution. The following content introduces some of the storage services provided by AWS for use with Linux.

Contents

- [Use Amazon S3 with Amazon EC2](#)
- [Use Amazon EFS with Amazon EC2](#)
- [Use Amazon FSx with Amazon EC2](#)
- [Use Amazon File Cache with Amazon EC2](#)

Use Amazon S3 with Amazon EC2

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 `DOC-EXAMPLE-BUCKET1` bucket, then it is addressable using the URL `https://DOC-EXAMPLE-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. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

GET or wget

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 in 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://my_bucket.s3.amazonaws.com/path-to-file
```

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. The AWS CLI enables users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the [AWS Command Line Interface detail page](#).

The **aws s3 cp** command is similar to the Unix **cp** command. You can copy files from Amazon S3 to your instance, copy files from your instance to Amazon S3, and copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
[ec2-user ~]$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
[ec2-user ~]$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
[ec2-user ~]$ aws s3 sync s3://remote_S3_bucket local_directory
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the [Amazon Simple Storage Service User Guide](#). You can use this API and its examples to help develop your application and integrate it with other APIs and SDKs, such as the boto Python interface.

Use Amazon EFS with Amazon EC2

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](#).
- [Mount targets in each default subnet](#) 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 check box, 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 check box, 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 instance using the new launch instance wizard](#).

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](#).
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

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 instance using the new launch instance wizard](#)
- 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	<i>file system security group</i>
UDP	20001 - 20003	<i>file system security group</i>
UDP	4049	<i>file system security group</i>
UDP	2049	<i>file system security group</i>
UDP	635	<i>file system security group</i>
UDP	4045 - 4046	<i>file system security group</i>
TCP	4049	<i>file system security group</i>
TCP	635	<i>file system security group</i>
TCP	2049	<i>file system security group</i>
TCP	111	<i>file system security group</i>
TCP	4045 - 4046	<i>file system security group</i>
TCP	20001 - 20003	<i>file system security group</i>
All	All	<i>file system security group</i>

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	<i>instance security group</i>
UDP	20001 - 20003	<i>instance security group</i>
UDP	4049	<i>instance security group</i>
UDP	111	<i>instance security group</i>
UDP	635	<i>instance security group</i>
UDP	4045 - 4046	<i>instance security group</i>
TCP	4045 - 4046	<i>instance security group</i>
TCP	635	<i>instance security group</i>
TCP	2049	<i>instance security group</i>
TCP	4049	<i>instance security group</i>
TCP	20001 - 20003	<i>instance security group</i>
TCP	111	<i>instance security group</i>

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} ${fsx_mount_point_1} nfs4
nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
0 0\n" >> /etc/fstab; else printf "\n${svm_id_1}.${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:${vol_path_1} ${fsx_mount_point_1} nfs4
nfsvers=4.1,rsz=1048576,wsz=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 check box. 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 check box. 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

Amazon File Cache is a fully managed, high-speed cache on AWS that's used 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 [What is Amazon File Cache?](#).

You can access your cache from your Amazon EC2 instances using the open-source Lustre client. Amazon EC2 instances can access your cache from other Availability Zones within the same Amazon Virtual Private Cloud (Amazon VPC), provided that your networking allows access across subnets within the VPC. After your cache is mounted, you can work with its files and directories as you do when using a local file system.

To get started, see [Getting started with Amazon File Cache](#).

Instance volume limits

The maximum number of Amazon EBS volumes that you can attach to an instance depends on the instance type and instance size. 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.

Topics

- [Volume limits for instances built on the Nitro System](#)
- [Volume limits for Xen-based instances](#)

Volume limits for instances built on the Nitro System

Topics

- [Dedicated Amazon EBS volume limit](#)
- [Shared Amazon EBS volume limit](#)

Dedicated Amazon EBS volume limit

The following Nitro instance types have a dedicated Amazon 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 Amazon 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
- Compute optimized: C7a, C7i
- Memory optimized: R7a, R7i, R7iz

For these instance types that support **dedicated volume limits**, the volume limits depend on the instance size. The following table shows the limit for each instance size.

Instance size	Volume limit
medium large xlarge 2xlarge 4xlarge 8xlarge 12xlarge	32
16xlarge	48
24xlarge	64
32xlarge	88
48xlarge	128
metal-16x1 metal-24x 1	39

Instance size	Volume limit
metal-32x1 metal-48x1	79

Shared Amazon EBS volume limit

All other Nitro instance types (not listed in [Dedicated Amazon 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 of these instances support a maximum of 28 attachments. For example, if you have no additional network interface attachments on an `m5.xlarge` instance, you can attach up to 27 EBS volumes (*28 volume limit - 1 network interface*). If you have two additional network interfaces on an `m5.xlarge` instance, you can attach up to 25 EBS volumes (*28 volume limit - 3 network interfaces*). Similarly, if you have two additional network interfaces on an `m5d.xlarge` instance, which has 1 NVMe instance store volume, you can attach up to 24 EBS volumes (*28 volume limit - 3 network interfaces - 1 NVMe instance store volume*).

The following exceptions for instance types that have **shared volume limits**:

- DL2q instances support a maximum of 19 EBS volumes.
- Most bare metal instances support a maximum of 31 EBS volumes.
- High memory virtualized instances support a maximum of 27 EBS volumes.
- High memory bare metal instances support a maximum of 19 EBS volumes.
- `inf1.xlarge` and `inf1.2xlarge` instances support a maximum of 26 EBS volumes.
- `inf1.6xlarge` instances support a maximum of 23 EBS volumes.
- `mac1.metal` instances support a maximum of 16 EBS volumes.
- `mac2.metal`, `mac2-m2.metal`, and `mac2-m2pro.metal` instances support a maximum of 10 EBS volumes.
- `inf1.24xlarge` instances support a maximum of 11 EBS volumes.
- `g5.48xlarge` instances support a maximum of 9 EBS volumes.

- `d3.8xlarge` and `d3en.12xlarge` instances support a maximum of 3 EBS volumes.
- For accelerated computing instances, the attached accelerators count towards the shared volume limit. For example, for `p4d.24xlarge` instances, which have a shared volume limit of 28, 8 GPUs, and 8 NVMe instance store volumes, you can attach up to 11 Amazon EBS volumes (*28 volume limit - 1 network interface - 8 GPUs - 8 NVMe instance store volumes*).

Volume limits for Xen-based 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.

Amazon EC2 instance root volume

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 when you launch them or after they are running.

We reserve specific device names for root volumes. For more information, see [Device names on Linux instances](#).

Contents

- [Root volume type](#)
- [Choose an AMI by root volume type](#)
- [Determine the root device type of your instance](#)
- [Change the root volume to persist](#)

- [Change the initial size of the root volume](#)
- [Replace an EC2 instance root volume](#)

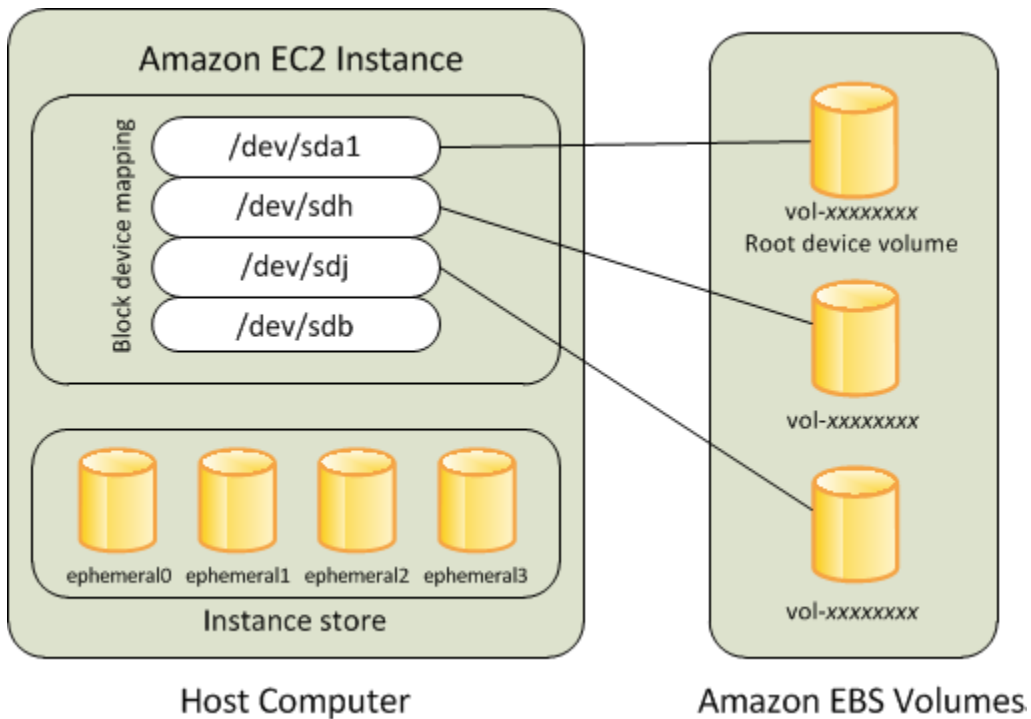
Root volume type

You can launch an instance from either an instance store-backed AMI or an Amazon EBS-backed AMI. The description of an AMI includes which type of AMI it is; you'll see the root device referred to in some places as either `ebs` (for Amazon EBS-backed) or `instance store` (for instance store-backed). This is important because there are significant differences between what you can do with each type of AMI. For more information about these differences, see [Storage for the root device](#). We recommend that you use AMIs backed by Amazon EBS, because these instances launch faster and use persistent storage.

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

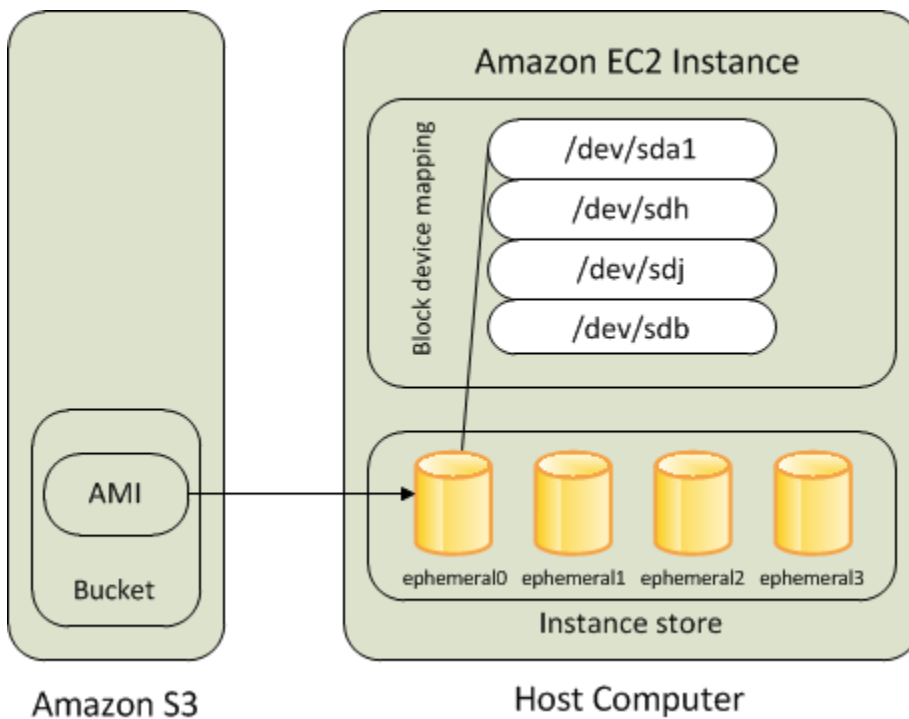
If an Amazon EBS-backed instance fails, you can restore your session by following one of these methods:

- 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 Linux 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

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 [Amazon EC2 instance store](#).



Requirement

Only the following instance types support an instance store volume as the root volume: C3, D2, I2, M3, and R3.

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.

Choose an AMI by root volume type

The AMI that you specify when you launch your instance determines the type of root device volume that your instance has. You can view AMIs by root device type using one of the following methods.

Console

To choose an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. From the filter lists, select the image type (such as **Public images**). In the search bar, choose **Platform** to select the operating system (such as **Amazon Linux**) and **Root device type** to select the root volume type (**ebs**).
4. (Optional) To get additional information to help you make your choice, choose the **Preferences** icon, toggle the columns to display, and choose **Confirm**.
5. Choose an AMI and write down its AMI ID.

To choose an instance store-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. From the filter lists, select the image type (such as **Public images**). In the search bar, choose **Platform** to select the operating system (such as **Amazon Linux**), and **Root device type** to select the root volume type (**instance-store**).
4. (Optional) To get additional information to help you make your choice, choose the **Preferences** icon, toggle the columns to display, and choose **Confirm**.
5. Choose an AMI and write down its AMI ID.

AWS CLI

To verify the type of the root device volume of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [describe-images](#) (AWS CLI)

- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

Determine the root device type of your instance

Console

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. On the **Storage** tab, under **Root device details**, check the value of **Root device type** as follows:
 - If the value is EBS, this is an Amazon EBS-backed instance.
 - If the value is INSTANCE-STORE, this is an instance store-backed instance.

AWS CLI

To determine the root device type of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

Change the root volume to persist

By default, the root volume for an AMI backed by Amazon EBS is deleted when the instance terminates. You can change the default behavior to ensure that the volume persists after the instance terminates. To change the default behavior, set the `DeleteOnTermination` attribute to `false` using a block device mapping.

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 using the Amazon EC2 console or the command line tools.

Console

To configure the root volume to persist when you launch an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then choose **Launch instances**.
3. Choose an Amazon Machine Image (AMI), choose an 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 using the AWS CLI

Use the [run-instances](#) command and include a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
$ aws ec2 run-instances --block-device-mappings file://mapping.json ...other
parameters...
```

Specify the following in `mapping.json`.

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

Tools for Windows PowerShell

To configure the root volume to persist when you launch an instance using the Tools for Windows PowerShell

Use the [New-EC2Instance](#) command and include a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
C:\> $ebs.DeleteOnTermination = $false
C:\> $bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
C:\> $bdm.DeviceName = "dev/xvda"
C:\> $bdm.Ebs = $ebs
C:\> New-EC2Instance -ImageId ami-0abcdef1234567890 -BlockDeviceMapping
    $bdm ...other parameters...
```

Configure the root volume to persist for an existing instance

You can configure the root volume to persist for a running instance using the command line tools only.

AWS CLI

To configure the root volume to persist for an existing instance using the AWS CLI

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
    }
  }
]
```


Tools for Windows PowerShell

To configure the root volume to persist for an existing instance using the AWS Tools for Windows PowerShell

Use the [Edit-EC2InstanceAttribute](#) command with a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsInstanceBlockDeviceSpecification
C:\> $ebs.DeleteOnTermination = $false
C:\> $bdm = New-Object Amazon.EC2.Model.InstanceBlockDeviceMappingSpecification
C:\> $bdm.DeviceName = "/dev/xvda"
C:\> $bdm.Ebs = $ebs
C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -BlockDeviceMapping
$bdm
```

Confirm that a root volume is configured to persist

You can confirm that a root volume is configured to persist using the Amazon EC2 console or the command line tools.

Console

To confirm that a root volume is configured to persist using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then select the instance.
3. In the **Storage** tab, under **Block devices**, locate the entry for the root volume. If **Delete on termination** is No, the volume is configured to persist.

AWS CLI

To confirm that a root volume is configured to persist using the AWS CLI

Use the [describe-instances](#) command and verify that the `DeleteOnTermination` attribute in the `BlockDeviceMappings` response element is set to `false`.

```
$ aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

```
...
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": false,
        "VolumeId": "vol-1234567890abcdef0",
        "AttachTime": "2013-07-19T02:42:39.000Z"
      }
    }
  ]
  ...
```

Tools for Windows PowerShell

To confirm that a root volume is configured to persist using the AWS Tools for Windows PowerShell

Use the [Get-EC2Instance](#) and verify that the `DeleteOnTermination` attribute in the `BlockDeviceMappings` response element is set to `false`.

```
C:\> (Get-EC2Instance -InstanceId i-
i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

Change the initial size of the root volume

By default, the size of the root volume is determined by the size of the snapshot. You can increase the initial size of the root volume using the block device mapping of the instance as follows.

1. Determine the device name of the root volume specified in the AMI, as described in [View the EBS volumes in an AMI block device mapping](#).
2. Confirm the size of the snapshot specified in the AMI block device mapping.
3. Override the size of the root volume using the instance block device mapping, as described in [Update the block device mapping when launching an instance](#), specifying a volume size that is larger than the snapshot size.

For example, the following entry for the instance block device mapping increases the size of the root volume, `/dev/xvda`, to 100 GiB. You can omit the snapshot ID in the instance block device mapping because the snapshot ID is already specified in the AMI block device mapping.

```
{
  "DeviceName": "/dev/xvda",
  "Ebs": {
    "VolumeSize": 100
  }
}
```

For more information, see [Block device mappings](#).

Replace an EC2 instance root volume

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.

Topics

- [How does it work?](#)
- [Replace a root volume](#)
- [View root volume replacement tasks](#)

How does it work?

When you replace the root volume for an instance, a new (replacement) root volume is restored in one of the following ways:

- **To the initial launch state** — the volume is restored to its initial state at instance launch. For more information, see [Restore a root volume to its launch state](#).
- **From a snapshot** from the same lineage as the current root volume — this allows you to fix issues, such as root volume corruption or guest operating system network configuration errors. For more information, see [Replace a root volume using a snapshot](#).
- **From an AMI** that has the same key attributes as the instance — this allows you to perform operating system and application patching or upgrades. For more information, see [Replace a root volume using an AMI](#).

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. You can choose whether or not 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 it when you no longer need it.

If the root volume replacement task fails, the instance is rebooted and the original root volume remains attached to the instance.

Considerations for root volume replacement

- 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. All other bare metal instance types are not supported.
- You can use any snapshot that belongs to the same lineage as any of the instance's 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.
- 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

	Original root volume	Specified snapshot or AMI	Encryption by default	Replacement root volume	Encryption key used for replacement root volume
	Unencrypted	Unencrypted	Enabled	Encrypted	Account's default KMS key for Amazon EBS encryption
	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.

Topics

- [Restore a root volume to its launch state](#)
- [Replace a root volume using a snapshot](#)
- [Replace a root volume using an AMI](#)

Restore a root volume to its launch state

You can perform a root volume replacement that replaces an instance's root volume with a replacement root volume that is restored to the original root volume's launch state. The replacement volume is automatically restored from the snapshot that was used to create the original volume during the instance launch.

The replacement root volume gets the same type, size, and delete on termination attributes as the original root volume.

Replace a root volume using a snapshot

You can perform a root volume replacement that replaces an instance's root volume with a replacement volume that is restored to a specific snapshot. This enables you to restore the root volume for an instance to a specific snapshot that you previously created from that root volume.

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 belong to the same lineage as the instance's current root volume.
- You can't use snapshot copies created from snapshots that were taken from the root volume.
- After successfully replacing the root volume, snapshots taken from the original root volume can still be used to replace the new (replacement) root volume.

Replace a root volume using an AMI


You can perform a root volume replacement using an AMI that you own or an AMI that is shared with you. The AMI must have the same product code, billing information, architecture type, and virtualization type as that of 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.

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. For more information, see [View root volume replacement tasks](#).

You can replace the root volume for an instance using one of the following methods.

Note

If you use the Amazon EC2 console, the functionality is available in the new console only.

New 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. In the **Replace root volume** screen, do one of the following:
 - To restore the replacement root volume to its initial launch state, choose **Create replacement task** without selecting a snapshot.
 - To restore the replacement root volume to a specific snapshot, for **Snapshot**, select the snapshot to use, and then choose **Create replacement task**.
 - To restore the replacement root volume using an AMI, for **AMI**, select the AMI to use, and then choose **Create replacement task**.
5. To delete the original root volume after the replacement task completes, select **Delete replaced root volume**.

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`.

```
$ aws ec2 create-replace-root-volume-task \  
--instance-id i-1234567890abcdef0 \  
--delete-replaced-root-volume true
```

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`.

```
$ aws ec2 create-replace-root-volume-task \  
--instance-id i-1234567890abcdef0 \  
--snapshot-id s-1234567890abcdef0 \  
--delete-replaced-root-volume true
```

```
--snapshot-id snap-9876543210abcdef0 \  
--delete-replaced-root-volume true
```

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`.

```
$ aws ec2 create-replace-root-volume-task \  
--instance-id i-01234567890abcdef \  
--image-id ami-09876543210abcdef \  
--delete-replaced-root-volume true
```

Tools for Windows 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`.

```
PS C:\> New-EC2ReplaceRootVolumeTask -InstanceId i-1234567890abcdef0 -  
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`.

```
PS C:\> New-EC2ReplaceRootVolumeTask -InstanceId i-1234567890abcdef0 -  
SnapshotId snap-9876543210abcdef0 -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`.

```
PS C:\> New-EC2ReplaceRootVolumeTask -InstanceId i-1234567890abcdef0 -  
ImageId ami-09876543210abcdef -DeleteReplacedRootVolume $true
```

View root volume replacement tasks

When you replace the root volume for an instance, a *root volume replacement task* is created. The root volume replacement task transitions through the following states during the process:

- `pending` — the replacement volume is being created.
- `in-progress` — the original volume is being detached and the replacement volume is being attached.
- `succeeded` — the replacement volume has been successfully attached to the instance and the instance is available.
- `failing` — the replacement task is in the process of failing.
- `failed` — the replacement task has failed, but the original root volume is still attached.
- `failing-detached` — the replacement task is in the process of failing 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.

You can view the root volume replacement tasks for an instance using one of the following methods.

Note

If you use the Amazon EC2 console, the functionality is available in the new console only.

Console

To view the root volume replacement tasks

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Select the instance for which to view the root volume replacement tasks, and then choose the **Storage** tab.
4. In the **Storage** tab, expand **Recent root volume replacement tasks**.

AWS CLI

To view the status of a root volume replacement task

Use the [describe-replace-root-volume-tasks](#) command and specify the IDs of the root volume replacement tasks to view.

```
$ aws ec2 describe-replace-root-volume-tasks \  
--replace-root-volume-task-ids replacevol-1234567890abcdef0
```

```
{  
  "ReplaceRootVolumeTasks": [  
    {  
      "ReplaceRootVolumeTaskId": "replacevol-1234567890abcdef0",  
      "InstanceId": "i-1234567890abcdef0",  
      "TaskState": "succeeded",  
      "StartTime": "2020-11-06 13:09:54.0",  
      "CompleteTime": "2020-11-06 13:10:14.0",  
      "SnapshotId": "snap-01234567890abcdef",  
      "DeleteReplacedRootVolume": "True"  
    }  
  ]  
}
```

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
```

Tools for Windows PowerShell

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.

```
PS C:\> Get-EC2ReplaceRootVolumeTask -  
ReplaceRootVolumeTaskIds replacevol-1234567890abcdef0
```

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 on Linux instances

When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

The number of volumes that your instance can support is determined by the operating system. For more information, see [Instance volume limits](#).

Contents

- [Available device names](#)
- [Device name considerations](#)

For information about device names on Windows instances, see [Device naming on Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Available device names

There are two types of virtualization available for Linux instances: paravirtual (PV) and hardware virtual machine (HVM). The virtualization type of an instance is determined by the AMI used to launch the instance. All instance types support HVM AMIs. Some previous generation instance types support PV AMIs. Be sure to note the virtualization type of your AMI because the recommended and available device names that you can use depend on the virtualization type of your instance. For more information, see [Linux AMI virtualization types](#).

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 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-d][a-z] /dev/xvd[e-z]	Differs by AMI /dev/sda1 or /dev/xvda	/dev/sd[f-p] *	/dev/sd[b-e] /dev/sd[b-h] (h1.16xlarge) /dev/sd[b-y] (d2.8xlarge) /dev/sd[b-i] (i2.8xlarge) **

* The device names that you specify for NVMe EBS volumes in a block device mapping are renamed using NVMe device names (`/dev/nvme[0-26]n1`). The block device driver can assign NVMe device names in a different order than you specified for the volumes in the block device mapping.

** NVMe instance store volumes are automatically enumerated and assigned an NVMe device name.

For more information about instance store volumes, see [Amazon EC2 instance store](#). 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:

- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- The number of NVMe instance store volumes for an instance depends on the size of the instance. NVMe instance store volumes are automatically enumerated and assigned an NVMe device name (`/dev/nvme[0-26]n1`).
- Depending on the block device driver of the kernel, the device could be attached with a different name than you specified. For example, if you specify a device name of `/dev/sdh`, your device could be renamed `/dev/xvdh` or `/dev/hdh`. In most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), the trailing letter could change (`/dev/sda` could become `/dev/xvde`). In these cases, the trailing letter of each device name is incremented the same number of times. For example, if `/dev/sdb` is renamed `/dev/xvdf`, then `/dev/sdc` is renamed `/dev/xvdg`. Amazon Linux creates a symbolic link for the name you specified to the renamed device. Other operating systems could behave differently.
- HVM AMIs do not support the use of trailing numbers on device names, except for `/dev/sda1`, which is reserved for the root device, and `/dev/sda2`. While using `/dev/sda2` is possible, we do not recommend using this device mapping with HVM instances.
- When using PV AMIs, you cannot attach volumes that share the same device letters both with and without trailing digits. For example, if you attach a volume as `/dev/sdc` and another volume as `/dev/sdc1`, only `/dev/sdc` is visible to the instance. To use trailing digits in device names, you must use trailing digits on all device names that share the same base letters (such as `/dev/sdc1`, `/dev/sdc2`, `/dev/sdc3`).
- Some custom kernels might have restrictions that limit use to `/dev/sd[f-p]` or `/dev/sd[f-p][1-6]`. If you're having trouble using `/dev/sd[q-z]` or `/dev/sd[q-z][1-6]`, try switching to `/dev/sd[f-p]` or `/dev/sd[f-p][1-6]`.

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. For example, on Linux, you can use the **lsblk** command to view the disk devices and their mount points. On Windows you can use the Disk Management utility or the **diskpart** command.

Block device mappings

Each instance that you launch has an associated root device volume, which is either an Amazon EBS volume or an instance store volume. You can use block device mapping to specify additional

EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance. However, the only way to attach instance store volumes to an instance is to use block device mapping to attach the volumes as the instance is launched.

For more information about root device volumes, see [Change the root volume to persist](#).

Contents

- [Block device mapping concepts](#)
- [AMI block device mapping](#)
- [Instance block device mapping](#)

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](#)
- [Block device mapping instance store caveats](#)

- [Example block device mapping](#)
- [How devices are made available in the operating system](#)

Block device mapping entries

When you create a block device mapping, you specify the following information for each block device that you need to attach to the instance:

- The device name used within Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume. The name assigned can be different from the name that Amazon EC2 recommends. For more information, see [Device names on Linux 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-xxxxxxx`). 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.

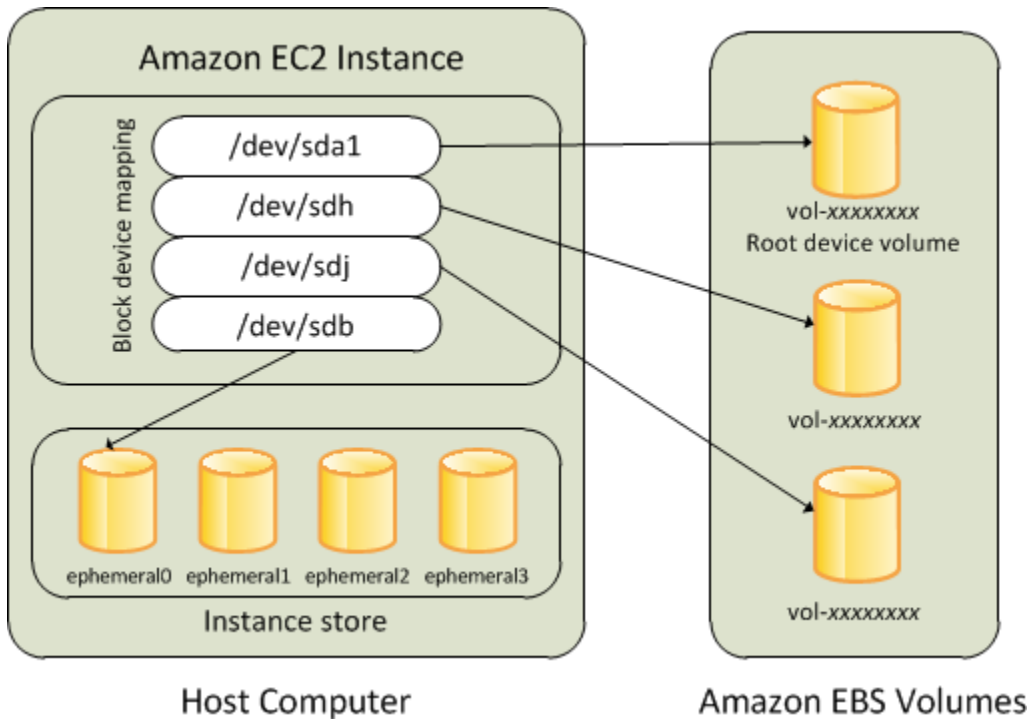
Note

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.

With a Linux instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots. The instance type determines which instance store volumes are formatted and mounted by default. You can mount additional instance store volumes at launch, as long as you don't exceed the number of instance store volumes

available for your instance type. For more information, see [Amazon EC2 instance store](#). The block device driver for the instance determines which devices are used when the volumes are formatted and mounted.

AMI block device mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. To add more block devices to an AMI, you must create your own AMI.

Contents

- [Specify a block device mapping for an AMI](#)
- [View the EBS volumes in an AMI block device mapping](#)

Specify a block device mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For EBS volumes, the existing data is saved to a new snapshot, and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an EBS-backed AMI, you can add EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add instance store volumes only by modifying the block device mapping entries in the image manifest file when registering the image.

Note

For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select an instance and choose **Actions, Image and templates, Create image**.

4. Enter a name and a description for the image.
5. The instance volumes appear under **Instance volumes**. To add another volume, choose **Add volume**.
6. For **Volume type**, choose the volume type. For **Device** choose the device name. For an EBS volume, you can specify additional details, such as a snapshot, volume size, volume type, IOPS, and encryption state.
7. Choose **Create image**.

To add volumes to an AMI using the command line

Use the [create-image](#) AWS CLI command to specify a block device mapping for an EBS-backed AMI. Use the [register-image](#) AWS CLI command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the `--block-device-mappings` parameter. Arguments encoded in JSON can be supplied either directly on the command line or by reference to a file:

```
--block-device-mappings [mapping, ...]  
--block-device-mappings [file://mapping.json]
```

To add an instance store volume, use the following mapping.

```
{  
  "DeviceName": "/dev/sdf",  
  "VirtualName": "ephemeral0"  
}
```

To add an empty 100 GiB gp2 volume, use the following mapping.

```
{  
  "DeviceName": "/dev/sdg",  
  "Ebs": {  
    "VolumeSize": 100  
  }  
}
```

To add an EBS volume based on a snapshot, use the following mapping.

```
{
```

```
"DeviceName": "/dev/sdh",
"Ebs": {
  "SnapshotId": "snap-xxxxxxxx"
}
}
```

To omit a mapping for a device, use the following mapping.

```
{
  "DeviceName": "/dev/sdj",
  "NoDevice": ""
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the following commands (AWS Tools for Windows PowerShell):

- [New-EC2Image](#)
- [Register-EC2Image](#)

View the EBS volumes in an AMI block device mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. Choose **EBS images** from the **Filter** list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:
 - **Root Device Type** (ebs)
 - **Root Device Name** (for example, `/dev/sda1`)
 - **Block Devices** (for example, `/dev/sda1=snap-1234567890abcdef0:8:true`)

If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (This screen doesn't display instance store volumes.)

To view the EBS volumes for an AMI using the command line

Use the [describe-images](#) (AWS CLI) command or [Get-EC2Image](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an AMI.

Instance block device mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI.

Limitations

- For the root volume, you can only modify the following: volume size, volume type, and the **Delete on Termination** flag.
- When you modify an EBS volume, you can't decrease its size. Therefore, you must specify a snapshot whose size is equal to or greater than the size of the snapshot specified in the block device mapping of the AMI.

Contents

- [Update the block device mapping when launching an instance](#)
- [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.

To add volumes to an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch Instance**.

3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, you can modify the root volume, EBS volumes, and instance store volumes as follows:
 - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
 - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
 - To add an EBS volume, choose **Add New Volume**, choose **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
 - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and choose its **Delete** icon.
 - To add an instance store volume, choose **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.
6. Complete the remaining wizard pages, and choose **Launch**.

To add volumes to an instance using the AWS CLI

Use the [run-instances](#) AWS CLI command with the `--block-device-mappings` option to specify a block device mapping for an instance at launch.

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- `/dev/sdb=ephemeral0`
- `/dev/sdh=snap-1234567890abcdef0`
- `/dev/sdj=:100`

To prevent `/dev/sdj` from attaching to an instance launched from this AMI, use the following mapping.

```
{
  "DeviceName": "/dev/sdj",
  "NoDevice": ""
}
```

```
}
```

To increase the size of `/dev/sdh` to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for `/dev/sdh`, because specifying the device name is enough to identify the volume.

```
{
  "DeviceName": "/dev/sdh",
  "Ebs": {
    "VolumeSize": 300
  }
}
```

To increase the size of the root volume at instance launch, first call [describe-images](#) with the ID of the AMI to verify the device name of the root volume. For example, `"RootDeviceName": "/dev/xvda"`. To override the size of the root volume, specify the device name of the root device used by the AMI and the new volume size.

```
{
  "DeviceName": "/dev/xvda",
  "Ebs": {
    "VolumeSize": 100
  }
}
```

To attach an additional instance store volume, `/dev/sdc`, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect. If the instance supports NVMe instance store volumes, they are automatically enumerated and assigned an NVMe device name.

```
{
  "DeviceName": "/dev/sdc",
  "VirtualName": "ephemeral1"
}
```

To add volumes to an instance using the AWS Tools for Windows PowerShell

Use the `-BlockDeviceMapping` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

Update the block device mapping of a running instance

You can use the [modify-instance-attribute](#) AWS CLI command to update the block device mapping of a running instance. You do not need to stop the instance before changing this attribute.

```
aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings
file://mapping.json
```

For example, to preserve the root volume at instance termination, specify the following in `mapping.json`.

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the [Edit-EC2InstanceAttribute](#) command (AWS Tools for Windows PowerShell).

View the EBS volumes in an instance block device mapping

You can easily enumerate the EBS volumes mapped to an instance.

Note

For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

To view the EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. In the search box, enter **Root device type**, and then choose **EBS**. This displays a list of EBS-backed instances.

4. Select the desired instance and look at the details displayed in the **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.

5. To display additional information about an EBS volume, choose its volume ID to go to the volume page.

To view the EBS volumes for an instance using the command line

Use the [describe-instances](#) (AWS CLI) command or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an instance.

View the instance block device mapping for instance store volumes

When you view the block device mapping for your instance, you can see only the EBS volumes, not the instance store volumes. The method you use to view the instance store volumes for your instance depends on the volume type.

NVMe instance store volumes

You can use the NVMe command line package, [nvme-cli](#), to query the NVMe instance store volumes in the block device mapping. Download and install the package on your instance, and then run the following command.

```
[ec2-user ~]$ sudo nvme list
```

The following is example output for an instance. The text in the Model column indicates whether the volume is an EBS volume or an instance store volume. In this example, both `/dev/nvme1n1` and `/dev/nvme2n1` are instance store volumes.

Node	SN	Model
Namespace		
-----	-----	-----

/dev/nvme0n1	vol106afc3f8715b7a597	Amazon Elastic Block Store	1
/dev/nvme1n1	AWS2C1436F5159EB6614	Amazon EC2 NVMe Instance Storage	1
/dev/nvme2n1	AWSB1F4FF0C0A6C281EA	Amazon EC2 NVMe Instance Storage	1
...			

HDD or SSD instance store volumes

You can use instance metadata to query the HDD or SSD instance store volumes in the block device mapping. NVMe instance store volumes are not included.

The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`. For more information, see [Instance metadata and user data](#).

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
```

The instance type determines the number of instance store volumes that are available to the instance. If the number of instance store volumes in a block device mapping exceeds the number of instance store volumes available to an instance, the additional volumes are ignored. To view the instance store volumes for your instance, run the **lsblk** command. To learn how many instance store volumes are supported by each instance type, see [Instance store volumes](#).

Torn write prevention

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.

Supported block sizes and block boundary alignments

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, I4gn, and I4gen instances supported by AWS Nitro SSD.	
Amazon EBS volumes	All Amazon EBS volumes attached to instances built on the AWS Nitro System .		

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 torn write prevention support and configuration](#).

Requirements

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 torn write prevention support and configuration

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 <code>/dev/xvda</code> . 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(<code>nvmeXn1</code>).
32:63	The volume ID. For example, <code>vol01234567890abcdef</code> . 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 guaranteed 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 $NTWPG * n \leq NTWPU$ where n is positive integer. The write operation LBA offset 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

Bytes	Description
	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 software stack for torn write prevention

Torn write prevention is enabled by default on [supported instance types with supported volumes](#). You do not need to enable 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 `0_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=0_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](#).

Resources and tags

Amazon EC2 provides different *resources* that you can create and use. Some of these resources include images, instances, volumes, and snapshots. When you create a resource, we assign the resource a unique resource ID.

Some resources can be tagged with values that you define, to help you organize and identify them.

The following topics describe resources and tags, and how you can work with them.

Contents

- [Recycle Bin](#)
- [Resource locations](#)
- [Resource IDs](#)
- [List and filter your resources](#)
- [Amazon EC2 Global View](#)
- [Tag your Amazon EC2 resources](#)
- [Amazon EC2 service quotas](#)
- [Amazon EC2 usage reports](#)

Recycle Bin

Recycle Bin is a data recovery feature that enables you to restore accidentally deleted Amazon EBS snapshots and EBS-backed AMIs. When using Recycle Bin, if your resources are deleted, they are retained in the Recycle Bin for a time period that you specify before being permanently deleted.

You can restore a resource from the Recycle Bin at any time before its retention period expires. After you restore a resource from the Recycle Bin, the resource is removed from the Recycle Bin and you can use it in the same way that you use any other resource of that type in your account. If the retention period expires and the resource is not restored, the resource is permanently deleted from the Recycle Bin and it is no longer available for recovery.

Using Recycle Bin helps to ensure business continuity by protecting your business-critical data against accidental deletion.

Topics

- [How does it work?](#)
- [Supported resources](#)
- [Considerations](#)
- [Quotas](#)
- [Related services](#)
- [Pricing](#)
- [Required IAM permissions](#)
- [Work with retention rules](#)
- [Work with resources in the Recycle Bin](#)
- [Monitor Recycle Bin](#)

How does it work?

To enable and use Recycle Bin, you must create *retention rules* in the AWS Regions in which you want to protect your resources. Retention rules specify the following:

- The resource type that you want to protect.
- The resources that you want to retain in the Recycle Bin when they are deleted.
- The retention period for which to retain resources in the Recycle Bin before they are permanently deleted.

With Recycle Bin, you can create two types of retention rules:

- **Tag-level retention rules** — A tag-level retention rule uses resource tags to identify the resources that are to be retained in the Recycle Bin. For each retention rule, you specify one or more tag key and value pairs. Resources of the specified type that are tagged with at least one of the tag key and value pairs that are specified in the retention rule are automatically retained in the Recycle Bin upon deletion. Use this type of retention rule if you want to protect specific resources in your account based on their tags.
- **Region-level retention rules** — A Region-level retention rule does not have any resource tags specified. It applies to all of the resources of the specified type in the Region in which the rule is created, even if the resources are not tagged. Use this type of retention rule if you want to protect all resources of a specific type in a specific Region.

While a resource is in the Recycle Bin, you have the ability to restore it for use at any time.

The resource remains in the Recycle Bin until one of the following happens:

- You manually restore it for use. When you restore a resource from the Recycle Bin, the resource is removed from the Recycle Bin and it immediately becomes available for use. You can use restored resources in the same way as any other resource of that type in your account.
- The retention period expires. If the retention period expires, and the resource has not been restored from the Recycle Bin, the resource is permanently deleted from the Recycle Bin and it can no longer be viewed or restored.

Supported resources

Recycle Bin supports the following resource types:

- Amazon EBS snapshots

Important

Recycle Bin retention rules also apply to archived snapshots in the archive storage tier. If you delete an archived snapshot that matches a retention rule, that snapshot is retained in the Recycle Bin for the period defined in the retention rule. Archived snapshots are billed at the rate for archived snapshots while they are in the Recycle Bin.

- Amazon EBS-backed Amazon Machine Images (AMIs)

Note

Retention rules also apply to disabled AMIs.

Considerations

The following considerations apply when working with Recycle Bin and retention rules.

General considerations

- **⚠ Important**

When you create your first retention rule, it can take up to 30 minutes for the rule to become active and for it to start retaining resources. After you create the first retention rule, subsequent retention rules become active and start retaining resources almost immediately.
- If a resource matches more than one retention rule upon deletion, then the retention rule with the longest retention period takes precedence.
- You can't manually delete a resource from the Recycle Bin. The resource will be automatically deleted when its retention period expires.
- While a resource is in the Recycle Bin, you can only view it, restore it, or modify its tags. To use the resource in any other way, you must first restore it.
- If any AWS service, such as AWS Backup or Amazon Data Lifecycle Manager, deletes a resource that matches a retention rule, that resource is automatically retained by Recycle Bin.
- When a resource is sent to the Recycle Bin, the following system-generated tag is assigned to the resource:
 - Tag key — `aws:recycle-bin:resource-in-bin`
 - Tag value — `true`

You can't manually edit or delete this tag. When the resource is restored from the Recycle Bin, the tag is automatically removed.

Considerations for snapshots

- **⚠ Important**

If you have retention rules for AMIs and for their associated snapshots, make the retention period for the snapshots the same or longer than the retention period for the AMIs. This ensures that Recycle Bin does not delete the snapshots associated with an AMI before deleting the AMI itself, as this would make the AMI unrecoverable.
- If a snapshot is enabled for fast snapshot restore when it is deleted, fast snapshot restore is automatically disabled shortly after the snapshot is sent to the Recycle Bin.

- If you restore the snapshot before fast snapshot restore is disabled for the snapshot, it remains enabled.
- If you restore the snapshot, after fast snapshot restore has been disabled, it remains disabled. If needed, you must manually re-enable fast snapshot restore.
- If a snapshot is shared when it is deleted, it is automatically unshared when it is sent to the Recycle Bin. If you restore the snapshot, all of the previous sharing permissions are automatically restored.
- If a snapshot that was created by another AWS service, such as AWS Backup is sent to the Recycle Bin and you later restore that snapshot from the Recycle Bin, it is no longer managed by the AWS service that created it. You must manually delete the snapshot if it is no longer needed.

Considerations for AMIs

- Only Amazon EBS-backed AMIs are supported.

• **⚠ Important**

If you have retention rules for AMIs and for their associated snapshots, make the retention period for the snapshots the same or longer than the retention period for the AMIs. This ensures that Recycle Bin does not delete the snapshots associated with an AMI before deleting the AMI itself, as this would make the AMI unrecoverable.

- If an AMI is shared when it is deleted, it is automatically unshared when it is sent to the Recycle Bin. If you restore the AMI, all of the previous sharing permissions are automatically restored.
- Before you can restore an AMI from the Recycle Bin, you must first restore all of its associated snapshots from the Recycle Bin and ensure that they are in the available state.
- If the snapshots that are associated with the AMI are deleted from the Recycle Bin, the AMI is no longer recoverable. The AMI will be deleted when the retention period expires.
- If an AMI that was created by another AWS service, such as AWS Backup, is sent to the Recycle Bin and you later restore that AMI from the Recycle Bin, it is no longer managed by the AWS service that created it. You must manually delete the AMI if it is no longer needed.

Considerations for Amazon Data Lifecycle Manager snapshot policies

- If Amazon Data Lifecycle Manager deletes a snapshot that matches a retention rule, that snapshot is automatically retained by Recycle Bin.

- If Amazon Data Lifecycle Manager deletes a snapshot and sends it to the Recycle Bin when the policy's retention threshold is reached, and you manually restore the snapshot from the Recycle Bin, you must manually delete that snapshot when it is no longer needed. Amazon Data Lifecycle Manager will no longer manage the snapshot.
- If you manually delete a snapshot that was created by a policy, and that snapshot is in the Recycle Bin when the policy's retention threshold is reached, Amazon Data Lifecycle Manager will not delete the snapshot. Amazon Data Lifecycle Manager does not manage the snapshots while they are stored in the Recycle Bin.

If the snapshot is restored from the Recycle Bin before the policy's retention threshold is reached, Amazon Data Lifecycle Manager will delete the snapshot when the policy's retention threshold is reached.

If the snapshot is restored from the Recycle Bin after the policy's retention threshold is reached, Amazon Data Lifecycle Manager will no longer delete the snapshot. You must manually delete the snapshot when it is no longer needed.

Considerations for AWS Backup

- If AWS Backup deletes a snapshot that matches a retention rule, that snapshot is automatically retained by Recycle Bin.

Considerations for archived snapshots

- Recycle Bin retention rules also apply to archived snapshots in the archive storage tier. If you delete an archived snapshot that matches a retention rule, that snapshot is retained in the Recycle Bin for the period defined in the retention rule.

Archived snapshots are billed at the rate for archived snapshots while they are in the Recycle Bin.

If a retention rule deletes an archived snapshot from the Recycle Bin before the minimum archive period of 90 days, you are billed for the remaining days. For more information, see [Archived snapshot pricing and billing](#) in the *Amazon EBS User Guide*.

To use an archived snapshot that is in the Recycle Bin, you must first recover the snapshot from the Recycle Bin and then restore it from the archive tier to the standard tier.

Quotas

The following quotas apply to Recycle Bin.

Quota	Default quota			
Retention rules per Region	250			
Tag key and value pairs per retention rule	50			

Related services

Recycle Bin works with the following services:

- **AWS CloudTrail** — Enables you to record events that occur in Recycle Bin. For more information, see [Monitor Recycle Bin using AWS CloudTrail](#).

Pricing

Resources in the Recycle Bin are billed at their standard rates. There are no additional charges for using Recycle Bin and retention rules. For more information, see [Amazon EBS pricing](#).

Note

Some resources might still appear in the Recycle Bin console or in the AWS CLI and API output for a short period after their retention periods have expired and they have been permanently deleted. You are not billed for these resources. Billing stops as soon as the retention period expires.

You can use the following AWS generated cost allocation tags for cost tracking and allocation purposes when using AWS Billing and Cost Management.

- Key: `aws:recycle-bin:resource-in-bin`
- Value: `true`

For more information, see [AWS-generated cost allocation tags](#) in the *AWS Billing and Cost Management User Guide*.

Required IAM permissions

By default, users don't have permission to work with Recycle Bin, retention rules, or with resources that are in the Recycle Bin. To allow users to work with these resources, you must create IAM policies that grant permission to use specific resources and API actions. Once the policies are created, you must add permissions to your users, groups, or roles.

Topics

- [Permissions for working with Recycle Bin and retention rules](#)
- [Permissions for working with resources in the Recycle Bin](#)
- [Condition keys for Recycle Bin](#)

Permissions for working with Recycle Bin and retention rules

To work with Recycle Bin and retention rules, users need the following permissions.

- `rbin:CreateRule`
- `rbin:UpdateRule`
- `rbin:GetRule`
- `rbin:ListRules`
- `rbin>DeleteRule`
- `rbin:TagResource`
- `rbin:UntagResource`
- `rbin:ListTagsForResource`
- `rbin:LockRule`
- `rbin:UnlockRule`

To use the Recycle Bin console, users need the `tag:GetResources` permission.

The following is an example IAM policy that includes the `tag:GetResources` permission for console users. If some permissions are not needed, you can remove them from the policy.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "rbin:CreateRule",
      "rbin:UpdateRule",
      "rbin:GetRule",
      "rbin:ListRules",
      "rbin>DeleteRule",
      "rbin:TagResource",
      "rbin:UntagResource",
      "rbin:ListTagsForResource",
      "rbin:LockRule",
      "rbin:UnlockRule",
      "tag:GetResources"
    ],
    "Resource": "*"
  }]
}
```

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 [Creating 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 [Creating 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*.

Permissions for working with resources in the Recycle Bin

For more information about the IAM permissions needed to work with resources in the Recycle Bin, see the following:

- [Permissions for working with snapshots in the Recycle Bin](#)
- [Permissions for working with AMIs in the Recycle Bin](#)

Condition keys for Recycle Bin

Recycle Bin defines the following condition keys that you can use in the `Condition` element of an IAM policy to control the conditions under which the policy statement applies. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.

Topics

- [rbin:Request/ResourceType condition key](#)
- [rbin:Attribute/ResourceType condition key](#)

rbin:Request/ResourceType condition key

The `rbin:Request/ResourceType` condition key can be used to filter access on [CreateRule](#) and [ListRules](#) requests based on the value specified for the `ResourceType` request parameter.

Example 1 - CreateRule

The following sample IAM policy allows IAM principals to make **CreateRule** requests only if the value specified for the `ResourceType` request parameter is `EBS_SNAPSHOT` or `EC2_IMAGE`. This allows the principal to create new retention rules for snapshots and AMIs only.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "rbin:CreateRule"
      ],
      "Resource" : "*"
    }
  ]
}
```

```

        "Condition" : {
            "StringEquals" : {
                "rbin:Request/ResourceType" : ["EBS_SNAPSHOT", "EC2_IMAGE"]
            }
        }
    ]
}

```

Example 2 - ListRules

The following sample IAM policy allows IAM principals to make **ListRules** requests only if the value specified for the `ResourceType` request parameter is `EBS_SNAPSHOT`. This allows the principal to list retention rules for snapshots only, and it prevents them from listing retention rules for any other resource type.

```

{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "rbin:ListRules"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "rbin:Request/ResourceType" : "EBS_SNAPSHOT"
        }
      }
    }
  ]
}

```

rbin:Attribute/ResourceType condition key

The `rbin:Attribute/ResourceType` condition key can be used to filter access on [DeleteRule](#), [GetRule](#), [UpdateRule](#), [LockRule](#), [UnlockRule](#), [TagResource](#), [UntagResource](#), and [ListTagsForResource](#) requests based on the value of the retention rule's `ResourceType` attribute.

Example 1 - UpdateRule

The following sample IAM policy allows IAM principals to make **UpdateRule** requests only if the `ResourceType` attribute of the requested retention rule is `EBS_SNAPSHOT` or `EC2_IMAGE`. This allows the principal to update retention rules for snapshots and AMIs only.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "rbin:UpdateRule"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "rbin:Attribute/ResourceType" : ["EBS_SNAPSHOT", "EC2_IMAGE"]
        }
      }
    }
  ]
}
```

Example 2 - DeleteRule

The following sample IAM policy allows IAM principals to make **DeleteRule** requests only if the `ResourceType` attribute of the requested retention rule is `EBS_SNAPSHOT`. This allows the principal to delete retention rules for snapshots only.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "rbin>DeleteRule"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "rbin:Attribute/ResourceType" : "EBS_SNAPSHOT"
        }
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Work with retention rules

To enable and use Recycle Bin, you must create *retention rules* in the AWS Regions in which you want to protect your resources. Retention rules specify the following:

- The resource type that you want to protect.
- The resources that you want to retain in the Recycle Bin when they are deleted.
- The retention period for which to retain resources in the Recycle Bin before they are permanently deleted.

With Recycle Bin, you can create two types of retention rules:

- **Tag-level retention rules** — A tag-level retention rule uses resource tags to identify the resources that are to be retained in the Recycle Bin. For each retention rule, you specify one or more tag key and value pairs. Resources of the specified type that are tagged with at least one of the tag key and value pairs that are specified in the retention rule are automatically retained in the Recycle Bin upon deletion. Use this type of retention rule if you want to protect specific resources in your account based on their tags.
- **Region-level retention rules** — A Region-level retention rule does not have any resource tags specified. It applies to all of the resources of the specified type in the Region in which the rule is created, even if the resources are not tagged. Use this type of retention rule if you want to protect all resources of a specific type in a specific Region.

After you create a retention rule, resources that match its criteria are automatically retained in the Recycle Bin for the specified retention period after they are deleted.

Topics

- [Create a retention rule](#)
- [View Recycle Bin retention rules](#)
- [Update retention rules](#)
- [Lock retention rules](#)
- [Unlock retention rules](#)

- [Tag retention rules](#)
- [View retention rule tags](#)
- [Remove tags from retention rules](#)
- [Delete Recycle Bin retention rules](#)

Create a retention rule

When you create a retention rule, you must specify the following required parameters:

- The resource type that is to be protected by the retention rule.
- The resources that are to be protected by the retention rule. You can create retention rules at the tag level and the Region level.
 - To create a tag-level retention rule, specify the resource tags that identify the resources to protect. You can specify up to 50 tags for each rule, and add the same tag key and value pair to a maximum of five retention rules.
 - To create a Region-level retention rule, do not specify any tag key and value pairs. In this case, all resources of the specified type are protected.
- The period to retain the resources in the Recycle Bin after they are deleted. The period can be up to 1 year (365 days).

You can also specify the following optional parameters:

- An optional name for the retention rule. The name can be up to 255 characters long.
- An optional description for the retention rule. The description can be up to 255 characters long.

Note

We recommend that you do not include personally identifying, confidential, or sensitive information in the retention rule description.

- Optional retention rule tags to help identify and organize your retention rules. You can assign up to 50 tags to each rule.

You can also optionally lock retention rules on creation. If you lock a retention rule on creation, you must also specify the unlock delay period, which can be 7 to 30 days. Retention rules remain unlocked by default unless you explicitly lock them.

Retention rules function only in the Regions in which they are created. If you intend to use Recycle Bin in other Regions, you must create additional retention rules in those Regions.

You can create a Recycle Bin retention rule using one of the following methods.

Recycle Bin console

To create a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**, and then choose **Create retention rule**.
3. In the **Rule details** section, do the following:
 - a. *(Optional)* For **Retention rule name**, enter a descriptive name for the retention rule.
 - b. *(Optional)* For **Retention rule description**, enter a brief description for the retention rule.
4. In the **Rule settings** section, do the following:
 - a. For **Resource type**, select choose the type of resource for the retention rule to protect. The retention rule will retain only resources of this type in the Recycle Bin.
 - b. Do one of the following:
 - To create a Region-level retention rule that matches all deleted resources of the specified type in the Region, select **Apply to all resources**. The retention rule will retain all deleted resources of the specified in the Recycle Bin upon deletion, even if the resources do not have any tags.
 - To create a tag-level retention rule, for **Resource tags to match**, enter the tag key and value pairs to use to identify resource of the specified type that are to be retained in the Recycle Bin. Only resources of the specified type that have at least one of the specified tag key and value pairs will be retained by the retention rule.
 - c. For **Retention period**, enter the number of days for which the retention rule is to retain resources in the Recycle Bin.
5. *(Optional)* To lock the retention rule, for **Rule lock settings**, select **Lock**, and then for **Unlock delay period**, specify the unlock delay period in days. A locked retention rule can't

be modified or deleted. To modify or delete the rule, you must first unlock it and then wait for the unlock delay period to expire. For more information, see [Lock retention rules](#)

To leave the retention rule unlocked, for **Rule lock settings**, keep **Unlock** selected. An unlocked retention rule can be modified or deleted at any time. For more information, see [Unlock retention rules](#).

6. (Optional) In the **Tags** section, do the following:
 - To tag the rule with custom tags, choose **Add tag** and then enter the tag key and value pair.
7. Choose **Create retention rule**.

AWS CLI

To create a retention rule

Use the [create-rule](#) AWS CLI command. For `--retention-period`, specify the number of days to retain deleted snapshots in the Recycle Bin. For `--resource-type`, specify `EBS_SNAPSHOT` for snapshots or `EC2_IMAGE` for AMIs. To create a tag-level retention rule, for `--resource-tags`, specify the tags to use to identify the snapshots that are to be retained. To create a Region-level retention rule, omit `--resource-tags`. To lock a retention rule, include `--lock-configuration`, and specify the unlock delay period in days.

```
$ aws rbin create-rule \  
--retention-period RetentionPeriodValue=number_of_days,RetentionPeriodUnit=DAYS \  
--resource-type EBS_SNAPSHOT|EC2_IMAGE \  
--description "rule_description" \  
--lock-configuration  
'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=unlock_delay_in_days}' \  
--resource-tags ResourceTagKey=tag_key,ResourceTagValue=tag_value
```

Example 1

The following example command creates an unlocked Region-level retention rule that retains all deleted snapshots for a period of 7 days.

```
$ aws rbin create-rule \  
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \  
--resource-type EBS_SNAPSHOT \  

```

```
--description "Match all snapshots"
```

Example 2

The following example command creates a tag-level rule that retains deleted snapshots that are tagged with `purpose=production` for a period of 7 days.

```
$ aws rbin create-rule \  
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \  
--resource-type EBS_SNAPSHOT \  
--description "Match snapshots with a specific tag" \  
--resource-tags ResourceTagKey=purpose,ResourceTagValue=production
```

Example 3

The following example command creates a locked Region-level retention rule that retains all deleted snapshots for a period of 7 days. The retention rule is locked with an unlock delay period of 7 days.

```
$ aws rbin create-rule \  
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \  
--resource-type EBS_SNAPSHOT \  
--description "Match all snapshots" \  
--lock-configuration 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=7}'
```

View Recycle Bin retention rules

You can view Recycle Bin retention rules using one of the following methods.

Recycle Bin console

To view retention rules

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**.
3. The grid lists all of the retention rules for the selected Region. To view more information about a specific retention rule, select it in the grid.

AWS CLI

To view all of your retention rules

Use the [list-rules](#) AWS CLI command, and for `--resource-type`, specify `EBS_SNAPSHOT` for snapshots or `EC2_IMAGE` for AMIs.

```
$ aws rbin list-rules --resource-type EBS_SNAPSHOT|EC2_IMAGE
```

Example

The following example command provides lists all retention rules that retain snapshots.

```
$ aws rbin list-rules --resource-type EBS_SNAPSHOT
```

To view information for a specific retention rule

Use the [get-rule](#) AWS CLI command.

```
$ aws rbin get-rule --identifier rule_ID
```

Example

The following example command provides information about retention rule `pwxIkFcvge4`.

```
$ aws rbin get-rule --identifier pwxIkFcvge4
```

Update retention rules

You can update an unlocked retention rule's description, resource tags, and retention period at any time after creation. You can't update a retention rule's resource type or unlock delay period, even if the retention rule is unlocked.

You can't update a locked retention rule in any way. If you need to modify a locked retention rule, you must first unlock it and wait for the unlock delay period to expire.

If you need to modify the unlock delay period for a locked retention rule, you must [unlock the retention rule](#), and wait for the current unlock delay period to expire. When the unlock delay period is expired, you must [relock the retention rule](#) and specify the new unlock delay period.

Note

We recommend that you do not include personally identifying, confidential, or sensitive information in the retention rule description.

After you update a retention rule, the changes only apply to new resources that it retains. The changes do not affect resources that it previously sent to the Recycle Bin. For example, if you update a retention rule's retention period, only snapshots that are deleted after the update are retained for the new retention period. Snapshots that it sent to the Recycle Bin before the update are still retained for the previous (old) retention period.

You can update a retention rule using one of the following methods.

Recycle Bin console**To update a retention rule**

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**.
3. In the grid, select the retention rule to update, and choose **Actions, Edit retention rule**.
4. In the **Rule details** section, update **Retention rule name** and **Retention rule description** as needed.
5. In the **Rule settings** section, update the **Resource type**, **Resource tags to match**, and **Retention period** as needed.
6. In the **Tags** section, add or remove retention rule tags as needed.
7. Choose **Save retention rule**.

AWS CLI**To update a retention rule**

Use the [update-rule](#) AWS CLI command. For `--identifier`, specify the ID of the retention rule to update. For `--resource-types`, specify `EBS_SNAPSHOT` for snapshots or `EC2_IMAGE` for AMIs.

```
$ aws rbin update-rule \  
--identifier rule_ID \  

```

```
--retention-period RetentionPeriodValue=number_of_days,RetentionPeriodUnit=DAYS \  
--resource-type EBS_SNAPSHOT|EC2_IMAGE \  
--description "rule_description"
```

Example

The following example command updates retention rule 6lsJ2Fa9nh9 to retain all snapshots for 7 days and updates its description.

```
$ aws rbin update-rule \  
--identifier 6lsJ2Fa9nh9 \  
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \  
--resource-type EBS_SNAPSHOT \  
--description "Retain for three weeks"
```

Lock retention rules

Recycle Bin lets you lock Region-level retention rules at any time.

Note

You can't lock tag-level retention rules.

A locked retention rule can't be modified or deleted, even by users who have the required IAM permissions. Lock your retention rules to help protect them against accidental or malicious modifications and deletions.

When you lock a retention rule, you must specify an unlock delay period. This is the period of time that you must wait after unlocking the retention rule before you can modify or delete it. You cannot modify or delete the retention rule during the unlock delay period. You can modify or delete the retention rule only after the unlock delay period has expired.

You can't change the unlock delay period after the retention rule has been locked. If your account permissions have been compromised, the unlock delay period gives you additional time to detect and respond to security threats. The length of this period should be longer than the time it takes for you to identify and respond to security breaches. To set the right duration, you can review previous security incidents and the time needed to identify and remediate an account breach.

We recommend that you use Amazon EventBridge rules to notify you of retention rule lock state changes. For more information, see [Monitor Recycle Bin using Amazon EventBridge](#).

Considerations

- You can lock Region-level retention rules only.
- You can lock an unlocked retention rule at any time.
- The unlock delay period must be 7 to 30 days.
- You can re-lock a retention rule during the unlock delay period. Relocking the retention rule resets the unlock delay period.

You can lock a Region-level retention rule using one of the following methods.

Recycle Bin console

To lock a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation panel, choose **Retention rules**.
3. In the grid, select the unlocked retention rule to lock, and choose **Actions, Edit retention rule lock**.
4. In the Edit retention rule lock screen, choose **Lock**, and then for **Unlock delay period**, specify the unlock delay period in days.
5. Select the **I acknowledge that locking the retention rule will prevent it from being modified or deleted** check box, and then choose **Save**.

AWS CLI

To lock an unlocked retention rule

Use the [lock-rule](#) AWS CLI command. For `--identifier`, specify the ID of the retention rule to lock. For `--lock-configuration`, specify the unlock delay period in days.

```
$ aws rbin lock-rule \  
--identifier rule_ID \  
--lock-configuration  
'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=number_of_days}'
```


Example

The following example command locks retention rule 61sJ2Fa9nh9 and sets the unlock delay period to 15 days.

```
$ aws rbin lock-rule \  
--identifier 61sJ2Fa9nh9 \  
--lock-configuration 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=15}'
```

Unlock retention rules

You can't modify or delete a locked retention rule. If you need to modify a locked retention rule, you must first unlock it. After you have unlocked the retention rule, you must wait for the unlock delay period to expire before you modify or delete it. You can't modify or delete a retention rule during the unlock delay period.

An unlocked retention rule can be modified and deleted at any time by a user who has the required IAM permissions. Leaving your retention rules unlocked could expose them to accidental or malicious modifications and deletions.

Considerations

- You can re-lock a retention rule during the unlock delay period.
- You can re-lock a retention rule after the unlock delay period has expired.
- You can't bypass the unlock delay period.
- You can't change the unlock delay period after the initial lock.

We recommend that you use Amazon EventBridge rules to notify you of retention rule lock state changes. For more information, see [Monitor Recycle Bin using Amazon EventBridge](#).

You can unlock a locked Region-level retention rule using one of the following methods.

Recycle Bin console

To unlock a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation panel, choose **Retention rules**.

3. In the grid, select the locked retention rule to unlock, and choose **Actions, Edit retention rule lock**.
4. On the Edit retention rule lock screen, choose **Unlock**, and then choose **Save**.

AWS CLI

To unlock a locked retention rule

Use the [unlock-rule](#) AWS CLI command. For `--identifier`, specify the ID of the retention rule to unlock.

```
$ aws rbin unlock-rule \  
--identifier rule_ID
```

Example

The following example command unlocks retention rule 61sJ2Fa9nh9

```
$ aws rbin unlock-rule \  
--identifier 61sJ2Fa9nh9
```

Tag retention rules

You can assign custom tags to your retention rules to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to efficiently find a specific retention rule based on the custom tags that you assigned.

You can assign a tag to a retention rule using one of the following methods.

Recycle Bin console

To tag a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**.
3. Select the retention rule to tag, choose the **Tags** tab, and then choose **Manage tags**.
4. Choose **Add tag**. For **Key**, enter the tag key. For **Value**, enter the tag value.
5. Chose **Save**.

AWS CLI

To tag a retention rule

Use the [tag-resource](#) AWS CLI command. For `--resource-arn`, specify the Amazon Resource Name (ARN) of the retention rule to tag, and for `--tags`, specify the tag key and value pair.

```
$ aws rbin tag-resource \  
--resource-arn retention_rule_arn \  
--tags key=tag_key,value=tag_value
```

Example

The following example command tags retention rule `arn:aws:rbin:us-east-1:123456789012:rule/n0oSBBtItF3` with tag `purpose=production`.

```
$ aws rbin tag-resource \  
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/n0oSBBtItF3 \  
--tags key=purpose,value=production
```

View retention rule tags

You can view the tags assigned to a retention rule using one of the following methods.

Recycle Bin console

To view tags for a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**.
3. Select the retention rule for which to view the tags, and choose the **Tags** tab.

AWS CLI

To view the tags assigned to a retention rule

Use the [list-tags-for-resource](#) AWS CLI command. For `--resource-arn`, specify the ARN of the retention rule.

```
$ aws rbin list-tags-for-resource \  
--resource-arn retention_rule_arn
```

Example

The following example command lists the tags for retention rule `arn:aws:rbin:us-east-1:123456789012:rule/n0oSBBtItF3`.

```
$ aws rbin list-tags-for-resource \  
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/n0oSBBtItF3
```

Remove tags from retention rules

You can remove tags from a retention rule using one of the following methods.

Recycle Bin console

To remove a tag from a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**.
3. Select the retention rule from which to remove the tag, choose the **Tags** tab, and then choose **Manage tags**.
4. Choose **Remove** next to the tag to remove.
5. Chose **Save**.

AWS CLI

To remove a tag from a retention rule

Use the [untag-resource](#) AWS CLI command. For `--resource-arn`, specify the ARN of the retention rule. For `--tagkeys`, specify the tags keys of the tags to remove.

```
$ aws rbin untag-resource \  
--resource-arn retention_rule_arn \  
--tagkeys tag_key
```

Example

The following example command removes tags that have a tag key of purpose from retention rule `arn:aws:rbin:us-east-1:123456789012:rule/n0oSBBtItF3`.

```
$ aws rbin untag-resource \  
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/n0oSBBtItF3 \  
--tagkeys purpose
```

Delete Recycle Bin retention rules

You can delete a retention rule at any time. When you delete a retention rule, it no longer retains new resources in the Recycle Bin after they have been deleted. Resources that were sent to the Recycle Bin before the retention rule was deleted continue to be retained in the Recycle Bin according to the retention period defined in the retention rule. When the period expires, the resource is permanently deleted from the Recycle Bin.

You can delete a retention rule using one of the following methods.

Recycle Bin console

To delete a retention rule

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Retention rules**.
3. In the grid, select the retention rule to delete, and choose **Actions, Delete retention rule**.
4. When prompted, enter the confirmation message and choose **Delete retention rule**.

AWS CLI

To delete a retention rule

Use the [delete-rule](#) AWS CLI command. For `--identifier`, specify the ID of the retention rule to delete.

```
$ aws rbin delete-rule --identifier rule_ID
```

Example

The following example command deletes retention rule 61sJ2Fa9nh9.

```
$ aws rbin delete-rule --identifier 61sJ2Fa9nh9
```

Work with resources in the Recycle Bin

Recycle Bin supports the following resource types:

- Amazon EBS snapshots
- Amazon EBS-backed Amazon Machine Images (AMIs)

Tasks

- [Recover snapshots from the Recycle Bin](#)
- [Recover AMIs from the Recycle Bin](#)

Recover snapshots from the Recycle Bin

Recycle Bin is a data recovery feature that enables you to restore accidentally deleted Amazon EBS snapshots and EBS-backed AMIs. When using Recycle Bin, if your resources are deleted, they are retained in the Recycle Bin for a time period that you specify before being permanently deleted.

You can restore a resource from the Recycle Bin at any time before its retention period expires. After you restore a resource from the Recycle Bin, the resource is removed from the Recycle Bin and you can use it in the same way that you use any other resource of that type in your account. If the retention period expires and the resource is not restored, the resource is permanently deleted from the Recycle Bin and it is no longer available for recovery.

Snapshots in the Recycle Bin are billed at the same rate as regular snapshots in your account. There are no additional charges for using Recycle Bin and retention rules. For more information, see [Amazon EBS pricing](#).

For more information, see [Recycle Bin](#).

Topics

- [Permissions for working with snapshots in the Recycle Bin](#)
- [View snapshots in the Recycle Bin](#)

- [Restore snapshots from the Recycle Bin](#)

Permissions for working with snapshots in the Recycle Bin

By default, users don't have permission to work with snapshots that are in the Recycle Bin. To allow users to work with these resources, you must create IAM policies that grant permission to use specific resources and API actions. Once the policies are created, you must add permissions to your users, groups, or roles.

To view and recover snapshots that are in the Recycle Bin, users must have the following permissions:

- `ec2:ListSnapshotsInRecycleBin`
- `ec2:RestoreSnapshotFromRecycleBin`

To manage tags for snapshots in the Recycle Bin, users need the following additional permissions.

- `ec2:CreateTags`
- `ec2>DeleteTags`

To use the Recycle Bin console, users need the `ec2:DescribeTags` permission.

The following is an example IAM policy. It includes the `ec2:DescribeTags` permission for console users, and it includes the `ec2:CreateTags` and `ec2>DeleteTags` permissions for managing tags. If the permissions are not needed, you can remove them from the policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:ListSnapshotsInRecycleBin",
        "ec2:RestoreSnapshotFromRecycleBin"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
```

```
    "Action": [
      "ec2:CreateTags",
      "ec2>DeleteTags",
      "ec2:DescribeTags"
    ],
    "Resource": "arn:aws:ec2:Region:account-id:snapshot/*"
  },
]
}
```

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 [Creating 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 [Creating 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*.

For more information about the permissions needed to use Recycle Bin, see [Permissions for working with Recycle Bin and retention rules](#).

View snapshots in the Recycle Bin

While a snapshot is in the Recycle Bin, you can view limited information about it, including:

- The ID of the snapshot.
- The snapshot description.
- The ID of the volume from which the snapshot was created.
- The date and time when the snapshot was deleted and it entered Recycle Bin.

- The date and time when the retention period expires. The snapshot will be permanently deleted from the Recycle Bin at this time.

You can view the snapshots in the Recycle Bin using one of the following methods.

Recycle Bin console

To view snapshots in the Recycle Bin using the console

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Recycle Bin**.
3. The grid lists all of the snapshots that are currently in the Recycle Bin. To view the details for a specific snapshot, select it in the grid and choose **Actions, View details**.

AWS CLI

To view snapshots in the Recycle Bin using the AWS CLI

Use the [list-snapshots-in-recycle-bin](#) AWS CLI command. Include the `--snapshot-id` option to view a specific snapshot. Or omit the `--snapshot-id` option to view all snapshots in the Recycle Bin.

```
$ aws ec2 list-snapshots-in-recycle-bin --snapshot-id snapshot_id
```

For example, the following command provides information about snapshot `snap-01234567890abcdef` in the Recycle Bin.

```
$ aws ec2 list-snapshots-in-recycle-bin --snapshot-id snap-01234567890abcdef
```

Example output:

```
{
  "SnapshotRecycleBinInfo": [
    {
      "Description": "Monthly data backup snapshot",
      "RecycleBinEnterTime": "2021-12-01T13:00:00.000Z",
      "RecycleBinExitTime": "2021-12-15T13:00:00.000Z",
      "VolumeId": "vol-abcdef09876543210",
    }
  ]
}
```

```
        "SnapshotId": "snap-01234567890abcdef"  
      }  
    ]  
  }
```

Restore snapshots from the Recycle Bin

You can't use a snapshot in any way while it is in the Recycle Bin. To use the snapshot, you must first restore it. When you restore a snapshot from the Recycle Bin, the snapshot is immediately available for use, and it is removed from the Recycle Bin. You can use a restored snapshot in the same way that you use any other snapshot in your account.

You can restore a snapshot from the Recycle Bin using one of the following methods.

Recycle Bin console

To restore a snapshot from the Recycle Bin using the console

1. Open the Recycle Bin console at <https://console.aws.amazon.com/rbin/home/>
2. In the navigation pane, choose **Recycle Bin**.
3. The grid lists all of the snapshots that are currently in the Recycle Bin. Select the snapshot to restore and choose **Recover**.
4. When prompted, choose **Recover**.

AWS CLI

To restore a deleted snapshot from the Recycle Bin using the AWS CLI

Use the [restore-snapshot-from-recycle-bin](#) AWS CLI command. For `--snapshot-id`, specify the ID of the snapshot to restore.

```
$ aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snapshot_id
```

For example, the following command restores snapshot `snap-01234567890abcdef` from the Recycle Bin.

```
$ aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snap-01234567890abcdef
```

Example output:

```
{
  "SnapshotId": "snap-01234567890abcdef",
  "Description": "Monthly data backup snapshot",
  "Encrypted": false,
  "OwnerId": "111122223333",
  "Progress": "100%",
  "StartTime": "2021-12-01T13:00:00.000000+00:00",
  "State": "recovering",
  "VolumeId": "vol-ffffffff",
  "VolumeSize": 30
}
```

Recover AMIs from the Recycle Bin

Recycle Bin is a data recovery feature that enables you to restore accidentally deleted Amazon EBS snapshots and EBS-backed AMIs. When using Recycle Bin, if your resources are deleted, they are retained in the Recycle Bin for a time period that you specify before being permanently deleted.

You can restore a resource from the Recycle Bin at any time before its retention period expires. After you restore a resource from the Recycle Bin, the resource is removed from the Recycle Bin and you can use it in the same way that you use any other resource of that type in your account. If the retention period expires and the resource is not restored, the resource is permanently deleted from the Recycle Bin and it is no longer available for recovery.

AMIs in the Recycle Bin do not incur any additional charges.

For more information, see [Recycle Bin](#).

Topics

- [Permissions for working with AMIs in the Recycle Bin](#)
- [View AMIs in the Recycle Bin](#)
- [Restore AMIs from the Recycle Bin](#)

Permissions for working with AMIs in the Recycle Bin

By default, users don't have permission to work with AMIs that are in the Recycle Bin. To allow users to work with these resources, you must create IAM policies that grant permission to use

specific resources and API actions. Once the policies are created, you must add permissions to your users, groups, or roles.

To view and recover AMIs that are in the Recycle Bin, users must have the following permissions:

- `ec2:ListImagesInRecycleBin`
- `ec2:RestoreImageFromRecycleBin`

To manage tags for AMIs in the Recycle Bin, users need the following additional permissions.

- `ec2:CreateTags`
- `ec2>DeleteTags`

To use the Recycle Bin console, users need the `ec2:DescribeTags` permission.

The following is an example IAM policy. It includes the `ec2:DescribeTags` permission for console users, and it includes the `ec2:CreateTags` and `ec2>DeleteTags` permissions for managing tags. If the permissions are not needed, you can remove them from the policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:ListImagesInRecycleBin",
        "ec2:RestoreImageFromRecycleBin"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2>DeleteTags",
        "ec2:DescribeTags"
      ],
      "Resource": "arn:aws:ec2:Region::image/*"
    }
  ]
}
```

```
}
```

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 [Creating 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 [Creating 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*.

For more information about the permissions needed to use Recycle Bin, see [Permissions for working with Recycle Bin and retention rules](#).

View AMIs in the Recycle Bin

While an AMI is in the Recycle Bin, you can view limited information about it, including:

- The name, description, and unique ID of the AMI.
- The date and time when the AMI was deleted and it entered Recycle Bin.
- The date and time when the retention period expires. The AMI will be permanently deleted at this time.

You can view the AMIs in the Recycle Bin using one of the following methods.

Recycle Bin console

To view deleted AMIs in the Recycle Bin using the console

1. Open the Recycle Bin console at console.aws.amazon.com/rbin/home/.
2. In the navigation pane, choose **Recycle Bin**.

3. The grid lists all of the resources that are currently in the Recycle Bin. To view the details for a specific AMI, select it in the grid, and choose **Actions, View details**.

AWS CLI

To view deleted AMIs in the Recycle Bin using the AWS CLI

Use the [list-images-in-recycle-bin](#) AWS CLI command. To view specific AMIs, include the `--image-id` option and specify the IDs of the AMIs to view. You can specify up to 20 IDs in a single request.

To view all of the AMIs in the Recycle Bin, omit the `--image-id` option. If you do not specify a value for `--max-items`, the command returns 1,000 items per page, by default. For more information, see [Pagination](#) in the *Amazon EC2 API Reference*.

```
$ aws ec2 list-images-in-recycle-bin --image-id ami_id
```

For example, the following command provides information about AMI `ami-01234567890abcdef` in the Recycle Bin.

```
$ aws ec2 list-images-in-recycle-bin --image-id ami-01234567890abcdef
```

Example output:

```
{
  "Images": [
    {
      "ImageId": "ami-0f740206c743d75df",
      "Name": "My AL2 AMI",
      "Description": "My Amazon Linux 2 AMI",
      "RecycleBinEnterTime": "2021-11-26T21:04:50+00:00",
      "RecycleBinExitTime": "2022-03-06T21:04:50+00:00"
    }
  ]
}
```

Important

If you receive the following error, you might need to update your AWS CLI version. For more information, see [Command not found errors](#).

```
aws.exe: error: argument operation: Invalid choice, valid choices are: ...
```

Restore AMIs from the Recycle Bin

You can't use an AMI in any way while it is in the Recycle Bin. To use the AMI, you must first restore it. When you restore an AMI from the Recycle Bin, the AMI is immediately available for use, and it is removed from the Recycle Bin. You can use a restored AMI in the same way that you use any other AMI in your account.

You can restore an AMI from the Recycle Bin using one of the following methods.

Recycle Bin console

To restore an AMI from the Recycle Bin using the console

1. Open the Recycle Bin console at console.aws.amazon.com/rbin/home/.
2. In the navigation pane, choose **Recycle Bin**.
3. The grid lists all of the resources that are currently in the Recycle Bin. Select the AMI to restore, and choose **Recover**.
4. When prompted, choose **Recover**.

AWS CLI

To restore a deleted AMI from the Recycle Bin using the AWS CLI

Use the [restore-image-from-recycle-bin](#) AWS CLI command. For `--image-id`, specify the ID of the AMI to restore.

```
$ aws ec2 restore-image-from-recycle-bin --image-id ami_id
```

For example, the following command restores AMI `ami-01234567890abcdef` from the Recycle Bin.

```
$ aws ec2 restore-image-from-recycle-bin --image-id ami-01234567890abcdef
```

The command returns no output on success.

⚠ Important

If you receive the following error, you might need to update your AWS CLI version. For more information, see [Command not found errors](#).

```
aws.exe: error: argument operation: Invalid choice, valid choices are: ...
```

Monitor Recycle Bin

You can use the following features to monitor the Recycle Bin.

Topics

- [Monitor Recycle Bin using Amazon EventBridge](#)
- [Monitor Recycle Bin using AWS CloudTrail](#)

Monitor Recycle Bin using Amazon EventBridge

Recycle Bin sends events to Amazon EventBridge for actions performed on retention rules. With EventBridge, you can establish rules that initiate programmatic actions in response to these events. For example, you can create a EventBridge rule that sends a notification to your email when a retention rule is unlocked and it enters its unlock delay period. For more information, see [Creating Amazon EventBridge rules that react to events](#).

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 initiated the event. For more information, see [Amazon EventBridge event patterns](#) in the *Amazon EventBridge User Guide*.

For more information about Amazon EventBridge, see [What Is Amazon EventBridge?](#) in the *Amazon EventBridge User Guide*.

Events

- [RuleLocked](#)
- [RuleChangeAttempted](#)
- [RuleUnlockScheduled](#)

- [RuleUnlockingNotice](#)
- [RuleUnlocked](#)

RuleLocked

The following is an example of an event that Recycle Bin generates when a retention rule is successfully locked. This event can be generated by **CreateRule** and **LockRule** requests. The API that generated the event is noted in the `api-name` field.

```
{
  "version": "0",
  "id": "exampleb-b491-4cf7-a9f1-bf370example",
  "detail-type": "Recycle Bin Rule Locked",
  "source": "aws.rbin",
  "account": "123456789012",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
  ],
  "detail": {
    "detail-version": " 1.0.0",
    "rule-id": "a12345abcde",
    "rule-description": "locked account level rule",
    "unlock-delay-period": "30 days",
    "api-name": "CreateRule"
  }
}
```

RuleChangeAttempted

The following is an example of an event that Recycle Bin generates for unsuccessful attempts to modify or delete a locked rule. This event can be generated by **DeleteRule** and **UpdateRule** requests. The API that generated the event is noted in the `api-name` field.

```
{
  "version": "0",
  "id": "exampleb-b491-4cf7-a9f1-bf370example",
  "detail-type": "Recycle Bin Rule Change Attempted",
  "source": "aws.rbin",
```

```
"account": "123456789012",
"time": "2022-08-10T16:37:50Z",
"region": "us-west-2",
"resources": [
  "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
],
"detail":
{
  "detail-version": " 1.0.0",
  "rule-id": "a12345abcde",
  "rule-description": "locked account level rule",
  "unlock-delay-period": "30 days",
  "api-name": "DeleteRule"
}
}
```

RuleUnlockScheduled

The following is an example of an event that Recycle Bin generates when a retention rule is unlocked and it starts its unlock delay period.

```
{
  "version": "0",
  "id": "exampleb-b491-4cf7-a9f1-bf370example",
  "detail-type": "Recycle Bin Rule Unlock Scheduled",
  "source": "aws.rbin",
  "account": "123456789012",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
  ],
  "detail":
  {
    "detail-version": " 1.0.0",
    "rule-id": "a12345abcde",
    "rule-description": "locked account level rule",
    "unlock-delay-period": "30 days",
    "scheduled-unlock-time": "2022-09-10T16:37:50Z",
  }
}
```

RuleUnlockingNotice

The following is an example of an event that Recycle Bin generates daily while a retention rule is in its unlock delay period, until the day before the unlock delay period expires.

```
{
  "version": "0",
  "id": "exampleb-b491-4cf7-a9f1-bf370example",
  "detail-type": "Recycle Bin Rule Unlocking Notice",
  "source": "aws.rbin",
  "account": "123456789012",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
  ],
  "detail": {
    "detail-version": " 1.0.0",
    "rule-id": "a12345abcde",
    "rule-description": "locked account level rule",
    "unlock-delay-period": "30 days",
    "scheduled-unlock-time": "2022-09-10T16:37:50Z"
  }
}
```

RuleUnlocked

The following is an example of an event that Recycle Bin generates when the unlock delay period for a retention rule expires and the retention rule can be modified or deleted.

```
{
  "version": "0",
  "id": "exampleb-b491-4cf7-a9f1-bf370example",
  "detail-type": "Recycle Bin Rule Unlocked",
  "source": "aws.rbin",
  "account": "123456789012",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
  ],
  "detail":
```

```
{
  "detail-version": " 1.0.0",
  "rule-id": "a12345abcde",
  "rule-description": "locked account level rule",
  "unlock-delay-period": "30 days",
  "scheduled-unlock-time": "2022-09-10T16:37:50Z"
}
```

Monitor Recycle Bin using AWS CloudTrail

The Recycle Bin service is integrated with AWS CloudTrail. CloudTrail is a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls performed in Recycle Bin as events. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket. If you don't configure a trail, you can still view the most recent management events in the CloudTrail console in **Event history**. You can use the information collected by CloudTrail to determine the request that was made to Recycle Bin, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

Recycle Bin information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in Recycle Bin, 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 Recycle Bin, create a trail. A *trail* enables CloudTrail to deliver log files to an S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see [Overview for creating a trail](#) in the *AWS CloudTrail User Guide*.

Supported API actions

For Recycle Bin, you can use CloudTrail to log the following API actions as *management events*.

- CreateRule

- UpdateRule
- GetRules
- ListRule
- DeleteRule
- TagResource
- UntagResource
- ListTagsForResource
- LockRule
- UnlockRule

For more information about logging management events, see [Logging management events for trails](#) in the *CloudTrail User Guide*.

Identity information

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentityElement](#).

Understand Recycle Bin log file entries

A trail is a configuration that enables delivery of events as log files to an S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following are example CloudTrail log entries.

CreateRule

```
{
```

```
"eventVersion": "1.08",
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "123456789012",
  "arn": "arn:aws:iam::123456789012:root",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "123456789012",
      "arn": "arn:aws:iam::123456789012:role/Admin",
      "accountId": "123456789012",
      "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-08-02T21:43:38Z"
    }
  }
},
"eventTime": "2021-08-02T21:45:22Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "CreateRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
"requestParameters": {
  "retentionPeriod": {
    "retentionPeriodValue": 7,
    "retentionPeriodUnit": "DAYS"
  },
  "description": "Match all snapshots",
  "resourceType": "EBS_SNAPSHOT"
},
"responseElements": {
  "identifier": "jkrnexample"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
```

```

"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}

```

GetRule

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-08-02T21:43:38Z"
      }
    }
  },
  "eventTime": "2021-08-02T21:45:33Z",
  "eventSource": "rbin.amazonaws.com",
  "eventName": "GetRule",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 boto-core/1.21.9",
  "requestParameters": {

```

```

    "identifier": "jkrnexample"
  },
  "responseElements": null,
  "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
  "eventID": "714fafex-2eam-42pl-913e-926d4example",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
  }
}

```

ListRules

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-08-02T21:43:38Z"
      }
    }
  },
  "eventTime": "2021-08-02T21:44:37Z",

```



```

"eventSource": "rbin.amazonaws.com",
"eventName": "ListRules",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 boto-core/1.21.9",
"requestParameters": {
  "resourceTags": [
    {
      "resourceTagKey": "test",
      "resourceTagValue": "test"
    }
  ]
},
"responseElements": null,
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}

```

UpdateRule

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",

```

```
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-08-02T21:43:38Z"
  }
}
},
"eventTime": "2021-08-02T21:46:03Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "UpdateRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 boto3/1.21.9",
"requestParameters": {
  "identifier": "jkrnexample",
  "retentionPeriod": {
    "retentionPeriodValue": 365,
    "retentionPeriodUnit": "DAYS"
  }
},
"description": "Match all snapshots",
"resourceType": "EBS_SNAPSHOT"
},
"responseElements": null,
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```

DeleteRule

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-08-02T21:43:38Z"
      }
    }
  },
  "eventTime": "2021-08-02T21:46:25Z",
  "eventSource": "rbin.amazonaws.com",
  "eventName": "DeleteRule",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 boto3/1.21.9",
  "requestParameters": {
    "identifier": "jkrnexample"
  },
  "responseElements": null,
  "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
  "eventID": "714fafex-2eam-42pl-913e-926d4example",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012",
  "tlsDetails": {
```

```

    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
  }
}

```

TagResource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-10-22T21:38:34Z"
      }
    }
  },
  "eventTime": "2021-10-22T21:43:15Z",
  "eventSource": "rbin.amazonaws.com",
  "eventName": "TagResource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 boto-core/1.21.26",
  "requestParameters": {
    "resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234",
    "tags": [
      {
        "key": "purpose",

```

```

        "value": "production"
      }
    ]
  },
  "responseElements": null,
  "requestID": "examplee-7962-49ec-8633-795efexample",
  "eventID": "example4-6826-4c0a-bdec-0bab1example",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
  }
}

```

UntagResource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      }
    },
    "webIdFederationData": {},
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-10-22T21:38:34Z"
    }
  }
}

```

```

},
"eventTime": "2021-10-22T21:44:16Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "UntagResource",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
"requestParameters": {
  "resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234",
  "tagKeys": [
    "purpose"
  ]
},
"responseElements": null,
"requestID": "example7-6c1e-4f09-9e46-bb957example",
"eventID": "example6-75ff-4c94-a1cd-4d5f5example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}

```

ListTagsForResource

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",

```

```

    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-10-22T21:38:34Z"
  }
}
},
"eventTime": "2021-10-22T21:42:31Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "ListTagsForResource",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 boto3/1.21.26",
"requestParameters": {
  "resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234"
},
"responseElements": null,
"requestID": "example8-10c7-43d4-b147-3d9d9example",
"eventID": "example2-24fc-4da7-a479-c9748example",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}

```

LockRule

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",

```

```
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "creationDate": "2022-10-25T00:45:11Z",
    "mfaAuthenticated": "false"
  }
},
"eventTime": "2022-10-25T00:45:19Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "LockRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "python-requests/2.25.1",
"requestParameters": {
  "identifier": "jkrnexample",
  "lockConfiguration": {
    "unlockDelay": {
      "unlockDelayValue": 7,
      "unlockDelayUnit": "DAYS"
    }
  }
},
"responseElements": {
  "identifier": "jkrnexample",
  "description": "",
  "resourceType": "EBS_SNAPSHOT",
  "retentionPeriod": {
    "retentionPeriodValue": 7,
    "retentionPeriodUnit": "DAYS"
  },
  "resourceTags": [],
  "status": "available",
  "lockConfiguration": {
```



```

    "unlockDelay": {
      "unlockDelayValue": 7,
      "unlockDelayUnit": "DAYS"
    }
  },
  "lockState": "locked"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}

```

UnlockRule

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-25T00:45:11Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```

```
    }
  }
},
"eventTime": "2022-10-25T00:46:17Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "UnlockRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "python-requests/2.25.1",
"requestParameters": {
  "identifier": "jkrnexample"
},
"responseElements": {
  "identifier": "jkrnexample",
  "description": "",
  "resourceType": "EC2_IMAGE",
  "retentionPeriod": {
    "retentionPeriodValue": 7,
    "retentionPeriodUnit": "DAYS"
  },
  "resourceTags": [],
  "status": "available",
  "lockConfiguration": {
    "unlockDelay": {
      "unlockDelayValue": 7,
      "unlockDelayUnit": "DAYS"
    }
  },
  "lockState": "pending_unlock",
  "lockEndTime": "Nov 1, 2022, 12:46:17 AM"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```

}

Resource locations

Amazon EC2 resources are specific to the AWS Region or Availability Zone in which they reside.

Resource	Type	Description
Amazon EC2 resource identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its Region and can be used only in the Region where you created the resource.
User-supplied resource names	Regional	Each resource name, such as a security group name or key pair name, is tied to its Region and can be used only in the Region where you created the resource. Although you can create resources with the same name in multiple Regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the Region where its files are located within Amazon S3. You can copy an AMI from one Region to another. For more information, see Copy an AMI .
EBS snapshots	Regional	An EBS snapshot is tied to its Region and can only be used to create volumes in the same Region. You can copy a snapshot from one Region to another.
EBS volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Elastic IP addresses	Regional	An Elastic IP address is tied to a Region and can be associated only with an instance in the same Region.

Resource	Type	Description
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, its instance ID is tied to the Region.
Key pairs	Global or Regional	<p>The key pairs that you create using Amazon EC2 are tied to the Region where you created them. You can create your own RSA key pair and upload it to the Region in which you want to use it; therefore, you can make your key pair globally available by uploading it to each Region.</p> <p>For more information, see Amazon EC2 key pairs and Amazon EC2 instances.</p>
Security groups	Regional	A security group is tied to a Region and can be assigned only to instances in the same Region. You can't enable an instance to communicate with an instance outside its Region using security group rules. Traffic from an instance in another Region is seen as WAN bandwidth.

Resource IDs

When resources are created, we assign each resource a unique resource ID. A resource ID takes the form of a resource identifier (such as snap for a snapshot) followed by a hyphen and a unique combination of letters and numbers.

Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its Region and can be used only in the Region where you created the resource.

You can use resource IDs to find your resources in the Amazon EC2 console. If you are using a command line tool or the Amazon EC2 API to work with Amazon EC2, resource IDs are required for certain commands. For example, if you are using the [stop-instances](#) AWS CLI command to stop an instance, you must specify the instance ID in the command.

Resource ID length

Prior to January 2016, the IDs assigned to newly created resources of certain resource types used 8 characters after the hyphen (for example, i-1a2b3c4d). From January 2016 to June 2018, we changed the IDs of these resource types to use 17 characters after the hyphen (for example, i-1234567890abcdef0). Depending on when your account was created, you might have some existing resources with short IDs, however, any new resources will receive the longer IDs.

List and filter your resources

You can get a list of some types of resources using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include, or exclude, only the resources that match certain criteria.

Contents

- [List and filter resources using the console](#)
- [List and filter using the CLI and API](#)
- [View resources across Regions using Amazon EC2 Global View](#)

List and filter resources using the console

Contents

- [List resources using the console](#)
- [Filter resources using the console](#)
 - [Supported filters](#)

List resources using the console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose the option that corresponds to the resource type. For example, to list your instances, choose **Instances**.

The page displays all resources of the selected resource type.

Filter resources using the console

To filter a list of resources

1. In the navigation pane, select a resource type (for example, **Instances**).
2. Choose the search field.
3. 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.

The screenshot shows the 'Edit filter' dialog box in the Amazon EC2 console. The dialog box is titled 'Edit filter' and has a close button (X) in the top right corner. It contains three fields: 'Property' with a dropdown menu showing 'Architecture', 'Operator' with a dropdown menu showing '=', and 'Value' with a search input field containing 'arm64'. Below these fields are 'Cancel' and 'Apply' buttons. In the background, a search bar is visible with the text 'Search', and a table with a header 'Name' is partially visible.

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** check box. If this check box is selected, client filtering is performed.
3. Select the **Use case sensitive matching** check box. If this check box 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** check box 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** check box 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.

List and filter using the CLI and API

Each resource type has a corresponding CLI command and API action that you use to list resources of that type. The resulting lists of resources can be long, so it can be faster and more useful to filter the results to include only the resources that match specific criteria.

Filtering considerations

- You can specify multiple filters and multiple filter values in a single request.
- You can use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches zero or one character.
- Filter values are case sensitive.
- Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of `*amazon\?\` searches for the literal string `*amazon?\`.

Supported filters

To see the supported filters for each Amazon EC2 resource, see the following documentation:

- AWS CLI: The `describe` commands in the [AWS CLI Command Reference-Amazon EC2](#).

- Tools for Windows PowerShell: The Get commands in the [AWS Tools for PowerShell Cmdlet Reference-Amazon EC2](#).
- Query API: The Describe API actions in the [Amazon EC2 API Reference](#).

Example Example: Specify a single filter

You can list your Amazon EC2 instances using [describe-instances](#). Without filters, the response contains information for all of your resources. You can use the following command to include only the running instances in your output.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=running
```

To list only the instance IDs for your running instances, add the `--query` parameter as follows.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=running --query "Reservations[*].Instances[*].InstanceId" --output text
```

The following is example output.

```
i-0ef1f57f78d4775a4  
i-0626d4edd54f1286d  
i-04a636d18e83cfacb
```

Example Example: Specify multiple filters or filter values

If you specify multiple filters or multiple filter values, the resource must match all filters to be included in the results.

You can use the following command to list all instances whose type is either `m5.large` or `m5d.large`.

```
aws ec2 describe-instances --filters Name=instance-type,Values=m5.large,m5d.large
```

You can use the following command to list all stopped instances whose type is `t2.micro`.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=stopped  
Name=instance-type,Values=t2.micro
```

Example Example: Use wildcards in a filter value

If you specify `database` as the filter value for the `description` filter when describing EBS snapshots using [describe-snapshots](#), the command returns only the snapshots whose description is "database".

```
aws ec2 describe-snapshots --filters Name=description,Values=database
```

The `*` wildcard matches zero or more characters. If you specify `*database*` as the filter value, the command returns only snapshots whose description includes the word `database`.

```
aws ec2 describe-snapshots --filters Name=description,Values=*database*
```

The `?` wildcard matches exactly 1 character. If you specify `database?` as the filter value, the command returns only snapshots whose description is "database" or "database" followed by one character.

```
aws ec2 describe-snapshots --filters Name=description,Values=database?
```

If you specify `database????`, the command returns only snapshots whose description is "database" followed by up to four characters. It excludes descriptions with "database" followed by five or more characters.

```
aws ec2 describe-snapshots --filters Name=description,Values=database????
```

Example Example: Filter based on date

With the AWS CLI, you can use JMESPath to filter results using expressions. For example, the following [describe-snapshots](#) command displays the IDs of all snapshots created by your AWS account (represented by `123456789012`) before the specified date (represented by `2020-03-31`). If you do not specify the owner, the results include all public snapshots.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]" --output text
```

The following command displays the IDs of all snapshots created in the specified date range.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query
  "Snapshots[?(StartTime>='2019-01-01') && (StartTime<='2019-12-31')].[SnapshotId]" --
output text
```

Filter based on tags

For examples of how to filter a list of resources according to their tags, see [Work with tags using the command line](#).

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 [Amazon EC2 Global View](#).

Amazon EC2 Global View

Amazon EC2 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. Amazon EC2 Global View also provides *global search* functionality that lets you search for specific resources or specific resource types across multiple Regions simultaneously.

Amazon EC2 Global View does not let you modify resources in any way.

Supported resources

Using Amazon EC2 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
- 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 Amazon EC2 Global View.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "ec2:DescribeRegions",
        "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 Amazon EC2 Global View

Open the Amazon EC2 Global View console at <https://console.aws.amazon.com/ec2globalview/home>.

Important

You cannot use a private window in Firefox to access Amazon EC2 Global View.

The console consists of the following:

- **Region explorer**—This tab includes the following sections:
 - **Summary**—Provides a high-level overview of your resources across all Regions.

Enabled Regions 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 region counts**—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 tab 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.

Tip

If you only use specific Regions or resource types, you can customize Amazon EC2 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 Amazon EC2 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](#)
- [Tag your resources](#)
- [Tag restrictions](#)

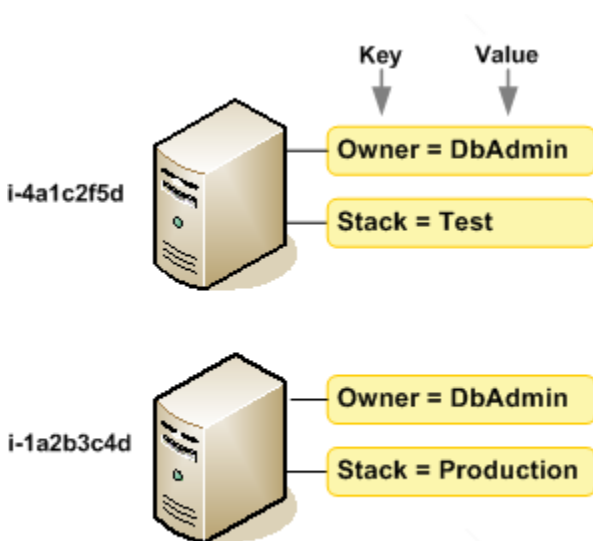
- [Tags and access management](#)
- [Tag your resources for billing](#)
- [Work with tags using the console](#)
- [Work with tags using the command line](#)
- [Work with instance tags in instance metadata](#)
- [Add tags to a resource using CloudFormation](#)

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

You can tag most Amazon EC2 resources that already exist in your account. The following [table](#) lists the resources that support tagging.

If you're using the Amazon EC2 console, you can apply tags to resources by using the **Tags** tab on the relevant resource screen, or you can use the **Tags 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 resources during creation](#).

The following table describes the Amazon EC2 resources that can be tagged, and the resources that can be tagged on creation using the Amazon EC2 API, the AWS CLI, or an AWS SDK.

Tagging support for Amazon EC2 resources

Resource	Supports tags	Supports tagging on creation
AFI	Yes	Yes
AMI	Yes	Yes
Bundle task	No	No
Capacity Reservation	Yes	Yes
Carrier gateway	Yes	Yes
Client VPN endpoint	Yes	Yes
Client VPN route	No	No
Customer gateway	Yes	Yes
Dedicated Host	Yes	Yes
Dedicated Host Reservation	Yes	Yes
DHCP options	Yes	Yes
EBS snapshot	Yes	Yes
EBS volume	Yes	Yes
EC2 Fleet	Yes	Yes
Egress-only internet gateway	Yes	Yes
Elastic IP address	Yes	Yes
Elastic Graphics accelerator	Yes	No
Instance	Yes	Yes
Instance event window	Yes	Yes

Resource	Supports tags	Supports tagging on creation
Instance store volume	N/A	N/A
Internet gateway	Yes	Yes
IP address pool (BYOIP)	Yes	Yes
Key pair	Yes	Yes
Launch template	Yes	Yes
Launch template version	No	No
Local gateway	Yes	No
Local gateway route table	Yes	No
Local gateway virtual interface	Yes	No
Local gateway virtual interface group	Yes	No
Local gateway route table VPC association	Yes	Yes
Local gateway route table virtual interface group association	Yes	No
NAT gateway	Yes	Yes
Network ACL	Yes	Yes
Network interface	Yes	Yes
Placement group	Yes	Yes
Prefix list	Yes	Yes

Resource	Supports tags	Supports tagging on creation
Reserved Instance	Yes	No
Reserved Instance listing	No	No
Route table	Yes	Yes
Spot Fleet request	Yes	Yes
Spot Instance request	Yes	Yes
Security group	Yes	Yes
Security group rule	Yes	No
Subnet	Yes	Yes
Traffic Mirror filter	Yes	Yes
Traffic Mirror session	Yes	Yes
Traffic Mirror target	Yes	Yes
Transit gateway	Yes	Yes
Transit gateway multicast domain	Yes	Yes
Transit gateway route table	Yes	Yes
Transit gateway VPC attachment	Yes	Yes
Virtual private gateway	Yes	Yes
VPC	Yes	Yes
VPC endpoint	Yes	Yes
VPC endpoint service	Yes	Yes

Resource	Supports tags	Supports tagging on creation
VPC endpoint service configuration	Yes	Yes
VPC flow log	Yes	Yes
VPC peering connection	Yes	Yes
VPN connection	Yes	Yes

You can tag instances, volumes, elastic graphics, network interfaces, and Spot Instance requests on creation using the Amazon EC2 [launch instance wizard](#) in the Amazon EC2 console. You can tag your EBS volumes on creation using the **Volumes** screen, or EBS snapshots using the **Snapshots** screen. Alternatively, use the resource-creating Amazon EC2 APIs (for example, [RunInstances](#)) to apply tags when creating your resource.

You can apply tag-based resource-level permissions in your IAM policies to the Amazon EC2 API actions that support tagging on creation to implement granular control over the users and groups that can tag resources on creation. Your resources are properly secured from creation—tags are applied immediately to your resources, therefore any tag-based resource-level permissions controlling the use of resources are immediately effective. Your resources can be tracked and reported on more accurately. You can enforce the use of tagging on new resources, and control which tag keys and values are set on your resources.

You can also apply resource-level permissions to the `CreateTags` and `DeleteTags` Amazon EC2 API actions in your IAM policies to control which tag keys and values are set on your existing resources. For more information, see [Example: 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 [Work with instance tags in 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.

You can't tag all resources. For more information, see [Tagging support for Amazon EC2 resources](#).

Tags and access management

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags. For more information, see [Grant permission to tag resources during creation](#).

You can also use resource tags to implement attribute-based control (ABAC). You can create IAM policies that allow operations based on the tags for the resource. For more information, see [Control access to EC2 resources using resource tags](#).

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.

Work with tags using the console

You can use the Amazon EC2 console to display the tags of an individual resource, and to apply or remove tags from one resource at a time.

You can use the **Tag Editor** in the AWS Resource Groups console to display the tags of all of your Amazon EC2 resources across all Regions. You can view tags by resource and by resource type, and you can see which resource types are associated with a specified tag. You can apply or remove tags from multiple resources and multiple resource types at a time. The **Tag Editor** provides a central,

unified way to create and manage your tags. For more information, see the [Tagging AWS Resources User Guide](#).

Tasks

- [Display tags](#)
- [Add and delete tags on an individual resource](#)
- [Add and delete tags for multiple resources](#)
- [Add a tag when you launch an instance](#)
- [Filter a list of resources by tag](#)

Display tags

You can display the tags of an individual resource in the Amazon EC2 console. To display the tags of all your resources, use the **Tag Editor** in the AWS Resource Groups console.

Display tags of an individual resource

When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays your Amazon EC2 instances. When you select a resource from one of these lists (for example, an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags by choosing the **Tags** tab.

You can add a column to the resource list to display all values for tags with the same key. You can use this column to sort and filter the resource list by the tag.

New console

To add a column to the resource list to display your tags

1. In the EC2 console, choose the **Preferences** gear-shaped icon in the top right corner of the screen.
2. In the **Preferences** dialog box, for **Tag columns** (at bottom left), select one or more tag keys, and then choose **Confirm**.

Old console

There are two ways to add a new column to the resource list to display your tags:

- On the **Tags** tab, select **Show Column**. A new column is added to the console.
- Choose the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**.

Display tags for multiple resources

You can display tags across multiple resources by using the **Tag Editor** in the [AWS Resource Groups console](#). For more information, see the [Tagging AWS Resources User Guide](#).

Add and delete tags on an individual resource

You can manage tags for an individual resource directly from the resource's page.

To add a tag to an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region where the resource to tag is located. For more information, see [Resource locations](#).
3. In the navigation pane, select a resource type (for example, **Instances**).
4. Select the resource from the resource list and choose the **Tags** tab.
5. Choose **Manage tags**, and then choose **Add new tag**. Enter the key and value for the tag. Choose **Add new tag** again for each additional tag to add. When you are finished adding tags, choose **Save**.

To delete a tag from an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region where the resource to untag is located. For more information, see [Resource locations](#).
3. In the navigation pane, choose a resource type (for example, **Instances**).
4. Select the resource from the resource list and choose the **Tags** tab.
5. Choose **Manage tags**. For each tag to remove, choose **Remove**. When you are finished removing tags, choose **Save**.

Add and delete tags for multiple resources

To add a tag to multiple resources

1. Open the **Tag Editor** in the AWS Resource Groups console at <https://console.aws.amazon.com/resource-groups/tag-editor>.
2. For **Regions**, select one or more Regions where the resources to tag are located.
3. For **Resource types**, select the type of resources to tag (for example, **AWS::EC2::Instance**).
4. Choose **Search resources**.
5. Under **Resource search results**, select the check box next to each resource to tag.
6. Choose **Manage tags of selected resources**.
7. Under **Edit tags of all selected resources**, choose **Add tag**, and then enter the new tag key and value. Choose **Add tag** again for each additional tag to add.

Note

If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

8. Choose **Review and apply tag changes**.
9. Choose **Apply changes to all selected**.

To remove a tag from multiple resources

1. Open the **Tag Editor** in the AWS Resource Groups console at <https://console.aws.amazon.com/resource-groups/tag-editor>.
2. For **Regions**, select the Regions where the resources to untag are located.
3. For **Resource types**, select the type of resources to untag (for example, **AWS::EC2::Instance**).
4. Choose **Search resources**.
5. Under **Resource search results**, select the check box next to each resource to untag.
6. Choose **Manage tags of selected resources**.
7. Under **Edit tags of all selected resources**, next to the tag to remove, choose **Remove tag**.
8. Choose **Review and apply tag changes**.
9. Choose **Apply changes to all selected**.

Add a tag when you launch an instance

New console

To add a tag using the launch instance wizard

1. From the navigation bar, select the Region for the instance. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. Select the Region that meets your needs. For more information, see [Resource locations](#).
2. Choose **Launch instance**.
3. Under **Name and tags**, you can enter a descriptive name for your instance and specify 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, elastic graphics, and network interfaces. For Spot Instances, you can tag the Spot Instance request only.

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.
4. Under **Application and OS Images (Amazon Machine Image)**, choose the operating system (OS) for your instance and an AMI. For more information, see [Application and OS Images \(Amazon Machine Image\)](#).
 5. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or create a new one.
 6. Either keep all the other fields at their default values or choose specific values for your desired instance configuration. For information about the fields, see [Launch an instance using defined parameters](#).
 7. In the **Summary** panel, review your settings, and then choose **Launch instance**.

Old console

To add a tag using the launch instance wizard

1. From the navigation bar, select the Region for the instance. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. Select the Region that meets your needs. For more information, see [Resource locations](#).
2. Choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Select the AMI to use and choose **Select**. For more information, see [Find a Linux AMI](#).
4. On the **Configure Instance Details** page, configure the instance settings as necessary, and then choose **Next: Add Storage**.
5. On the **Add Storage** page, you can specify additional storage volumes for your instance. Choose **Next: Add Tags** when done.
6. On the **Add Tags** page, specify tags for the instance, the volumes, or both. Choose **Add another tag** to add more than one tag to your instance. Choose **Next: Configure Security Group** when you are done.
7. On the **Configure Security Group** page, you can choose from an existing security group that you own, or let the wizard create a new security group for you. Choose **Review and Launch** when you are done.
8. Review your settings. When you're satisfied with your selections, choose **Launch**. Select an existing key pair or create a new one, select the acknowledgment check box, and then choose **Launch Instances**.

Filter a list of resources by tag

You can filter your list of resources based on one or more tag keys and tag values.

To filter a list of resources by tag in the Amazon EC2 console

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 [List and filter your resources](#).

To filter multiple resources across multiple Regions by tag using the Tag Editor

You can use the **Tag Editor** in the AWS Resource Groups console to filter multiple resources across multiple Regions by tag. For more information, see [Finding resources to tag](#) in the *Tagging AWS Resources User Guide*.

Work with tags using the command line

You can add tags to many EC2 resources when you create them, using the tag specifications parameter for the create command. You can view the tags for a resource using the describe command for the resource. You can also add, update, or delete tags for your existing resources using the following commands.

Task	AWS CLI	AWS Tools for Windows PowerShell
Add or overwrite one or more tags	create-tags	New-EC2Tag
Delete one or more tags	delete-tags	Remove-EC2Tag
Describe one or more tags	describe-tags	Get-EC2Tag

Tasks

- [Add tags on resource creation](#)
- [Add tags to an existing resource](#)
- [Describe tagged resources](#)

Add tags on resource creation

The following examples demonstrate how to apply tags when you create resources.

Note

The way you enter JSON-formatted parameters on the command line differs depending on your operating system.

- Linux, macOS, or Unix and Windows PowerShell – Use single quotes (') to enclose the JSON data structure.
- Windows – Omit the single quotes when using the commands with the Windows command line.

For more information, see [Specifying parameter values for the AWS CLI](#).

Example Example: Launch an instance and apply tags to the instance and volume

The following [run-instances](#) command launches an instance and applies a tag with the key **webserver** and the value **production** to the instance. The command also applies a tag with the key **cost-center** and the value **cc123** to any EBS volume that's created (in this case, the root volume).

```
aws ec2 run-instances \  
  --image-id ami-abc12345 \  
  --count 1 \  
  --instance-type t2.micro \  
  --key-name MyKeyPair \  
  --subnet-id subnet-6e7f829e \  
  --tag-specifications  
'ResourceType=instance,Tags=[{Key=webserver,Value=production}]'  
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

You can apply the same tag keys and values to both instances and volumes during launch. The following command launches an instance and applies a tag with a key of **cost-center** and a value of **cc123** to both the instance and any EBS volume that's created.

```
aws ec2 run-instances \  
  --image-id ami-abc12345 \  
  --count 1 \  
  --instance-type t2.micro \  
  --key-name MyKeyPair \  
  --tag-specifications  
'ResourceType=instance,Tags=[{Key=cost-center,Value=cc123}]'  
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```



```
--subnet-id subnet-6e7f829e \  
--tag-specifications 'ResourceType=instance,Tags=[{Key=cost-center,Value=cc123}]' \  
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

Example Example: Create a volume and apply a tag

The following [create-volume](#) command creates a volume and applies two tags: **purpose=production** and **cost-center=cc123**.

```
aws ec2 create-volume \  
--availability-zone us-east-1a \  
--volume-type gp2 \  
--size 80 \  
--tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production}, \  
{Key=cost-center,Value=cc123}]'
```

Add tags to an existing resource

The following examples demonstrate how to add tags to an existing resource using the [create-tags](#) command.

Example Example: Add a tag to a resource

The following command adds the tag **Stack=production** to the specified image, or overwrites an existing tag for the AMI where the tag key is **Stack**. If the command succeeds, no output is returned.

```
aws ec2 create-tags \  
--resources ami-78a54011 \  
--tags Key=Stack,Value=production
```

Example Example: Add tags to multiple resources

This example adds (or overwrites) two tags for an AMI and an instance. One of the tags contains just a key (**webserver**), with no value (we set the value to an empty string). The other tag consists of a key (**stack**) and value (**Production**). If the command succeeds, no output is returned.

```
aws ec2 create-tags \  
--resources ami-1a2b3c4d i-1234567890abcdef0 \  
--tags Key=webserver,Value= Key=stack,Value=Production
```

Example Example: Add tags with special characters

This example adds the tag **[Group]=test** to an instance. The square brackets (**[** and **]**) are special characters, which must be escaped.

If you are using Linux or OS X, to escape the special characters, enclose the element with the special character with double quotes (**"**), and then enclose the entire key and value structure with single quotes (**'**).

```
aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags 'Key="[Group]",Value=test'
```

If you are using Windows, to escape the special characters, enclose the element that has special characters with double quotes (**"**), and then precede each double quote character with a backslash (****) as follows:

```
aws ec2 create-tags ^\  
  --resources i-1234567890abcdef0 ^\  
  --tags Key="\[Group]",Value=test
```

If you are using Windows PowerShell, to escape the special characters, enclose the value that has special characters with double quotes (**"**), precede each double quote character with a backslash (****), and then enclose the entire key and value structure with single quotes (**'**) as follows:

```
aws ec2 create-tags `\  
  --resources i-1234567890abcdef0 `\  
  --tags 'Key="\[Group]",Value=test'
```

Describe tagged resources

The following examples show you how to use filters with the [describe-instances](#) to view instances with specific tags. All EC2 describe commands use this syntax to filter by tag across a single resource type. Alternatively, you can use the [describe-tags](#) command to filter by tag across EC2 resource types.

Example Example: Describe instances with the specified tag key

The following command describes the instances with a **Stack** tag, regardless of the value of the tag.

```
aws ec2 describe-instances \  
  --filters Name=tag-key,Values=Stack
```

Example Example: Describe instances with the specified tag

The following command describes the instances with the tag **Stack=production**.

```
aws ec2 describe-instances \  
  --filters Name=tag:Stack,Values=production
```

Example Example: Describe instances with the specified tag value

The following command describes the instances with a tag with the value **production**, regardless of the tag key.

```
aws ec2 describe-instances \  
  --filters Name=tag-value,Values=production
```

Example Example: Describe all EC2 resources with the specified tag

The following command describes all EC2 resources with the tag **Stack=Test**.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Stack Name=value,Values=Test
```

Work with instance tags in 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.

Topics

- [Allow access to tags in instance metadata](#)
- [Turn off access to tags in instance metadata](#)
- [View if access to tags in instance metadata is allowed](#)
- [Retrieve tags from instance metadata](#)

Allow 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 allow access by using one of the following methods.

To allow access to tags in instance metadata using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, and then choose **Actions, Instance settings, Allow tags in instance metadata**.
4. To allow access to tags in instance metadata, select the **Allow** check box.
5. Choose **Save**.

To allow access to tags in instance metadata at launch using the AWS CLI

Use the [run-instances](#) command and set InstanceMetadataTags to enabled.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type c3.large \  
  ...  
  --metadata-options "InstanceMetadataTags=enabled"
```

To allow access to tags in instance metadata on a running or stopped instance using the AWS CLI

Use the [modify-instance-metadata-options](#) command and set `--instance-metadata-tags` to enabled.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-123456789example \  
  --instance-metadata-tags enabled
```

Turn off access to tags in instance metadata

To turn off access to instance tags in the instance metadata, use one of the following methods. You don't need to turn off access to instance tags on instance metadata at launch because it's turned off by default.

To turn off access to tags in instance metadata using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, and then choose **Actions, Instance settings, Allow tags in instance metadata**.
4. To turn off access to tags in instance metadata, clear the **Allow** check box.
5. Choose **Save**.

To turn off access to tags in instance metadata using the AWS CLI

Use the [modify-instance-metadata-options](#) command and set `--instance-metadata-tags` to disabled.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-123456789example \  
  --instance-metadata-tags disabled
```

View if access to tags in instance metadata is allowed

For each instance, you can use the Amazon EC2 console or AWS CLI to view whether access to instance tags from the instance metadata is allowed.

To view if access to tags in instance metadata is allowed using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**, and then select an instance.
3. On the **Details** tab, check the **Allow tags in instance metadata** field. If the value is **Enabled**, tags in instance metadata is allowed. If the value is **Disabled**, tags in instance metadata is not allowed.

To view if access to tags in instance metadata is allowed using the AWS CLI

Use the [describe-instances](#) command and specify the instance ID.

```
aws ec2 describe-instances \  
  --instance-ids i-1234567890abcdef0
```

The following example output is truncated for space. The "InstanceMetadataTags" parameter indicates whether tags in instance metadata is allowed. If the value is enabled, tags in instance metadata is allowed. If the value is disabled, tags in instance metadata is not allowed.

```
{  
  "Reservations": [  
    {  
      "Groups": [],  
      "Instances": [  
        {  
          "AmiLaunchIndex": 0,  
          "ImageId": "ami-0abcdef1234567890",  
          "InstanceId": "i-1234567890abcdef0",  
          ...  
        }  
      ],  
      "MetadataOptions": {  
        "State": "applied",  
        "HttpTokens": "optional",  
        "HttpPutResponseHopLimit": 1,  
        "HttpEndpoint": "enabled",  
        "HttpProtocolIpv6": "disabled",  
        "InstanceMetadataTags": "enabled"  
      },  
      ...  
    }  
  ],  
  ...  
}
```

Retrieve tags from instance metadata

If instance tags are allowed in the instance metadata, the `tags/instance` category is accessible from the instance metadata. For examples on how to retrieve tags from the instance metadata, see [Get the instance tags for an instance](#).

Add tags to a resource using CloudFormation

With Amazon EC2 resource types, you specify tags using either a `Tags` or `TagSpecifications` property.

The following examples add the tag `Stack=Production` to [AWS::EC2::Instance](#) using its `Tags` property.

Example Example: Tags in YAML

```
Tags:
  - Key: "Stack"
    Value: "Production"
```

Example Example: Tags in JSON

```
"Tags": [
  {
    "Key": "Stack",
    "Value": "Production"
  }
]
```

The following examples add the tag `Stack=Production` to [AWS::EC2::LaunchTemplate](#) [LaunchTemplateData](#) using its `TagSpecifications` property.

Example Example: TagSpecifications in YAML

```
TagSpecifications:
  - ResourceType: "instance"
    Tags:
      - Key: "Stack"
        Value: "Production"
```

Example Example: TagSpecifications in JSON

```
"TagSpecifications": [  
  {  
    "ResourceType": "instance",  
    "Tags": [  
      {  
        "Key": "Stack",  
        "Value": "Production"  
      }  
    ]  
  }  
]
```

Amazon EC2 service quotas

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default *quotas* (also referred to as limits) on these resources on a per-Region basis. For example, there is a maximum number of instances that you can launch in a Region. So if you were to launch an instance in the US West (Oregon) Region, for example, the request must not cause your usage to exceed your maximum number of instances in that Region.

The 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.

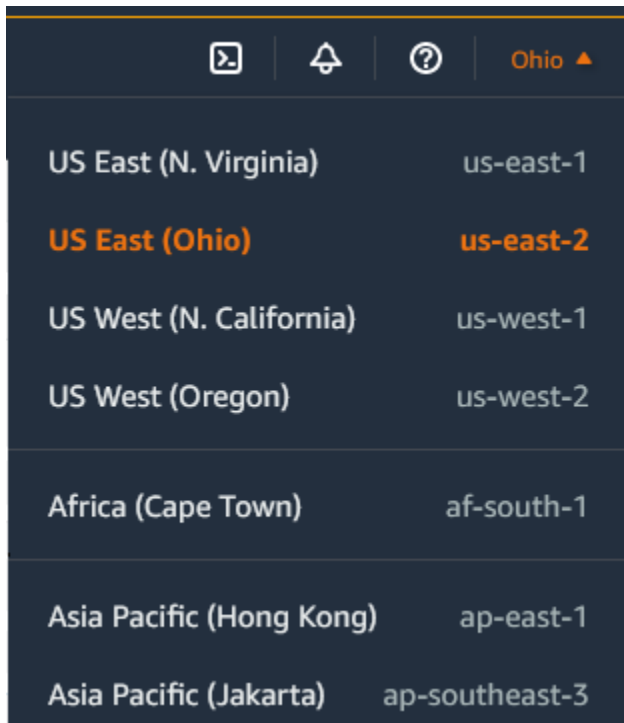
For more information, see [Amazon EC2 endpoints and quotas](#) and [Amazon EBS endpoints and quotas](#) in the *Amazon Web Services General Reference*.

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.



Region	Region Code
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Jakarta)	ap-southeast-3

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. 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 AWS 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

On all instances, Amazon EC2 restricts outbound traffic to public IP addresses over port 25 by default. You can request that this restriction be removed. For more information, see [How do I remove the restriction on port 25 from my Amazon EC2 instance or AWS Lambda function?](#) on AWS re:Post.

Note

This restriction does not apply to outbound traffic sent over port 25 to:

- 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](#).

Amazon EC2 usage reports

AWS provides a free reporting tool called AWS Cost Explorer that enables you to analyze the cost and usage of your EC2 instances and the usage of your Reserved Instances. You can view data up to the last 12 months, and forecast how much you are likely to spend for the next three months. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs. You also can specify time ranges for the data, and view time data by day or by month.

Here's an example of some of the questions that you can answer when using Cost Explorer:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?

- How is my instance usage distributed across AWS accounts?
- How well am I using my Reserved Instances?
- Are my Reserved Instances helping me save money?

For more information about working with reports in Cost Explorer, including saving reports, see [Analyzing your costs with AWS Cost Explorer](#) in the *AWS Cost Management User Guide*.

Track your Free Tier usage

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.

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).

EC2 Free Tier [Info](#)

Offers for all AWS Regions.

3 EC2 free tier offers in use


End of month forecast
⚠️ 2 offers forecasted to exceed free tier limit.

Exceeds free tier
⚠️ 1 offers exceeded and is now pay-as-you-go pricing.

[View Global EC2 resources](#)

Offer usage (monthly)


Windows EC2 Instances	<div style="width: 12%;"><div style="width: 12%;"></div></div>	12%
662 hours remaining		
Linux EC2 Instances	<div style="width: 100%;"><div style="width: 100%;"></div></div>	100%
⚠️ Offer limit reached		
Storage space on EBS	<div style="width: 85%;"><div style="width: 85%;"></div></div>	85%
4.59 GB remaining		

[View all AWS Free Tier offers](#) 

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, go to the next step in this tutorial.
 - 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 [Amazon EC2 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 [Using the AWS Free Tier](#) in the *AWS Billing User Guide*.

Troubleshoot EC2 instances

The following documentation can help you troubleshoot problems that you might have with your instance.

Contents

- [Troubleshoot instance launch issues](#)
- [Troubleshoot connecting to your instance](#)
- [Troubleshoot stopping your instance](#)
- [Troubleshoot instance termination \(shutting down\)](#)
- [Troubleshoot instances with failed status checks](#)
- [Troubleshoot an unreachable instance](#)
- [Boot from the wrong volume](#)
- [Use EC2Rescue for Linux](#)
- [EC2 Serial Console for Linux instances](#)
- [Send a diagnostic interrupt \(for advanced users\)](#)

For additional help with Windows instances, see [Troubleshoot Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Troubleshoot instance launch issues

The following issues prevent you from launching an instance.

Launch Issues

- [Invalid device name](#)
- [Instance limit exceeded](#)
- [Insufficient instance capacity](#)
- [The requested configuration is currently not supported. Please check the documentation for supported configurations.](#)
- [Instance terminates immediately](#)
- [Insufficient permissions](#)

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_id --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 [Change the instance type](#).
- If you are launching instances into a cluster placement group, you can get an insufficient capacity error. For more information, see [Work with placement groups](#).

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 [Instance volume limits](#).
- An EBS snapshot is corrupted.
- The root EBS volume is encrypted and you do not have permissions to access the KMS key for decryption.
- A snapshot specified in the block device mapping for the AMI is encrypted and you do not have permissions to access the KMS key for decryption or you do not have access to the KMS key to encrypt the restored volumes.
- The instance store-backed AMI that you used to launch the instance is missing a required part (an `image.part.xx` file).

For more information, get the termination reason using one of the following methods.

To get the termination reason using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. On the first tab, find the reason next to **State transition reason**.

To get the termination reason using the AWS Command Line Interface

1. Use the [describe-instances](#) command and specify the instance ID.

```
aws ec2 describe-instances --instance-id instance_id
```

2. Review the JSON response returned by the command and note the values in the StateReason response element.

The following code block shows an example of a StateReason response element.

```
"StateReason": {  
  "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",  
  "Code": "Server.InternalError"  
},
```

To 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 can I decode an authorization failure message after receiving an "UnauthorizedOperation" error during an EC2 instance launch?](#)

Troubleshoot connecting to your instance

The following information and common errors can help you troubleshoot connecting to your Linux instance.

To troubleshoot connecting to Windows instances, see [Troubleshoot Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Connection problems and errors

- [Common causes for connection issues](#)
- [Error connecting to your instance: Connection timed out](#)
- [Error: unable to load key ... Expecting: ANY PRIVATE KEY](#)
- [Error: User key not recognized by server](#)
- [Error: Permission denied or connection closed by \[instance\] port 22](#)
- [Error: Unprotected private key file](#)
- [Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"](#)
- [Error: Server refused our key or No supported authentication methods available](#)
- [Cannot ping instance](#)
- [Error: Server unexpectedly closed network connection](#)
- [Error: Host key validation failed for EC2 Instance Connect](#)
- [Can't connect to Ubuntu instance using EC2 Instance Connect](#)
- [I've lost my private key. How can I connect to my Linux instance?](#)

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 users on your Linux instance](#).

- **Get the default username for the AMI that you used to launch your instance:**

AMI used to launch instance	Default username
AL2023	ec2-user
Amazon Linux 2	
Amazon Linux	
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 so that you can connect to it. Check your instance to make sure it is running and has passed its status checks.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Verify 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 the two status checks.

Verify that you've satisfied all prerequisites to connect

Ensure that you have all the information you need to connect. For more information, see [Connect to your Linux instance](#).

For prerequisites specific to connection types, such as SSH, EC2 Instance Connect, OpenSSH, PuTTY, and more, see the following options.

Linux or macOS X

If your local computer operating system is Linux or macOS X, check the specific prerequisites for the following connection options:

- [SSH client](#)
- [EC2 Instance Connect](#)
- [AWS Systems Manager Session Manager](#)

Windows

If your local computer operating system is Windows, check the specific prerequisites for the following connection options:

- [OpenSSH](#)
- [PuTTY](#)
- [AWS Systems Manager Session Manager](#)

- [Windows Subsystem for Linux](#)

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.
 - For Linux instances: Verify that there is a rule that allows traffic from your local computer to port 22 (SSH).
 - For Windows instances: Verify that there is a rule that allows traffic from your local computer to port 3389 (RDP).

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-xxxxxxx) to navigate to the route table.
 - b. On the **Routes** tab, choose **Edit routes**. Choose **Add route**, use `0.0.0.0/0` as the destination and the internet gateway as the target. For IPv6, choose **Add route**, use `::/0` as the destination and the internet gateway as the target.
 - c. Choose **Save routes**.

Check the network access control list (ACL) for the subnet.

The network ACLs must allow inbound traffic from your local IP address on port 22 (for Linux instances) or port 3389 (for Windows instances). 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 (for Linux instances) or port 3389 (for Windows instances).

If you have a firewall on your computer, verify that it allows inbound and outbound traffic from your computer on port 22 (for Linux instances) or port 3389 (for Windows instances).

Check that your instance has a public IPv4 address.

If not, you can associate an Elastic IP address with your instance. For more information, see [Elastic IP addresses](#).

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 [Change the instance type](#).

To connect to your instance using an IPv6 address, check the following:

- Your subnet must be associated with a route table that has a route for IPv6 traffic (: : /0) to an internet gateway.
- Your security group rules must allow inbound traffic from your local IPv6 address on the proper port (22 for Linux and 3389 for Windows).
- Your network ACL rules must allow inbound and outbound IPv6 traffic.

- If you launched your instance from an older AMI, it 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 Linux 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 from Windows with 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 user name for your AMI. Enter the user name in the **Host name** box in the **PuTTY Configuration** window.

AMI used to launch instance	Default username
AL2023	ec2-user
Amazon Linux 2	
Amazon Linux	
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 user name for your AMI *and* that you have specified the proper private key (.pem) file for your instance.

The appropriate user names are as follows:

AMI used to launch instance	Default username
AL2023	ec2-user
Amazon Linux 2	
Amazon Linux	
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 Linux 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 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 via 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 via 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: Server refused our key *or* No supported authentication methods available

If you use PuTTY to connect to your instance and get either of the following errors, Error: Server refused our key or Error: No supported authentication methods available, verify that you are connecting with the appropriate user name for your AMI. Type the user name in **User name** in the **PuTTY Configuration** window.

The appropriate user names are as follows:

AMI used to launch instance	Default username
AL2023	ec2-user
Amazon Linux 2	
Amazon Linux	
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

AMI used to launch instance	Default username
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 from Windows with 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 user name of the AMI that you used to launch your instance. For Amazon Linux 2, the default user name is `ec2-user`.

For example, if your instance was launched using Amazon Linux 2, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see [Connect to your Linux instance from Linux or macOS using SSH](#).

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 user name of the AMI that you used to launch your instance. For Ubuntu, the default user name 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 from Linux or macOS using SSH](#).

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 Linux 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 [Instance lifecycle](#).

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

New console

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 Linux 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**.

Old console

Choose **Launch instances**, and then use the launch wizard to launch a *temporary* instance with the following options:

- On the **Choose an AMI** page, select the same AMI that you used to launch the original instance. If this AMI is unavailable, you can create an AMI that you can use from the stopped instance. For more information, see [Create an Amazon EBS-backed Linux AMI](#).

- On the **Choose an Instance Type** page, leave the default instance type that the wizard selects for you.
- On the **Configure Instance Details** page, specify the same Availability Zone as the original instance. If you're launching an instance in a VPC, select a subnet in this Availability Zone.
- On the **Add Tags** page, add the tag Name=Temporary to the instance to indicate that this is a temporary instance.
- On the **Review** page, choose **Launch**. Choose the key pair that you created in Step 1, then choose **Launch Instances**.

Step 5: Detach the root volume from the original instance and attach it to the temporary instance

1. In the navigation pane, choose **Volumes** and select the root device volume for the original instance (you 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 user name `ec2-user`. You might need to substitute a different user name, such as `ubuntu` for Ubuntu instances.

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/  
authorized_keys
```

If this copy succeeded, you can go to the next step.

(Optional) Otherwise, if you don't have permission to edit files in `/mnt/tempvol`, you must update the file using **sudo** and then check the permissions on the file to verify that you are able to log into the original instance. Use the following command to check the permissions on the file.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh  
total 4  
-rw----- 1 222 500 398 Sep 13 22:54 authorized_keys
```

In this example output, *222* is the user ID and *500* is the group ID. Next, use **sudo** to re-run the copy command that failed.

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/  
authorized_keys
```

Run the following command again to determine whether the permissions changed.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

If the user ID and group ID have changed, use the following command to restore them.

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance

1. From the temporary instance, unmount the volume that you attached so that you can reattach it to the original instance. For example, use the following command to unmount the volume at `/mnt/tempvol`.

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

2. Detach the volume from the temporary instance (you unmounted it in the previous step): From the Amazon EC2 console, 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 instance**.

Troubleshoot stopping your instance

If you have stopped your Amazon EBS-backed instance and it appears stuck in the stopping state, there may be an issue with the underlying host computer.

There is no cost for instance usage while an instance is in the stopping state or in any other state except running. You are only charged for instance usage when an instance is in the running state.

Force stop the instance

Force the instance to stop using either the console or the AWS CLI.

Note

You can force an instance to stop using the console only while the instance is in the stopping state. You can force an instance to stop using the AWS CLI while the instance is in any state, except shutting-down and terminated.

Console

To force stop the instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance state, Force stop instance, Stop**.

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.

AWS CLI

To force stop the instance using the AWS CLI

Use the [stop-instances](#) command and the `--force` option as follows:

```
aws ec2 stop-instances --instance-ids i-0123ab456c789d01e --force
```

If, after 10 minutes, the instance has not stopped, post a request for help 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](#).

Create a replacement instance

To attempt to resolve the problem while you are waiting for assistance from [AWS re:Post](#) or the [Support Center](#), create a replacement instance. Create an AMI of the stuck instance, and launch a new instance using the new AMI.

Important

Creating a replacement instance is recommended if it's registering [system status checks](#) only, as instance status checks will result in the AMI copying over an exact replica of the broken OS. Once you've confirmed the status message, create the AMI and launch a new instance using the new AMI.

Console

To create a replacement instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Actions, Image and templates, Create image**.
4. On the **Create image** page, do the following:
 - a. Enter a name and description for the AMI.
 - b. Choose **No reboot**.
 - c. Choose **Create image**.

For more information, see [Create a Linux 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 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 using the CLI

1. Create an AMI from the stuck instance using the [create-image](#) (AWS CLI) command and the `--no-reboot` option as follows:

```
aws ec2 create-image --instance-id i-0123ab456c789d01e --name "AMI" --  
description "AMI for replacement instance" --no-reboot
```

2. Launch a new instance from the AMI using the [run-instances](#) (AWS CLI) command as follows:

```
aws ec2 run-instances --image-id ami-1a2b3c4d --count 1 --instance-type c3.large  
--key-name MyKeyPair --security-groups MySecurityGroup
```

3. Verify that the new instance is working.
4. Terminate the stuck instance using the [terminate-instances](#) (AWS CLI) command as follows:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

If you are unable to create an AMI from the instance as described in the previous procedure, you can set up a replacement instance as follows:

(Alternate) To create a replacement instance using the console

1. Select the instance and choose **Description, Block devices**. Select each volume and make note of its volume ID. Be sure to note which volume is the root volume.
2. In the navigation pane, choose **Volumes**. Select each volume for the instance, and choose **Actions, Create Snapshot**.
3. In the navigation pane, choose **Snapshots**. Select the snapshot that you just created, and choose **Actions, Create Volume**.
4. Launch an instance with the same operating system as the stuck instance. Note the volume ID and device name of its root volume.
5. In the navigation pane, choose **Instances**, select the instance that you just launched, and choose **Instance state, Stop instance**.
6. In the navigation pane, choose **Volumes**, select the root volume of the stopped instance, and choose **Actions, Detach Volume**.
7. Select the root volume that you created from the stuck instance, choose **Actions, Attach Volume**, and attach it to the new instance as its root volume (using the device name that you made note of). Attach any additional non-root volumes to the instance.
8. In the navigation pane, choose **Instances** and select the replacement instance. Choose **Instance state, Start instance**. Verify that the instance is working.
9. Select the stuck instance, choose **Instance state, Terminate instance**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

Troubleshoot instance termination (shutting down)

You are not billed for any instance usage while an instance is not in the `running` state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to `shutting-down`.

Instance terminates immediately

Several issues can cause your instance to terminate immediately on start-up. See [Instance terminates immediately](#) for more information.

Delayed instance termination

If your instance remains in the shutting-down state longer than a few minutes, it might be delayed due to shutdown scripts being run by the instance.

Another possible cause is a problem with the underlying host computer. If your instance remains in the shutting-down state for several hours, Amazon EC2 treats it as a stuck instance and forcibly terminates it.

If it appears that your instance is stuck terminating and it has been longer than several hours, post a request for help to [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](#).

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 *instance_id* 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. For more information, see [Enable termination protection](#).

You must disable termination protection before you can terminate the instance.

To disable termination protection using the Amazon EC2 console, select the instance and then choose **Actions, Instance Settings, Change Termination Protection**.

To disable termination protection using the AWS CLI, use the following command.


```
$ aws ec2 modify-instance-attribute --instance-id instance_id --no-disable-api-termination
```

Instances automatically launched or terminated

Generally, the following behaviors mean that you've used Amazon EC2 Auto Scaling, EC2 Fleet, or Spot Fleet to scale your computing resources automatically based on criteria that you've defined:

- You terminate an instance and a new instance launches automatically.
- You launch an instance and one of your instances terminates automatically.
- You stop an instance and it terminates and a new instance launches automatically.

To stop automatic scaling, see the [Amazon EC2 Auto Scaling User Guide](#), [EC2 Fleet](#), or [Create a Spot Fleet request](#).

Troubleshoot instances with failed status checks

The following information can help you troubleshoot issues if your instance fails a status check. First determine whether your applications are exhibiting any problems. If you verify that the instance is not running your applications as expected, review the status check information and the system logs.

For examples of problems that can cause status checks to fail, see [Status checks for your instances](#).

Contents

- [Review status check information](#)
- [Retrieve the system logs](#)
- [Troubleshoot system log errors for Linux-based instances](#)
- [Out of memory: kill process](#)
- [ERROR: mmu_update failed \(Memory management update failed\)](#)
- [I/O error \(block device failure\)](#)
- [I/O ERROR: neither local nor remote disk \(Broken distributed block device\)](#)
- [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 \(EC2 incompatible kernel\)](#)
- [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... \(Missing device\)](#)
- [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. In the details pane, choose **Status and alarms** to see the individual results for all **System status checks** and **Instance status checks**.

If a system status check has failed, you can try one of the following options:

- Create an instance recovery alarm. For more information, see [Create alarms that stop, terminate, reboot, or recover an instance](#).

- If you changed the instance type to an [instance built on the AWS Nitro System](#), 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 instance using an Amazon EBS-backed AMI, stop and restart the instance.
- For an instance using an instance-store backed AMI, terminate the instance and launch a replacement.
- Wait for Amazon EC2 to resolve the issue.
- Post your issue to [AWS re:Post](#).
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service automatically launches a replacement instance. For more information, see [Health Checks for Auto Scaling Instances](#) in the *Amazon EC2 Auto Scaling User Guide*.
- Retrieve the system log and look for errors.

Retrieve the system logs

If an instance status check fails, you can reboot the instance and retrieve the system logs. The logs may reveal an error that can help you troubleshoot the issue. Rebooting clears unnecessary information from the logs.

To reboot an instance and retrieve the system log

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select your instance.
3. Choose **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-based instances

For Linux-based instances that have failed an instance status check, such as the instance reachability check, verify that you followed the steps above to retrieve the system log. The following list contains some common system log errors and suggested actions you can take to resolve the issue for each error.

Memory Errors

- [Out of memory: kill process](#)
- [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	<p>Do one of the following:</p> <ul style="list-style-type: none">• Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, a larger or a memory-optimized instance type.• Reboot the instance to return it to a non-impaired status. The problem will probably

For this instance type	Do this
	occur again unless you change the instance type.
Instance store-backed	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Terminate the instance and launch a new instance, specifying a different instance type. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.

ERROR: mmu_update failed (Memory management update failed)

Memory management update failures are indicated by a system log entry similar to the following:

```

...
Press `ESC' to enter the menu... 0  [H[J] Booting 'Amazon Linux 2011.09
(2.6.35.14-95.38.amzn1.i686)'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=
en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22

```

Potential cause

Issue with Amazon Linux

Suggested action

Post your issue to the [Developer Forums](#) or contact [AWS Support](#).

I/O error (block device failure)




An input/output error is indicated by a system log entry similar to the following example:

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Attempt to recover the volume. <div data-bbox="867 611 1507 921"><p> Note</p><p>It's good practice to snapshot your Amazon EBS volumes often. This dramatically decreases the risk of data loss as a result of failure.</p></div> <ol style="list-style-type: none">4. Re-attach the volume to the instance.5. Start the instance.
Instance store-backed	<p>Terminate the instance and launch a new instance.</p> <div data-bbox="829 1203 1507 1421"><p> Note</p><p>Data cannot be recovered. Recover from backups.</p></div> <div data-bbox="829 1488 1507 1845"><p> Note</p><p>It's a good practice to use either Amazon S3 or Amazon EBS for backups. Instance store volumes are directly tied to single host and single disk failures.</p></div>

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 I/O failed in request_timer_fn. Detaching...

Aborting journal on device drbd1-8.

block drbd1: I/O 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	<p>Use a newer kernel, either GRUB-based or static, using one of the following options:</p> <p>Option 1: Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters.</p> <p>Option 2:</p> <ol style="list-style-type: none"> 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 <code>-kernel</code> and <code>-ramdisk</code> parameters.

"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

Incompatible kernel and userland

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none"> 1. Stop the instance. 2. Modify the configuration to use a newer kernel. 3. Start the instance.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none"> 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: <ol style="list-style-type: none"> 1. Select corrected ramdisk for the Amazon EBS volume. 2. Stop the instance. 3. Detach the volume and repair it. 4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none"> 1. Terminate the instance and launch a new instance with the correct ramdisk. 2. Create a new AMI with the correct ramdisk.

ERROR Invalid kernel (EC2 incompatible kernel)

This condition is indicated by a system log similar to the one shown below.

```
...
root (hd0)

  Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

  Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

Potential causes

One or both of the following conditions can cause this problem:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure:

For this instance type	Do this
	<ol style="list-style-type: none"> 1. Stop the instance. 2. Replace with working kernel. 3. Install a fallback kernel. 4. Modify the AMI by correcting the kernel.
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> 1. Terminate the instance and launch a new instance with the correct kernel. 2. Create an AMI with the correct kernel. 3. (Optional) Seek technical assistance for data recovery using AWS Support.

fsck: No such file or directory while trying to open... (File system not found)

This condition is indicated by a system log similar to the one shown below.

```

Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [ OK ]

Starting udev: [ OK ]

Setting hostname localhost: [ OK ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[ OK ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

```

```
/dev/sdh:
```

```
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
```

```
    e2fsck -b 8193 <device>
```

```
[FAILED]
```

```
*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

Potential causes

- A bug exists in ramdisk filesystem definitions `/etc/fstab`
- Misconfigured filesystem definitions in `/etc/fstab`
- Missing/failed drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: <ol style="list-style-type: none">1. Stop the instance, detach the root volume, repair/modify <code>/etc/fstab</code> the volume, attach the volume to the instance, and start the instance.2. Fix ramdisk to include modified <code>/etc/fstab</code> (if applicable).3. Modify the AMI to use a newer ramdisk.

For this instance type	Do this
	<p>The sixth field in the <code>fstab</code> defines availability requirements of the mount – a nonzero value implies that an <code>fsck</code> will be done on that volume and <i>must</i> succeed. Using this field can be problematic in Amazon EC2 because a failure typically results in an interactive console prompt that is not currently available in Amazon EC2. Use care with this feature and read the Linux man page for <code>fstab</code>.</p>
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Terminate the instance and launch a new instance.2. Detach any errant Amazon EBS volumes and the reboot instance.3. (Optional) Seek technical assistance for data recovery using AWS Support.

General error mounting filesystems (failed mount)

This condition is indicated by a system log similar to the one shown below.

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds
```

```
EXT3-fs: mounted filesystem with ordered data mode.
```

```
Setting up other filesystems.
```

```
Setting up new root fs
```

```
no fstab.sys, mounting internal defaults
```

```
Switching to new root and running init.
```

```
unmounting old /dev
```

```
unmounting old /proc
```

```
unmounting old /sys
```

```
mountall:/proc: unable to mount: Device or resource busy
```

```
mountall:/proc/self/mountinfo: No such file or directory
```

```
mountall: root filesystem isn't mounted
```

```
init: mountall main process (221) terminated with status 1
```

General error mounting filesystems.

A maintenance shell will now be started.

CONTROL-D will terminate this shell and re-try.

Press enter for maintenance

(or type Control-D to continue):

Potential causes

Instance type	Potential cause
Amazon EBS-backed	<ul style="list-style-type: none"> • Detached or failed Amazon EBS volume. • Corrupted filesystem. • Mismatched ramdisk and AMI combination (such as Debian ramdisk with a SUSE AMI).
Instance store-backed	<ul style="list-style-type: none"> • A failed drive. • A corrupted file system. • A mismatched ramdisk and combination (for example, a Debian ramdisk with a SUSE AMI).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the root volume.3. Attach the root volume to a known working instance.4. Run filesystem check (<code>fsck -a /dev/...</code>).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	<p>Try one of the following:</p> <ul style="list-style-type: none">• Start a new instance.• (Optional) Seek technical assistance for data recovery using AWS Support.

VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line: root=/dev/sda1 ro 4
...
Registering block device major 8
...
```

Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)

Potential causes

Instance type	Potential cause
Amazon EBS-backed	<ul style="list-style-type: none">• Device not attached correctly.• Root device not attached at correct device point.• Filesystem not in expected format.• Use of legacy kernel (such as 2.6.16-XenU).• A recent kernel update on your instance (faulty update, or an update bug)
Instance store-backed	Hardware device failure.

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Stop and then restart the instance.• Modify root volume to attach at the correct device point, possible <code>/dev/sda1</code> instead of <code>/dev/sda</code>.• 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>udev[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: <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Fix the device mapping problem.

For this instance type	Do this
	<ol style="list-style-type: none"> 4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> 1. Create a new AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance from the AMI you created.

XENBUS: Device with no driver...

This condition is indicated by a system log similar to the one shown below.

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udev[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: <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Fix the device mapping problem.4. Start the instance.5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: <ol style="list-style-type: none">1. Create an AMI with the appropriate fix (map block device correctly).2. Terminate the instance and launch a new instance using the AMI you created.

... days without being checked, check forced (File system check required)

This condition is indicated by a system log similar to the one shown below.

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

Potential causes

Filesystem check time passed; a filesystem check is being forced.

Suggested actions

- Wait until the filesystem check completes. A filesystem check can take a long time depending on the size of the root filesystem.
- Modify your filesystems to remove the filesystem check (fsck) enforcement using tune2fs or tools appropriate for your filesystem.

fsck died with exit status... (Missing device)

This condition is indicated by a system log similar to the one shown below.

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

Potential causes

- Ramdisk looking for missing drive
- Filesystem consistency check forced
- Drive failed or detached

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Try one or more of the following to resolve the issue:</p> <ul style="list-style-type: none">• Stop the instance, attach the volume to an existing running instance.• Manually run consistency checks.• Fix ramdisk to include relevant utilities.

For this instance type	Do this
	<ul style="list-style-type: none">• Modify filesystem tuning parameters to remove consistency requirements (not recommended).
Instance store-backed	<p>Try one or more of the following to resolve the issue:</p> <ul style="list-style-type: none">• Rebundle ramdisk with correct tooling.• Modify file system tuning parameters to remove consistency requirements (not recommended).• Terminate the instance and launch a new instance.• (Optional) Seek technical assistance for data recovery using AWS Support.

GRUB prompt (grubdom>)

This condition is indicated by a system log similar to the one shown below.

```
GNU GRUB version 0.97 (629760K lower / 0K upper memory)
```

```
[ Minimal BASH-like line editing is supported. For  
the first word, TAB lists possible command  
completions. Anywhere else TAB lists the possible  
completions of a device/filename. ]
```

```
grubdom>
```


Potential causes

Instance type	Potential causes
Amazon EBS-backed	<ul style="list-style-type: none"> • Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).
Instance store-backed	<ul style="list-style-type: none"> • Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Option 1: Modify the AMI and relaunch the instance:</p> <ol style="list-style-type: none"> 1. Modify the source AMI to create a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.

For this instance type	Do this
	<ol style="list-style-type: none">Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition).Terminate the instance and launch a new one using the AMI that you created. <p>Option 2: Fix the existing instance:</p> <ol style="list-style-type: none">Stop the instance.Detach the root filesystem.Attach the root filesystem to a known working instance.Mount filesystem.Create a GRUB configuration file.Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.Detach filesystem.Attach to the original instance.Modify kernel attribute to use the appropriate GRUB image (1st disk or 1st partition on 1st disk).Start the instance.

For this instance type	Do this
Instance store-backed	<p>Option 1: Modify the AMI and relaunch the instance:</p> <ol style="list-style-type: none">1. Create the new AMI with a GRUB configuration file at the standard location (/boot/grub/menu.lst).2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition).3. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.4. Terminate the instance and launch a new instance using the AMI you created. <p>Option 2: Terminate the instance and launch a new instance, specifying the correct kernel.</p> <div data-bbox="829 1045 1507 1262"><p> Note</p><p>To recover data from the existing instance, contact AWS Support.</p></div>

Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)

This condition is indicated by a system log similar to the one shown below.

```
...
Bringing up loopback interface: [ OK ]

Bringing up interface eth0: Device eth0 has different MAC address than expected,
ignoring.
[FAILED]
```

```
Starting auditd: [ OK ]
```

Potential causes

There is a hardcoded interface MAC in the AMI configuration

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Modify the AMI to remove the hardcoding and relaunch the instance.• Modify the instance to remove the hardcoded MAC address. <p>OR</p> <p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the root volume.3. Attach the volume to another instance and modify the volume to remove the hardcoded MAC address.4. Attach the volume to the original instance.5. Start the instance.
Instance store-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Modify the instance to remove the hardcoded MAC address.• Terminate the instance and launch a new instance.

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)

This condition is indicated by a system log similar to the one shown below.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

SELinux has been enabled in error:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the failed instance.2. Detach the failed instance's root volume.3. Attach the root volume to another running Linux instance (later referred to as a recovery instance).4. Connect to the recovery instance and mount the failed instance's root volume.5. Disable SELinux on the mounted root volume. This process varies across Linux distributions; for more information, consult your OS-specific documentation.

For this instance type	Do this
	<div data-bbox="867 210 1507 667" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>Note</p> <p>On some systems, you disable SELinux by setting <code>SELINUX=disabled</code> in the <code>/mount_point/etc/sysconfig/selinux</code> file, where <code>mount_point</code> is the location that you mounted the volume on your recovery instance.</p> </div> <ol style="list-style-type: none"> 6. Unmount and detach the root volume from the recovery instance and reattach it to the original instance. 7. Start the instance.
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> 1. Terminate the instance and launch a new instance. 2. (Optional) Seek technical assistance for data recovery using AWS Support.

XENBUS: Timeout connecting to devices (Xenbus timeout)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to kernel.
```

Potential causes

- The block device is not connected to the instance
- This instance is using an old instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: <ul style="list-style-type: none">• Modify the AMI and instance to use a modern kernel and relaunch the instance.• Reboot the instance.
Instance store-backed	Do one of the following: <ul style="list-style-type: none">• Terminate the instance.• Modify the AMI to use a modern kernel, and launch a new instance using this AMI.

Troubleshoot an unreachable instance

You can use the following methods to troubleshoot an unreachable Linux instance. For information about troubleshooting an unreachable Windows instance, see [Troubleshoot an unreachable instance](#) in the *EC2 User Guide for Windows*.

Contents

- [Instance reboot](#)
- [Instance console output](#)
- [Capture a screenshot of an unreachable instance](#)
- [Instance recovery when a host computer fails](#)

Instance reboot

The ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

Just as you can reset a computer by pressing the reset button, you can reset EC2 instances using the Amazon EC2 console, CLI, or API. For more information, see [Reboot your instance](#).

Instance console output

Console output is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started.

For Linux/Unix, the instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. The console output returns buffered information that was posted shortly after an instance transition state (start, stop, reboot, and terminate). The posted output is not continuously updated; only when it is likely to be of the most value.

You can optionally retrieve the latest serial console output at any time during the instance lifecycle. This option is only supported on [instances built on the AWS Nitro System](#). It is not supported through the Amazon EC2 console.

Note

Only the most recent 64 KB of posted output is stored, which is available for at least 1 hour after the last posting.

Only the instance owner can access the console output.

Use one of the following methods to get console output.

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 and then choose **Actions, Monitor and troubleshoot, Get system log**.

Command line

To get console output

You can use one of the following commands. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [get-console-output](#) (AWS CLI)
- [Get-EC2ConsoleOutput](#) (AWS Tools for Windows PowerShell)

For more information about common system log errors, see [Troubleshoot system log errors for Linux-based instances](#).

Capture a screenshot of an unreachable instance

If you are unable to reach your instance via SSH, 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

Supported Regions

This feature is available in the following Regions:

- US East (N. Virginia) Region
- US East (Ohio) Region
- US West (N. California) Region
- US West (Oregon) Region
- Africa (Cape Town) Region
- Asia Pacific (Hong Kong) Region
- Asia Pacific (Hyderabad) Region
- Asia Pacific (Jakarta) Region
- Asia Pacific (Melbourne) Region
- Asia Pacific (Mumbai) Region
- Asia Pacific (Osaka) Region
- Asia Pacific (Seoul) Region
- Asia Pacific (Singapore) Region
- Asia Pacific (Sydney) Region
- Asia Pacific (Tokyo) Region
- Canada (Central) Region
- Canada West (Calgary) Region
- China (Beijing) Region
- China (Ningxia) Region
- Europe (Frankfurt) Region
- Europe (Ireland) Region
- Europe (London) Region
- Europe (Milan) Region
- Europe (Paris) Region
- Europe (Spain) Region
- Europe (Stockholm) Region
- Europe (Zurich) Region
- Israel (Tel Aviv) Region
- South America (São Paulo) Region

- Middle East (Bahrain) Region
- Middle East (UAE) Region

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.

Command line

To capture a screenshot of an instance

You can use one of the following commands. The returned content is base64-encoded. For more information about these command line interfaces, see [Access Amazon EC2](#).

- [get-console-screenshot](#) (AWS CLI)
- [GetConsoleScreenshot](#) (Amazon EC2 Query API)

Instance recovery when a host computer fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You are notified of such an event ahead of time by email.

To recover an Amazon EBS-backed instance running on a host computer that failed

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.
4. Restore any important data.

For more information, see [Stop and start 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.

For more information, see [Create an instance store-backed Linux AMI](#).

Boot from the wrong volume

In some situations, you may find that a volume other than the volume attached to `/dev/xvda` or `/dev/sda` has become the root volume of your instance. This can happen when you have attached the root volume of another instance, or a volume created from the snapshot of a root volume, to an instance with an existing root volume.

This is due to how the initial ramdisk in Linux works. It chooses the volume defined as `/` in the `/etc/fstab`, and in some distributions, this is determined by the label attached to the volume partition. Specifically, you find that your `/etc/fstab` looks something like the following:

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

If you check the label of both volumes, you see that they both contain the `/` label:

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

In this example, you could end up having `/dev/xvdf1` become the root device that your instance boots to after the initial ramdisk runs, instead of the `/dev/xvda1` volume from which you

had intended to boot. To solve this, use the same **e2label** command to change the label of the attached volume that you do not want to boot from.

In some cases, specifying a UUID in `/etc/fstab` can resolve this. However, if both volumes come from the same snapshot, or the secondary is created from a snapshot of the primary volume, they share a UUID.

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

To change the label of an attached ext4 volume

1. Use the **e2label** command to change the label of the volume to something other than `/`.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1 old/
```

2. Verify that the volume has the new label.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1
old/
```

To change the label of an attached xfs volume

- Use the **xfs_admin** command to change the label of the volume to something other than `/`.

```
[ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
writing all SBs
new label = "old/"
```

After changing the volume label as shown, you should be able to reboot the instance and have the proper volume selected by the initial ramdisk when the instance boots.

Important

If you intend to detach the volume with the new label and return it to another instance to use as the root volume, you must perform the above procedure again and change

the volume label back to its original value. Otherwise, the other instance does not boot because the ramdisk is unable to find the volume with the label /.

Use EC2Rescue for Linux

EC2Rescue for Linux is an easy-to-use, open-source tool that can be run on an Amazon EC2 Linux instance to diagnose and troubleshoot common issues using its library of over 100 modules. A few generalized use cases for EC2Rescue for Linux include gathering syslog and package manager logs, collecting resource utilization data, and diagnosing/remediating known problematic kernel parameters and common OpenSSH issues.

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 [AWS Support-TroubleshootSSH](#).

If you are using a Windows instance, see [EC2Rescue for Windows Server](#).

Contents

- [Install EC2Rescue for Linux](#)
- [Work with EC2Rescue for Linux](#)
- [Develop EC2Rescue modules](#)

Install EC2Rescue for Linux

The EC2Rescue for Linux tool can be installed on an Amazon EC2 Linux instance that meets the following prerequisites.

Prerequisites

- Supported operating systems:
 - Amazon Linux 2
 - Amazon Linux 2016.09+
 - SUSE Linux Enterprise Server 12+
 - RHEL 7+

- Ubuntu 16.04+
- Software requirements:
 - Python 2.7.9+ or 3.2+

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 [AWS 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/ec2r1.tgz
```

2. (Optional) Before proceeding, you can optionally verify the signature of the `EC2Rescue` for Linux installation file. For more information, see [\(Optional\) Verify the signature of EC2Rescue for Linux](#).
3. Download the sha256 hash file:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2r1.tgz.sha256
```

4. Verify the integrity of the tarball:

```
sha256sum -c ec2r1.tgz.sha256
```

5. Unpack the tarball:

```
tar -xzvf ec2r1.tgz
```

6. Verify the installation by listing out the help file:

```
cd ec2r1-<version_number>  
./ec2r1 help
```

To install the bundled build

For a link to the download and a list of limitations, see [EC2Rescue for Linux](#) on github.

(Optional) Verify the signature of EC2Rescue for Linux

The following is the recommended process of verifying the validity of the EC2Rescue for Linux package for Linux-based operating systems.

When you download an application from the internet, we recommend that you authenticate the identity of the software publisher and check that the application has not been altered or corrupted after it was published. This protects you from installing a version of the application that contains a virus or other malicious code.

If, after running the steps in this topic, you determine that the software for EC2Rescue for Linux is altered or corrupted, do not run the installation file. Instead, contact Amazon Web Services.

EC2Rescue for Linux files for Linux-based operating systems are signed using GnuPG, an open-source implementation of the Pretty Good Privacy (OpenPGP) standard for secure digital signatures. GnuPG (also known as GPG) provides authentication and integrity checking through a digital signature. AWS publishes a public key and signatures that you can use to verify the downloaded EC2Rescue for Linux package. For more information about PGP and GnuPG (GPG), see <http://www.gnupg.org>.

The first step is to establish trust with the software publisher. Download the public key of the software publisher, check that the owner of the public key is who they claim to be, and then add the public key to your keyring. Your keyring is a collection of known public keys. After you establish the authenticity of the public key, you can use it to verify the signature of the application.

Tasks

- [Install the GPG tools](#)
- [Authenticate and import the public key](#)
- [Verify the signature of the package](#)

Install the GPG tools

If your operating system is Linux or Unix, the GPG tools may already be installed. To test whether the tools are installed on your system, enter **gpg2** at a command prompt. If the GPG tools are installed, you see a GPG command prompt. If the GPG tools are not installed, you see an error stating that the command cannot be found. You can install the GnuPG package from a repository.

To install GPG tools on Debian-based Linux

- From a terminal, run the following command:

```
apt-get install gnupg2
```

To install GPG tools on Red Hat-based Linux

- From a terminal, run the following command:

```
yum install gnupg2
```

Authenticate and import the public key

The next step in the process is to authenticate the EC2Rescue for Linux public key and add it as a trusted key in your GPG keyring.

To authenticate and import the EC2Rescue for Linux public key

- 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/ec2r1.key
```

- At a command prompt in the directory where you saved `ec2r1.key`, use the following command to import the EC2Rescue for Linux public key into your keyring:

```
gpg2 --import ec2r1.key
```

The command returns results similar to the following:

```
gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
imported
gpg: Total number processed: 1
gpg:             imported: 1 (RSA: 1)
```

Verify the signature of the package

After you've installed the GPG tools, authenticated and imported the EC2Rescue for Linux public key, and verified that the EC2Rescue for Linux public key is trusted, you are ready to verify the signature of the EC2Rescue for Linux installation script.

To verify the EC2Rescue for Linux installation script signature

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/ec2r1.tgz.sig
```

2. Verify the signature by running the following command at a command prompt in the directory where you saved `ec2r1.tgz.sig` and the EC2Rescue for Linux installation file. Both files must be present.

```
gpg2 --verify ./ec2r1.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 AECC 1146  7A9D 8851 1153 6991 ED45
```

If the output contains the phrase `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"`, it means that the signature has successfully been verified, and you can proceed to run the EC2Rescue for Linux installation script.

If the output includes the phrase `BAD signature`, check whether you performed the procedure correctly. If you continue to get this response, contact Amazon Web Services and do not run the installation file that you downloaded previously.

The following are details about the warnings that you might see:

- **WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.** This refers to your personal level of trust in your belief that you possess an authentic public key for EC2Rescue for Linux. In an ideal world, you would visit an Amazon Web Services office and receive the key in person. However, more often you download it from a website. In this case, the website is an Amazon Web Services website.
- **gpg2: no ultimately trusted keys found.** This means that the specific key is not "ultimately trusted" by you (or by other people whom you trust).

For more information, see <http://www.gnupg.org>.

Work with EC2Rescue for Linux

The following are common tasks you can perform to get started using this tool.

Tasks

- [Run EC2Rescue for Linux](#)
- [Upload the results](#)
- [Create backups](#)
- [Get help](#)

Run EC2Rescue for Linux

You can run EC2Rescue for Linux as shown in the following examples.

Example Example: Run all modules

To run all modules, run EC2Rescue for Linux with no options:

```
./ec2r1 run
```

Some modules require root access. If you are not a root user, use **sudo** to run these modules as follows:

```
sudo ./ec2r1 run
```

Example Example: Run a specific module

To run only specific modules, use the **--only-modules** parameter:

```
./ec2r1 run --only-modules=module_name --arguments
```

For example, this command runs the **dig** module to query the `amazon.com` domain:

```
./ec2r1 run --only-modules=dig --domain=amazon.com
```

Example Example: View the results

You can view the results in `/var/tmp/ec2r1`:

```
cat /var/tmp/ec2r1/logfile_location
```

For example, view the log file for the **dig** module:

```
cat /var/tmp/ec2r1/2017-05-11T15_39_21.893145/mod_out/run/dig.log
```

Upload the results

If AWS Support has requested the results or to share the results from an S3 bucket, upload them using the EC2Rescue for Linux CLI tool. The output of the EC2Rescue for Linux commands should provide the commands that you need to use.

Example Example: Upload results to AWS Support

```
./ec2r1 upload --upload-directory=/var/tmp/ec2r1/2017-05-11T15_39_21.893145 --support-url="URLProvidedByAWSSupport"
```

Example Example: Upload results to an S3 bucket

```
./ec2r1 upload --upload-directory=/var/tmp/ec2r1/2017-05-11T15_39_21.893145 --presigned-url="YourPresignedS3URL"
```

For more information about generating pre-signed URLs for Amazon S3, see [Uploading Objects Using Pre-Signed URLs](#).

Create backups

Create a backup for your instance, one or more volumes, or a specific device ID using the following commands.

Example Example: Back up an instance using an Amazon Machine Image (AMI)

```
./ec2r1 run --backup=ami
```

Example Example: Back up all volumes associated with the instance

```
./ec2r1 run --backup=allvolumes
```

Example Example: Back up a specific volume

```
./ec2r1 run --backup=volumeID
```

Get help

EC2Rescue for Linux includes a help file that gives you information and syntax for each available command.

Example Example: Display the general help

```
./ec2r1 help
```

Example Example: List the available modules

```
./ec2r1 list
```

Example Example: Display the help for a specific module

```
./ec2r1 help module_name
```

For example, use the following command to show the help file for the **dig** module:

```
./ec2r1 help dig
```


Develop EC2Rescue modules

Modules are written in YAML, a data serialization standard. A module's YAML file consists of a single document, representing the module and its attributes.

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	<p>The extended description of the module. Each line should be less than or equal to 75 characters in length. If the module consumes arguments, required or optional, include them in the helptext value.</p> <p>For example:</p> <pre>helptext: !!str Collect output from ps for system analysis Consumes --times= for number of times to repeat Consumes --period= for time period between repetition</pre>
placement	<p>The stage in which the module should be run. Supported values:</p> <ul style="list-style-type: none">• prediagnostic• run• postdiagnostic

Attribute	Description
language	<p>The language that the module code is written in. Supported values:</p> <ul style="list-style-type: none">• bash• python <div data-bbox="829 520 1507 737"><p> Note</p><p>Python code must be compatible with both Python 2.7.9+ and Python 3.2+.</p></div>
remediation	<p>Indicates whether the module supports remediation. Supported values are <code>True</code> or <code>False</code>.</p> <p>The module defaults to <code>False</code> if this is absent, making it an optional attribute for those modules that do not support remediation.</p>
content	The entirety of the script code.
constraint	The name of the object containing the constraint values.
domain	<p>A descriptor of how the module is grouped or classified. The set of included modules uses the following domains:</p> <ul style="list-style-type: none">• application• net• os• performance

Attribute	Description
class	<p>A descriptor of the type of task performed by the module. The set of included modules uses the following classes:</p> <ul style="list-style-type: none">• collect (collects output from programs)• diagnose (pass/fail based on a set of criteria)• gather (copies files and writes to specific file)
distro	<p>The list of Linux distributions that this module supports. The set of included modules uses the following distributions:</p> <ul style="list-style-type: none">• alami (Amazon Linux)• rhel• ubuntu• suse
required	<p>The required arguments that the module is consuming from the CLI options.</p>
optional	<p>The optional arguments that the module can use.</p>
software	<p>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.</p>

Attribute	Description
package	The source software package for an executable. This attribute is intended to provide extended details on the package with the software, including a URL for downloading or getting further information.
sudo	<p>Indicates whether root access is required to run the module.</p> <p>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.</p>
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 <code>--perfimpact=true</code> argument is not present, then the module is skipped.
parallelexclusive	Specifies a program that requires mutual exclusivity. For example, all modules specifying "bpf" run in a serial manner.

Add environment variables

The following table lists the available environment variables.

Environment Variable	Description
EC2RL_CALLPATH	The path to <code>ec2rl.py</code> . This path can be used to locate the lib directory and use vendored Python modules.

Environment Variable	Description
EC2RL_WORKDIR	<p>The main tmp directory for the diagnostic tool.</p> <p>Default value: <code>/var/tmp/ec2rl</code> .</p>
EC2RL_RUNDIR	<p>The directory where all output is stored.</p> <p>Default value: <code>/var/tmp/ec2rl/<date&timestamp></code> .</p>
EC2RL_GATHEREDDIR	<p>The root directory for placing gathered module data.</p> <p>Default value: <code>/var/tmp/ec2rl/<date&timestamp>/mod_out/gathered/</code> .</p>
EC2RL_NET_DRIVER	<p>The driver in use for the first, alphabetically ordered, non-virtual network interface on the instance.</p> <p>Examples:</p> <ul style="list-style-type: none">• <code>xen_netfront</code>• <code>ixgbevf</code>• <code>ena</code>
EC2RL_SUDO	<p>True if EC2Rescue for Linux is running as root; otherwise, false.</p>
EC2RL_VIRT_TYPE	<p>The virtualization type as provided by the instance metadata.</p> <p>Examples:</p> <ul style="list-style-type: none">• <code>default-hvm</code>• <code>default-paravirtual</code>

Environment Variable	Description
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
```

EC2 Serial Console for Linux 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.

If you are using a Windows instance, see [EC2 Serial Console for Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Topics

- [Prerequisites](#)
- [Configure access to the EC2 Serial Console](#)
- [Connect to the EC2 Serial Console](#)
- [Disconnect from the EC2 Serial Console](#)
- [Troubleshoot your Linux instance using the EC2 Serial Console](#)

Prerequisites

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](#)
- [sshd server](#)
- [Configure your chosen troubleshooting tool](#)

AWS Regions

Supported in all AWS Regions except Canada West (Calgary).

Wavelength Zones and AWS Outposts

Not supported.

Local Zones

Supported in all Local Zones.

Instance types

Supported instance types:

- 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`

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 [Instance lifecycle](#).

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*.

sshd server

You do not need an sshd server installed or running on your instance.

Configure your chosen troubleshooting tool

To troubleshoot your Linux instance via the serial console, you can use GRUB or SysRq. Before you can use these tools, you must first perform configuration steps on every instance on which you'll use them.

For the instructions to configure your chosen troubleshooting tool on Windows, see [Configure your chosen troubleshooting tool](#) in the *Amazon EC2 User Guide for Windows Instances*.

Tools

- [Configure GRUB](#)
- [Configure SysRq](#)

Configure GRUB

Before you can use GRUB via the serial console, you must configure your instance to use GRUB via the serial console.

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](#)
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
```

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"
```

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 user name 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**.

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. 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](#).

Topics

- [Levels of access to the EC2 Serial Console](#)
- [Manage account access to the EC2 Serial Console](#)
- [Configure IAM policies for EC2 Serial Console access](#)
- [Set an OS user password](#)

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

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](#).

Manage account access to the EC2 Serial Console

By default, there is no access to the serial console at the account level. You need to explicitly grant access to the serial console at the account level.

Topics

- [Grant permission to users to manage account access](#)
- [View account access status to the serial console](#)
- [Grant account access to the serial console](#)
- [Deny account access to the serial console](#)

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:GetSerialConsoleAccessStatus",
        "ec2:EnableSerialConsoleAccess",
        "ec2:DisableSerialConsoleAccess"
      ],
      "Resource": "*"
    }
  ]
}
```

For more information, see [Creating IAM policies](#) in the *IAM User Guide*.

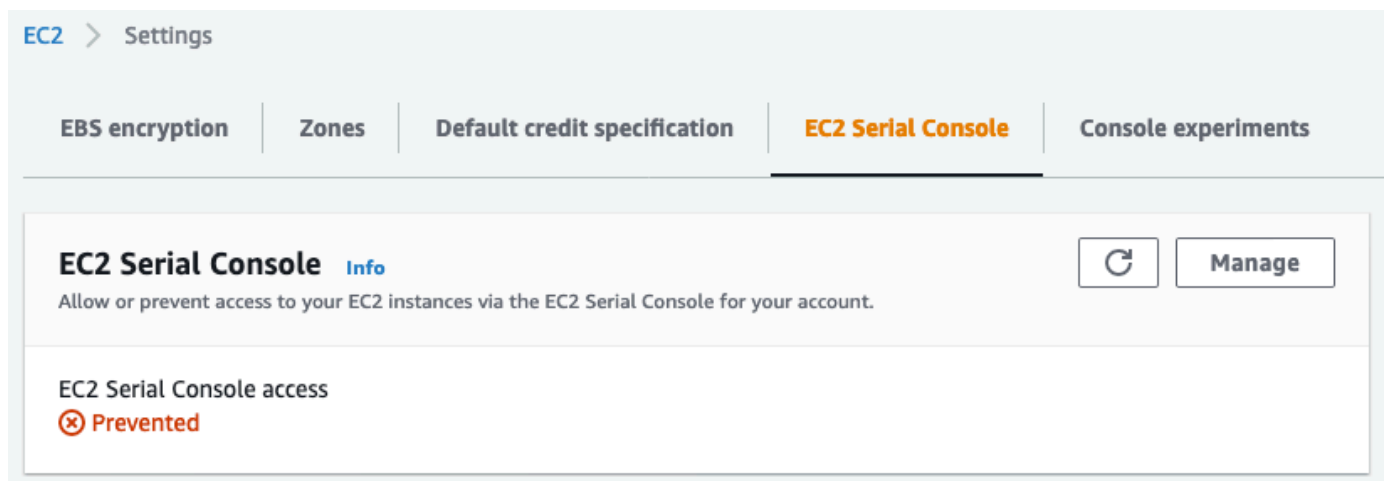
View account access status to the serial console

To view account access status to the serial console (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.

The **EC2 Serial Console access** field indicates whether account access is **Allowed** or **Prevented**.

The following screenshot shows that the account is prevented from using the EC2 serial console.



To view account access status to the serial console (AWS CLI)

Use the [get-serial-console-access-status](#) command to view account access status to the serial console.

```
aws ec2 get-serial-console-access-status --region us-east-1
```

In the following output, `true` indicates that the account is allowed access to the serial console.

```
{
  "SerialConsoleAccessEnabled": true
}
```

Grant account access to the serial console

To grant account access to the serial console (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. Choose **Manage**.
5. To allow access to the EC2 serial console of all instances in the account, select the **Allow** check box.
6. Choose **Update**.

To grant account access to the serial console (AWS CLI)

Use the [enable-serial-console-access](#) command to allow account access to the serial console.

```
aws ec2 enable-serial-console-access --region us-east-1
```

In the following output, `true` indicates that the account is allowed access to the serial console.

```
{  
  "SerialConsoleAccessEnabled": true  
}
```

Deny account access to the serial console

To deny account access to the serial console (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. Choose **Manage**.
5. To prevent access to the EC2 serial console of all instances in the account, clear the **Allow** check box.
6. Choose **Update**.

To deny account access to the serial console (AWS CLI)

Use the [disable-serial-console-access](#) command to prevent account access to the serial console.

```
aws ec2 disable-serial-console-access --region us-east-1
```

In the following output, `false` indicates that the account is denied access to the serial console.

```
{  
  "SerialConsoleAccessEnabled": false  
}
```

Configure IAM policies for EC2 Serial Console access

By default, your 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.

```
{
  "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:region:account-id: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.

```
{
  "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:region:account-id: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.

```

{
  "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:region:account-id:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/SerialConsole":
            "${aws:PrincipalTag/SerialConsole}"
        }
      }
    }
  ]
}
```

Set an OS user password

You can connect to the serial console without a password. However, to *use* the serial console for troubleshooting an 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 an 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

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 Linux 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.
- It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.
- Supported serial console port for Linux: `ttyS0`
- 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](#).
- **Use SysRq to connect to the serial console.** SysRq does not require that you connect via the browser-based client. For more information, see [Troubleshoot your Linux instance using SysRq](#).
- **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, the EC2 console, or the AWS CLI. For more information, see [Troubleshoot your Linux instance using SysRq](#) or [Reboot your instance](#).

5. At the login prompt, enter the user name of the password-based user that you [set up previously](#), and then press **Enter**.
6. 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 user name format is `instance-id.port0`, which comprises the instance ID and port 0. In the following example, the user name is `i-001234a4bf70dec41EXAMPLE.port0`.

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
```

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](#).
- **Use SysRq to connect to the serial console.** SysRq does not require that you connect via SSH. For more information, see [Troubleshoot your Linux instance using SysRq](#).
- **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, the EC2 console, or the AWS CLI. For more information, see [Troubleshoot your Linux instance using SysRq](#) or [Reboot your instance](#).
5. At the login prompt, enter the user name of the password-based user that you [set up previously](#), and then press **Enter**.
 6. 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:Eh wPktzRtTY 7TRSzz26XbB0/ HvV9jRM7mCZNOxw/ d/0
US East (N. Virginia)	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/ xadVMeBZGEru 5l2gx+yI5LDiJaLUcz 0FMmw
US West (N. California)	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OH ldlcMET8u 7QLSX3jmR TRAPFHvtq byoLZBMUCqiH3Y
US West (Oregon)	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EM Cle23TqKaBl6yGHain

Region Name	Region	Endpoint	Fingerprint
			qZcMwqNkD hhAVHa1O2JxVUc
Africa (Cape Town)	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RM WWZ2fVePe JUqzjO5jL2KlgXsczo 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+lfns32XITqL/4O0zlfbx4bZgsYFqy3o8mlk
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 gH8lsO51rezTPiSM35 BsU40
Asia Pacific (Osaka)	ap-northeast-3	ec2-serial-console.ap-northeast-3.api.aws	SHA256:Am0/ jiBKBnBuFnHr9aXs gEV3G8Tu/ vVHFxE/3UcyjsQ

Region Name	Region	Endpoint	Fingerprint
Asia Pacific (Seoul)	ap-northeast-2	serial-console.ec2-instance-connect.ap-northeast-2.aws	SHA256:FoqWXNX+DZ++GuNTztg9PK49WYMqBX+FrcZM2dSrql
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:yFvMwUK9IEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4
Asia Pacific (Tokyo)	ap-northeast-1	serial-console.ec2-instance-connect.ap-northeast-1.aws	SHA256:RQfsDCZTOfQawewTRDV1t9Em/HMrFQe+CRlIOT5um4k
Canada (Central)	ca-central-1	serial-console.ec2-instance-connect.ca-central-1.aws	SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gczYMMO7s4
China (Beijing)	cn-north-1	ec2-serial-console.cn-north-1.api.amazonwebservices.com.cn	SHA256:2gHVFy4H7uU3+WaFUxD28v/ggMeqjvSlgngpgLgGT+Y

Region Name	Region	Endpoint	Fingerprint
China (Ningxia)	cn-northwest-1	ec2-serial-console .cn-northwest-1.ap i.amazonwebservice s.com.cn	SHA256:Td grNZkiQOd VfYEBUhO4 SzUA09VWI 5rYOZGTogpwmIM
Europe (Frankfurt)	eu-central-1	serial-console.ec2- instance-connect.eu- central-1.aws	SHA256:aCMFS/ ylcOdOlkXvOl8A mZ1Toe+bB nrJJ3Fy0k0De2c
Europe (Ireland)	eu-west-1	serial-console.ec2- instance-connect.eu- west-1.aws	SHA256:h2 AaGAWO4Ha thhtm6ezs3Bj7udgUx i2qTrHjZAwCW6E
Europe (London)	eu-west-2	serial-console.ec2- instance-connect.eu- west-2.aws	SHA256:a69rd5CE/ AEG4Amm53I6 lkD1ZPvS/ BCV3tTPW2RnJg8
Europe (Milan)	eu-south-1	ec2-serial-console.eu- south-1.api.aws	SHA256:lC 0kOVJnpgF yBVrxn0A7 n99ecLbXS X95cuuS7X7QK30
Europe (Paris)	eu-west-3	serial-console.ec2- instance-connect.eu- west-3.aws	SHA256:q8 ldnAf9pym eNe8BnFVngY3RPAr/ kxswJUzfrlxeEWs

Region Name	Region	Endpoint	Fingerprint
Europe (Spain)	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:Go CW2DFRlu6 69QNxqFxE csR6fZUz/4F4n7T45Z cwoEc
Europe (Stockholm)	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tk GFFUVUDvo cDiGSS3Cu8Gdl6w2ul 32EPNpKFKLwX84
Europe (Zurich)	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8P px2mBMf6W dCw0NulzKfwM4/ IfRz4OaXFutQX Wp6mk
Israel (Tel Aviv)	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR 6q8v6kNNP i8+QSFQ4d j5dimNmZP TgwgsM1SNvtYyU
Middle East (Bahrain)	me-south-1	ec2-serial-console.me-south-1.api.aws	SHA256:nP jLLKHu2Qn LdUq2kVAr soK5xvPJO MRJKCBzCDqC3k8
Middle East (UAE)	me-central-1	ec2-serial-console.me-central-1.api.aws	SHA256:zpb5duKiBZ +lOdFwPeyy kB4MPBYhl/ XzXNeFSDKBvLE

Region Name	Region	Endpoint	Fingerprint
South America (São Paulo)	sa-east-1	ec2-serial-console.sa-east-1.api.aws	SHA256:rd2+/32Ognj ew1yVlemENaQzC +Botbih62OqAP Dq1dl
AWS GovCloud (US-East)	us-gov-east-1	serial-console.ec2 -instance-connect. us-gov-east-1.amaz onaws.com	SHA256:tl we19GWsoy LClrtvu38YEEh+DHik qnDcZnmtebvF28
AWS GovCloud (US-West)	us-gov-west-1	serial-console.ec2 -instance-connect. us-gov-west-1.amaz onaws.com	SHA256:kf OFRWLaOZfB +utbd3bRf8OlPf8nG O2YZLqXZilw5DQ

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 Linux 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.

Before commencing, be sure to check the [prerequisites](#).

Topics

- [Troubleshoot your Linux instance using GRUB](#)
- [Troubleshoot your Linux instance using SysRq](#)

For information about troubleshooting your Windows instance, see [Troubleshoot your Windows instance using the EC2 Serial Console](#) in the *Amazon EC2 User Guide for Windows Instances*.

Troubleshoot your Linux instance using GRUB

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.

Before you can use GRUB, make sure you have completed the [prerequisites](#), which include granting access to the serial console and configuring GRUB.

Use GRUB

After GRUB is configured, connect to the serial console and reboot the instance with the reboot command. During reboot, you see the GRUB menu. Press any key when the GRUB menu appears to stop the boot process, allowing you to interact with the GRUB menu.

Topics

- [Single user mode](#)
- [Emergency mode](#)

Single user 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.

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 user name 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**.

Emergency mode

Emergency mode is similar to single user mode except that the kernel runs at the lowest runlevel possible.

To boot into emergency mode, follow the steps in [Single user mode](#) in the preceding section, but at step 6 add the word `emergency` instead of `single`.

Troubleshoot your Linux instance using SysRq

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.

Before you can use SysRq, make sure you have completed the [prerequisites](#), which include granting access to the serial console and configuring SysRq.

Use SysRq

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-filestems
(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-filestems
(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.

Send a diagnostic interrupt (for advanced users)

Warning

Diagnostic interrupts are intended for use by advanced users. Incorrect usage could negatively impact your instance. Sending a diagnostic interrupt to an instance could trigger an instance to crash and reboot, which could lead to the loss of data.

You can send a diagnostic interrupt to an unreachable or unresponsive Linux instance to manually trigger a *kernel panic*.

Linux operating systems typically crash and reboot when a kernel panic occurs. The specific behavior of the operating system depends on its configuration. A kernel panic can also be used to cause the instance's operating system kernel to perform tasks, such as generating a crash dump file. You can then use the information in the crash dump file to conduct root cause analysis and debug the instance.

The crash dump data is generated locally by the operating system on the instance itself.

Before sending a diagnostic interrupt to your instance, we recommend that you consult the documentation for your operating system and then make the necessary configuration changes.

Contents

- [Supported instance types](#)
- [Prerequisites](#)
- [Send a diagnostic interrupt](#)

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 occurs.

To configure Amazon Linux 2 to generate a crash dump when a kernel panic occurs

1. Connect to your instance.
2. Install **kexec** and **kdump**.

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. Configure the kernel to reserve an appropriate amount of memory for the secondary kernel. The amount of memory to reserve depends on the total available memory of your instance. Open the `/etc/default/grub` file using your preferred text editor, locate the line that starts with `GRUB_CMDLINE_LINUX_DEFAULT`, and then add the `crashkernel` parameter in the following format: `crashkernel=memory_to_reserve`. For example, to reserve 160MB, modify the grub file as follows:

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=160M console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. Save the changes and close the grub file.
5. Rebuild the GRUB2 configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Open the `/etc/sysctl.conf` file using your preferred text editor and add the following.

```
kernel.unknown_nmi_panic=1
```

7. Reboot and reconnect to your instance.
8. Verify that the kernel has been booted with the correct `crashkernel` parameter.

```
$ grep crashkernel /proc/cmdline
```

The following example output indicates successful configuration.

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=160M console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
rd.shell=0
```

9. Verify that the **kdump** service is running.

```
[ec2-user ~]$ systemctl status kdump.service
```

The following example output shows the result if the **kdump** service is running.

```
kdump.service - Crash recovery kernel arming
  Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
  enabled)
  Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
  Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
  Main PID: 2503 (code=exited, status=0/SUCCESS)
```

 **Note**

By default, the crash dump file is saved to `/var/crash/`. To change the location, modify the `/etc/kdump.conf` file using your preferred text editor.

To configure Amazon Linux to generate a crash dump when a kernel panic occurs

1. Connect to your instance.
2. Install **kexec** and **kdump**.

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. Configure the kernel to reserve an appropriate amount of memory for the secondary kernel. The amount of memory to reserve depends on the total available memory of your instance.

```
$ sudo grubby --args="crashkernel=memory_to_reserve" --update-kernel=ALL
```

For example, to reserve 160MB for the crash kernel, use the following command.

```
$ sudo grubby --args="crashkernel=160M" --update-kernel=ALL
```

4. On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Open the `/etc/sysctl.conf` file using your preferred text editor and add the following.

```
kernel.unknown_nmi_panic=1
```

5. Reboot and reconnect to your instance.
6. Verify that the kernel has been booted with the correct `crashkernel` parameter.

```
$ grep crashkernel /proc/cmdline
```

The following example output indicates successful configuration.

```
root=LABEL=/ console=tty1 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295  
LANG=en_US.UTF-8 KEYTABLE=us crashkernel=160M
```

7. Verify that the **kdump** service is running.

```
[ec2-user ~]$ sudo service kdump status
```

If the service is running, the command returns the `Kdump is operational` response.

Note

By default, the crash dump file is saved to `/var/crash/`. To change the location, modify the `/etc/kdump.conf` file using your preferred text editor.

To configure SUSE Linux Enterprise, Ubuntu, or Red Hat Enterprise Linux

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 \(RHEL\)](#)

Send a diagnostic interrupt

After you have completed the necessary configuration changes, you can send a diagnostic interrupt to your instance using the AWS CLI or Amazon EC2 API.

To send a diagnostic interrupt to your instance (AWS CLI)

Use the [send-diagnostic-interrupt](#) command and specify the instance ID.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

Related information

The following related resources can help you as you work with this service.

Linux on AWS

- [Linux from AWS](#) – Portfolio of modern Linux-based operating systems from AWS.
- [EC2 Image Builder](#) – Automate the creation, management, and deployment of customized, secure, and up-to-date server images that are pre-installed and pre-configured with software settings to meet specific IT standards.

Tutorials for developers

- [Deploy a Web Application on Amazon EC2](#) – Create an Amazon EC2 instance using AWS CDK and deploy a web application on the instance.
- [Amazon EC2 Backup & Restore using AWS Backup](#) – Create an on-demand backup of an Amazon EC2 instance, then learn how to create a backup plan to backup Amazon EC2 instances.
- [Break a Monolith Application into Microservices with Amazon Elastic Container Service, Docker, and Amazon EC2](#) – Deploy a monolithic node.js application to a Docker container, then decouple the application into microservices without any downtime.

AWS re:Post

[AWS re:Post](#) – AWS managed question and answer (Q & A) service offering crowd-sourced, expert-reviewed answers to your technical questions.

Pricing

[Amazon EC2 pricing](#) – Pricing information for Amazon EC2.

General AWS resources

The following general resources can help you as you work with AWS.

- [Classes & Workshops](#) – Links to role-based and specialty courses, in addition to self-paced labs to help sharpen your AWS skills and gain practical experience.
- [AWS Developer Center](#) – Explore tutorials, download tools, and learn about AWS developer events.

- [AWS Developer Tools](#) – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- [Getting Started Resource Center](#) – Learn how to set up your AWS account, join the AWS community, and launch your first application.
- [Hands-On Tutorials](#) – Follow step-by-step tutorials to launch your first application on AWS.
- [AWS Whitepapers](#) – Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.
- [AWS Support Center](#) – The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
- [AWS Support](#) – The primary webpage for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
- [Contact Us](#) – A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
- [AWS Site Terms](#) – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

Document history

The following table describes important additions to the Amazon EC2 documentation starting in 2019. We also update the documentation frequently to address the feedback that you send us.

Change	Description	Date
AMI deregistration protection	You can turn on deregistration 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 C7a, C7i, M7a, M7g, M7i, R7a, and R7i instance types.	April 22, 2024
Introduced two new accelerated instance types: G6 and Gr6	New high performance GPU-based instance types for deep learning inference and graphics-intensive applications.	April 4, 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
PTP hardware clock – US East (N. Virginia)	The PTP hardware clock is now available in the US East (N. Virginia) Region.	March 26, 2024
C7gn bare metal instance	New c7gn.metal bare metal instance type powered by the latest generation AWS Graviton3E processors and the new AWS Nitro cards.	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 (IMDSv2) 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
New bare metal instances	Bare metal instances for C7gd, M7gd, and R7gd.	March 6, 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-launched 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 exhaustion 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

[AWSDataLifecycleManagerSSMFullAccess AWS managed policy](#)

Updated the policy to support application-consistent snapshots for SAP HANA using the `AWSSystemManagerSAP-CreateDLMSnapshotForSAPHANA` SSM document.

November 17, 2023

[VolumeStalledIOCheck metric](#)

You can use the `VolumeStalledIOCheck` metric to check whether a volume has passed or failed a stalled IO check in the last minute.

November 16, 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 hibernation](#)

You can now change the instance type of an instance enabled for hibernation when it is in the stopped state.

November 16, 2023

[Amazon Data Lifecycle Manager default policies](#)

You can now create Amazon Data Lifecycle Manager default policies for EBS snapshots and EBS-backed AMIs to backup all volumes and instances in a Region.

November 16, 2023

[DL2q instances](#)

DL2q instances use Qualcomm AI100 inference accelerators, which feature 7th generation Qualcomm Edge AI cores. They can be used to cost-efficiently deploy deep learning (DL) workloads in the cloud or validate performance and accuracy of DL workloads that will be deployed on Qualcomm edge devices.

November 15, 2023

[Amazon EBS snapshot lock](#)

You can lock your Amazon EBS snapshots to protect them against accidental or malicious deletions, or to store them in WORM format for a specific duration.

November 15, 2023

[Instance topology](#)

You can use the DescribeInstanceTopology 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 physically closer to each other.

November 13, 2023

Block public access for snapshots	You can now use block public access for snapshots to prevent the public sharing of your snapshots.	November 9, 2023
Amazon Data Lifecycle Manager pre and post scripts	You can now use pre and post scripts in your Amazon Data Lifecycle Manager snapshot policies to automate the lifecycle of application-consistent snapshots.	November 7, 2023
AWSDataLifecycleManagerSSMFullAccess AWS managed policy	Amazon Data Lifecycle Manager added the AWSDataLifecycleManagerSSMFullAccess AWS managed policy.	November 7, 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
New bare metal instances	.metal-16x1 and .metal-32x1 bare metal instances for R7iz.	October 30, 2023
New bare metal instances	.metal-24x1 and .metal-48x1 bare metal instances for M7i, R7i, and C7i.	October 30, 2023
New I4i instances	i4i.12xlarge and i4i.24xlarge instances are now available.	October 26, 2023

Mac2-m2 instances	New general purpose instance type that features Apple M2 processors.	October 25, 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
R7i instances	New memory optimized instance types that feature 4th generation Intel Xeon Scalable processors.	October 16, 2023
Hibernation support for Ubuntu 22.04.2 LTS (Jammy Jellyfish)	Hibernate your newly-launched 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
New bare metal instance	<code>r7a.metal-48x1</code> bare metal instance for R7a. Bare metal instances provide your applications with direct access to the physical resources of the host server.	October 4, 2023
C7a instances	New compute optimized instances powered by 4th generation AMD EPYC processors.	October 4, 2023
Hibernation support for Red Hat Enterprise Linux 9	Hibernate your newly-launched instances that were launched from the Red Hat Enterprise Linux 9 AMI.	October 2, 2023
Hibernation support for AL2023	Hibernate your newly-launched 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

NVMe reservations	Multi-Attach enabled io2 volumes support NVMe reservations, which is a set of industry-standard storage fencing protocols.	September 18, 2023
Mac2-m2pro instances	New general purpose instance type that features Apple M2 Pro processors.	September 18, 2023
C7i instances	New compute optimized instance types that feature 4th generation Intel Xeon Scalable processors.	September 14, 2023
Block public access to AMIs	You can enable <i>block public access for AMIs</i> at the account level to block any attempts to make your AMIs public.	September 12, 2023
R7a instances	New memory optimized instance types featuring 4th generation AMD EPYC 9R14 processors and up to 1536 GiB of system memory.	September 11, 2023
R7iz instances	New high-frequency and high memory instances powered by 4th generation Intel Xeon processors.	September 7, 2023
Hibernation support for M7i and M7i-flex	Hibernate your newly-launched instances running on M7i and M7i-flex instance types.	August 22, 2023

Hpc7a instances	New compute optimized instance types that feature 4th generation AMD EPYC processors. These instance types support up to 300 Gbps networking bandwidth, and up to 192 CPU cores with up to 768 GB of system memory.	August 17, 2023
M7a instances	New general purpose instances powered by 4th generation AMD EPYC processors.	August 15, 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
M7i-flex instances	New general purpose instances that offer a balance of compute, memory, and network resources for a broad spectrum of general purpose applications. They deliver a baseline CPU performance of 40 percent with the ability to deliver up to 100 percent CPU performance for 95 percent of the time over a 24-hour period.	August 2, 2023

M7i instances	New general purpose instance types that feature 4th generation Intel Xeon Scalable processors.	August 2, 2023
R7gd instances	New memory optimized instances featuring the latest AWS Graviton3 processors.	July 28, 2023
M7gd instances	New general purpose instances featuring the latest AWS Graviton3 processors.	July 28, 2023
C7gd instances	New compute optimized instances featuring the latest AWS Graviton3 processors.	July 28, 2023
P5 instances	New accelerated computing instances that feature 8 NVIDIA H100 GPUs with 640 GB high-bandwidth GPU memory, 3rd generation AMD EPYC processors, and 2 TB system memory.	July 26, 2023
Amazon EBS performance updates	Updated Amazon EBS performance for R6a instances.	June 29, 2023
Hpc7g instances	New high-performance computing instances powered by AWS Graviton3E processors that provide up to 35 percent higher vector-instruction processing performance than Graviton3 processors.	June 20, 2023

Dedicated Hosts	You can allocate Dedicated Hosts on specific hardware assets on an Outpost.	June 20, 2023
C7gn instances	New compute optimized instances powered by the latest generation AWS Graviton3E processors and the new AWS Nitro cards. These instances offer up to 200 Gbps network bandwidth.	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
New storage optimized instances	I4g is a new storage optimized instance that features the AWS Graviton2 processor and AWS Nitro SSDs.	May 9, 2023
Trn1n instances	New accelerated computing instances optimized for machine learning training powered by AWS Trainium accelerators.	April 13, 2023
Inf2 instances	New instances featuring AWS Inferentia2 accelerators, the latest machine learning chip designed by AWS.	April 13, 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 utilization notifications	AWS Health now sends notifications when capacity utilization for Capacity Reservations in your account drops below 20 percent.	April 3, 2023
Amazon EBS performance updates	Updated Amazon EBS performance for M6a and C6a instances.	April 3, 2023
Capacity Reservation groups	You can now add Capacity Reservations that are shared with you to Capacity Reservation groups that you own.	March 30, 2023
New bare metal instances	Bare metal instances for C6in, M6idn, M6in, R6idn, and R6in.	March 21, 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
Add supported instances for ENA Express	Added a table with new and existing supported instance types for ENA Express.	February 13, 2023
Fault testing on Amazon EBS	Use AWS FIS to temporarily stop I/O between an EBS volume and the instances to which it is attached to test how your workloads handle I/O interruptions.	January 27, 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
Hpc6id instance	New memory optimized instance featuring 3rd generation Intel Xeon Scalable processors (Ice Lake).	November 29, 2022
R6in and R6idn instances	New memory optimized instances for network-intensive workloads.	November 28, 2022
M6in and M6idn instances	New general computing instances types.	November 28, 2022
ENA Express	Increase throughput and minimize tail latency of network traffic between EC2 instances with ENA Express.	November 28, 2022
C6in instances	New compute optimized instances ideal for running high performance computing.	November 28, 2022
Recycle Bin retention rule lock	You can lock retention rules to help protect them against accidental or malicious modifications and deletions.	November 23, 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
Trn1 instances	New accelerated computing instances optimized for deep learning powered by AWS Trainium chips.	October 10, 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

R6a instances	New memory optimized instances featuring 3rd generation AMD EPYC processors.	July 19, 2022
Placement groups on AWS Outposts	Added a host spread strategy for placement groups on an Outpost.	June 30, 2022
Condition keys for Recycle Bin	You can use the <code>rbin:Request/ResourceType</code> and <code>rbin:Attribute/ResourceType</code> condition keys to filter access on Recycle Bin requests.	June 14, 2022
R6id instances	New memory optimized instances featuring 3rd generation Intel Xeon Scalable processors (Ice Lake).	June 9, 2022
io2 Block Express volumes	You can modify the size and provisioned IOPS of io2 Block Express volumes and you can enable them for fast snapshot restore.	May 31, 2022
Dedicated Hosts on AWS Outposts	You can allocate Dedicated Hosts on AWS Outposts.	May 31, 2022
M6id instances	New general purpose instances featuring 3rd generation Intel Xeon Scalable processors (Ice Lake).	May 26, 2022

C6id instances	New compute optimized instances featuring 3rd generation Intel Xeon Scalable processors (Ice Lake).	May 26, 2022
Instance stop protection	To prevent your instance from being accidentally stopped, you can enable stop protection for the instance.	May 24, 2022
C7g instances	New compute optimized instances featuring the latest AWS Graviton3 processors.	May 23, 2022
UEFI Secure Boot	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.	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
I4i instances	New storage optimized instances featuring 3rd generation Intel Xeon Scalable processors (Ice Lake).	April 27, 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

X2idn and X2iedn instances	New memory optimized instances featuring Intel Xeon Scalable processors (Ice Lake).	March 10, 2022
AMI last launched time	The <code>lastLaunchedTime</code> indicates when your AMI was last used to launch an instance.	February 28, 2022
C6a instances	New compute optimized instances featuring 3rd generation AMD EPYC processors (Milan).	February 14, 2022
Recycle Bin for AMIs	Recycle Bin enables you to restore accidentally deleted AMIs.	February 3, 2022
X2iezn instances	New memory optimized instances featuring Intel Xeon Platinum processors (Cascade Lake).	January 26, 2022
ED25519 keys	ED25519 keys are now supported for EC2 Instance Connect and EC2 Serial Console.	January 20, 2022
New Local Zones added	Add Local Zones in Atlanta, Phoenix, and Seattle.	January 11, 2022
Additional RHEL platforms for Capacity Reservations	Additional Red Hat Enterprise Linux platforms for On-Demand Capacity Reservations.	January 11, 2022

Hpc6a instances	New compute optimized instances featuring AMD EPYC processors.	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
Im4gn and Is4gen instances	New storage optimized instances.	November 30, 2021
Recycle Bin for Amazon EBS snapshots	Recycle Bin for Amazon EBS snapshots is a snapshot recovery feature that enables you to restore accidentally deleted snapshots.	November 29, 2021
M6a instances	New general purpose instances powered by AMD 3rd Generation EPYC processors.	November 29, 2021
G5g instances	New accelerated computing instances featuring AWS Graviton2 processors based on 64-bit Arm architecture.	November 29, 2021
Amazon EBS Snapshots Archive	Amazon EBS Snapshots Archive is a new storage tier that you can use for low-cost, long-term storage of your rarely-accessed snapshots.	November 29, 2021

R6i instances	New memory optimized instances.	November 22, 2021
G5 instances	New accelerated computing instances featuring up to 8 NVIDIA A10G GPUs and second generation AMD EPY processors.	November 11, 2021
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
C6i instances	New compute optimized instances featuring Intel Xeon Scalable processors (Ice Lake).	October 28, 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
New Local Zones added	Add Local Zones in Las Vegas, New York City, and Portland.	October 26, 2021
DL1 instances	New accelerated computing instances featuring Habana Gaudi accelerators and Intel Xeon Platinum processors (Cascade Lake).	October 26, 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-launched instances that were launched from the Ubuntu 20.04 LTS - Focal AMI.	October 4, 2021

EC2 Fleet and targeted On-Demand Capacity Reservations	EC2 Fleet can launch On-Demand Instances into targeted Capacity Reservations.	September 22, 2021
T3 instances on Dedicated Hosts	Support for T3 instances on Amazon EC2 Dedicated Host.	September 14, 2021
VT1 instances	New accelerated computing instances that use Xilinx Alveo U30 media accelerators and are designed for live video transcoding workloads.	September 13, 2021
Hibernation support for RHEL, Fedora, and CentOS	Hibernate your newly-launched instances that were launched from RHEL, Fedora, and CentOS AMIs.	September 9, 2021
New Local Zones added	Add Local Zones in Chicago, Minneapolis, and Kansas City.	September 8, 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
AMI deprecation support for Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager EBS-backed AMI policies can deprecate AMIs. The <code>AWSDataLifecycleManagerServiceRoleForAMIManagement</code> AWS managed policy has been updated to support this feature.	August 23, 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
M6i instances	New general purpose instances featuring third generation Intel Xeon Scalable processors (Ice Lake).	August 16, 2021
CloudWatch metrics for Amazon Data Lifecycle Manager	You can monitor your Amazon Data Lifecycle Manager policies using Amazon CloudWatch.	July 28, 2021
New Local Zone added	Add Local Zone in Denver.	July 27, 2021
CloudTrail data events for EBS direct APIs	The ListSnapshotBlocks, ListChangedBlocks, GetSnapshotBlock, and PutSnapshotBlock APIs can be logged data events in CloudTrail.	July 27, 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

io2 Block Express volumes	io2 Block Express volumes are now generally available in all Regions and Availability Zones that support R5b instances.	July 19, 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
New Local Zones added	Add Local Zones in Dallas and Philadelphia.	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

[High memory virtualized instances](#)

Virtualized high memory instances purpose-built to run large in-memory databases. The new types are u-6tb1.56xlarge, u-6tb1.112xlarge, u-9tb1.112xlarge, and u-12tb1.112xlarge.

May 11, 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 connection 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

[X2gd instances](#)

New memory optimized instances featuring an AWS Graviton2 processor based on 64-bit Arm architecture.

March 16, 2021

Amazon EBS local snapshots on Outposts	You can now use Amazon EBS local snapshots on Outposts to store snapshots of volumes on an Outpost locally in Amazon S3 on the Outpost itself.	February 4, 2021
Create a reverse DNS record	You can now set up reverse DNS lookup for your Elastic IP addresses.	February 3, 2021
Multi-Attach support for io2 volumes	You can now enable Provisioned IOPS SSD (io2) volumes for Amazon EBS Multi-Attach.	December 18, 2020
C6gn instances	New computed optimized instances featuring an AWS Graviton2 processor based on 64-bit Arm architecture. These instances can utilize up to 100 Gbps of network bandwidth.	December 18, 2020
Amazon Data Lifecycle Manager	Use Amazon Data Lifecycle Manager to automate the process of sharing snapshots and copying them across AWS accounts.	December 17, 2020
G4ad instances	New instances powered by AMD Radeon Pro V520 GPUs and AMD 2nd Generation EPYC processors.	December 9, 2020

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
io2 Block Express preview	You can opt in to the io2 Block Express volumes preview. io2 Block Express volumes provide sub-millisecond latency, and support higher IOPS, higher throughput, and larger capacity than io2 volumes.	December 1, 2020
gp3 volumes	A new Amazon EBS General Purpose SSD volume type. You can specify provisioned IOPS and throughput when you create or modify the volume.	December 1, 2020
D3, D3en, M5zn, and R5b instances	New instance types built on the Nitro System.	December 1, 2020
Throughput Optimized HDD and Cold HDD volume sizes	Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes can range in size from 125 GiB to 16 TiB.	November 30, 2020
Mac1 instances	New instances built on Apple Mac mini computers that support running macOS workloads on Amazon EC2.	November 30, 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-launched instances running on T3 and T3a instance types.	November 17, 2020
Amazon EFS Quick Create	You can create and mount an Amazon Elastic File System (Amazon EFS) file system to an instance at launch using Amazon EFS Quick Create.	November 9, 2020
Amazon Data Lifecycle Manager	You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of EBS-backed AMIs.	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 recommendation.	November 4, 2020
P4d instances	New accelerated computing instances that provide a high-performance platform for machine learning and HPC workloads.	November 2, 2020
Hibernation support for I3, M5ad, and R5ad	Hibernate your newly-launched 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

Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager policies can be configured with up to four schedules.	September 17, 2020
T4g instances	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	September 14, 2020
Hibernation support for M5a and R5a	Hibernate your newly-launched instances running on M5a and R5a instance types.	August 28, 2020
Provisioned IOPS SSD (io2) volumes for Amazon EBS	Provisioned IOPS SSD (io2) volumes are designed to provide 99.999 percent volume durability with an AFR no higher than 0.001 percent.	August 24, 2020
Instance metadata provides instance location and placement information	New instance metadata fields under the placement category: Region, placement group name, partition number, host ID, and Availability Zone ID.	August 24, 2020
C5ad instances	New compute optimized instances featuring second-generation AMD EPYC processors.	August 13, 2020

Wavelength Zones	A Wavelength Zone is an isolated zone in the carrier location where the Wavelength infrastructure is deployed.	August 6, 2020
Capacity Reservation groups	You can use AWS Resource Groups to create logical collections of Capacity Reservations, and then target instance launches into those groups.	July 29, 2020
C6gd, M6gd, and R6gd instances	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	July 27, 2020
Fast snapshot restore	You can enable fast snapshot restore for snapshots that are shared with you.	July 21, 2020
C6g and R6g instances	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	June 10, 2020

Bare metal instances for G4dn	New instances that provide your applications with direct access to the physical resources of the host server.	June 5, 2020
C5a instances	New compute optimized instances featuring second-generation AMD EPYC processors.	June 4, 2020
Bring your own IPv6 addresses	You can bring part or all of your IPv6 address range from your on-premises network to your AWS account.	May 21, 2020
M6g instances	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	May 11, 2020
Launch instances using a Systems Manager parameter	You can specify a AWS Systems Manager parameter instead of an AMI when you launch an instance.	May 5, 2020
Customize scheduled event notifications	You can customize scheduled event notifications to include tags in the email notification.	May 4, 2020

Amazon Linux 2 Kernel Live Patching	Kernel Live Patching for Amazon Linux 2 enables you to apply security vulnerability and critical bug patches to a running Linux kernel, without reboots or disruptions to running applications.	April 28, 2020
Amazon EBS Multi-Attach	You can now attach a single Provisioned IOPS SSD (io1) volume to up to 16 Nitro-based instances that are in the same Availability Zone.	February 14, 2020
Stop and start a Spot Instance	Stop your Spot Instances backed by Amazon EBS and start them at will, instead of relying on the stop interruption behavior.	January 13, 2020
Resource tagging	You can tag egress-only internet gateways, local gateways, local gateway route tables, local gateway virtual interfaces, local gateway virtual interface groups, local gateway route table VPC associations, and local gateway route table virtual interface group associations.	January 10, 2020
Connect to your instance using Session Manager	You can start a Session Manager session with an instance from the Amazon EC2 console.	December 18, 2019

Inf1 instances	New instances featuring AWS Inferentia, a machine learning inference chip designed to deliver high performance at a low cost.	December 3, 2019
Dedicated Hosts and host resource groups	Dedicated Hosts can now be used with host resource groups.	December 2, 2019
Dedicated Host sharing	You can now share your Dedicated Hosts across AWS accounts.	December 2, 2019
Default credit specification at the account level	You can set the default credit specification per burstable performance instance family at the account level per AWS Region.	November 25, 2019
Instance type discovery	You can find an instance type that meets your needs.	November 22, 2019
Dedicated Hosts	You can now configure a Dedicated Host to support multiple instance types in an instance family.	November 21, 2019
Amazon EBS fast snapshot restores	You can enable fast snapshot restores on an EBS snapshot to ensure that EBS volumes created from the snapshot are fully-initialized at creation and instantly deliver all of their provisioned performance.	November 20, 2019

Instance Metadata Service Version 2	You can use Instance Metadata Service Version 2, which is a session-oriented method for requesting instance metadata.	November 19, 2019
Elastic Fabric Adapter	Elastic Fabric Adapters can now be used with Intel MPI 2019 Update 6.	November 15, 2019
Queued purchases of Reserved Instances	You can queue the purchase of a Reserved Instance up to three years in advance.	October 4, 2019
G4dn instances	New instances featuring NVIDIA Tesla GPUs.	September 19, 2019
Diagnostic interrupt	You can send a diagnostic interrupt to an unreachable or unresponsive instance to trigger a kernel panic.	August 14, 2019
Capacity optimized allocation strategy	Using EC2 Fleet or Spot Fleet, you can 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
Resource tagging	Launch templates on creation.	July 24, 2019

EC2 Instance Connect	EC2 Instance Connect is a simple and secure way to connect to your instances using Secure Shell (SSH).	June 27, 2019
Host recovery	Automatically restart your instances on a new host in the event of an unexpected hardware failure on a Dedicated Host.	June 5, 2019
Amazon EBS multi-volume snapshots	You can take exact point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance.	May 29, 2019
Resource tagging	You can tag Dedicated Host Reservations.	May 27, 2019
Amazon EBS encryption by default	After you enable encryption by default in a Region, all new EBS volumes you create in the Region are encrypted using the default KMS key for EBS encryption.	May 23, 2019
Resource tagging	You can tag VPC endpoints , endpoint services, and endpoint service configurations.	May 13, 2019
Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases	Move existing Microsoft SQL Server workloads from a Windows to a Linux operating system.	May 8, 2019

I3en instances	New I3en instances can utilize up to 100 Gbps of network bandwidth.	May 8, 2019
Elastic Fabric Adapter	You can attach an Elastic Fabric Adapter to your instances to accelerate High Performance Computing (HPC) applications.	April 29, 2019
T3a instances	New instances featuring AMD EPYC processors.	April 24, 2019
M5ad and R5ad instances	New instances featuring AMD EPYC processors.	March 27, 2019
Resource tagging	You can assign custom tags to your Dedicated Host Reservations to categorize them in different ways.	March 14, 2019
Bare metal instances for M5, M5d, R5, R5d, and z1d	New instances that provide your applications with direct access to the physical resources of the host server.	February 13, 2019

History for previous years

The following table describes important additions to the Amazon EC2 documentation in 2018 and earlier years.

Feature	API version	Description	Release date
Partition placement groups	2016-11-15	Partition placement groups spread instances across logical partitions, ensuring that instances in one partition do not share underlying	20 December 2018

Feature	API version	Description	Release date
		hardware with instances in other partitions. For more information, see Partition placement groups .	
p3dn.24xlarge instances	2016-11-15	New p3dn.24xlarge instances provide 100 Gbps of network bandwidth.	7 December 2018
Hibernate EC2 Linux instances	2016-11-15	You can hibernate a Linux instance if it's enabled for hibernation and it meets the hibernation prerequisites. For more information, see Hibernate your 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
Instances featuring 100 Gbps of network bandwidth	2016-11-15	New C5n instances can utilize up to 100 Gbps of network bandwidth.	26 November 2018
Instances featuring Arm-based Processors	2016-11-15	New A1 instances deliver significant cost savings and are ideally suited for scale-out and Arm-based workloads.	26 November 2018
Spot console recommends a fleet of instances	2016-11-15	The Spot console recommends a fleet of instances based on Spot best practice (instance diversification) to meet the minimum hardware specifications (vCPUs, memory, and storage) for your application need. For more information, see Create a Spot Fleet request .	20 November 2018

Feature	API version	Description	Release date
New EC2 Fleet request type: <code>instant</code>	2016-11-15	EC2 Fleet now supports a new request type, <code>instant</code> , that you can use to synchronously provision capacity across instance types and purchase models. The <code>instant</code> request returns the launched instances in the API response, and takes no further action, enabling you to control if and when instances are launched. For more information, see EC2 Fleet request types .	14 November 2018
Instances featuring AMD EPYC processors	2016-11-15	New general purpose (M5a) and memory optimized instances (R5a) offer lower-priced options for microservices, small to medium databases, virtual desktops, development and test environments, business applications, and more.	6 November 2018
Spot savings information	2016-11-15	You can view the savings made from using Spot Instances for a single Spot Fleet or for all Spot Instances. For more information, see Savings from purchasing Spot Instances .	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 Optimize CPU options .	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 a launch template .	30 October 2018

Feature	API version	Description	Release date
On-Demand Capacity Reservations	2016-11-15	You can reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This allows you to create and manage capacity reservations independently from the billing discounts offered by Reserved Instances (RI). For more information, see On-Demand Capacity Reservations .	25 October 2018
Bring Your Own IP Addresses (BYOIP)	2016-11-15	You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources. For more information, see Bring your own IP addresses (BYOIP) in Amazon EC2 .	23 October 2018
g3s.xlarge instances	2016-11-15	Expands the range of the accelerated-computing G3 instance family with the introduction of g3s.xlarge instances.	11 October 2018
Dedicated Host tag on create and console support	2016-11-15	You can tag your Dedicated Hosts on creation, and you can manage your Dedicated Host tags using the Amazon EC2 console. For more information, see Allocate Dedicated Hosts .	08 October 2018
High memory instances	2016-11-15	These instances are purpose-built to run large in-memory databases. They offer bare metal performance with direct access to host hardware.	27 September 2018
f1.4xlarge instances	2016-11-15	Expands the range of the accelerated-computing F1 instance family with the introduction of f1.4xlarge instances.	25 September 2018

Feature	API version	Description	Release date
Console support for scheduled scaling for Spot Fleet	2016-11-15	Increase or decrease the current capacity of the fleet based on the date and time. For more information, see Scale Spot Fleet using scheduled scaling .	20 September 2018
T3 instances	2016-11-15	T3 instances are burstable general-purpose instance type that provide a baseline level of CPU performance with the ability to burst CPU usage at any time for as long as required. For more information, see Burstable performance instances .	21 August 2018
Allocation strategies for EC2 Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Allocation strategies for Spot Instances .	26 July 2018
Allocation strategies for Spot Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Allocation strategies for Spot Instances .	26 July 2018
R5 and R5d instances	2016-11-15	R5 and R5d instances are ideally suited for high-performance databases, distributed in-memory caches, and in-memory analytics. R5d instances come with NVMe instance store volumes.	25 July 2018

Feature	API version	Description	Release date
z1d instances	2016-11-15	These instances are designed for applications that require high per-core performance with a large amount of memory, such as electronic design automation (EDA) and relational databases. These instances come with NVME instance store volumes.	25 July 2018
Automate snapshot lifecycle	2016-11-15	You can use Amazon Data Lifecycle Manager to automate creation and deletion of snapshots for your EBS volumes. For more information, see 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 a launch template .	11 July 2018
Tag Dedicated Hosts	2016-11-15	You can tag your Dedicated Hosts. For more information, see Tag Dedicated Hosts .	3 July 2018
i3.metal instances	2016-11-15	i3.metal instances provide your applications with direct access to the physical resources of the host server, such as processors and memory.	17 May 2018
Get latest console output	2016-11-15	You can retrieve the latest console output for some instance types when you use the get-console-output AWS CLI command.	9 May 2018
Optimize CPU options	2016-11-15	When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs. For more information, see Optimize CPU options .	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 .	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 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 group for an instance .	1 March 2018
Longer resource IDs	2016-11-15	You can enable the longer ID format for more resource types. For more information, see Resource IDs .	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. For more information, see Tag an Elastic IP address .	21 December 2017

Feature	API version	Description	Release date
Amazon Linux 2	2016-11-15	Amazon Linux 2 is a new version of Amazon Linux. It provides a high performance, stable, and secure foundation for your applications. For more information, see Amazon Linux .	13 December 2017
Amazon Time Sync Service	2016-11-15	You can use the Amazon Time Sync Service to keep accurate time on your instance. For more information, see Set the time for your Linux 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 Launch an instance from a launch template .	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
H1 instances	2016-11-15	H1 instances are designed for high-performance big data workloads.	28 November 2017
M5 instances	2016-11-15	M5 instances are general purpose compute instances. They provide a balance of compute, memory, storage, and network resources.	28 November 2017

Feature	API version	Description	Release date
Spot Instance hibernation	2016-11-15	The Spot service can hibernate Spot Instances in the event of an interruption. For more information, see Hibernate interrupted Spot Instances .	28 November 2017
Spot Fleet target tracking	2016-11-15	You can set up target tracking scaling policies for your Spot Fleet. For more information, see Scale Spot Fleet using a target tracking policy .	17 November 2017
Spot Fleet integrates with Elastic Load Balancing	2016-11-15	You can attach one or more load balancers to a Spot Fleet.	10 November 2017
X1e instances	2016-11-15	X1e instances are ideally suited for high-performance databases, in-memory databases, and other memory-intensive enterprise applications.	28 November 2017
C5 instances	2016-11-15	C5 instances are designed for compute-heavy applications.	6 November 2017
Merge and split Convertible Reserved Instances	2016-11-15	You can exchange (merge) two or more Convertible Reserved Instances for a new Convertible Reserved Instance. You can also use the modification process to split a Convertible Reserved Instance into smaller reservations. For more information, see Exchange Convertible Reserved Instances .	6 November 2017
P3 instances	2016-11-15	P3 instances are compute-optimized GPU instances.	25 October 2017

Feature	API version	Description	Release date
Modify VPC tenancy	2016-11-15	You can change the instance tenancy attribute of a VPC from <code>dedicated</code> to <code>default</code> . For more information, see Change the 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 Interruption behavior .	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. For more information, see Security group rules .	31 August 2017
Recover Elastic IP addresses	2016-11-15	If you release an Elastic IP address for use in a VPC, you might be able to recover it. For more information, see Recover an Elastic IP address .	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

Feature	API version	Description	Release date
G3 instances	2016-11-15	G3 instances provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. G3 instances also provide NVIDIA GRID Virtual Workstation features, supporting 4 monitors with resolutions up to 4096x2160.	13 July 2017
F1 instances	2016-11-15	F1 instances are accelerated computing instances.	19 April 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 resources during creation .	28 March 2017
I3 instances	2016-11-15	I3 instances are storage optimized instances.	23 February 2017
Perform modifications on attached EBS volumes	2016-11-15	With most EBS volumes attached to most EC2 instances, you can modify volume size, type, and IOPS without detaching the volume or stopping the instance.	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 Specify a tenancy for your Spot Instances .	19 January 2017

Feature	API version	Description	Release date
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
R4 instances	2016-09-15	R4 instances are memory optimized instances . R4 instances are well-suited for memory-intensive, latency-sensitive workloads such as business intelligence (BI), data mining and analysis, in-memory databases, distributed web scale in-memory caching, and applications performance real-time processing of unstructured big data.	30 November 2016
New t2.xlarge and t2.2xlarge instance types	2016-09-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances .	30 November 2016
P2 instances	2016-09-15	P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models.	29 September 2016
m4.16xlarge instances	2016-04-01	Expands the range of the general-purpose M4 family with the introduction of m4.16xlarge instances, with 64 vCPUs and 256 GiB of RAM.	6 September 2016

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 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 support .	28 June 2016
Enhanced support for viewing and modifying longer IDs	2016-04-01	You can now view and modify longer ID settings for other IAM users, IAM roles, or the root user. For more information, see Resource IDs .	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
X1 instances	2015-10-01	Memory-optimized instances designed for running in-memory databases, big data processing engines, and high performance computing (HPC) applications.	18 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

Feature	API version	Description	Release date
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 CloudWatch metrics for Spot Fleet .	21 March 2016
Scheduled Instances	2015-10-01	Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration.	13 January 2016
Longer resource IDs	2015-10-01	We're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. During the opt-in period, you can enable the longer ID format for supported resource types. For more information, see Resource IDs .	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

Feature	API version	Description	Release date
New t2.nano instance type	2015-10-01	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances .	15 December 2015
Dedicated hosts	2015-10-01	An Amazon EC2 Dedicated host is a physical server with instance capacity dedicated for your use. For more information, see Dedicated Hosts .	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
Spot Fleet diversified allocation strategy	2015-04-15	You can now allocate Spot Instances in multiple Spot pools using a single Spot Fleet request. For more information, see Allocation strategies for Spot Instances .	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 Spot Fleet instance weighting .	31 August 2015

Feature	API version	Description	Release date
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
New t2.large instance type		T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances .	16 June 2015
M4 instances		General-purpose instances that provide a balance of compute, memory, and network resources. M4 instances are powered by a custom Intel 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processor with AVX2.	11 June 2015
Spot Fleets	2015-04-15	You can manage a collection, or fleet, of Spot Instances instead of managing separate Spot Instance requests. For more information, see 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 <i>VM Import/Export User Guide</i> .	23 April 2015

Feature	API version	Description	Release date
New g2.8xlarge instance type		The new g2.8xlarge instance is backed by four high-performance NVIDIA GPUs, making it well suited for GPU compute workloads including large scale rendering, transcoding, machine learning, and other server-side workloads that require massive parallel processing power.	7 April 2015
D2 instances		Dense-storage instances that are optimized for applications requiring sequential access to large amount of data on direct attached instance storage. D2 instances are designed to offer best price/performance in the dense-storage family. Powered by 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processors, D2 instances improve on HS1 instances by providing additional compute power, more memory, and Enhanced Networking. In addition, D2 instances are available in four instance sizes with 6TB, 12TB, 24TB, and 48TB storage options.	24 March 2015
Automatic recovery for EC2 instances		<p>You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, IP addresses, and all instance metadata.</p> <p>For more information, see Recover your instance.</p>	12 January 2015

Feature	API version	Description	Release date
C4 instances		Next-generation compute-optimized instances that provide very high CPU performance at an economical price. C4 instances are based on custom 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) processors. With additional Turbo boost, the processor clock speed in C4 instances can reach as high as 3.5GHz with 1 or 2 core turbo. Expanding on the capabilities of C3 compute-optimized instances, C4 instances offer customers the highest processor performance among EC2 instances. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics.	11 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		<p>The best way to protect against Spot Instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot Instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot Instance.</p> <p>For more information, see Spot Instance interruption notices.</p>	5 January 2015
DescribeVolumes pagination support	2014-09-01	<p>The DescribeVolumes API call now supports the pagination of results with the MaxResults and NextToken parameters. For more information, see DescribeVolumes in the <i>Amazon EC2 API Reference</i>.</p>	23 October 2014
T2 instances	2014-06-15	<p>T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances.</p>	30 June 2014
New EC2 Service Limits page		<p>Use the EC2 Service Limits page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.</p>	19 June 2014

Feature	API version	Description	Release date
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
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

Feature	API version	Description	Release date
R3 instances	2014-02-01	<p>Memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances.</p> <p>For detailed instance type specifications, see Specifications in the <i>Amazon EC2 Instance Types Guide</i>. For pricing information, see Amazon EC2 On-Demand Pricing.</p>	9 April 2014
New Amazon Linux AMI release		Amazon Linux AMI 2014.03 is released.	27 March 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2. For more information, see Amazon EC2 usage reports .	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes <code>m3.medium</code> and <code>m3.large</code> are now supported. For detailed instance type specifications, see Specifications in the <i>Amazon EC2 Instance Types Guide</i> . For pricing information, see Amazon EC2 On-Demand Pricing .	20 January 2014

Feature	API version	Description	Release date
I2 instances	2013-10-15	These instances provide very high IOPS and support TRIM on Linux instances for better successive SSD write performance. I2 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance.	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, <code>m3.xlarge</code> and <code>m3.2xlarge</code> now support instance store with SSD volumes.	19 December 2013
Importing Linux virtual machines	2013-10-15	The VM Import process now supports the importation of Linux instances. For more information, see the VM Import/Export User Guide .	16 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Identity and access management for Amazon EC2 .	20 November 2013

Feature	API version	Description	Release date
C3 instances	2013-10-15	<p>Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics.</p> <p>For detailed instance type specifications, see Specifications in the <i>Amazon EC2 Instance Types Guide</i>. For pricing information, see Amazon EC2 On-Demand Pricing.</p>	14 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launch an AWS Marketplace instance .	11 November 2013
G2 instances	2013-10-01	These instances are ideally suited for video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side workloads requiring massive parallel processing power.	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launch an instance using the old launch instance wizard .	10 October 2013

Feature	API version	Description	Release date
Modifying Instance Types of Amazon EC2 Reserved Instances	2013-10-01	You can now modify the instance type of Linux Reserved Instances within the same family (for example, M1, M2, M3, C1). For more information, see Modify Reserved Instances .	09 October 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.09 is released.	30 September 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a Region. For more information, see Modify Reserved Instances .	11 September 2013
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assign a public IPv4 address during instance 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 IAM policies for Amazon EC2 .	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies.	11 June 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
New Amazon Linux AMI release		Amazon Linux AMI 2013.03 is released.	27 March 2013

Feature	API version	Description	Release date
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: <code>c1.xlarge</code> , <code>m2.2xlarge</code> , <code>m3.xlarge</code> , and <code>m3.2xlarge</code> .	19 March 2013
Copy an AMI from one Region to another	2013-02-01	You can copy an AMI from one Region to another, enabling you to launch consistent instances in more than one AWS Region quickly and easily. For more information, see Copy an AMI .	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either EC2-Classic or a VPC, or only into a VPC, on a region-by-region basis. If you can launch instances only into a VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013
High storage (hs1.8xlarge) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems.	20 December 2012

Feature	API version	Description	Release date
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
Linux Kernels		Updated AKI IDs; reorganized distribution kernels; updated PVOps section.	13 November 2012
M3 instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For detailed instance type specifications, see Specifications in the <i>Amazon EC2 Instance Types Guide</i> . For pricing information, see Amazon EC2 On-Demand Pricing .	31 October 2012
Spot Instance request status	2012-10-01	Spot Instance request status makes it easy to determine the state of your Spot requests.	14 October 2012
New Amazon Linux AMI release		Amazon Linux AMI 2012.09 is released.	11 October 2012

Feature	API version	Description	Release date
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012
Provisioned IOPS SSD for Amazon EBS	2012-07-20	Provisioned IOPS SSD volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times.	31 July 2012
High I/O instances for Amazon EC2	2012-06-15	High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage.	18 July 2012
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: <ul style="list-style-type: none"> • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services. 	11 June 2012

Feature	API version	Description	Release date
Spot Instance features that make it easier to get started and handle the potential of interruption.		<p>You can now manage your Spot Instances as follows:</p> <ul style="list-style-type: none"> Specify the amount you are willing to pay for Spot Instances using Auto Scaling launch configurations, and set up a schedule for specifying the amount you are willing to pay for Spot Instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the <i>Amazon EC2 Auto Scaling User Guide</i>. Get notifications when instances are launched or terminated. Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources. 	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	<p>Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere.</p> <p>Added support for timestamps in instance and system status checks.</p>	25 May 2012
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for cc2.8xlarge instances in a VPC.	26 April 2012

Feature	API version	Description	Release date
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
New Linux AMI release		Amazon Linux AMI 2012.03 is released.	28 March 2012
New AKI version		We've released AKI version 1.03 and AKIs for the AWS GovCloud (US) region.	28 March 2012
Medium instances, support for 64-bit on all AMIs, and a Java-based SSH Client	2011-12-15	Added support for a new instance type and 64-bit information. Added procedures for using the Java-based SSH client to connect to Linux instances.	7 March 2012
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
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 GRU Region and AKIs		Added information about the release of new AKIs for the SA-East-1 Region. This release deprecates the AKI version 1.01. AKI version 1.02 will continue to be backward compatible.	14 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.	01 December 2011

Feature	API version	Description	Release date
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 instances .	16 November 2011
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (cc2.8xlarge) to Amazon EC2.	14 November 2011
New PDX Region and AKIs		Added information about the release of new AKIs for the new US-West 2 Region.	8 November 2011
Spot Instances in Amazon VPC	2011-07-15	Added information about the support for Spot Instances in Amazon VPC. With this update, users can launch Spot Instances a virtual private cloud (VPC). By launching Spot Instances in a VPC, users of Spot Instances can enjoy the benefits of Amazon VPC.	11 October 2011
New Linux AMI release		Added information about the release of Amazon Linux AMI 2011.09. This update removes the beta tag from the Amazon Linux AMI, supports the ability to lock the repositories to a specific version, and provides for notification when updates are available to installed packages including security updates.	26 September 2011

Feature	API version	Description	Release date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of <code>ImportInstance</code> and <code>ImportVolume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of <code>ResumeImport</code> , users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide .	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes.	27 June 2011
Enabling Linux AMI to run user-provided kernels		Added information about the AKI version change from 1.01 to 1.02. This version updates the PVGRUB to address launch failures associated with t1.micro Linux instances. For more information, see User provided kernels .	20 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
Enabling Linux AMI to run user-provided kernels		Added information about enabling a Linux AMI to use PVGRUB Amazon Kernel Image (AKI) to run a user-provided kernel. For more information, see User provided kernels .	26 April 2011
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Dedicated Instances .	27 March 2011

Feature	API version	Description	Release date
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances. For more information, see Reserved Instances .	27 March 2011
New Amazon Linux reference AMI		The new Amazon Linux reference AMI replaces the CentOS reference AMI. Removed information about the CentOS reference AMI, including the section named Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI.	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance metadata and user data 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 integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2.	3 March 2011
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance.	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 Enable termination protection .	23 February 2011

Feature	API version	Description	Release date
Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI		Added information about how to correct clock drift for cluster instances running on Amazon's CentOS 5.4 AMI.	25 January 2011
VM Import	2010-11-15	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see the VM Import/Export User Guide .	15 December 2010
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see List and filter your resources and Tag your Amazon EC2 resources .	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances. For more information, see Ensure idempotency in the <i>Amazon EC2 API Reference</i> .	19 September 2010
Micro instances	2010-06-15	Amazon EC2 offers the t1.micro instance type for certain types of applications. For more information, see Burstable performance instances .	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Identity and access management for Amazon EC2 .	2 September 2010

Feature	API version	Description	Release date
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For detailed instance type specifications, see Specifications in the <i>Amazon EC2 Instance Types Guide</i> . For pricing information, see Amazon EC2 On-Demand Pricing .	12 July 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes.	14 June 2010
High-memory extra large instances	2009-11-30	Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For detailed instance type specifications, see Specifications in the <i>Amazon EC2 Instance Types Guide</i> . For pricing information, see Amazon EC2 On-Demand Pricing .	22 February 2010