User Guide for Windows Instances

Amazon Elastic Compute Cloud

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

Amazon Elastic Compute Cloud: User Guide for Windows 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 ... 5
Access EC2 ... 6
Pricing ... 7
Estimates, billing, and cost optimization ... 8
Set up ... 10
Sign up for an AWS account ... 10
Create an administrative user ... 11
Create a key pair ... 12
Create a security group ... 13
Get started tutorial ... 19
Overview ... 20
Prerequisites ... 20
Step 1: Launch an instance ... 21
Step 2: Connect to your instance ... 22
Step 3: Track your Free Tier usage ... 28
Step 4: Clean up your instance ... 30
Next steps ... 31
Best practices ... 32
Amazon Machine Images ... 37
Boot modes ... 37
Launch an instance ... 39
AMI boot mode parameter ... 44
Instance type boot mode ... 46
Instance boot mode ... 48
Operating system boot mode ... 50
UEFI variables ... 50
UEFI Secure Boot ... 51
AWS Windows AMIs ... 55
Select an initial Windows AMI ... 56
Keep your AMIs up to date ... 57
Virtualization types ... 57

Windows fast launch ... 58
Managed AWS Windows AMIs ... 81
Specialized Windows AMIs ... 91
AWS Windows AMI version history ... 104
Find a Windows AMI ... 239
Find a Windows AMI using the Amazon EC2 console ... 240
Find an AMI using the AWS Tools for Windows PowerShell ... 241
Find an AMI using the AWS CLI ... 241
Find the latest Windows AMI using Systems Manager ... 242
Use a Systems Manager parameter to find an AMI ... 243
Shared AMIs ... 247
Verified provider ... 247
Find shared AMIs ... 248
Make an AMI public ... 251
Share an AMI with organizations or OUs ... 260
Share an AMI with specific AWS accounts ... 270
Cancel having an AMI shared with your account ... 274
Use bookmarks ... 276
Best practices for shared Windows AMIs ... 277
Paid AMIs ... 278
Sell your AMI ... 279
Find a paid AMI ... 279
Purchase a paid AMI ... 281
Get the product code for your instance ... 282
Use paid support ... 282
Bills for paid and supported AMIs ... 283
Manage your AWS Marketplace subscriptions ... 283
AMI lifecycle ... 284
Create a custom Windows AMI ... 284
Modify an AMI ... 306
Copy an AMI ... 307
Store and restore an AMI ... 318
Deprecate an AMI ... 328
Disable an AMI ... 335
Archive AMI snapshots ... 341
Deregister your AMI ... 342

Automate the EBS-backed AMI lifecycle ... 347
Use encryption with EBS-backed AMIs ... 347
Instance-launching scenarios ... 347
Image-copying scenarios ... 351
Monitor AMI events ... 353
AMI events ... 354
Create Amazon EventBridge rules ... 357
Understand AMI billing ... 360
AMI billing fields ... 361
Find AMI billing information ... 363
Verify AMI charges on your bill ... 366
AMI quotas ... 366
Request a quota increase for AMIs ... 367
Instances ... 369
Windows instances ... 370
Instances and AMIs ... 370
Differences between Windows Server and Windows instances ... 371
Design your applications to run on Windows instances ... 373
Instance types ... 374
Instance type naming convention ... 375
Available instance types ... 377
Hardware specifications ... 387
Instances built on the Nitro System ... 388
Networking and storage features ... 390
Instance limits ... 398
General purpose ... 398
Compute optimized ... 481
Memory optimized ... 503
Storage optimized ... 541
Accelerated computing ... 553
High-performance computing ... 580
Find an instance type ... 585
Get recommendations ... 587
Change the instance type ... 595
EBS optimization ... 603
Supported instance types ... 604

Get maximum performance ... 657
View instances types that support EBS optimization ... 657
Enable EBS optimization at launch ... 659
Enable EBS optimization for an existing instance ... 660
Instance purchasing options ... 661
Determine the instance lifecycle ... 662
On-Demand Instances ... 663
Reserved Instances ... 666
Spot Instances ... 729
Dedicated Hosts ... 828
Dedicated Instances ... 889
Capacity Reservations ... 898
Instance lifecycle ... 981
Instance launch ... 984
Instance stop and start (Amazon EBS-backed instances only) ... 984
Instance hibernate (Amazon EBS-backed instances only) ... 985
Instance reboot ... 986
Instance retirement ... 986
Instance termination ... 986
Differences between reboot, stop, hibernate, and terminate ... 987
Launch ... 989
Stop and start ... 1066
Hibernate ... 1077
Reboot ... 1100
Retire ... 1102
Terminate ... 1105
Recover ... 1115
Connect ... 1121
Connect to your instance ... 1122
Connect to instances without requiring a public IPv4 address ... 1139
Connect your instance to a resource ... 1169
Configure instances ... 1211
Instance launch ... 1211
PV drivers ... 1364
AWS NVMe drivers ... 1397
Optimize CPU options ... 1405

Set the time ... 1484
Set the password ... 1493
Add Windows components ... 1494
Configure a secondary private IPv4 Address ... 1499
Run commands at launch ... 1506
Instance metadata and user data ... 1520
SQL Server Clustering in EC2 ... 1642
Install WSL ... 1642
Upgrade Windows instances ... 1643
Perform an in-place upgrade ... 1644
Perform an automated upgrade ... 1648
Migrate to latest generation instance types ... 1660
Migrate Microsoft SQL Server from Windows to Linux ... 1670
Troubleshoot an upgrade ... 1670
Identify instances ... 1671
Inspect the instance identity document ... 1671
Inspect the system UUID ... 1671
Inspect the system virtual machine generation identifier ... 1672
Set up a Windows HPC cluster ... 1672
Prerequisites ... 1673
Step 1: Create your security groups ... 1673
Step 2: Set up your Active Directory domain controller ... 1677
Step 3: Configure your head node ... 1678
Step 4: Set up the compute node ... 1680
Step 5: Scale your HPC compute nodes (optional) ... 1682
Fleets ... 1684
EC2 Fleet ... 1685
EC2 Fleet limitations ... 1687
Burstable performance instances ... 1687
EC2 Fleet request types ... 1688
EC2 Fleet configuration strategies ... 1715
Work with EC2 Fleets ... 1753
Spot Fleet ... 1780
Spot Fleet request types ... 1780
Spot Fleet configuration strategies ... 1781
Work with Spot Fleets ... 1820

CloudWatch metrics for Spot Fleet ... 1853
Automatic scaling for Spot Fleet ... 1857
Monitor fleet events ... 1866
EC2 Fleet event types ... 1866
Spot Fleet event types ... 1873
Create EventBridge rules ... 1880
Tutorials ... 1890
Tutorial: Use EC2 Fleet with instance weighting ... 1890
Tutorial: Use EC2 Fleet with On-Demand as the primary capacity ... 1894
Tutorial: Launch On-Demand Instances using targeted Capacity Reservations ... 1896
Tutorial: Launch instances into Capacity Blocks ... 1902
Tutorial: Use Spot Fleet with instance weighting ... 1904
Example configurations ... 1908
EC2 Fleet example configurations ... 1908
Spot Fleet example configurations ... 1929
Fleet quotas ... 1947
Request a quota increase for target capacity ... 1949
Elastic Graphics ... 1950
Elastic Graphics basics ... 1951
Pricing for Elastic Graphics ... 1954
Elastic Graphics limitations ... 1954
Work with Elastic Graphics ... 1955
Configure your security groups ... 1955
Launch an instance with an Elastic Graphics accelerator ... 1957
Install the required software for Elastic Graphics ... 1958
Verify Elastic Graphics functionality on your instance ... 1958
View Elastic Graphics information ... 1961
Submit feedback ... 1962
Elastic Graphics maintenance ... 1963
How will I be notified? ... 1963
What do I need to do? ... 1964
What happens when an accelerator reaches its retirement date? ... 1964
Use CloudWatch metrics to monitor Elastic Graphics ... 1965
Elastic Graphics metrics ... 1965
Elastic Graphics dimensions ... 1966
View CloudWatch metrics for Elastic Graphics ... 1966

Create CloudWatch alarms to monitor Elastic Graphics ... 1967
Troubleshoot ... 1967
Investigate application performance issues ... 1968
Resolve unhealthy status issues ... 1970
Why am I seeing multiple ENIs? ... 1972
Monitor ... 1973
Automated and manual monitoring ... 1974
Automated monitoring tools ... 1975
Manual monitoring tools ... 1976
Best practices for monitoring ... 1977
Monitor the status of your instances ... 1977
Instance status checks ... 1978
State change events ... 1985
Scheduled events ... 1988
Monitor your instances using CloudWatch ... 2020
Instance alarms ... 2021
Enable detailed monitoring ... 2022
List available metrics ... 2024
Get statistics for metrics ... 2050
Graph metrics ... 2061
Create an alarm ... 2062
Create alarms that stop, terminate, reboot, or recover an instance ... 2063
Automate using EventBridge ... 2076
Amazon EC2 event types ... 2077
Log API calls with AWS CloudTrail ... 2078
Amazon EC2 and Amazon EBS information in CloudTrail ... 2078
Understand Amazon EC2 and Amazon EBS log file entries ... 976
Audit users that connect via EC2 Instance Connect ... 2080
Monitor your .NET and SQL Server applications ... 2082
Networking ... 2084
Regions and Zones ... 2085
Regions ... 2085
Availability Zones ... 2092
Local Zones ... 2097
Wavelength Zones ... 2099
AWS Outposts ... 2102

Instance IP addressing ... 2104
Private IPv4 addresses ... 2105
Public IPv4 addresses ... 2106
Elastic IP addresses (IPv4) ... 2107
IPv6 addresses ... 2107
Work with the IPv4 addresses for your instances ... 2109
Work with the IPv6 addresses for your instances ... 2111
Multiple IP addresses ... 2114
EC2 instance hostnames ... 2126
Link-local addresses ... 2126
Instance hostname types ... 2127
Types of EC2 hostnames ... 2127
Where you see Resource name and IP name ... 2129
How to decide whether to choose Resource name or IP name ... 2131
Modify Hostname type and DNS Hostname configurations ... 2131
Bring your own IP addresses ... 2133
BYOIP definitions ... 2134
Requirements and quotas ... 2134
Onboarding prerequisites ... 2135
Onboard your BYOIP ... 2144
Work with your address range ... 2149
Validate your BYOIP ... 2150
Regional availability ... 2154
Local Zone availability ... 2154
Learn more ... 2155
Elastic IP addresses ... 2155
Elastic IP address pricing ... 2156
Elastic IP address basics ... 2156
Work with Elastic IP addresses ... 2157
Elastic IP address quota ... 2172
Network interfaces ... 2173
Network interface basics ... 2174
IP addresses per network interface per instance type ... 2176
Work with network interfaces ... 2220
Best practices for configuring network interfaces ... 2231
Scenarios for network interfaces ... 2232

Requester-managed network interfaces ... 2236
Assign prefixes ... 2237
Network bandwidth ... 2254
Available instance bandwidth ... 2255
Monitor instance bandwidth ... 2256
Enhanced networking ... 2257
Enhanced networking support ... 2258
Enable enhanced networking on your instance ... 2258
Elastic Network Adapter (ENA) ... 2258
ENA Express ... 2279
Intel 82599 VF ... 2295
Operating system optimizations ... 2300
Network performance metrics ... 2302
Troubleshoot ENA Windows driver ... 2306
Instance topology ... 2322
How it works ... 2323
Prerequisites ... 2327
Examples ... 2328
Placement groups ... 2340
Placement strategies ... 2341
Rules and limitations ... 2344
Work with placement groups ... 2347
Share a placement group ... 2360
Placement groups on AWS Outposts ... 2366
Network MTU ... 2367
Jumbo frames (9001 MTU) ... 2368
Path MTU Discovery ... 2369
Check the path MTU between two hosts ... 2370
Check and set the MTU on your Windows instance ... 2371
Troubleshoot ... 2373
Virtual private clouds ... 2373
Your default VPCs ... 2374
Create additional VPCs ... 2374
Access the internet from your instances ... 2375
Shared subnets ... 2376
IPv6-only subnets ... 2376

RDP access to your instances ... 2376
Ports and Protocols ... 2376
AllJoyn Router ... 2377
Cast to Device ... 2378
Core Networking ... 2382
Delivery Optimization ... 2444
Diag Track ... 2444
DIAL Protocol Server ... 2445
Distributed File System (DFS) Management ... 2445
File and Printer Sharing ... 2446
File Server Remote Management ... 2451
ICMP v4 All ... 2452
Microsoft Edge ... 2452
Microsoft Media Foundation Network Source ... 2453
Multicast ... 2453
Remote Desktop ... 2454
Windows Device Management ... 2457
Windows Feature Experience Pack ... 2460
Windows Firewall Remote Management ... 2460
Windows Remote Management ... 2460
Security ... 2462
Infrastructure security ... 2463
Network isolation ... 2464
Isolation on physical hosts ... 2464
Controlling network traffic ... 2464
Resilience ... 2466
Data protection ... 2467
Amazon EBS data security ... 2468
Encryption at rest ... 2469
Encryption in transit ... 2470
Windows VBS ... 2472
Credential Guard ... 2472
Identity and access management ... 2478
Network access to your instance ... 2479
Amazon EC2 permission attributes ... 2479
IAM and Amazon EC2 ... 2479

IAM policies ... 2481
AWS managed policies ... 2552
IAM roles ... 2554
Network access ... 2570
Key pairs ... 2574
Create a key pair ... 2575
Tag a key pair ... 2582
Describe your key pairs ... 2584
Delete your key pair ... 2589
Verify the fingerprint ... 2590
Security groups ... 2592
Security group rules ... 2594
Connection tracking ... 2597
Default and custom security groups ... 2602
Work with security groups ... 2604
Security group rules for different use cases ... 2614
AWS PrivateLink ... 2621
Create an interface VPC endpoint ... 2621
Create an endpoint policy ... 2621
Configuration management ... 2623
Update management ... 2624
Change management ... 2624
Compliance validation ... 2625
Audit and accountability ... 2626
NitroTPM ... 2627
Considerations ... 2628
Prerequisites ... 2628
Verify whether an AMI is enabled for NitroTPM ... 2629
Enable or stop using NitroTPM on an instance ... 2631
Storage ... 2633
Amazon EBS ... 2634
Instance store ... 2635
Instance store volume and data lifetime ... 2636
Instance store volumes ... 2638
Add instance store volumes ... 2659
SSD instance store volumes ... 2664

File storage ... 2665
Amazon S3 ... 2666
Amazon EFS ... 2668
Amazon FSx ... 2668
Amazon File Cache ... 2674
Instance volume limits ... 2674
Volume limits for instances built on the Nitro System ... 2675
Volume limits for Xen-based instances ... 2677
Root device volume ... 2677
Configure the root volume to persist ... 2678
Confirm that a root volume is configured to persist ... 2680
Change the initial size of the root volume ... 2682
Device names ... 2682
Available device names ... 2683
Device name considerations ... 2684
Block device mappings ... 2684
Block device mapping concepts ... 2685
AMI block device mapping ... 2689
Instance block device mapping ... 2692
Map disks to volumes ... 2697
List NVMe volumes ... 2698
List volumes ... 2704
VSS-based snapshots ... 2713
What is AWS VSS? ... 2714
Prerequisites ... 2715
Create VSS snapshots ... 2726
Troubleshooting ... 2737
Restore volumes from VSS snapshots ... 2740
Version history ... 2741
Resources and tags ... 2745
Recycle Bin ... 2745
How does it work? ... 2746
Supported resources ... 2747
Considerations ... 2747
Quotas ... 2751
Related services ... 2751

Pricing ... 2751
Required IAM permissions ... 2752
Work with retention rules ... 2757
Work with resources in the Recycle Bin ... 2771
Monitor Recycle Bin ... 2781
Resource locations ... 2800
Resource IDs ... 2801
List and filter your resources ... 2802
Console steps ... 2802
CLI and API steps ... 2808
Global View (cross-Region) ... 2811
Global View ... 2811
Tag your resources ... 2814
Tag basics ... 2815
Tag your resources ... 2816
Tag restrictions ... 2820
Tags and access management ... 2821
Tag your resources for billing ... 2822
Work with tags using the console ... 2822
Work with tags using the command line ... 2828
Work with instance tags in instance metadata ... 2832
Add tags to a resource using CloudFormation ... 2836
Service quotas ... 2837
View your current quotas ... 2837
Request an increase ... 2838
Restriction on email sent using port 25 ... 2839
Usage reports ... 2839
Track your Free Tier usage ... 2840
Troubleshoot ... 2843
Common issues ... 2843
EBS volumes don't initialize on Windows Server 2016 and 2019 ... 2844
Boot an EC2 Windows instance into Directory Services Restore Mode (DSRM) ... 2845
Instance loses network connectivity or scheduled tasks don't run when expected ... 2848
Unable to get console output ... 2848
Windows Server 2012 R2 not available on the network ... 2849
Disk signature collision ... 2849

Common messages ... 2851
"Password is not available" ... 2851
"Password not available yet" ... 2852
"Cannot retrieve Windows password" ... 2852
"Waiting for the metadata service" ... 2853
"Unable to activate Windows" ... 2857
"Windows is not genuine (0x80070005)" ... 2859
"No Terminal Server License Servers available to provide a license" ... 2859
"Some settings are managed by your organization" ... 2860
Troubleshoot launch issues ... 2860
Invalid device name ... 2861
Instance limit exceeded ... 2862
Insufficient instance capacity ... 2862
The requested configuration is currently not supported. Please check the documentation for supported configurations. ... 2863
Instance terminates immediately ... 2863
High CPU usage shortly after Windows starts ... 2865
Insufficient permissions ... 2865
Connect to your instance ... 2866
Remote Desktop can't connect to the remote computer ... 2867
Error using the macOS RDP client ... 2871
RDP displays a black screen instead of the desktop ... 2871
Unable to remotely log on to an instance with a user that is not an administrator ... 2872
Troubleshooting Remote Desktop issues using AWS Systems Manager ... 2872
Enable Remote Desktop on an EC2 instance with remote registry ... 2876
I've lost my private key. How can I connect to my Windows instance? ... 2877
Troubleshoot an unreachable instance ... 2878
Instance reboot ... 2878
Instance console output ... 2878
Capture a screenshot of an unreachable instance ... 2879
Common screenshots ... 2881
Instance recovery when a host computer fails ... 2890
Reset a lost or expired Windows administrator password ... 2891
Reset using EC2Launch v2 ... 2892
Reset Using EC2Config ... 2897
Reset using EC2Launch ... 2903

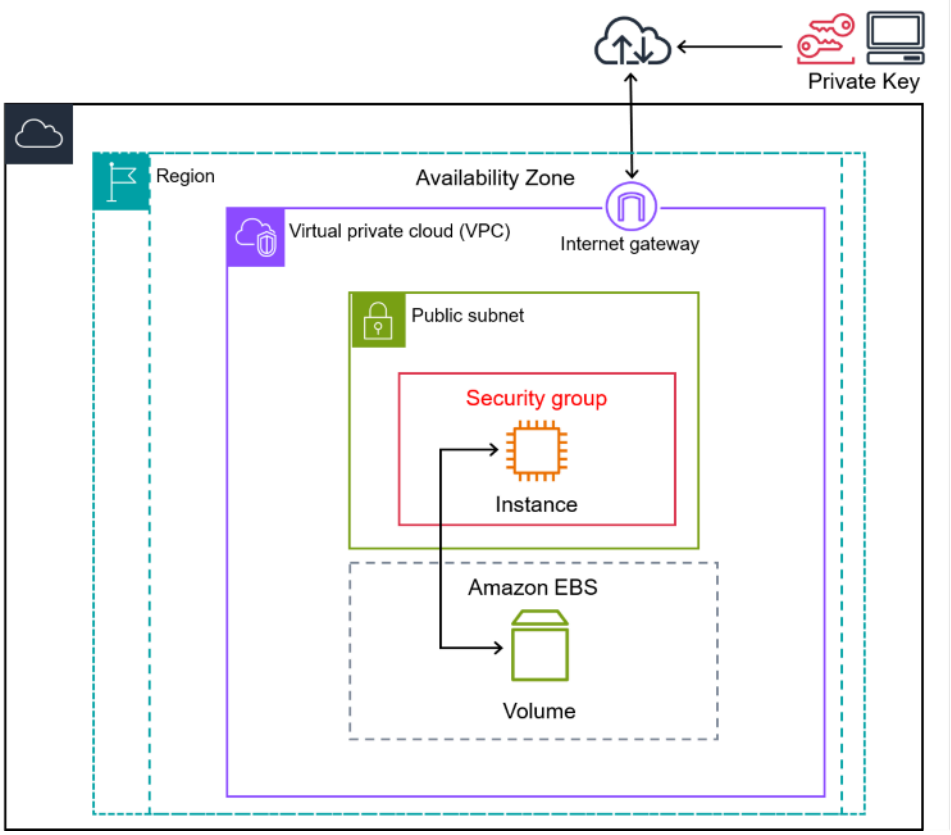
Stop your instance ... 2908
Force stop the instance ... 2908
Create a replacement instance ... 2909
Terminate your instance ... 2911
Instance terminates immediately ... 2911
Delayed instance termination ... 2911
Terminated instance still displayed ... 2912
Error: The instance may not be terminated. Modify its 'disableApiTermination' instance attribute ... 2912
Instances automatically launched or terminated ... 2912
Troubleshoot Sysprep ... 2913
EC2Rescue for Windows Server ... 2914
Use the GUI ... 2915
Use the command line ... 2921
Use Systems Manager ... 2928
EC2 Serial Console ... 2932
Prerequisites ... 2933
Configure access to the EC2 Serial Console ... 2936
Connect to the EC2 Serial Console ... 2944
Disconnect from the EC2 Serial Console ... 2952
Troubleshoot your instance using the EC2 Serial Console ... 2953
Send a diagnostic interrupt ... 2959
Supported instance types ... 2959
Prerequisites ... 2959
Send a diagnostic interrupt ... 2960
AWS Systems Manager for Microsoft System Center VMM ... 2961
Features ... 891
Limitations ... 323
Requirements ... 2962
Get started ... 2962
Set up ... 2962
Sign up for AWS ... 2963
Set up access for users ... 2963
Deploy the add-in ... 2967
Provide your AWS credentials ... 2967
Manage EC2 Instances ... 2968

Create an EC2 Instance 2969
View your instances 2972
Connect to your instance 2972
Reboot your instance 2973
Stop your instance 2974
Start your instance 2974
Terminate your instance 2974
Import Your VM 2975
Prerequisites 2975
Import your virtual machine 2976
Check the import task status 2977
Back up your imported instance 2978
Troubleshoot 2978
Error: Add-in cannot be installed 2978
Installation errors 2979
Check the log file 2980
Errors importing a virtual machine 2980
Uninstall the add-in 2980
Related information 2982
Document history 2984
History for previous years 3017

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 Windows-based instances on Amazon EC2. See the EC2 User Guide for Linux Instances for information to help you run Linux-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?

Topics

- Features of Amazon EC2
- Get started with Amazon EC2
- Related services
- Access Amazon EC2
- Pricing for Amazon EC2

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, Availability Zones, Local Zones, AWS Outposts, and Wavelength 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. Windows-specific features and use case information can be found at Windows Server on AWS.

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 Windows 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 Windows instances
- Connect to your Windows instance
- Transfer files to Windows instances

Learn the basics of Amazon EC2

- Amazon EC2 Windows instances
- Instance types
- Tags

Read about networking and security

- Key pairs
- Security groups
- Elastic IP addresses
- Virtual private clouds

Review your storage options

- Amazon EBS
- Instance store

Work with Windows instances

- AWS Systems Manager Run Command in the *AWS Systems Manager User Guide*
- Tutorial: Get started with Amazon EC2 Windows instances

Troubleshoot EC2

- Troubleshoot EC2 Windows instances
- AWS re:Post

Related services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. In addition, you can provision EC2 resources using other AWS services, such as the following:

- Amazon EC2 Auto Scaling

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

- AWS CloudFormation

  Helps you model and set up your AWS resources using templates.

- AWS Elastic Beanstalk

  Deploy and manage applications in the AWS Cloud without having to understand the underlying infrastructure.

- AWS OpsWorks

  Automate how servers are configured, deployed, and managed across your Amazon EC2 instances using Chef and Puppet.

- 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.

Additional related services

- Amazon Lightsail

To build websites or web applications, you can deploy and manage basic cloud resources using Amazon Lightsail. To compare the features of Amazon EC2 and Lightsail for your use case, see Amazon Lightsail or Amazon EC2.

- Elastic Load Balancing

  Automatically distribute incoming application traffic across multiple instances.

- Amazon Relational Database Service (Amazon RDS)

  Set up, operate, and scale a managed relational database in the cloud. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups.

- Amazon Elastic Container Service (Amazon ECS)

  Deploy, manage, and scale containerized applications on a cluster of EC2 instances.

- Amazon Elastic Kubernetes Service (Amazon EKS)

  Run your Kubernetes applications on AWS.

- Amazon CloudWatch

  Monitor your instances and Amazon EBS volumes.

- Amazon GuardDuty

  Detect potentially unauthorized or malicious use of your EC2 instances.

- AWS Backup

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

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

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

If you have questions concerning AWS billing, accounts, and events, contact AWS Support.

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

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

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 an administrative user
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 an administrative 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 an administrative user

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 an administrative user

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 an administrative 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 administrative user

- 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*.

Create a key pair

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

If you haven't created a key pair already, you can create one by using the Amazon EC2 console. Note that if you plan to launch instances in multiple 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.

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 RDP. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple 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 **RDP**. For **Source**, do one of the following:

      - Choose **My IP** to automatically add the public IPv4 address of your local computer.
      - Choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix `/32`, for example, `203.0.113.25/32`. If your company 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 RDP. This would allow access to your instance from all IP addresses on the internet. This is acceptable for a short time in a test environment, but it is unsafe for production environments.

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

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=3389,ToPort=3389,IpRanges='[{CidrIp=172.31.0.0/16,Description="RDP
 from private network"}]' \

 IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges='[{CidrIp=203.0.113.25/32,Description="RD
 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 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.

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 Windows instances.

Tutorial: Get started with Amazon EC2 Windows 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 Windows instance. An *instance* is a virtual server in the AWS Cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

When you sign up for AWS, you can get started with Amazon EC2 using the AWS Free Tier. If you created your AWS account less than 12 months ago, and have not already exceeded the Free Tier benefits for Amazon EC2, it 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 Linux instance, see this tutorial in the *Amazon EC2 User Guide for Linux Instances*: Get started with Amazon EC2 Linux instances.
- If you'd prefer to use the command line, see this tutorial in the *AWS Command Line Interface User Guide*: Using Amazon EC2 through the AWS CLI.

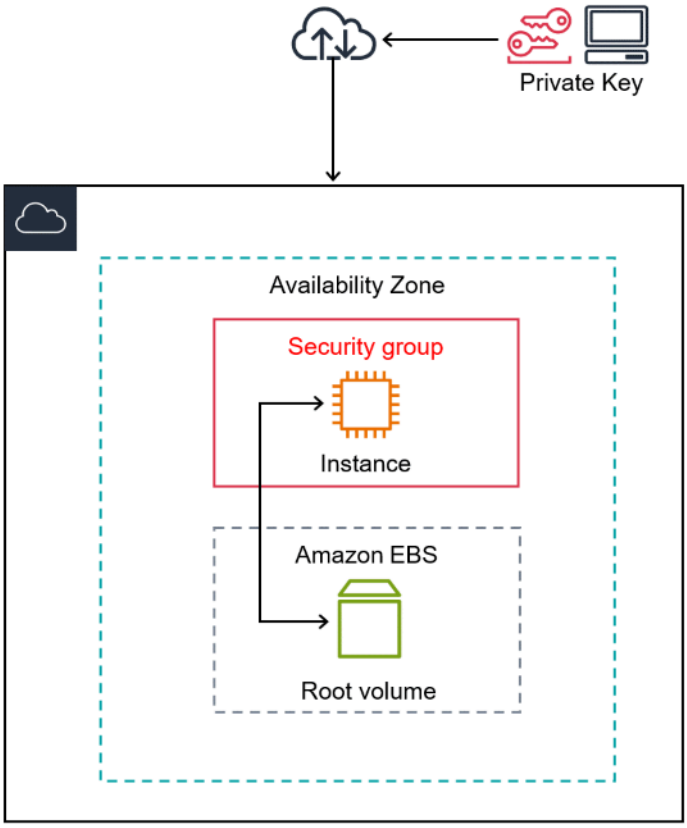
Contents

- 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 Windows 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 Windows. This is the operating system (OS) for your instance.
   b. From **Amazon Machine Image (AMI)**, select the AMI for Windows Server 2016 Base or later.. 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. Note that you must select an **RSA** key. **ED25519** keys are not supported for Windows instances.

> **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

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

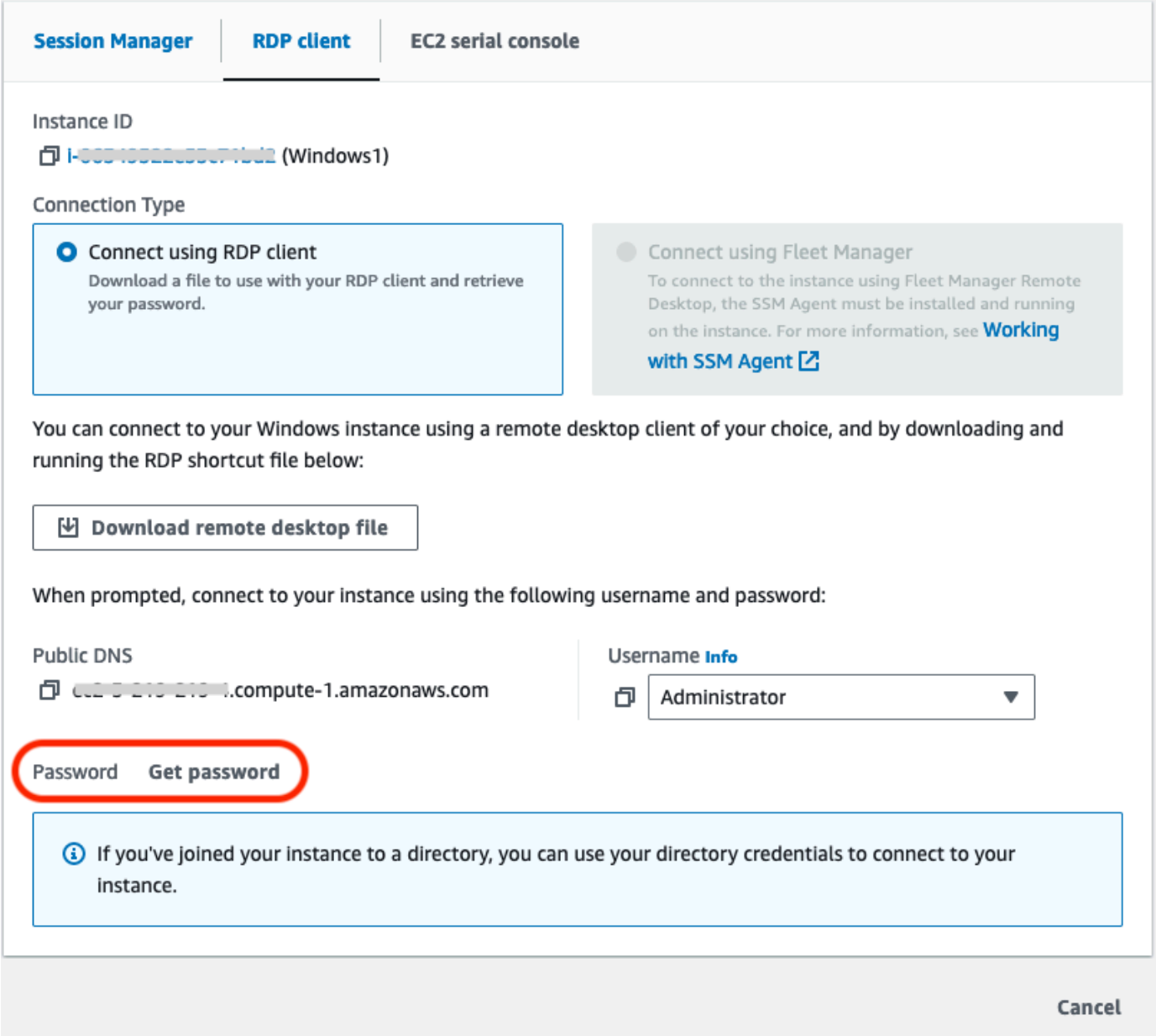
The default username for the Administrator account depends on the language of the operating system (OS) contained in the AMI. To ascertain the correct username, identify the language of your AMI's OS, and then choose the corresponding username. For example, for an English OS, the username is `Administrator`,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

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

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

To connect to your Windows instance using an RDP client

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and then choose **Connect**.
4. On the **Connect to instance** page, choose the **RDP client** tab.
5. For **Username**, choose the default username for the Administrator account. The username you choose must match the language of the operating system (OS) contained in the AMI that you used to launch your instance. If there is no username in the same language as your OS, choose **Administrator (Other)**.
6. Choose **Get password**.

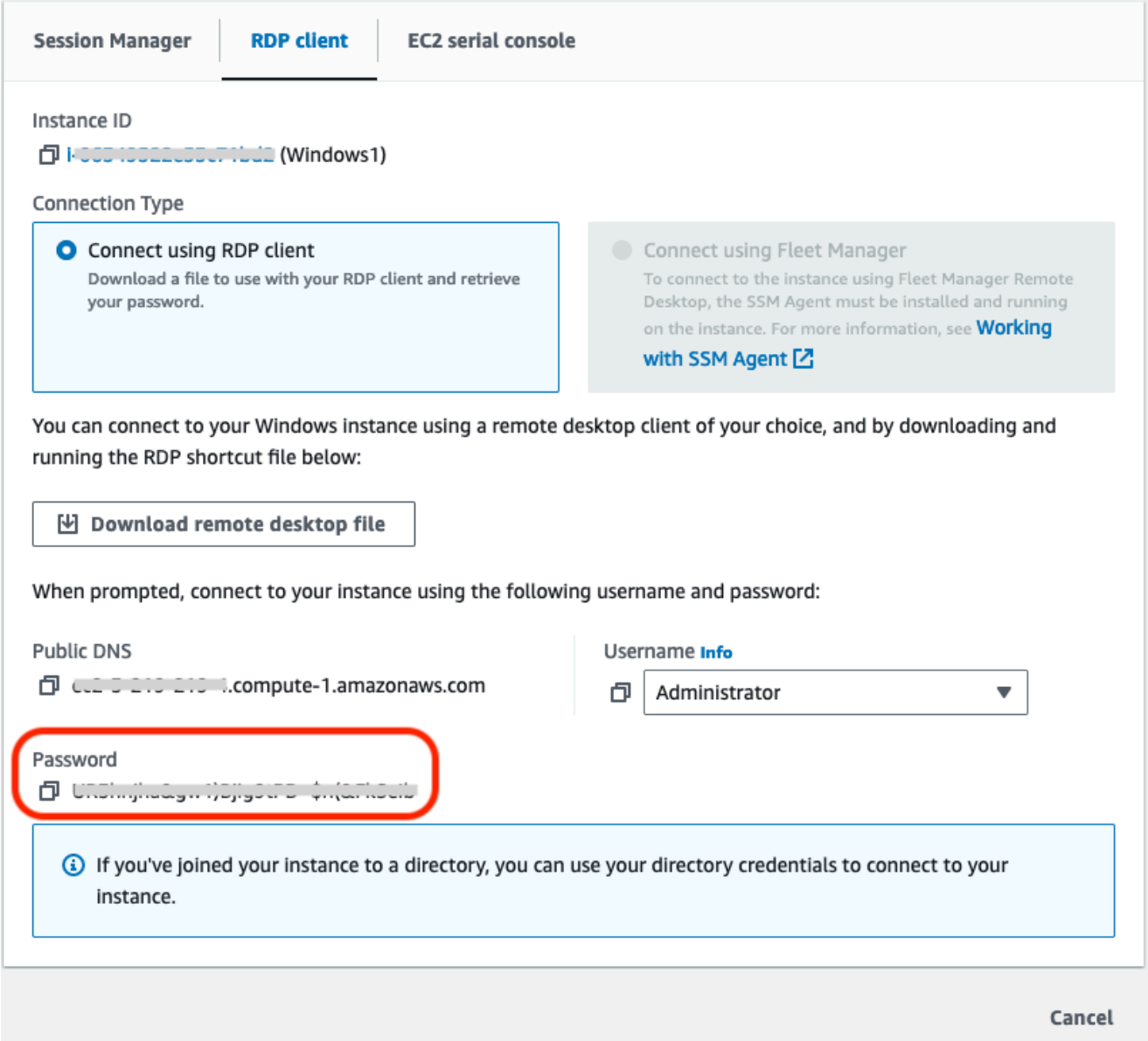


7. On the **Get Windows password** page, do the following:

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

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

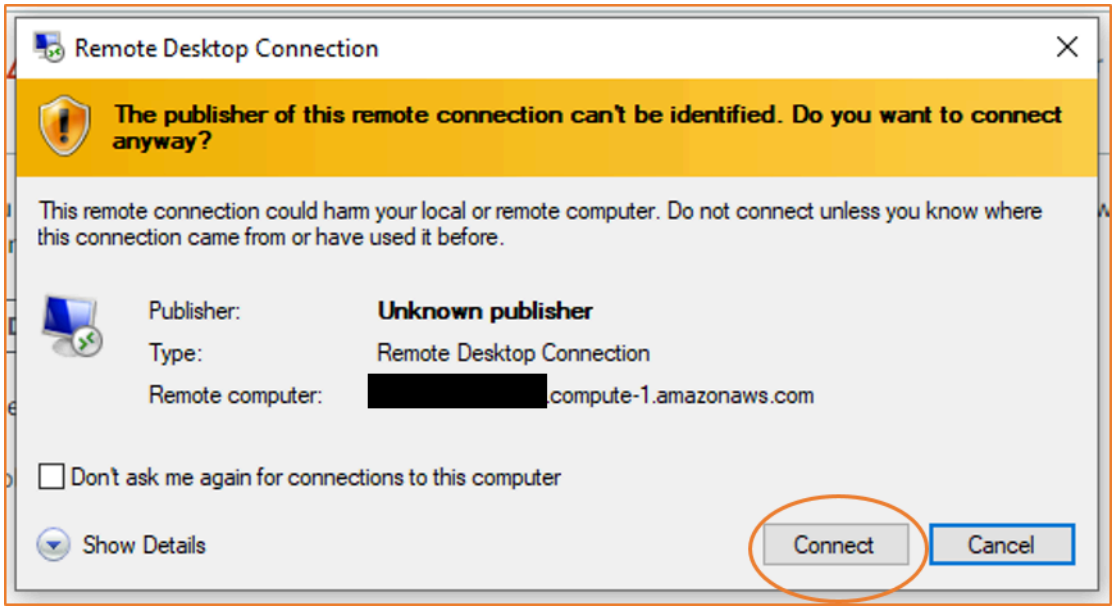
   c. Copy the password and save it in a safe place. This password is required to connect to the instance.



8. Choose **Download remote desktop file**. Your browser prompts you to either open or save the RDP shortcut file. When you have finished downloading the file, choose **Cancel** to return to the **Instances** page.

   - If you opened the RDP file, you'll see the **Remote Desktop Connection** dialog box.
   - If you saved the RDP file, navigate to your downloads directory, and open the RDP file to display the dialog box.

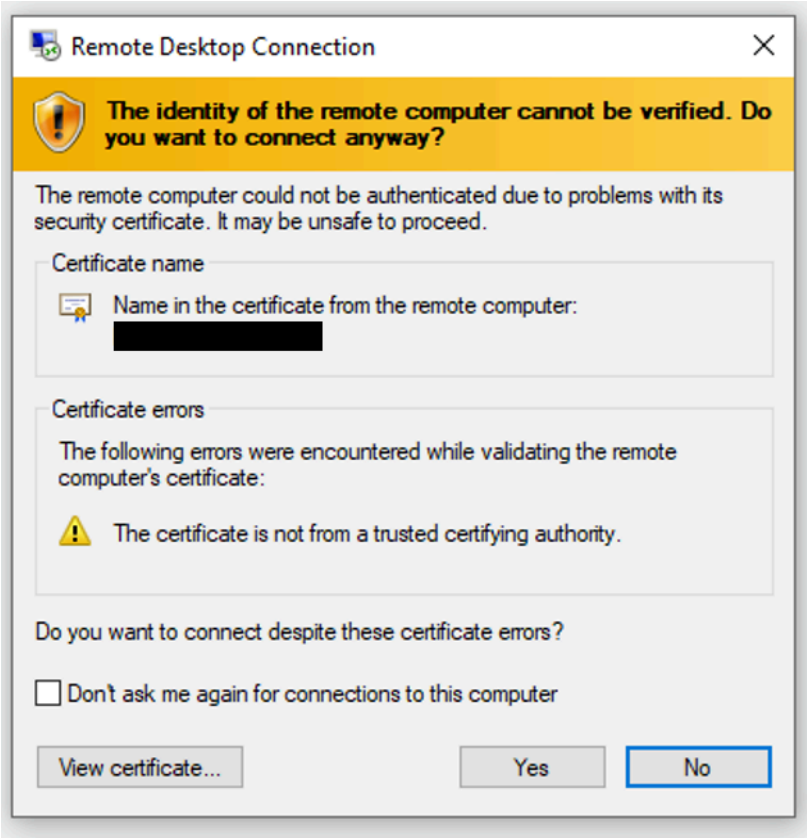
9. You might get a warning that the publisher of the remote connection is unknown. Choose **Connect** to continue to connect to your instance.



10. The administrator account is chosen by default. Paste the password that you copied previously, and then choose **Continue**.

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

11. Due to the nature of self-signed certificates, you might get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer. Alternatively, if you trust the certificate, choose **Yes** (Windows) or **Continue** (Mac OS X) to skip the following steps.



a. [Windows] Choose **View certificate**.

   [Mac OS X] Choose **Show Certificate**.

b. [Windows] Choose the **Details** tab, and scroll down to **Thumbprint**.

   [Mac OS X] Expand **Details**, and scroll down to **SHA1 Fingerprints**.

   This is the unique identifier for the remote computer's security certificate.

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

d. In the system log output, look for `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint (Windows) or fingerprint (Mac OS X) of the certificate, you have verified the identity of the remote computer.

e. [Windows] Return to the **Certificate** dialog box and choose **OK**.

   [Mac OS X computer] Return to the **Verify Certificate** dialog box and choose **Continue**.

f. [Windows] Choose **Yes** in the **Remote Desktop Connection** window to connect to your instance.

[Mac OS X] The process automatically commences connecting to your instance. Note that you might need to switch spaces to see the Windows instance screen. For more information, see See open windows and spaces in Mission Control on Mac.

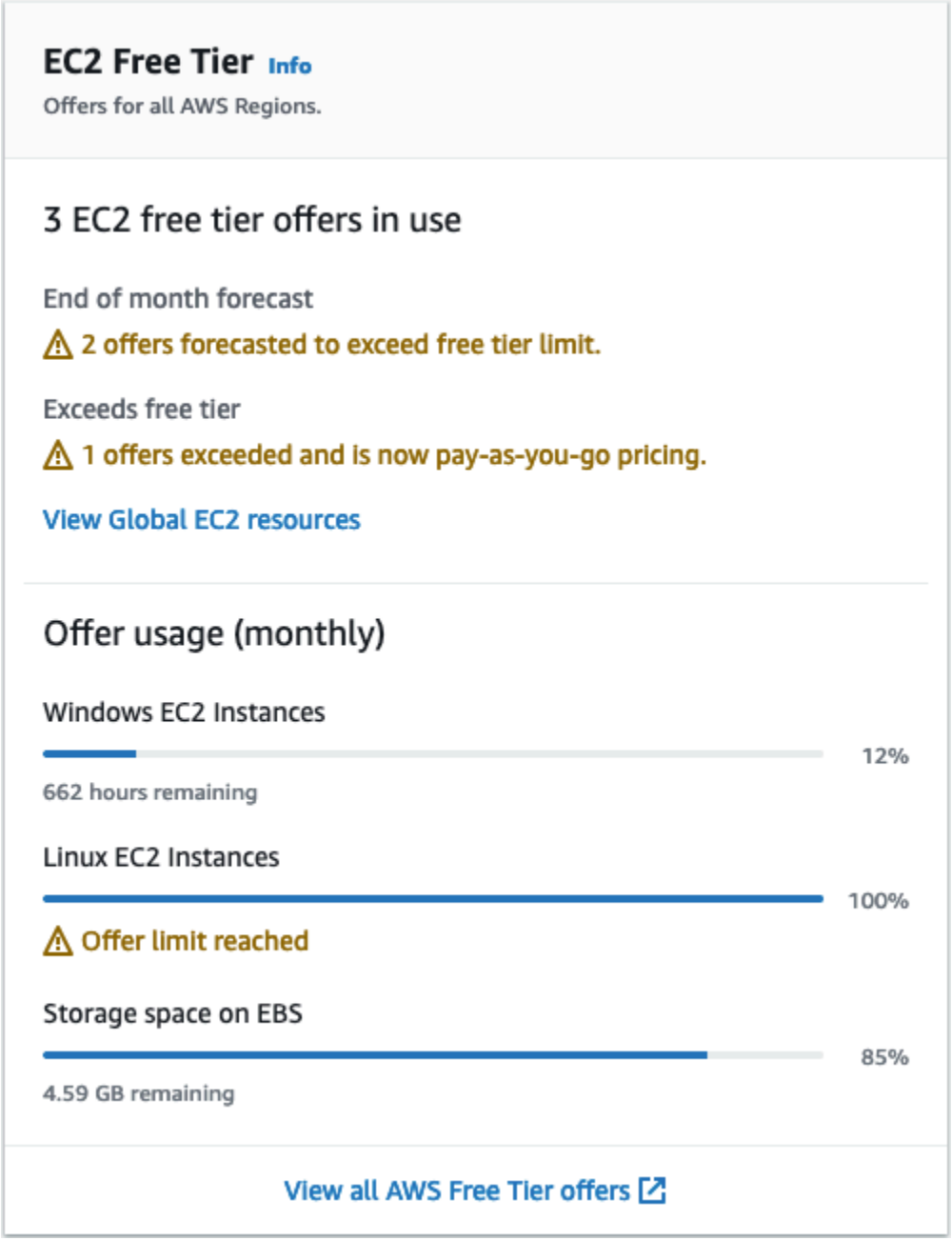
Step 3: Track your Free Tier usage

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).



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 your instance.

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*.
- 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 Windows on Amazon EC2

To ensure the best results from running Windows on Amazon EC2, we recommend that you perform the following best practices.

- Update drivers
- Use the latest Windows AMIs
- Security
- Storage
- Resource management
- Backup and recovery
- Networking

Update Windows drivers

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

- Use SNS topics to receive updates for new driver releases.
- Use the AWS Systems Manager Automation runbook AWSSupport-UpgradeWindowsAWSDrivers to easily apply the updates across your instances.

Launch new instances with the latest Windows AMIs

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

- To view updates to each release of the AWS Windows AMIs, see AWS Windows AMI version history.
- To build with the latest available AMIs, see Query for the Latest Windows AMI Using Systems Manager Parameter Store.

Test system/application performance before migration

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

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

Update launch agents

Update to the latest EC2Launch v2 agent to ensure that the latest enhancements are applied across your fleet. For more information, see Migrate to EC2Launch v2.

If you have a mixed fleet, or if you want to continue to use the EC2Launch (Windows Server 2016 and 2019) or EC2 Config (legacy OS only) agents, update to the latest versions of the respective agents.

Automatic updates are supported on the following combinations of Windows Server version and launch agents. You can opt in to automatic updates in the SSM Quick Setup Host Management console under **Amazon EC2 Launch Agents**.

Windows Version	EC2Launch v1	EC2Launch v2
2016	✓	✓
2019	✓	✓
2022		✓

- For more information about updating to EC2Launch v2, see Install the latest version of EC2Launch v2.
- For information to manually update EC2Config, see Install the Latest Version of EC2Config.
- For information to manually update EC2Launch, see Install the Latest Version of EC2Launch.

Security

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

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

- **Least Access:** Grant access only to systems and locations that are trusted and expected. This applies to all Microsoft products such as Active Directory, Microsoft business productivity servers, and infrastructure services such as Remote Desktop Services, reverse proxy servers, IIS web servers,and more. Use AWS capabilities such as Amazon EC2 instance security groups, network access control lists (ACLs), and Amazon VPC public/private subnets to layer security across multiple locations in an architecture. Within a Windows instance, customers can use Windows Firewall to further layer a defense-in-depth strategy within their deployment. Install only the OS components and applications that are necessary for the system to function as designed. Configure infrastructure services such as IIS to run under service accounts, or to use features such as application pool identities to access resources locally and remotely across your infrastructure.
- **Least Privilege:** Determine the minimum set of privileges that instances and accounts need in order to perform their functions. Restrict these servers and users to only allow these defined permissions. Use techniques such as Role Based Access Controls to reduce the surface area of administrative accounts, and create the most limited roles to accomplish a task. Use OS features such as Encrypting File System (EFS) within NTFS to encrypt sensitive data at rest, and control application and user access to it.
- **Configuration Management:** Create a baseline server configuration that incorporates up-to-date security patches and host-based protection suites that include anti-virus, anti-malware, intrusion detection/prevention, and file integrity monitoring. Assess each server against the current recorded baseline to identify and flag any deviations. Ensure each server is configured to generate and securely store appropriate log and audit data. For more information, see AWS Windows AMIs.
- **Change Management:** Create processes to control changes to server configuration baselines and work toward fully automated change processes. Also, leverage Just Enough Administration (JEA) with Windows PowerShell DSC to limit administrative access to the minimum required functions.

- **Patch Management:** Implement processes that regularly patch, update, and secure the operating system and applications on your EC2 instances. For more information, see Update your Windows instance.
- **Audit Logs:** Audit access and all changes to Amazon EC2 instances to verify server integrity and ensure only authorized changes are made. Leverage features such as Enhanced Logging for IIS to enhance default logging capabilities. AWS capabilities such as VPC Flow Logs and AWS CloudTrail are also available to audit network access, including allowed/denied requests and API calls, respectively.

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

- 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.

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

- Boot modes
- AWS Windows AMIs
- Find a Windows AMI
- Shared AMIs
- Paid AMIs
- AMI lifecycle
- Use encryption with EBS-backed AMIs
- Monitor AMI events using Amazon EventBridge
- Understand AMI billing information
- AMI quotas

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
- 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

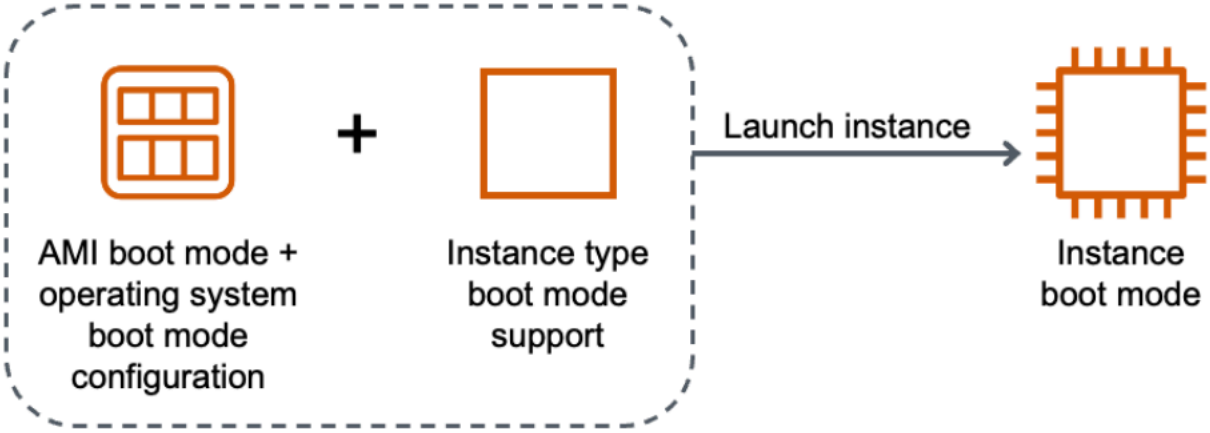
Limitations

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 Windows 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

```
C:\> aws ec2 describe-instance-types --filters Name=supported-boot-mode,Values=uefi
 Name=processor-info.supported-architecture,Values=x86_64 --query "InstanceTypes[*].
[InstanceType]" --output text | sort

c5.12xlarge
c5.18xlarge
c5.24xlarge
c5.2xlarge
c5.4xlarge
c5.9xlarge
c5.large
...
```

PowerShell

```
PS C:\> Get-EC2InstanceType | `
 Where-Object {
     $_.SupportedBootModes -Contains "uefi" -and `
     $_.ProcessorInfo.SupportedArchitectures -eq "x86_64"
     } | `
 Sort-Object InstanceType | `
 Format-Table InstanceType -GroupBy CurrentGeneration

CurrentGeneration: True

InstanceType
------------
c5.12xlarge
c5.18xlarge
c5.24xlarge
c5.2xlarge
c5.4xlarge
...
```

To see the available instance types for Windows 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

```
C:\> aws ec2 describe-instance-types --filters Name=supported-boot-
mode,Values=uefi Name=bare-metal,Values=false Name=processor-info.supported-
architecture,Values=x86_64 --query "InstanceTypes[*].[InstanceType]" --output text |
 sort

c5.12xlarge
c5.18xlarge
c5.24xlarge
c5.2xlarge
```

```
...
```

PowerShell

```
PS C:\> Get-EC2InstanceType | `
    Where-Object { `
        $_.SupportedBootModes -Contains "uefi" -and `
        $_.BareMetal -eq $False -and `
        $_.ProcessorInfo.SupportedArchitectures -eq "x86_64"
        } | `
    Sort-Object InstanceType  | `
    Format-Table InstanceType, SupportedBootModes, BareMetal,
 @{Name="SupportedArchitectures";
 Expression={$_.ProcessorInfo.SupportedArchitectures}}

InstanceType    SupportedBootModes  BareMetal SupportedArchitectures
------------    ------------------  --------- ----------------------
c5.12xlarge     {legacy-bios, uefi}     False x86_64
c5.18xlarge     {legacy-bios, uefi}     False x86_64
c5.24xlarge     {legacy-bios, uefi}     False x86_64
c5.2xlarge      {legacy-bios, uefi}     False x86_64
c5.4xlarge      {legacy-bios, uefi}     False x86_64
c5.9xlarge      {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.

The following Windows AMIs support UEFI:

- TPM-Windows_Server-2022-English-Full-Base
- TPM-Windows_Server-2022-English-Core-Base
- TPM-Windows_Server-2019-English-Full-Base
- TPM-Windows_Server-2019-English-Core-Base
- TPM-Windows_Server-2016-English-Full-Base
- TPM-Windows_Server-2016-English-Core-Base

For information about Linux AMIs, see Requirements for launching an instance on UEFI in the *Amazon EC2 User Guide for Linux 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.

```
C:\> 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 the create the instance.

```
C:\> 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 the 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.

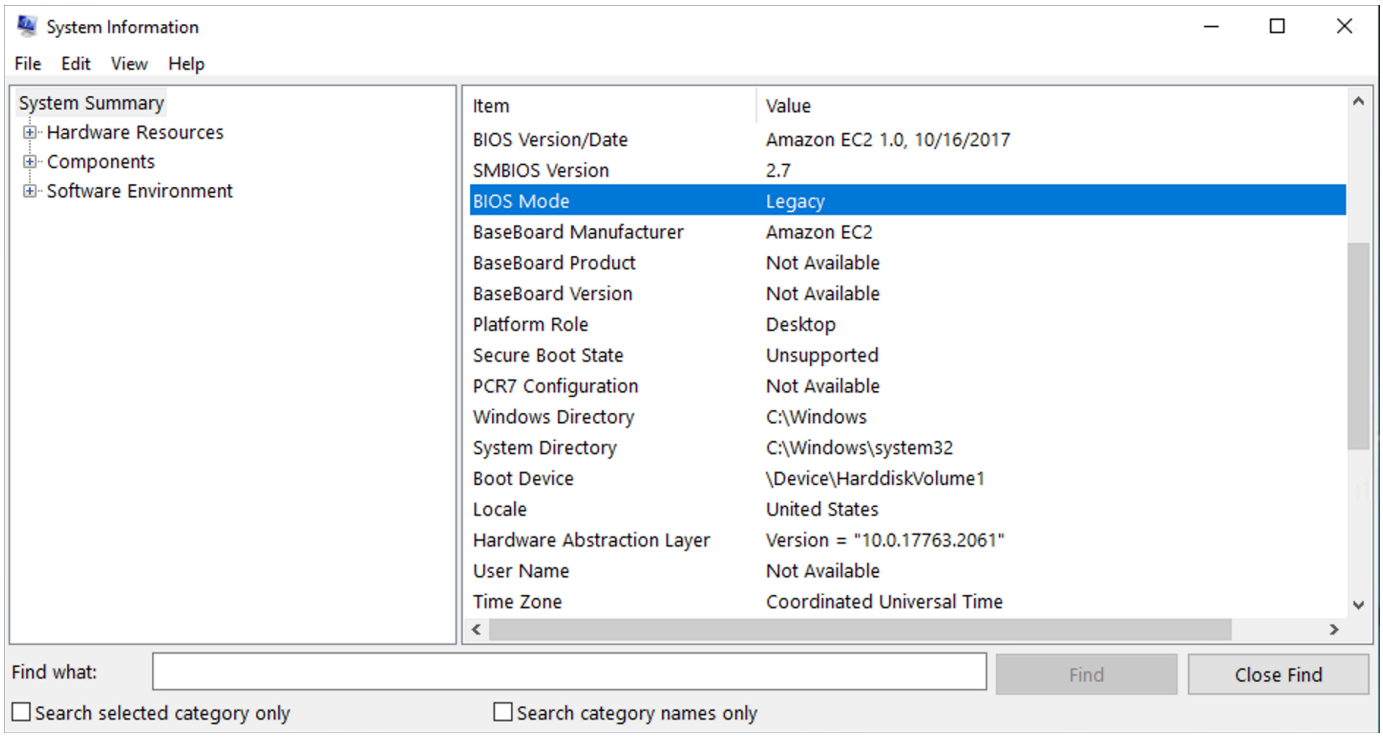
- `CurrentInstanceBootMode` – The boot mode that is used to boot the instance at launch or start.

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 RDP.

To determine the boot mode of the instance's operating system

1. Connect to your Windows instance using RDP.
2. Go to **System Information** and check the **BIOS Mode** row.



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 a Windows instance with UEFI Secure Boot support
- Verify whether a Windows instance is enabled for UEFI Secure Boot

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 a Windows 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 for Windows instances

AMI

Requires an AMI with 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

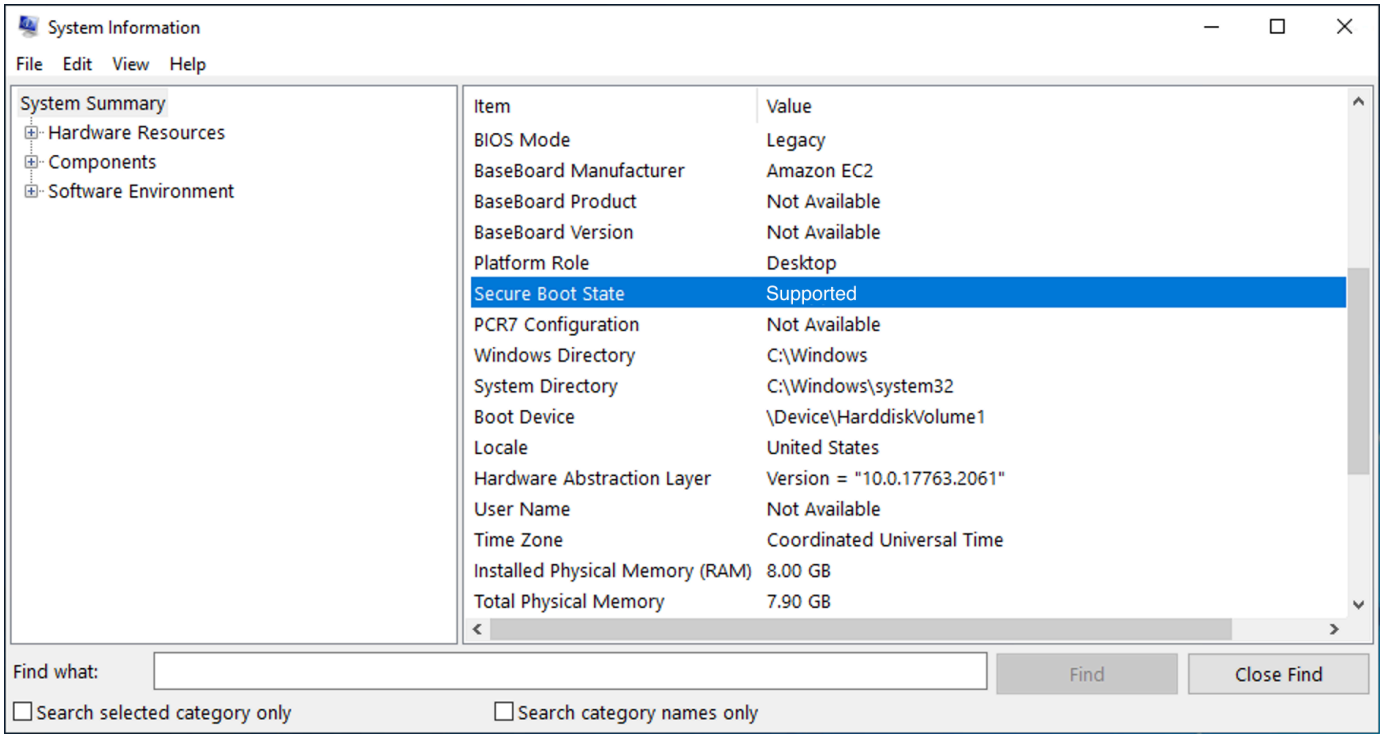
- Supported: All virtualized instance types that support UEFI also support UEFI Secure Boot. For the instance types that support UEFI Secure Boot, see Considerations.
- Not supported: Bare metal instance types do not support UEFI Secure Boot.

For the prerequisites for Linux instances, see Launch an instance with UEFI Secure Boot support in the *Amazon EC2 User Guide for Linux Instances*.

Verify whether a Windows instance is enabled for UEFI Secure Boot

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*.

To verify whether a Linux instance is enabled, see Verify whether a Linux instance is supported for UEFI Secure Boot in the *Amazon EC2 User Guide for Linux Instances*.

AWS Windows AMIs

AWS provides a set of publicly available AMIs that contain software configurations specific to the Windows platform. You can quickly start building and deploying your applications with Amazon EC2 by using these AMIs. First choose the AMI that meets your specific requirements, and then launch an instance using that AMI. You retrieve the password for the administrator account and then log in to the instance using Remote Desktop Connection, just as you would with any other Windows server.

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

Windows AMIs that have been configured for faster launching are pre-provisioned, using snapshots to launch instances up to 65% faster. To learn more about faster launching for Windows AMIs, including how you can configure faster launching for your Windows AMI, see Configure Windows fast launch for your Windows Server AMI.

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

> **Note**
>
> Microsoft no longer supports Windows Server 2003, 2008, and 2008 R2. We recommend that you launch new EC2 instances using a supported version of Windows Server. If you have existing EC2 instances that are running an unsupported version of Windows Server, we recommend that you upgrade those instances to a supported version of Windows Server. For more information, see Upgrade an Amazon EC2 Windows instance to a newer version of Windows Server.

Windows AMI topics

- Select an initial Windows AMI
- Keep your AMIs up to date
- Virtualization types
- Configure Windows fast launch for your Windows Server AMI
- Managed AWS Windows AMIs
- Specialized Windows AMIs
- AWS Windows AMI version history

Select an initial Windows AMI

To view the Windows AMIs provided by AWS, you can use the Amazon EC2 console or AWS Marketplace. For more information, see Find a Windows AMI.

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

- AWS Application Migration Service
- VM Import/Export

Keep your AMIs up to date

AWS provides updated and fully-patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). The AWS Windows AMIs contain the latest security updates available at the time they were created. For more information, see Details about AWS Windows AMI versions and Patches, security updates, and AMI IDs.

Use the AWS Systems Manager Automation runbook AWS-UpdateWindowsAmi to update an AMI by installing Windows updates, Amazon software, and Amazon drivers. You can also use EC2 Image Builder, a fully managed AWS service, to help automate creating up-to-date AMIs. For more information, see the *EC2 Image Builder User Guide*.

For EC2 instances in an Auto Scaling group, you can create and use the `PatchAMIAndUpdateASG` runbook to update an Auto Scaling group with a newly patched AMI. For more information, see Updating AMIs for Auto Scaling groups in the *AWS Systems Manager User Guide*.

For existing EC2 instances, we recommend that you regularly patch, update, and secure the operating system and applications. For more information, see Update your Windows instance.

Virtualization types

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

HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.

HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. HVM AMIs are required to take advantage of enhanced networking and GPU processing. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform; HVM virtualization provides this access.

Paravirtual guests traditionally performed better with storage and network operations than HVM guests because they could leverage special drivers for I/O that avoided the overhead of emulating network and disk hardware, whereas HVM guests had to translate these instructions to emulated hardware. Now PV drivers are available for HVM guests, so Windows instances can get performance advantages in storage and network I/O by using them. With these PV on HVM drivers, HVM guests can get the same performance as paravirtual guests, or better.

Configure Windows fast launch for your Windows Server AMI

Every Amazon EC2 Windows instance must go through the standard Windows operating system (OS) launch steps, which include several reboots, and often take 15 minutes or longer to complete. Amazon EC2 Windows Server AMIs that have the Windows fast launch feature enabled complete some of those steps and reboots in advance to reduce the time it takes to launch an instance.

When you configure a Windows Server AMI for Windows fast launch, Amazon EC2 creates a set of pre-provisioned snapshots to use for faster launching, as follows.

1. Amazon EC2 launches a set of temporary t3 instances, based on your settings.
2. As each temporary instance completes the standard launch steps, Amazon EC2 creates a pre-provisioned snapshot of the instance. It stores the snapshot in your Amazon S3 bucket.
3. When the snapshot is ready, Amazon EC2 terminates the associated t3 instance to keep resource costs as low as possible.
4. The next time Amazon EC2 launches an instance from the Windows fast launch enabled AMI, it uses one of the snapshots to significantly reduce the time it takes to launch.

Amazon EC2 automatically replenishes the snapshots you have on hand as it uses them to launch instances from the Windows fast launch enabled AMI.

Any account that has access to an AMI with Windows fast launch enabled can benefit from reduced launch times. When the AMI owner grants access for you to launch instances, the pre-provisioned snapshots come from the AMI owner’s account.

If an AMI that supports Windows fast launch is shared with you, you can enable or disable faster launching on the shared AMI yourself. If you enable a shared AMI for Windows fast launch, Amazon EC2 creates the pre-provisioned snapshots directly in your account. If you deplete the snapshots in your account, you can still use snapshots from the AMI owner's account.

> **Note**
>
> Windows fast launch deletes pre-provisioned snapshots as soon as they're consumed by a launch to minimize storage costs and prevent reuse. However, if the deleted snapshots match a retention rule, Recycle Bin automatically retains them. We recommend that you review the scope of your Recycle Bin retention rules so that this doesn't happen. For more information, see Considerations.
>
> This feature is not the same as EBS fast snapshot restore. You must explicitly enable EBS fast snapshot restore on a per-snapshot basis, and it has its own associated costs.

The following video demonstrates how to configure your Windows AMI for faster launching with a quick overview of the related key terms and their definitions: Launching EC2 Windows instances up to 65% faster on AWS.

Resource costs

There is no service charge to configure Windows AMIs for Windows fast launch. However, standard pricing applies for any underlying AWS resources that Amazon EC2 uses. To learn more about associated resource costs and how to manage them, see Manage resource costs.

Contents

- Key terms
- Prerequisites
- Configure Windows fast launch settings for your Amazon EC2 Windows Server AMI
- View AMIs with Windows fast launch enabled (AWS CLI)
- Manage resource costs
- Monitor Windows fast launch
- Service-linked role for Windows fast launch

Key terms

The Windows fast launch feature uses the following key terms:

Pre-provisioned snapshot

A snapshot of an instance that was launched from a Windows AMI with Windows fast launch enabled, and that has completed the following Windows launch steps, rebooting as required.

- Sysprep specialize
- Windows Out of Box Experience (OOBE)

When these steps are complete, Windows fast launch stops the instance, and creates a snapshot that is later used for faster launching from the AMI, based on your configuration.

Launch frequency

Controls the number of pre-provisioned snapshots that Amazon EC2 can launch within the specified timeframe. When you enable Windows fast launch for your AMI, Amazon EC2 creates the initial set of pre-provisioned snapshots in the background. For example, if the launch frequency is set to five launches per hour, which is the default, then Windows fast launch creates an initial set of five pre-provisioned snapshots.

When Amazon EC2 launches an instance from an AMI with Windows fast launch enabled, it uses one of the pre-provisioned snapshots to reduce the launch time. As snapshots are used, they are automatically replenished, up to the number specified by the launch frequency.

If you expect a spike in the number of instances that are launched from your AMI – during a special event, for example – you can increase the launch frequency in advance to cover the additional instances that you'll need. When your launch rate returns to normal, you can adjust the frequency back down.

When you experience a higher number of launches than anticipated, you might use up all the pre-provisioned snapshots that you have available. This doesn't cause any launches to fail. However, it can result in some instances going through the standard launch process, until snapshots can be replenished.

Target resource count

The number of pre-provisioned snapshots to keep on hand for an Amazon EC2 Windows Server AMI with Windows fast launch enabled.

Max parallel launches

Controls how many instances Amazon EC2 can launch at the same time to create the pre-provisioned snapshots for Windows fast launch. If your target resource count is higher than the maximum parallel launches that you've configured, then Amazon EC2 launches the number of instances specified by **Max parallel launches** to start creating the snapshots. As those instances complete the process, Amazon EC2 takes the snapshot and stops the instance. Then it continues to launch more instances until the total number of snapshots available has reached the target resource count. The value for **Max parallel launches** must be 6 or greater.

Prerequisites

Before you set up Windows fast launch, verify that you've met the following prerequisites that are required to create snapshots for the AMIs in your AWS account:

- If you don't use a launch template to configure your settings, ensure that a default VPC is configured for the Region in which you use Windows fast launch.

  > **Note**
  >
  > If you accidentally delete your default VPC in the Region where you plan to configure Windows fast launch, you can create a new default VPC in that Region. To learn more, see Create a default VPC in the *Amazon VPC User Guide*.

- To specify a non-default VPC, you must use a launch template when you configure Windows fast launch. For more information, see Use a launch template when you set up Windows fast launch.
- If your account includes a policy that enforces IMDSv2 for Amazon EC2 instances, you must create a launch template that specifies the metadata configuration to enforce IMDSv2.
- Private Windows fast launch AMIs must support user data script execution.
- To configure Windows fast launch for an AMI, you must create the AMI using **Sysprep** with the shutdown option. The Windows fast launch feature doesn't currently support AMIs that were created from a running instance.

  To create an AMI using **Sysprep**, see Create a custom Windows AMI.
- The default quota for **Max parallel launches** across all AMIs in an AWS account is 40 per Region. You can request a Service Quotas increase for your account, as follows.

1. Sign in to the AWS Management Console and open the Service Quotas console at https://console.aws.amazon.com/servicequotas/.
2. In the navigation pane, choose AWS services.
3. In the search bar, enter `EC2 Fast Launch`, and select the result.
4. Select the link for `Parallel instance launches`. This takes you to the **Parallel instance launches** service quota detail page.
5. Choose **Request quota increase**.

For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Configure Windows fast launch settings for your Amazon EC2 Windows Server AMI

You can configure Windows fast launch for Windows AMIs that you own, or AMIs that are shared with you from the AWS Management Console, API, SDKs, CloudFormation, or AWS Command Line Interface (AWS CLI). Before you configure Windows fast launch, verify that your AMI meets all of the prerequisites that are required to create the pre-provisioned snapshots. For more information, see Prerequisites.

The following sections cover configuration steps for the Amazon EC2 console and AWS CLI.

Enable Windows fast launch

To enable Windows fast launch, choose the tab that matches your environment, and follow the steps.

> **Note**
>
> Before changing these settings, make sure that your AMI, and the Region that you run in meet all Prerequisites.

Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.

3. Choose the AMI to update by selecting the check box next to the **Name**.
4. From the **Actions** menu above the list of AMIs, choose **Configure fast launch**. This opens the **Configure fast launch** page, where you configure the settings for Windows fast launch.
5. To start using pre-provisioned snapshots to launch instances from your Windows AMI faster, select the **Enable fast launch for Windows** check box.
6. From the **Set anticipated launch frequency** drop-down list, choose a value to specify the number of snapshots that are created and maintained to cover your expected instance launch volume.
7. When you're done making changes, choose **Save changes**.

> **Note**
>
> If you need to use a launch template to specify a non-default VPC, or to configure metadata settings for IMDSv2, see Use a launch template when you set up Windows fast launch.

AWS CLI

The **enable-fast-launch** command calls the Amazon EC2 EnableFastLaunch API operation.

Syntax:

```
aws ec2 enable-fast-launch \
  --image-id <value> \
  --resource-type <value> \ (optional)
  --snapshot-configuration <value> \ (optional)
  --launch-template <value> \ (optional)
  --max-parallel-launches <value> \ (optional)
  --dry-run | --no-dry-run \  (optional)
  --cli-input-json <value> \ (optional)
  --generate-cli-skeleton <value> \ (optional)
```

Example:

The following enable-fast-launch example enables Windows fast launch for the specified AMI, launching six parallel instances for pre-provisioning. The `ResourceType` is set to `snapshot`, which is the default value.

```
aws ec2 enable-fast-launch \
         --image-id ami-01234567890abcedf  \
         --max-parallel-launches 6 \
         --resource-type snapshot
```

Output:

```
{
     "ImageId": "ami-01234567890abcedf",
     "ResourceType": "snapshot",
     "SnapshotConfiguration": {
         "TargetResourceCount": 10
     },
     "LaunchTemplate": {},
     "MaxParallelLaunches": 6,
     "OwnerId": "0123456789123",
     "State": "enabling",
     "StateTransitionReason": "Client.UserInitiated",
     "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
 }
```

Tools for PowerShell

The **Enable-EC2FastLaunch** cmdlet calls the Amazon EC2 EnableFastLaunch API operation to enable Windows fast launch on your Windows AMI.

Syntax:

```
Enable-EC2FastLaunch
      -ImageId <String>
      -LaunchTemplate_LaunchTemplateId <String>
      -LaunchTemplate_LaunchTemplateName <String>
      -MaxParallelLaunch <Int32>
      -ResourceType <String>
      -SnapshotConfiguration_TargetResourceCount <Int32>
      -LaunchTemplate_Version <String>
      -Select <String>
      -PassThru <SwitchParameter>
      -Force <SwitchParameter>
```

Example:

The following Enable-EC2FastLaunch example enables Windows fast launch for the specified AMI, launching six parallel instances for pre-provisioning. The `ResourceType` is set to `snapshot`, which is the default value.

```
Enable-EC2FastLaunch `
 -ImageId ami-01234567890abcedf `
 -MaxParallelLaunch 6 `
 -Region us-west-2 `
 -ResourceType snapshot
```

Output:

```
ImageId               : ami-01234567890abcedf
LaunchTemplate        :
MaxParallelLaunches   : 6
OwnerId               : 0123456789123
ResourceType          : snapshot
SnapshotConfiguration : Amazon.EC2.Model.FastLaunchSnapshotConfigurationResponse
State                 : enabling
StateTransitionReason : Client.UserInitiated
StateTransitionTime   : 2/25/2022 12:24:11 PM
```

Disable Windows fast launch

To disable Windows fast launch, choose the tab that matches your environment, and follow the steps.

> **Note**
>
> Before changing these settings, make sure that your AMI, and the Region that you run in meet all Prerequisites.

Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Choose the AMI to update by selecting the check box next to the **Name**.

4. From the **Actions** menu above the list of AMIs, choose **Configure fast launch**. This opens the **Configure fast launch** page, where you configure the settings for Windows fast launch.
5. Clear the **Enable fast launch for Windows** check box to disable Windows fast launch and to remove pre-provisioned snapshots. This results in the AMI using the standard launch process for each instance, going forward.

> **Note**
>
> When you disable Windows image optimization, any existing pre-provisioned snapshots are automatically deleted. This step must be completed before you can start using the feature again.

6. When you're done making changes, choose **Save changes**.

AWS CLI

The **disable-fast-launch** command calls the Amazon EC2 DisableFastLaunch API operation.

Syntax:

```
aws ec2 disable-fast-launch \
  --image-id <value> \
  --force | --no-force \ (optional)
  --dry-run | --no-dry-run \ (optional)
  --cli-input-json <value> \ (optional)
  --generate-cli-skeleton <value> \ (optional)
```

Example:

The following disable-fast-launch example disables Windows fast launch on the specified AMI, and cleans up existing pre-provisioned snapshots.

```
aws ec2 disable-fast-launch \
         --image-id ami-01234567890abcedf
```

Output:

```
{
     "ImageId": "ami-01234567890abcedf",
     "ResourceType": "snapshot",
```

```
    "SnapshotConfiguration": {},
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
 }
```

Tools for PowerShell

The **Disable-EC2FastLaunch** cmdlet calls the Amazon EC2 DisableFastLaunch API operation.

Syntax:

```
Disable-EC2FastLaunch
      -ImageId <String>
      -ForceStop <Boolean>
      -Select <String>
      -PassThru <SwitchParameter>
      -Force <SwitchParameter>
```

Example:

The following Disable-EC2FastLaunch example disables Windows fast launch on the specified AMI, and cleans up existing pre-provisioned snapshots.

```
Disable-EC2FastLaunch -ImageId ami-01234567890abcedf
```

Output:

```
ImageId               : ami-01234567890abcedf
LaunchTemplate        :
Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
MaxParallelLaunches   : 6
OwnerId               : 0123456789123
ResourceType          : snapshot
```

```
SnapshotConfiguration :
State                 : disabling
StateTransitionReason : Client.UserInitiated
StateTransitionTime   : 2/25/2022 1:10:08 PM
```

Use a launch template when you set up Windows fast launch

With a launch template, you can configure a set of launch parameters that Amazon EC2 uses each time it launches an instance from that template. You can specify such things as an AMI to use for your base image, instance types, storage, network settings, and more.

Launch templates are optional, except for the following specific cases, where you must use a launch template for your Windows AMI when you configure faster launching:

- You must use a launch template to specify a non-default VPC for your Windows AMI.
- If your account includes a policy that enforces IMDSv2 for Amazon EC2 instances, you must create a launch template that specifies the metadata configuration to enforce IMDSv2.

  Use the launch template that includes your metadata configuration from the EC2 console, or when you run the enable-fast-launch command in the AWS CLI, or call the EnableFastLaunch API action.

> **Note**
>
> Amazon EC2 Windows fast launch doesn't support user data scripts in the launch template. If you use a launch template for Windows fast launch, you must not specify user data.

Specify a non-default VPC

Step 1: Create a launch template

Create a launch template that specifies the following details for your Windows instances:

- The VPC subnet.
- An instance type of `t3.xlarge`.

For more information, see Create a launch template.

Step 2: Specify the launch template for your Windows fast launch AMI

Choose the tab that matches your process:

Console

To specify the launch template for Windows fast launch from the AWS Management Console, follow these steps:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Choose the AMI to update by selecting the check box next to the **Name**.
4. From the **Actions** menu above the list of AMIs, choose **Configure fast launch**. This opens the **Configure fast launch** page, where you configure the settings for Windows fast launch.
5. The **Launch template** box performs a filtered search that finds launch templates in your account in the current Region that match the text you've entered. Specify all or part of the launch template name or ID in the box to show a list of matching launch templates. For example, if you enter `fast` in the box, Amazon EC2 finds all of the launch templates in your account in the current Region that have "fast" in the name.

   To create a new launch template, you can choose **Create launch template**.
6. When you select a launch template, Amazon EC2 shows the default version for that template in the **Source template version** box. To specify a different version, highlight the default version to replace it, and enter the version number you want in the box.
7. When you're done making changes, choose **Save changes**.

AWS CLI, API

To specify the launch template for Windows fast launch from the AWS CLI, specify the launch template name or ID in the `--launch-template` parameter when you run the enable-fast-launch command in the AWS CLI.

To specify the launch template for Windows fast launch in an API request, specify the launch template name or ID in the `LaunchTemplate` parameter when you call the EnableFastLaunch API action.

For more information about EC2 launch templates, see Launch an instance from a launch template.

Create a custom image with Windows fast launch enabled

Amazon EC2 Windows fast launch integrates with EC2 Image Builder to help you create custom images with Windows fast launch enabled. For more information, see Create distribution settings for a Windows AMI with EC2 Fast Launch enabled (AWS CLI) in the *EC2 Image Builder User Guide*.

View AMIs with Windows fast launch enabled (AWS CLI)

You can use the describe-fast-launch-images command in the AWS CLI, or the Get-EC2FastLaunchImage Tools for PowerShell Cmdlet to get details for AMIs that have Windows fast launch enabled.

Amazon EC2 provides the following details for each Windows AMI that is returned in the results:

- The image ID for an AMI with Windows fast launch enabled.
- The resource type that is used for pre-provisioning the associated Windows AMI. Supported value: `snapshot`.
- The snapshot configuration, which is a group of parameters that configure pre-provisioning for the associated Windows AMI using snapshots.
- Launch template information, including the ID, name, and version of the launch template that the associated AMI uses when it launches Window instances from pre-provisioned snapshots.
- The maximum number of instances that can be launched at the same time for creating resources.
- The owner ID for the associated AMI. This is not populated for AMIs that are shared with you.
- The current state of Windows fast launch for the associated AMI. Supported values include: `enabling | enabling-failed | enabled | enabled-failed | disabling | disabling-failed`.

> **Note**
>
> You can also see the current state displayed in the **Manage image optimization** page in the EC2 console, as **Image optimization state**.

- The reason that Windows fast launch for the associated AMI changed to the current state.
- The time that Windows fast launch for the associated AMI changed to the current state.

Choose the tab that matches your command line environment:

AWS CLI

The **describe-fast-launch-images** command calls the Amazon EC2 DescribeFastLaunchImages API operation.

Syntax:

```
aws ec2 describe-fast-launch-images \
 --image-ids <value> \ (optional)
 --filters <value> \ (optional)
 --dry-run | --no-dry-run \ (optional)
 --cli-input-json <value> \ (optional)
 --starting-token <value> \ (optional)
 --page-size <value> \ (optional)
 --max-items <value> \ (optional)
 --generate-cli-skeleton <value> \ (optional)
```

Example:

The following describe-fast-launch-images example describes the details for each of the AMIs in the account that are configured for Windows fast launch. In this example, only one AMI in the account is configured for Windows fast launch.

```
aws ec2 describe-fast-launch-images
```

Output:

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
```

```
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
    ]
}
```

Tools for PowerShell

The **Get-EC2FastLaunchImage** cmdlet calls the Amazon EC2 DescribeFastLaunchImages API operation.

Syntax:

```
Get-EC2FastLaunchImage
-Filter <Filter[]>
-ImageId <String[]>
-MaxResult <Int32>
-NextToken <String>
-Select <String>
-NoAutoIteration <SwitchParameter>
```

Example:

The following Get-EC2FastLaunchImage example describes the details for each of the AMIs in the account that are configured for Windows fast launch. In this example, only one AMI in the account is configured for Windows fast launch.

```
Get-EC2FastLaunchImage -ImageId ami-01234567890abcedf
```

Output:

```
ImageId               : ami-01234567890abcedf
LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
MaxParallelLaunches   : 6
OwnerId               : 0123456789123
ResourceType          : snapshot
SnapshotConfiguration :
State                 : enabled
```

```
StateTransitionReason : Client.UserInitiated
StateTransitionTime   : 2/25/2022 12:54:43 PM
```

Manage resource costs

There is no service charge to configure Windows AMIs for Windows fast launch. However, when you enable Windows fast launch for an Amazon EC2 Windows AMI, standard pricing applies for underlying AWS resources that Amazon EC2 uses to prepare and store the pre-provisioned snapshots. You can configure cost allocation tags to help you track and manage the costs that are associated with Windows fast launch resources. For more information about how to configure cost allocation tags, see Track Windows fast launch costs on your bill.

The following example demonstrates how the costs associated with Windows fast launch snapshots costs might be allocated.

Example scenario: The AtoZ Example company has a Windows AMI with a 50 GiB EBS root volume. They enable Windows fast launch for their AMI, and set the target resource count to five. Over the course of a month, using Windows fast launch for their AMI costs them around $5.00, and the cost breakdown is as follows:

1. When AtoZ Example enables Windows fast launch, Amazon EC2 launches five small instances. Each instance runs through the Sysprep and OOBE Windows launch steps, rebooting as required. This takes several minutes for each instance (time can vary, based on how busy that Region or Availability Zone (AZ) is, and on the size of the AMI).

   Costs

   - Instance runtime costs (or minimum runtime, if applicable): five instances
   - Volume costs: five EBS root volumes

2. When the pre-provisioning process completes, Amazon EC2 takes a snapshot of the instance, which it stores in Amazon S3. Snapshots are typically stored for 4–8 hours before they are consumed by a launch. In this case, the cost is roughly $0.02 to $0.05 per snapshot.

   Costs

   - Snapshot storage (Amazon S3): five snapshots

3. After Amazon EC2 takes the snapshot, it stops the instance. At that point, the instance is no longer accruing costs. However EBS volume costs continue to accrue.

Costs

- EBS volumes: costs continue for the associated EBS root volumes.

> **Note**
>
> The costs shown here are for demonstration purposes only. Your costs will vary, depending on your AMI configuration and pricing plan.

Track Windows fast launch costs on your bill

Cost allocation tags can help you organize your AWS bill to reflect the costs associated with Windows fast launch. You can use the following tag that Amazon EC2 adds to the resources it creates when it prepares and stores pre-provisioned snapshots for Windows fast launch:

Tag key: `CreatedBy`, **Value:** `EC2 Fast Launch`

After you activate the tag in the Billing and Cost Management console, and set up your detailed billing report, the `user:CreatedBy` column appears on the report. The column includes values from all services. However, if you download the CSV file, you can import the data into a spreadsheet, and filter for `EC2 Fast Launch` in the value. This information also appears in the AWS Cost and Usage Report when the tag is activated.

Step 1: Activate user-defined cost allocation tags

To include resource tags in your cost reports, you must first activate the tag in the Billing and Cost Management console. For more information, see Activating User-Defined Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*.

> **Note**
>
> Activation can take up to 24 hours.

Step 2: Set up a cost report

If you already have a cost report set up, a column for your tag appears the next time the report runs after activation is complete. To set up cost reports for the first time, choose one of the following.

- See Setting up a monthly cost allocation report in the *AWS Billing and Cost Management User Guide*.
- See Creating Cost and Usage Reports in the *AWS Cost and Usage Report User Guide*.

> **Note**
>
> It can take up to 24 hours for AWS to start delivering reports to your S3 bucket.

You can configure Windows fast launch for Windows AMIs that you own, or AMIs that are shared with you from the Amazon EC2 console, API, SDKs, CloudFormation, or **ec2** commands in the AWS CLI. The following sections cover configuration steps for the Amazon EC2 console and AWS CLI.

You can also create custom Windows AMIs that are configured for Windows fast launch with EC2 Image Builder. For more information, see Create distribution settings for a Windows AMI with Windows fast launch enabled (AWS CLI).

Monitor Windows fast launch

This section covers how to monitor the Amazon EC2 Windows Server AMIs in your account that have Windows fast launch enabled.

Monitor Windows fast launch state changes with EventBridge

When the state changes for a Windows AMI with Windows fast launch enabled, Amazon EC2 generates an `EC2 Fast Launch State-change Notification` event. Then Amazon EC2 sends the state change event to Amazon EventBridge (formerly known as Amazon CloudWatch Events).

You can create EventBridge rules that trigger one or more actions in response to the state change event. For example, you can create an EventBridge rule that detects when Windows fast launch is enabled and performs the following actions:

- Sends a message to an Amazon SNS topic that notifies its subscribers.
- Invokes a Lambda function that performs some action.
- Sends the state change data to Amazon Data Firehose for analytics.

For more information, see Creating Amazon EventBridge rules that react to events in the *Amazon EventBridge User Guide*.

State change events

The Windows fast launch feature emits JSON formatted state change events on a best-effort basis. Amazon EC2 sends the events to EventBridge in near real time. This section describes the event fields and shows an example of the event format.

`EC2 Fast Launch State-change Notification`

imageId

Identifies the AMI with the Windows fast launch state change.

resourceType

The type of resource to use for pre-provisioning. Supported value: `snapshot`. The default value is `snapshot`.

state

The current state of the Windows fast launch feature for the specified AMI. Valid values include the following:

- **enabling** – You've enabled the Windows fast launch feature for the AMI, and Amazon EC2 has started creating snapshots for the pre-provisioning process.
- **enabling-failed** – Something went wrong that caused the pre-provisioning process to fail the first time that you enabled the Windows fast launch for an AMI. This can happen anytime during the pre-provisioning process.
- **enabled** – The Windows fast launch feature is enabled. The state changes to `enabled` as soon as Amazon EC2 creates the first pre-provisioned snapshot for a newly enabled Windows fast launch AMI. If the AMI was already enabled and goes through pre-provisioning again, the state change happens right away.
- **enabled-failed** – This state applies only if this is not the first time your Windows fast launch AMI goes through the pre-provisioning process. This can happen if the Windows fast launch feature is disabled and then later enabled again, or if there is a configuration change or other error after pre-provisioning is completed for the first time.
- **disabling** – The AMI owner has turned off the Windows fast launch feature for the AMI, and Amazon EC2 has started the clean up process.

- **disabled** – The Windows fast launch feature is disabled. The state changes to `disabled` as soon as Amazon EC2 completes the clean up process.
- **disabling-failed** – Something went wrong that caused the clean up process to fail. This means that some pre-provisioned snapshots may still remain in the account.

stateTransitionReason

The reason that the state changed for the Windows fast launch AMI.

> **Note**
>
> All fields in this event message are required.

The following example shows a newly enabled Windows fast launch AMI that has launched the first instance to start the pre-provisioning process. At this point, the state is `enabling`. After Amazon EC2 creates the first pre-provisioned snapshot, the state changes to `enabled`.

```
{
 "version": "0",
 "id": "01234567-0123-0123-0123-012345678901",
 "detail-type": "EC2 Fast Launch State-change Notification",
 "source": "aws.ec2",
 "account": "123456789012",
 "time": "2022-08-31T20:30:12Z",
 "region": "us-east-1",
 "resources": [
  "arn:aws:ec2:us-east-1:123456789012:image/ami-123456789012"
 ],
 "detail": {
  "imageId": "ami-123456789012",
  "resourceType": "snapshot",
  "state": "enabling",
  "stateTransitionReason": "Client.UserInitiated"
 }
}
```

Monitor Windows fast launch metrics with CloudWatch

Amazon EC2 AMIs with Windows fast launch enabled send metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Windows fast

launch sends to CloudWatch. The `AWS/EC2` namespace includes the following Windows fast launch metrics:

Metric	Description
NumberOfAvailableFastLaunchSnapshots	The number of pre-provisioned snapshots available per Windows fast launch enabled AMI.
NumberOfInstancesFastLaunched	The number of instances per Windows fast launch enabled AMI that were launched from pre-provisioned snapshots.
NumberOfInstancesNotFastLaunched	The number of instances per Windows fast launch enabled AMI that resulted in a cold boot due to the lack of available pre-provi sioned snapshots at launch time.
FastLaunchSnapshotUsedToRefillStartTime	The timestamp when Amazon EC2 launched a new image from a Windows fast launch enabled AMI to create another snapshot after an existing snapshot was used.
FastLaunchSnapshotCreationTime	Measures the time it took for Amazon EC2 to launch an instance and create a snapshot for a Windows fast launch enabled AMI.

Service-linked role for Windows fast launch

Amazon EC2 uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service. Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information about how Amazon EC2 uses IAM roles, including service-linked roles, see IAM roles for Amazon EC2.

Amazon EC2 uses the service-linked role named `AWSServiceRoleForEC2FastLaunch` to create and manage a set of pre-provisioned snapshots that reduce the time it takes to launch instances from your Windows AMI.

You don't need to create this service-linked role manually. When you start using Windows fast launch for your AMI, Amazon EC2 creates the service-linked role for you, if it doesn't already exist.

> **Note**
>
> If the service-linked role is deleted from your account, you can enable Windows fast launch for another Windows AMI to re-create the role in your account. Alternatively, you can disable Windows fast launch for your current AMI, and then enable it again. However, disabling the feature results in your AMI using the standard launch process for all new instances while Amazon EC2 removes all of your pre-provisioned snapshots. After all of the pre-provisioned snapshots are gone, you can enable using Windows fast launch for your AMI again.

Amazon EC2 does not allow you to edit the `AWSServiceRoleForEC2FastLaunch` 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 by using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

You can delete a service-linked role only after first deleting all of the related resources. This protects the Amazon EC2 resources that are associated with your Amazon EC2 Windows Server AMI with Windows fast launch enabled, because you can't inadvertently remove permission to access the resources.

Amazon EC2 supports the Windows fast launch service-linked role in all of the Regions where the Amazon EC2 service is available. For more information, see Regions.

Permissions granted by `AWSServiceRoleForEC2FastLaunch`

Amazon EC2 uses the `EC2FastLaunchServiceRolePolicy` managed policy to complete the following actions:

- `cloudwatch:PutMetricData` – Post metric data associated with Windows fast launch to the Amazon EC2 namespace.
- `ec2:CreateLaunchTemplate` – Create a launch template for your Amazon EC2 Windows Server AMI with Windows fast launch enabled.
- `ec2:CreateSnapshot` – Create pre-provisioned snapshots for your Amazon EC2 Windows Server AMI with Windows fast launch enabled.

- `ec2:CreateTags` – Create tags for resources that are associated with launching and pre-provisioning Windows instances for your Amazon EC2 Windows Server AMI with Windows fast launch enabled.
- `ec2:DeleteSnapshots` – Delete all associated pre-provisioned snapshots if Windows fast launch is turned off for a previously enabled AMI.
- `ec2:DescribeImages` – Describe images for all resources.
- `ec2:DescribeInstanceAttribute` – Describe instance attributes for all resources.
- `ec2:DescribeInstanceStatus` – Describe instance status for all resources.
- `ec2:DescribeInstances` – Describe instances for all resources.
- `ec2:DescribeInstanceTypeOfferings` – Describe instance type offerings for all resources.
- `ec2:DescribeLaunchTemplates` – Describe launch templates for all resources.
- `ec2:DescribeLaunchTemplateVersions` – Describe launch template versions for all resources.
- `ec2:DescribeSnapshots` – Describe snapshot resources for all resources.
- `ec2:DescribeSubnets` – Describe subnets for all resources.
- `ec2:RunInstances` – Launch instances from an Amazon EC2 Windows Server AMI with Windows fast launch enabled, in order to perform provisioning steps.
- `ec2:StopInstances` – Stop instances that were launched from an Amazon EC2 Windows Server AMI with Windows fast launch enabled, in order to create pre-provisioned snapshots.
- `ec2:TerminateInstances` – Terminate an instance that was launched from an Amazon EC2 Windows Server AMI with Windows fast launch enabled, after creating the pre-provisioned snapshot from it.
- `iam:PassRole` – Allows the `AWSServiceRoleForEC2FastLaunch` service-linked role to launch instances on your behalf using the instance profile from your launch template.

For more information about using managed policies for Amazon EC2, see AWS managed policies for Amazon Elastic Compute Cloud.

Access to customer managed keys for use with encrypted AMIs and EBS snapshots

Prerequisite

- To enable Amazon EC2 to access an encrypted AMI on your behalf, you must have permission for the `createGrant` action in the customer managed key.

When you enable Windows fast launch for an encrypted AMI, Amazon EC2 ensures that permission is granted for the `AWSServiceRoleForEC2FastLaunch` role to use the customer managed key to access your AMI. This permission is needed to launch instances and create pre-provisioned snapshots on your behalf.

Managed AWS Windows AMIs

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

- EC2Launch v2 (Windows Server 2022)
- EC2Launch (Windows Server 2016 and 2019)
- EC2Config service (through Windows Server 2012 R2)
- AWS Systems Manager
- AWS CloudFormation
- AWS Tools for Windows PowerShell
- Network drivers (SRIOV, ENA, Citrix PV)
- Storage drivers (NVMe, AWS PV, Citrix PV)
- Graphics drivers (NVidia GPU, Elastic GPU)
- Spot Instance hibernation

For information about other customizations, see AWS Windows AMIs.

Managed Windows AMIs topics

- Details about AWS Windows AMI versions
  - Where AWS gets the Windows Server installation media
  - What to expect in an official AWS Windows AMI
  - How AWS validates security, integrity, and authenticity of software on AMIs
  - How AWS decides which Windows AMIs to offer
  - Patches, security updates, and AMI IDs
- Configuration changes for AWS Windows AMIs
- Update your Windows instance

- Upgrade or migrate to a newer version of Windows Server
- Subscribe to Windows AMI notifications
- Changes in Windows Server 2016 and later AMIs

Details about AWS Windows AMI versions

Where AWS gets the Windows Server installation media

When a new version of Windows Server is released, we download the Windows ISO from Microsoft and validate the hash Microsoft publishes. An initial AMI is then created from the Windows distribution ISO. The drivers needed to boot on EC2 are included in addition to our EC2 launch agent. To prepare this initial AMI for public release, we perform automated processes to convert the ISO to an AMI. This prepared AMI is used for the monthly automated update and release process.

What to expect in an official AWS Windows AMI

AWS provides AMIs with a variety of configurations for popular versions of Microsoft supported Windows Server Operating Systems. As outlined in the previous section, we start with the Windows Server ISO from Microsoft's Volume Licensing Service Center (VLSC) and validates the hash to ensure it matches Microsoft's documentation for new Windows Server operating systems.

We perform the following changes using automation on AWS to take the current Windows Server AMIs and update them:

- Install all Microsoft recommended Windows security patches. We release images shortly after the monthly Microsoft patches are made available.
- Install the latest drivers for AWS hardware, including network and disk drivers, EC2WinUtil for troubleshooting, as well as GPU drivers in selected AMIs.
- Include the following AWS launch agent software by default:
  - EC2Launch v2 for Windows Server 2022 and optionally for Windows Server 2019 and 2016 with specific AMIs. For more information, see Configure a Windows instance using EC2Launch v2.
  - EC2Launch for Windows Server 2016 and 2019.
  - EC2Config for Windows Server 2012 R2 and earlier.
- Configure Windows Time to use the Amazon Time Sync Service.

- Make changes in all power schemes to set the display to never turn off.
- Perform minor bug fixes – generally one-line registry changes to enable or disable features that we have found to improve performance on AWS.
- Tests and validates AMIs across new and existing EC2 platforms to ensure compatibility, stability, and consistency prior to release.
- Other than the previously mentioned changes, we keep the AMIs as close as possible to the Microsoft default installation of Windows Server. For example, we keep the PowerShell and .NET Framework installations as they are and don't install additional Windows roles, role services, or features.

How AWS validates security, integrity, and authenticity of software on AMIs

We take a number of steps during the image build process, to maintain the security, integrity, and authenticity of AWS provided Windows AMIs. A few examples include:

- AWS provided Windows AMIs are built using source media obtained directly from Microsoft.
- Windows Updates are downloaded directly from Microsoft's Windows Update Service by Windows, and installed on the instance used to create the AMI during the image build process.
- AWS Software is downloaded from secure S3 buckets and installed in the AMIs.
- Drivers—such as for the chipset and GPU—are obtained directly from the vendor, stored in secure S3 buckets, and installed on the AMIs during the image build process.

How AWS decides which Windows AMIs to offer

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

- New AMI offerings are created for new OS releases. You can count on AWS releasing "Base," "Core/Container," and "SQL Express/Standard/Web/Enterprise" offerings in English and other widely used languages. The primary difference between Base and Core offerings is that Base offerings have a desktop/GUI whereas Core offerings are PowerShell command line only. For more information about Windows Server Core, see https://docs.microsoft.com/en-us/windows-server/administration/server-core/what-is-server-core.
- New AMI offerings are created to support new platforms – for example, the Deep Learning and "NVidia" AMIs were created to support customers using our GPU-based instance types (P2 and P3, G2 and G3, and more).

- Less popular AMIs are sometimes removed. If we see a particular AMI is launched only a few times in its entire lifespan, we will remove it in favor of more widely used options.

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

Patches, security updates, and AMI IDs

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

> **Note**
>
> Instances launched from Windows Server 2019 and later AMIs may show a Windows Update dialog message stating "Some settings are managed by your organization." This message appears as a result of changes in Windows Server 2019 and does not impact the behavior of Windows Update or your ability to manage update settings.
> To remove this warning, see "Some settings are managed by your organization".

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

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

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

Configuration changes for AWS Windows AMIs

The following configuration changes are applied to each AWS Windows AMI.

Clean and prepare

Change	Applies to
Check for pending file renames or reboots, and reboot as needed	All AMIs
Delete `.dmp` files	All AMIs
Delete logs (event logs, Systems Manager, EC2Config)	All AMIs
Delete temporary folders and files for Sysprep	All AMIs
Clear recent history (Start menu, Windows Explorer, and more)	Windows Server 2012 R2 and earlier
Perform virus scan	All AMIs
Pre-compile queued .NET assemblies (before Sysprep)	All AMIs
Run Windows maintenance tools	Windows Server 2012 R2 and later
Restore default values for Internet Explorer	All AMIs
Restore default values for EC2Config	Windows Server 2012 R2 and earlier
Set EC2Launch to run at the next launch	Windows Server 2016 and 2019
Reset the Windows wallpaper	All AMIs
Run Sysprep	All AMIs

Install and configure

Change	Applies to
Add links to the Amazon EC2 Windows Guide	All AMIs
Attach instance storage volumes to extended mount points	All AMIs
Install the current AWS Tools for Windows PowerShell	All AMIs
Install the current AWS CloudFormation helper scripts	All AMIs
Install the current EC2Config and SSM Agent	Windows Server 2012 R2 and earlier
Install the current EC2Launch and SSM Agent	Windows Server 2016 and 2019
Install the current EC2Launch v2 and SSM Agent	Windows Server 2022 and later
Install the current AWS PV, ENA, and NVMe drivers	Windows Server 2008 R2 and later
Install the current SRIOV drivers	Windows Server 2012 R2 and later
Install the current Citrix PV driver	Windows Server 2008 SP2 and earlier
Install the current EC2WinUtil driver	Windows Server 2008 R2 and later
Install PowerShell 2.0 and 3.0	Windows Server 2008 SP2 and R2
If Microsoft SQL Server is installed: • Install service packs • Configure to start automatically • Add BUILTIN\Administrators to the SysAdmin role	All AMIs

Change	Applies to
• Open TCP port 1433 and UDP port 1434	
Apply the following hotfixes: • MS15-011 • KB2582281 • KB2634328 • KB2394911 • KB2780879	Windows Server 2008 SP2 and R2
Allow ICMP traffic through the firewall	Windows Server 2012 R2 and earlier
Enable file and printer sharing	Windows Server 2012 R2 and earlier
Disable RunOnce for Internet Explorer	All AMIs
Enable remote PowerShell	All AMIs
Configure a paging file on the system volume as follows: • Windows Server 2016 and later - Managed by the system • Windows Server 2012 R2 - Initial size and max size are 8 GB • Windows Server 2012 and earlier - Initial size is 512 MB, max size is 8 GB	All AMIs
Configure an additional system managed paging file on `Z:`, if available	Windows Server 2012 R2 and earlier
Disable hibernation and delete the hibernation file	All AMIs
Disable the Connected User Experiences and Telemetry service	All AMIs
Set the performance options for best performance	All AMIs
Set the power setting to high performance	All AMIs

Change	Applies to
Disable the screen saver password	All AMIs
Set the RealTimeIsUniversal registry key	All AMIs
Set the timezone to UTC	All AMIs
Disable Windows updates and notifications	All AMIs
Run Windows Update and reboot until there are no pending updates	All AMIs
Set the display in all power schemes to never turn off	All AMIs
Set the PowerShell execution policy to "Unrestricted"	All AMIs

Update your Windows instance

After you launch a Windows instance, you are responsible for installing updates on it. For more information, see Update management in Amazon EC2.

You can manually install only the updates that interest you, or you can start from a current AWS Windows AMI and build a new Windows instance. For information about finding the current AWS Windows AMIs, and keeping your AMIs up to date, see Find a Windows AMI and Keep your AMIs up to date.

> **Note**
>
> Instances should be stateless when updating. For more information, see Managing Your AWS Infrastructure at Scale.

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

- Windows Server
- Microsoft SQL Server
- Windows PowerShell
- Install the latest version of EC2Launch v2

- Install the latest version of EC2Launch
- Install the latest version of EC2Config
- AWS Systems Manager SSM Agent
- Enable enhanced networking on Windows
- Install or upgrade AWS NVMe drivers using PowerShell
- Upgrade PV drivers on Windows instances
- AWS Tools for Windows PowerShell
- AWS CloudFormation helper scripts

We recommend that you reboot your Windows instance after installing updates. For more information, see Reboot your instance.

Upgrade or migrate to a newer version of Windows Server

For information about how to upgrade or migrate a Windows instance to a newer version of Windows Server, see Upgrade an Amazon EC2 Windows instance to a newer version of Windows Server.

Subscribe to Windows AMI notifications

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

To subscribe to Windows 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-1:801119661308:ec2-windows-ami-update`
      - `arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-private`

For **AWS GovCloud (US)**:

`arn:aws-us-gov:sns:us-gov-west-1:077303321853:ec2-windows-ami-update`

b. For **Protocol**, choose **Email**.

c. For **Endpoint**, type an email address that you can use to receive the notifications.

d. Choose **Create subscription**.

6. You'll receive a confirmation email with the subject line `AWS Notification - Subscription Confirmation`. Open the email and choose **Confirm subscription** to complete your subscription.

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

To unsubscribe from Windows AMI notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/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 **Delete**. When prompted for confirmation, choose **Delete**.

Changes in Windows Server 2016 and later AMIs

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

- To accommodate the change from .NET Framework to .NET Core, the EC2Config service has been deprecated on Windows Server 2016 AMIs and replaced by EC2Launch. EC2Launch is a bundle of Windows PowerShell scripts that perform many of the tasks performed by the EC2Config service. For more information, see Configure a Windows instance using EC2Launch. EC2Launch v2 replaces EC2Launch in Windows Server 2022 and later. For more information, see Configure a Windows instance using EC2Launch v2.

- On earlier versions of Windows Server AMIs, you can use the EC2Config service to join an EC2 instance to a domain and configure integration with Amazon CloudWatch. On Windows Server 2016 and later AMIs, you can use the CloudWatch agent to configure integration with Amazon CloudWatch. For more information about configuring instances to send log data to CloudWatch, see Collect Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent. For information about joining an EC2 instance to a domain, see Join an Instance to a Domain Using the `AWS-JoinDirectoryServiceDomain` JSON Document in the *AWS Systems Manager User Guide*.

Other differences

Note the following additional important differences for instances created from Windows Server 2016 and later AMIs.

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

For more information, see Windows Server on Microsoft.com.

Specialized Windows AMIs

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

Topics

- SQL Server AMIs provided by AWS
- STIG Hardened Amazon EC2 Windows Server AMIs

SQL Server AMIs provided by AWS

To discover available SQL Server license-included AMIs, see Find a SQL Server license-included AMI in the *Microsoft SQL Server on Amazon EC2 User Guide*.

To view changes to each release of the AWS Windows AMIs, including SQL Server updates, see the AWS Windows AMI version history in the *Amazon EC2 User Guide*.

STIG Hardened Amazon EC2 Windows Server AMIs

Security Technical Implementation Guides (STIGs) are the configuration standards created by the Defense Information Systems Agency (DISA) to secure information systems and software. DISA documents three levels of compliance risk, known as categories:

- **Category I** — The highest level of risk. It covers the most severe risks, and includes any vulnerability that can result in a loss of confidentiality, availability, or integrity.
- **Category II** — Medium risk.
- **Category III** — Low risk.

Each compliance level includes all STIG settings from lower levels. This means that the highest level includes all applicable settings from all levels.

To ensure that your systems are compliant with STIG standards, you must install, configure, and test a variety of security settings. STIG Hardened EC2 Windows Server AMIs are pre-configured with over 160 required security settings. Amazon EC2 supports the following operating systems for STIG Hardened AMIs:

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

The STIG Hardened AMIs include updated Department of Defense (DoD) certificates to help you get started and achieve STIG compliance. STIG Hardened AMIs are available in all public AWS and GovCloud Regions. You can launch instances from these AMIs directly from the Amazon EC2 console. They are billed using standard Windows pricing. There are no additional charges for using STIG Hardened AMIs.

You can find the STIG Hardened EC2 Windows Server AMIs in the Community AMIs when you launch an instance, as follows.

Launch an EC2 instance with a STIG Hardened Windows Server AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Instances** from the navigation pane. This opens a list of your EC2 instances in the current AWS Region.
3. Choose **Launch instances** from the upper right corner above the list. This opens the **Launch an instance** page.
4. To find a STIG Hardened AMI, choose **Browse more AMIs** on the right side of the **Application and OS Images (Amazon Machine Image)** section. This displays an advanced AMI search.
5. Select the **Community AMIs** tab, and enter part or all of one of the following name patterns in the search bar. Our AMIs indicate that they are "provided by Amazon".

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

Name patterns for STIG Hardened AMI names

- Windows_Server-2022-English-STIG-Full-*YYYY.MM.DD*
- Windows_Server-2022-English-STIG-Core-*YYYY.MM.DD*
- Windows_Server-2019-English-STIG-Full-*YYYY.MM.DD*
- Windows_Server-2019-English-STIG-Core-*YYYY.MM.DD*
- Windows_Server-2016-English-STIG-Full-*YYYY.MM.DD*
- Windows_Server-2016-English-STIG-Core-*YYYY.MM.DD*
- Windows_Server-2012-R2-English-STIG-Full-*YYYY.MM.DD*
- Windows_Server-2012-R2-English-STIG-Core-*YYYY.MM.DD*

The following sections list the STIG settings that Amazon applies to Windows Operating Systems and components.

Topics

- Core and base operating systems
- Microsoft .NET Framework 4.0 STIG Version 2 Release 2

- Windows Firewall STIG Version 2 Release 1
- Internet Explorer (IE) 11 STIG Version 2 Release 3
- Microsoft Edge STIG Version 1 Release 6
- Microsoft Defender STIG Version 2 Release 4
- Version history

Core and base operating systems

STIG Hardened EC2 AMIs are designed for use as standalone servers, and have the highest level of STIG settings applied.

The following list contains STIG settings that apply for STIG Hardened Windows AMIs. Not all settings apply in all cases. For example, some STIG settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

Windows Server 2022 STIG Version 1 Release 1

This release includes the following STIG settings for Windows operating systems:

V-254247, V-254265, V-254269, V-254270, V-254271, V-254272, V-254273, V-254274, V-254276, V-254277, V-254278, V-254285, V-254286, V-254287, V-254288, V-254289, V-254290, V-254291, V-254292, V-254293, V-254300, V-254301, V-254302, V-254303, V-254304, V-254305, V-254306, V-254307, V-254308, V-254309, V-254310, V-254311, V-254312, V-254313, V-254314, V-254315, V-254316, V-254317, V-254318, V-254319, V-254320, V-254321, V-254322, V-254323, V-254324, V-254325, V-254326, V-254327, V-254328, V-254329, V-254330, V-254331, V-254332, V-254333, V-254334, V-254335, V-254336, V-254337, V-254338, V-254339, V-254341, V-254342, V-254344, V-254345, V-254346, V-254347, V-254348, V-254349, V-254350, V-254351, V-254352, V-254353, V-254354, V-254355, V-254356, V-254357, V-254358, V-254359, V-254360, V-254361, V-254362, V-254363, V-254364, V-254365, V-254366, V-254367, V-254368, V-254369, V-254370, V-254371, V-254372, V-254373, V-254374, V-254375, V-254376, V-254377, V-254378, V-254379, V-254380, V-254381, V-254382, V-254383, V-254431, V-254432, V-254433, V-254434, V-254435, V-254436, V-254438, V-254439, V-254442, V-254443,

V-254444, V-254445, V-254446, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254465, V-254466, V-254467, V-254468, V-254469, V-254470, V-254471, V-254472, V-254473, V-254474, V-254475, V-254476, V-254477, V-254478, V-254479, V-254480, V-254481, V-254482, V-254483, V-254484, V-254485, V-254486, V-254487, V-254488, V-254489, V-254490, V-254493, V-254494, V-254495, V-254497, V-254499, V-254500, V-254501, V-254502, V-254503, V-254504, V-254505, V-254507, V-254508, V-254509, V-254510, V-254511, and V-254512

Windows Server 2019 STIG Version 2 Release 5

This release includes the following STIG settings for Windows operating systems:

V-205625, V-205626, V-205627, V-205628, V-205629, V-205630, V-205631, V-205632, V-205633, V-205634, V-205635, V-205636, V-205637, V-205638, V-205639, V-205640, V-205641, V-205642, V-205643, V-205644, V-205645, V-205646, V-205647, V-205648, V-205649, V-205650, V-205651, V-205652, V-205653, V-205654, V-205655, V-205656, V-205657, V-205658, V-205659, V-205660, V-205661, V-205662, V-205663, V-205664, V-205665, V-205666, V-205667, V-205668, V-205669, V-205670, V-205671, V-205672, V-205673, V-205674, V-205675, V-205676, V-205677, V-205678, V-205679, V-205680, V-205681, V-205682, V-205683, V-205684, V-205685, V-205686, V-205687, V-205688, V-205689, V-205690, V-205691, V-205692, V-205693, V-205694, V-205695, V-205696, V-205697, V-205698, V-205699, V-205700, V-205701, V-205702, V-205703, V-205704, V-205705, V-205706, V-205707, V-205708, V-205709, V-205710, V-205711, V-205712, V-205713, V-205714, V-205715, V-205716, V-205717, V-205718, V-205719, V-205720, V-205721, V-205722, V-205723, V-205724, V-205725, V-205726, V-205727, V-205728, V-205729, V-205730, V-205731, V-205732, V-205733, V-205734, V-205735, V-205736, V-205737, V-205738, V-205739, V-205740, V-205741, V-205742, V-205743, V-205744, V-205745, V-205746, V-205747, V-205748, V-205749, V-205750, V-205751, V-205752, V-205753, V-205754, V-205755, V-205756, V-205757, V-205758, V-205759, V-205760, V-205761, V-205762, V-205763, V-205764, V-205765, V-205766, V-205767, V-205768, V-205769, V-205770, V-205771, V-205772, V-205773, V-205774, V-205775, V-205776, V-205777, V-205778, V-205779, V-205780, V-205781, V-205782, V-205783, V-205784, V-205785, V-205786, V-205787, V-205788, V-205789, V-205790, V-205791, V-205792, V-205793, V-205794, V-205795, V-205796, V-205797, V-205798, V-205799, V-205800, V-205801, V-205802, V-205803, V-205804, V-205805, V-205806, V-205807, V-205808, V-205809, V-205810, V-205811, V-205812, V-205813, V-205814, V-205815, V-205816, V-205817, V-205818, V-205819, V-205820, V-205821, V-205822, V-205823, V-205824,

V-205825, V-205826, V-205827, V-205828, V-205829, V-205830, V-205832, V-205833, V-205834, V-205835, V-205836, V-205837, V-205838, V-205839, V-205840, V-205841, V-205842, V-205843, V-205844, V-205845, V-205846, V-205847, V-205848, V-205849, V-205850, V-205851, V-205852, V-205853, V-205854, V-205855, V-205858, V-205859, V-205860, V-205861, V-205862, V-205863, V-205865, V-205866, V-205867, V-205868, V-205869, V-205870, V-205871, V-205872, V-205873, V-205874, V-205875, V-205876, V-205877, V-205882, V-205883, V-205884, V-205885, V-205886, V-205887, V-205888, V-205890, V-205892, V-205893, V-205894, V-205895, V-205896, V-205897, V-205898, V-205899, V-205900, V-205901, V-205902, V-205903, V-205904, V-205906, V-205907, V-205908, V-205909, V-205910, V-205911, V-205912, V-205913, V-205914, V-205915, V-205916, V-205917, V-205918, V-205919, V-205920, V-205921, V-205922, V-205923, V-205924, V-205925, V-214936, and V-236001

Windows Server 2016 STIG Version 2 Release 5

This release includes the following STIG settings for Windows operating systems:

V-224828, V-224832, V-224833, V-224834, V-224835, V-224850, V-224851, V-224852, V-224853, V-224854, V-224855, V-224856, V-224857, V-224858, V-224859, V-224866, V-224867, V-224868, V-224869, V-224870, V-224871, V-224872, V-224873, V-224874, V-224877, V-224878, V-224879, V-224880, V-224881, V-224882, V-224883, V-224884, V-224885, V-224886, V-224887, V-224888, V-224889, V-224890, V-224891, V-224892, V-224893, V-224894, V-224895, V-224896, V-224897, V-224898, V-224899, V-224900, V-224901, V-224902, V-224903, V-224904, V-224905, V-224906, V-224907, V-224908, V-224909, V-224910, V-224911, V-224912, V-224913, V-224914, V-224915, V-224916, V-224917, V-224918, V-224919, V-224920, V-224922, V-224924, V-224925, V-224926, V-224927, V-224928, V-224929, V-224930, V-224931, V-224932, V-224933, V-224934, V-224935, V-224936, V-224937, V-224938, V-224939, V-224940, V-224941, V-224942, V-224943, V-224944, V-224945, V-224946, V-224947, V-224948, V-224949, V-224951, V-224952, V-224953, V-224954, V-224955, V-224956, V-224957, V-224958, V-224959, V-224960, V-224961, V-224962, V-224963, V-225010, V-225013, V-225014, V-225015, V-225016, V-225017, V-225018, V-225019, V-225020, V-225021, V-225022, V-225023, V-225024, V-225025, V-225028, V-225029, V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040, V-225041, V-225042, V-225043, V-225044, V-225045, V-225046, V-225047, V-225048, V-225049, V-225050, V-225051, V-225052, V-225053, V-225054, V-225055, V-225056, V-225057, V-225058, V-225060, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066, V-225067, V-225068, V-225069, V-225070, V-225071, V-225072, V-225073, V-225074, V-225076, V-225077,

V-225078, V-225079, V-225080, V-225081, V-225082, V-225083, V-225084, V-225085, V-225086, V-225087, V-225088, V-225089, V-225091, V-225092, V-225093, and V-236000

Windows Server 2012 R2 MS STIG Version 3 Release 5

This release includes the following STIG settings for Windows operating systems:

V-225574, V-225573, V-225572, V-225571, V-225570, V-225569, V-225568, V-225567, V-225566, V-225565, V-225564, V-225563, V-225562, V-225561, V-225560, V-225559, V-225558, V-225557, V-225556, V-225555, V-225554, V-225553, V-225552, V-225551, V-225550, V-225549, V-225548, V-225547, V-225546, V-225545, V-225544, V-225543, V-225542, V-225541, V-225540, V-225539, V-225538, V-225537, V-225536, V-225535, V-225534, V-225533, V-225532, V-225531, V-225530, V-225529, V-225528, V-225527, V-225526, V-225525, V-225524, V-225523, V-225522, V-225521, V-225520, V-225519, V-225518, V-225517, V-225516, V-225515, V-225514, V-225513, V-225512, V-225511, V-225510, V-225509, V-225508, V-225507, V-225506, V-225505, V-225504, V-225503, V-225502, V-225501, V-225500, V-225499, V-225498, V-225497, V-225496, V-225495, V-225494, V-225493, V-225492, V-225491, V-225490, V-225489, V-225488, V-225487, V-225486, V-225485, V-225484, V-225483, V-225482, V-225481, V-225480, V-225479, V-225478, V-225477, V-225476, V-225475, V-225474, V-225473, V-225472, V-225471, V-225470, V-225469, V-225468, V-225467, V-225466, V-225465, V-225464, V-225463, V-225462, V-225461, V-225460, V-225459, V-225458, V-225457, V-225456, V-225455, V-225454, V-225453, V-225452, V-225451, V-225450, V-225449, V-225448, V-225447, V-225446, V-225445, V-225444, V-225443, V-225442, V-225441, V-225440, V-225439, V-225438, V-225437, V-225436, V-225435, V-225434, V-225433, V-225432, V-225431, V-225430, V-225429, V-225428, V-225427, V-225426, V-225425, V-225424, V-225423, V-225422, V-225421, V-225420, V-225419, V-225418, V-225417, V-225416, V-225415, V-225414, V-225413, V-225412, V-225411, V-225410, V-225409, V-225408, V-225407, V-225406, V-225405, V-225404, V-225402, V-225401, V-225400, V-225399, V-225398, V-225397, V-225396, V-225395, V-225394, V-225393, V-225392, V-225391, V-225390, V-225389, V-225388, V-225387, V-225386, V-225385, V-225384, V-225383, V-225382, V-225381, V-225380, V-225379, V-225378, V-225377, V-225376, V-225375, V-225374, V-225373, V-225372, V-225371, V-225370, V-225369, V-225368, V-225367, V-225366, V-225365, V-225364, V-225363, V-225362, V-225361, V-225360, V-225359, V-225358, V-225357, V-225356, V-225355, V-225354, V-225353, V-225352, V-225351, V-225350, V-225349, V-225348, V-225347, V-225346, V-225345, V-225344, V-225343, V-225342, V-225341, V-225340, V-225339, V-225338, V-225337, V-225336, V-225335, V-225334, V-225333, V-225332, V-225331, V-225330, V-225329, V-225328, V-225327, V-225326,

V-225325, V-225324, V-225319, V-225318, V-225317, V-225316, V-225315, V-225314, V-225313, V-225312, V-225311, V-225310, V-225309, V-225308, V-225307, V-225306, V-225305, V-225304, V-225303, V-225302, V-225301, V-225300, V-225299, V-225298, V-225297, V-225296, V-225295, V-225294, V-225293, V-225292, V-225291, V-225290, V-225289, V-225288, V-225287, V-225286, V-225285, V-225284, V-225283, V-225282, V-225281, V-225280, V-225279, V-225278, V-225277, V-225276, V-225275, V-225274, V-225273, V-225272, V-225271, V-225270, V-225269, V-225268, V-225267, V-225266, V-225265, V-225264, V-225263, V-225262, V-225261, V-225260, V-225259, V-225258, V-225257, V-225256, V-225255, V-225254, V-225253, V-225252, V-225251, V-225250, V-225249, V-225248, V-225247, V-225246, V-225245, V-225244, V-225243, V-225242, V-225241, V-225240, and V-225239

Microsoft .NET Framework 4.0 STIG Version 2 Release 2

The following list contains STIG settings that apply to Windows operating system components for STIG Hardened EC2 AMIs. The following list contains STIG settings that apply for STIG Hardened Windows AMIs. Not all settings apply in all cases. For example, some STIG settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

.NET Framework on Windows Server 2019, 2016, and 2012 R2 MS

V-225238

Windows Firewall STIG Version 2 Release 1

The following list contains STIG settings that apply to Windows operating system components for STIG Hardened EC2 AMIs. The following list contains STIG settings that apply for STIG Hardened Windows AMIs. Not all settings apply in all cases. For example, some STIG settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

Windows Firewall on Windows Server 2019, 2016, and 2012 R2 MS

V-241989, V-241990, V-241991, V-241992, V-241993, V-241994, V-241995, V-241996, V-241997, V-241998, V-241999, V-242000, V-242001, V-242002, V-242003, V-242004, V-242005, V-242006, V-242007, and V-242008

Internet Explorer (IE) 11 STIG Version 2 Release 3

The following list contains STIG settings that apply to Windows operating system components for STIG Hardened EC2 AMIs. The following list contains STIG settings that apply for STIG Hardened Windows AMIs. Not all settings apply in all cases. For example, some STIG settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

IE 11 on Windows Server 2019, 2016, and 2012 R2 MS

V-46473, V-46475, V-46477, V-46481, V-46483, V-46501, V-46507, V-46509, V-46511, V-46513, V-46515, V-46517, V-46521, V-46523, V-46525, V-46543, V-46545, V-46547, V-46549, V-46553, V-46555, V-46573, V-46575, V-46577, V-46579, V-46581, V-46583, V-46587, V-46589, V-46591, V-46593, V-46597, V-46599, V-46601, V-46603, V-46605, V-46607, V-46609, V-46615, V-46617, V-46619, V-46621, V-46625, V-46629, V-46633, V-46635, V-46637, V-46639, V-46641, V-46643, V-46645, V-46647, V-46649, V-46653, V-46663, V-46665, V-46669, V-46681, V-46685, V-46689, V-46691, V-46693, V-46695, V-46701, V-46705, V-46709, V-46711, V-46713, V-46715, V-46717, V-46719, V-46721, V-46723, V-46725, V-46727, V-46729, V-46731, V-46733, V-46779, V-46781, V-46787, V-46789, V-46791, V-46797, V-46799, V-46801, V-46807, V-46811, V-46815, V-46819, V-46829, V-46841, V-46847, V-46849, V-46853, V-46857, V-46859, V-46861, V-46865, V-46869, V-46879, V-46883, V-46885, V-46889, V-46893, V-46895, V-46897, V-46903, V-46907, V-46921, V-46927, V-46939, V-46975, V-46981, V-46987, V-46995, V-46997, V-46999, V-47003, V-47005, V-47009, V-64711, V-64713, V-64715, V-64717, V-64719, V-64721, V-64723, V-64725, V-64729, V-72757, V-72759, V-72761, V-72763, V-75169, V-75171, and V-97527

Microsoft Edge STIG Version 1 Release 6

The following list contains STIG settings that apply to Windows operating system components for STIG Hardened EC2 AMIs. The following list contains STIG settings that apply for STIG Hardened Windows AMIs. Not all settings apply in all cases. For example, some STIG settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

Microsoft Edge on Windows Server 2022

V-235720, V-235721, V-235723, V-235724, V-235725, V-235726, V-235727, V-235728, V-235729, V-235730, V-235731, V-235732, V-235733, V-235734, V-235735, V-235736, V-235737, V-235738, V-235739, V-235740, V-235741, V-235742, V-235743, V-235744, V-235745, V-235746, V-235747, V-235748, V-235749, V-235750, V-235751, V-235752, V-235754, V-235756, V-235758, V-235759, V-235760, V-235761, V-235763, V-235764, V-235765, V-235766, V-235767, V-235768, V-235769, V-235770, V-235771, V-235772, V-235773, V-235774, and V-246736

Microsoft Defender STIG Version 2 Release 4

The following list contains STIG settings that apply to Windows operating system components for STIG Hardened EC2 AMIs. The following list contains STIG settings that apply for STIG Hardened Windows AMIs. Not all settings apply in all cases. For example, some STIG settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

Microsoft Defender on Windows Server 2022

V-213426, V-213427, V-213429, V-213430, V-213431, V-213432, V-213433, V-213434, V-213435, V-213436, V-213437, V-213438, V-213439, V-213440, V-213441, V-213442, V-213443, V-213444, V-213445, V-213446, V-213447, V-213448, V-213449, V-213450, V-213451, V-213452, V-213453, V-213455, V-213464, V-213465, and V-213466

Version history

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

Date	AMIs	Details
04/24/2023	Windows Server 2022 STIG Version 1 Release 1	Added support for Windows Server 2022, Microsoft Edge, and Microsoft Defender.

Date	AMIs	Details
	Microsoft Edge STIG Version 1 Release 6 Microsoft Defender STIG Version 2 Release 4	
03/01/203	Windows Server 2019 STIG Version 2 Release 5 Windows Server 2016 STIG Version 2 Release 5 Windows Server 2012 R2 MS STIG Version 3 Release 5 Microsoft .NET Framework 4.0 STIG Version 2 Release 2 Windows Firewall STIG Version 2 Release 1 Internet Explorer 11 STIG Version 2 Release 3	AMIs released for 2022 Q4 with updated versions where applicable, and applied STIGs.
07/21/202	Windows Server 2019 STIG Version 2 R4 Windows Server 2016 STIG Version 2 R4 Windows Server 2012 R2 MS STIG Version 3 R3 Microsoft .NET Framework 4.0 STIG Version 2 R1 Windows Firewall STIG Version 2 R1 Internet Explorer 11 STIG V1 R19	AMIs released with updated versions where applicable, and applied STIGs.

Date	AMIs	Details
12/15/2021	Windows Server 2019 STIG Version 2 R3 Windows Server 2016 STIG Version 2 R3 Windows Server 2012 R2 STIG Version 3 R3 Microsoft .NET Framework 4.0 STIG Version 2 R1 Windows Firewall STIG Version 2 R1 Internet Explorer 11 STIG V1 R19	AMIs released with updated versions where applicable, and applied STIGs.
6/9/2021	Windows Server 2019 STIG Version 2 R2 Windows Server 2016 STIG Version 2 R2 Windows Server 2012 R2 STIG Version 3 R2 Microsoft .NET Framework 4.0 STIG Version 2 R1 Windows Firewall STIG V1 R7 Internet Explorer 11 STIG V1 R19	Updated versions where applicable, and applied STIGs.

Date	AMIs	Details
4/5/2021	Windows Server 2019 STIG Version 2 R 1 Windows Server 2016 STIG Version 2 R 1 Windows Server 2012 R2 STIG Version 3 R 1 Microsoft .NET Framework 4.0 STIG Version 2 R 1 Windows Firewall STIG V1 R 7 Internet Explorer 11 STIG V1 R 19	Updated versions where applicable, and applied STIGs.
9/18/202	Windows Server 2019 STIG V1 R 5 Windows Server 2016 STIG V1 R 12 Windows Server 2012 R2 STIG Version 2 R 19 Internet Explorer 11 STIG V1 R 19 Microsoft .NET Framework 4.0 STIG V1 R 9 Windows Firewall STIG V1 R 7	Updated versions and applied STIGs.
12/6/201	Server 2012 R2 Core and Base V2 R17 Server 2016 Core and Base V1 R11 Internet Explorer 11 V1 R18 Microsoft .NET Framework 4.0 V1 R9 Windows Firewall STIG V1 R17	Updated versions and applied STIGs.

Date	AMIs	Details
9/17/201	Server 2012 R2 Core and Base V2 R16 Server 2016 Core and Base V1 R9 Server 2019 Core and Base V1 R2 Internet Explorer 11 V1 R17 Microsoft .NET Framework 4.0 V1 R8	Initial release.

AWS Windows AMI version history

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

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

- EC2Launch v2 version history
- EC2Launch version history
- EC2Config version history
- Systems Manager SSM Agent Release Notes
- Amazon ENA driver versions
- AWS NVME driver versions
- AWS PV driver package history
- AWS Tools for PowerShell Change Log

Monthly AMI updates for 2024 (to date)

For more information about Microsoft updates, see Description of Software Update Services and Windows Server Update Services changes in content for 2024.

Release	Changes
2024.02.14	

Release	Changes
	All AMIs • Windows Security Updates current to Febuary 13th, 2024 • AWS Tools for Windows PowerShell version 4.1.512 • cfn-init version 2.0.29 • SSM Agent version 3.2.2222.0 • SQL Server CUs installed: • SQL_2022: CU11
2024.01.16	**All AMIs** • EC2Launch v2 version 2.0.1739 • EC2Launch v1 version 1.3.2004617

Release	Changes
2024.01.10 (*Deprecated*)	**Note** *Due to functional issues with EC2Launch and EC2Launch v2, this AMI version is marked as deprecated. The AMIs are still available for launch, and are described by directly referencing their AMI ID. However, they will no longer appear in search results for public AMIs. We recommend that you use the latest AMI version, dated 2024.01.16.* **All AMIs** • Windows Security Updates current to January 9th, 2024 *Note:* Due to a known update installation issue, we excluded the standalone Windows update `KB5034439` on Windows Server 2022 Core AMIs. The update only applies to Windows installations with a separate WinRE partition. These partitions are not included with our EC2 Windows Server AMIs. For more details, see KB5034439: Windows Recovery Environment update for Azure Stack HCI, version 22H2 and Windows Server 2022: January 9, 2024 in the Microsoft documentation. • AWS Tools for PowerShell version 4.1.486 • EC2Launch v1 version 1.3.2004592 • EC2Launch v2 version 2.0.1702 • SQL Server CUs installed: • SQL_2019: CU24

Release	Changes
	Previous versions of Amazon-published Windows AMIs dated October 11th, 2023 and earlier will be made private after February 12th 2024, 10 AM Pacific.

Monthly AMI updates for 2023

For more information about Microsoft updates, see Description of Software Update Services and Windows Server Update Services changes in content for 2023.

Release	Changes
2023.12.13	**All AMIs** • Windows Security Updates current to December 12th, 2023 • AWS Tools for PowerShell version 4.1.468 • AMD Radeon Pro Driver version 22.10.01.12 • NVIDIA GRID Driver version 537.70 • NVIDIA Tesla Driver version 474.64 • SQL Server CUs installed: • SQL_2022: CU10 Previous versions of Amazon-published Windows AMIs dated September 13th, 2023 and earlier will be made private after January 8th 2024, 10 AM Pacific.
2023.11.15	

Release	Changes

	All AMIs
	
	• Windows Security Updates current to November 14th, 2023
	• AWS Tools for PowerShell version 4.1.447
	• EC2Launch version 1.3.2004491
	• SSM Agent version 3.2.1705.0
	• SQL Server CUs installed:
	  • SQL_2022: CU9
	  • SQL_20219: CU23
	• SQL Server GDRs installed:
	  • SQL 2017: KB5029376
	  • SQL 2016: KB5029186
	  • SQL 2014: KB5029185
	
	Previous versions of Amazon-published Windows AMIs dated August 10th, 2023 and earlier were made private.

Release	Changes
2023.10.11	**All AMIs** • Windows Security Updates current to October 10th, 2023 • cfn-init version 2.0.28 • EC2Launch version 1.3.2004438 • EC2Launch v2 version 2.0.1643 • SSM version 3.2.1630.0 • AWS Tools for PowerShell version 4.1.426 • SQL Server CUs installed: • SQL_2022: CU8 Previous versions of Amazon-published Windows AMIs dated July 12th, 2023 and earlier were made private.

Release	Changes

2023.09.13	**All AMIs**
	• Windows Security Updates current to September 12th, 2023
	• EC2Launch v2 version 2.0.1580
	• SSM version 3.2.1377.0
	• AWS Tools for PowerShell version 4.1.407
	• AWS NVMe driver version 1.5.0
	• SQL Server CUs installed:
	  • SQL_2022: CU7
	  • SQL_2019: CU22
	
	Windows Server 2012 RTM and Window Server 2012 R2 will reach End of Support (EOS) on October 10th, 2023 and will no longer receive regular security updates from Microsoft. On this date, AWS will no longer publish or distribute Windows Server 2012 RTM or Windows Server 2012 R2 AMIs. Existing instances running Windows Server 2012 RTM and Windows Server 2012 R2 will not be impacted. Custom AMIs in your account will also not be impacted. You can continue to use them normally after the EOS date.
	
	Previous versions of Amazon-published Windows AMIs dated June 14th, 2023 and earlier were made private.

Release	Changes
2023.08.10	**All AMIs** • Windows Security Updates current to August 8th, 2023 • AWS Tools for PowerShell version 4.1.383 • EC2Config version 4.9.5467 • SSM version 3.1.2282.0 • AWS ENA version 2.6.0 • cfn-init version 2.0.26 • SQL Server CUs installed: • SQL_2022: CU6 Windows Server 2012 RTM and Window Server 2012 R2 will reach End of Support (EOS) on October 10th, 2023 and will no longer receive regular security updates from Microsoft. On this date, AWS will no longer publish or distribute Windows Server 2012 RTM or Windows Server 2012 R2 AMIs. Existing instances running Windows Server 2012 RTM and Windows Server 2012 R2 will not be impacted. Custom AMIs in your account will also not be impacted. You can continue to use them normally after the EOS date. Previous versions of Amazon-published Windows AMIs dated May 10th, 2023 and earlier were made private.

Release	Changes

2023.07.12	**All AMIs**
	• Windows Security Updates current to July 11th, 2023
	• AWS Tools for Windows PowerShell version 4.1.366
	• EC2Launch version 1.3.2004256
	• EC2Launch v2 version 2.0.1521
	• SQL Server CUs installed:
	  • SQL_2022: CU5
	  • SQL_2019: CU21
	
	.NET Framework 3.5 is now enabled in Windows Server 2012 R2 AMIs due to Microsoft security updates. If these updates are applied before .NET 3.5 is enabled, it is no longer possible to enable the feature. If you prefer to disable .NET 3.5, you can do so through Server Manager or `dism` commands.
	
	Previous versions of Amazon-published Windows AMIs dated April 12th, 2023 and earlier were made private.

Release	Changes
2023.06.14	**All AMIs** • Windows Security Updates current to June 13th, 2023 • AWS Tools for Windows PowerShell version 4.1.346 • SQL Server CUs installed: • SQL_2022: CU4 The AWS Tools for Windows installation package has been deprecated, and no longer appears as an installed program in Windows AMIs provided by AWS. The AWSPowerShell Module is now installed at `C:\ProgramFiles\WindowsPowerShell\Modules\AWSPowerShell`. The .NET SDK remains located at `C:\ProgramFiles (x86)\AWS SDK for .NET`. For more information see the blog announcement. Windows Server 2012 RTM and Windows Server 2012 R2 will reach End of Support (EOS) on October 10, 2023 and will no longer receive regular security updates from Microsoft. On this date, AWS will no longer publish or distribute Windows Server 2012 RTM or Windows Server 2012 R2 AMIs. Existing RTM/R2 instances and custom AMIs in your account will not be impacted, and you can continue to use them after the EOS date. For more information about Microsoft End of Support on AWS, including upgrade and import options, as well as a full list of AMIs that will no longer be published or distributed on October 10, 2023, see the End of Support for Microsoft Products FAQ. Previous versions of Amazon-published Windows AMIs dated March 15th, 2023 and earlier were made private.

Release	Changes

2023.05.10	**All AMIs**
	• Windows Security Updates current to May 9th, 2023
	• AWS Tools for Windows PowerShell version 3.15.2072
	• EC2Launch v2 version 2.0.1303
	• cfn-init version 2.0.25
	• SQL Server CUs installed:
	  • SQL_2022: CU3
	  • SQL_2019: CU20
	
	Previous versions of Amazon-published Windows AMIs dated February 15th, 2023 and earlier were made private.

Release	Changes
2023.04.12	**All AMIs** • Windows Security Updates current to April 11th, 2023 • AWS Tools for Windows PowerShell version 3.15.2035 • AWS NVMe driver version 1.4.2 • SQL Server CUs installed: • SQL_2022: CU 2 • SSM version 3.1.2144.0 **Windows Server 2016, 2019, and 2022** • Intel 82599 VF driver version 2.1.249.0 **Windows Server 2012 R2** • Intel 82599 VF driver version 1.2.317.0 Previous versions of Amazon-published Windows AMIs dated January 19th, 2023 and earlier were made private.

Release	Changes

2023.03.15	**All AMIs**
	• Windows Security Updates current to March 14th, 2023
	• AWS Tools for Windows PowerShell version 3.15.1998
	• EC2Config version 4.9.5288
	• EC2Launch version 1.3.2004052
	• EC2Launch v2 version 2.0.1245
	• cfn-init version 2.0.24
	• SQL Server CUs installed:
	  • SQL_2022: CU 1
	  • SQL_2019: CU 19
	• SQL Server GDRs installed:
	  • SQL_2017: KB5021126
	  • SQL_2016: KB5021129
	  • SQL_2014: KB5021045
	
	Previous versions of Amazon-published Windows AMIs dated December 28th, 2022 and earlier were made private.

Release	Changes
2023.02.15	**All AMIs** • Windows Security Updates current to February 14th, 2023 • AWS Tools for Windows PowerShell version 3.15.1958 • AWS PV version 8.4.3 **New Windows AMIs** • TPM-Windows_Server-2019-English-Full-SQL_2019_Enterprise • TPM-Windows_Server-2019-English-Full-SQL_2019_Standard • TPM-Windows_Server-2022-English-Full-SQL_2022_Enterprise • TPM-Windows_Server-2022-English-Full-SQL_2022_Standard New Windows AMIs with Microsoft SQL Server with support for NitroTPM and UEFI Secure Boot have been released. The images include Windows Server 2019 or Windows Server 2022 with SQL Server 2019 or SQL Server 2022. Each SQL Server version is available in Standard and Enterprise editions. Previous versions of Amazon-published Windows AMIs dated November 21st, 2022 and earlier were made private.

Release	Changes
2023.01.19	**All AMIs** • cfn-init version 2.0.21 Previous versions of Amazon-published Windows AMIs dated October 27th, 2022 and earlier were made private.
2023.01.11	**All AMIs** • Windows security updates current to January 10th, 2023 • AWS Tools for Windows PowerShell version 3.15.1919 • EC2Launch version 1.3.2003975 • EC2Launch v2 version 2.0.1121

Monthly AMI updates for 2022

For more information about Microsoft updates, see Description of Software Update Services and Windows Server Update Services changes in content for 2022.

Release	Changes
2022.12.28	**Windows Server 2016 and 2019 AMIs** • EC2Launch version 1.3.2003975
2022.12.14	

Release	Changes
	All AMIs • Windows security updates current to December 13th, 2022 • AWS Tools for Windows PowerShell version 3.15.1886 • EC2Config version 4.9.5103 • EC2Launch version 1.3.2003961 • EC2Launch v2 version 2.0.1082 • SSM version 3.1.1856.0 • cfn-init version 2.0.19

Release	Changes
2022.11.21	**New Windows AMIs** • Windows_Server-2019-English-Full-SQL_2022_Enterprise • Windows_Server-2019-English-Full-SQL_2022_Express • Windows_Server-2019-English-Full-SQL_2022_Standard • Windows_Server-2019-English-Full-SQL_2022_Web • Windows_Server-2019-Japanese-Full-SQL_2022_Enterprise • Windows_Server-2019-Japanese-Full-SQL_2022_Standard • Windows_Server-2019-Japanese-Full-SQL_2022_Web • Windows_Server-2022-English-Full-SQL_2022_Enterprise • Windows_Server-2022-English-Full-SQL_2022_Express • Windows_Server-2022-English-Full-SQL_2022_Standard • Windows_Server-2022-English-Full-SQL_2022_Web • Windows_Server-2022-Japanese-Full-SQL_2022_Enterprise • Windows_Server-2022-Japanese-Full-SQL_2022_Standard • Windows_Server-2022-Japanese-Full-SQL_2022_Web Previous versions of Amazon-published Windows AMIs dated August 10, 2022 and earlier were made private.

Release	Changes
2022.11.17	**All AMIs** • EC2Config version 4.9.5064. This is an out of band release for images that use EC2Config as the default launch agent. This includes all Windows Server 2012 RTM and Windows Server 2012 R2 AMIs. This release updates EC2Config to the latest version to improve support for our newest EC2 instance types.

2022.11.10	**All AMIs**
	• Windows security updates current to November 8th, 2022
	• AWS Tools for Windows PowerShell version 3.15.1846
	• EC2Launch version 1.3.2003923
	• EC2Launch v2 version 2.0.1011
	• SQL Server CUs installed:
	  • SQL_2019: CU 18
	  • SQL_2017: CU 31
	• cfn-init version 2.0.18

Release	Changes
2022.10.27	**All AMIs** • Out-of-band updates applied to resolve issues resulting from October patches. For additional details, see https://learn.microsoft.com/en-us/windows/release-health/status-windows-10-20h2#2924msgdesc. Previous versions of Amazon-published Windows AMIs dated July 13, 2022 and earlier were made private.
2022.10.12	**All AMIs** • Windows security updates current to October 11th, 2022 • AWS Tools for Windows PowerShell version 3.15.1809 • EC2Launch version 1.3.2003857 • SSM version 3.1.1732.0 • cfn-init version 2.0.16

Release	Changes
2022.09.14	**All AMIs** • Windows security updates current to September 13th, 2022 • AWS Tools for Windows PowerShell version 3.15.1772 • EC2Launch version 1.3.2003824 • SQL Server CU installed: • SQL_2019: CU17 Previous versions of Amazon-published Windows AMIs dated June 15, 2022 and earlier were made private.
2022.08.10	**All AMIs** • Windows security updates current to August 9th, 2022 • AWS Tools for Windows PowerShell version 3.15.1737 • cfn-init version 2.0.15 • SSM version 3.1.1634.0 (only AMIs that include EC2Launch v1 or v2) • SQL Server CU installed: • SQL_2017: CU30 Previous versions of Amazon-published Windows AMIs dated May 25, 2022 and earlier were made private.

Release	Changes

2022.07.13	**All AMIs**
	
	• Windows security updates current to July 12th, 2022
	• AWS Tools for Windows PowerShell version 3.15.1706
	• cfn-init version 2.0.12
	• EC2Launch version 1.3.2003691
	• EC2Launch v2 version 2.0.863
	• SQL Server GDRs installed:
	  • SQL_2019: KB5014353
	  • SQL_2017: KB5014553
	  • SQL_2016: KB5014355
	  • SQL_2014: KB5014164
	
	Windows Server version 20H2 will reach end-of-support on August 9th, 2022. Existing instances and custom images owned by your account that are based on Windows Server version 20H2 will not be impacted. If you would like to retain access to Windows Server version 20H2, create a custom image in your account prior to August 9th, 2022. All public versions of the following images will be made private on the end-of-support date.
	
	• Windows_Server-20H2-English-Core-Base
	• Windows_Server-20H2-English-Core-ContainersLatest

Release	Changes
	Previous versions of Amazon-published Windows AMIs dated April 13th, 2022 and earlier were made private.

Release	Changes
2022.06.15	**All AMIs** • Windows security updates current to June 14th, 2022 • AWS Tools for Windows PowerShell version 3.15.1678 • AWS NVMe version 1.4.1 • EC2Config version 4.9.4588 • EC2Launch version 1.3.2003639 • SSM version 3.1.1188.0 Microsoft SQL Server 2012 is reaching end-of-support on July 12th, 2022. All public versions of the following images have been made private. Existing instances and custom images owned by your account that are based on Windows Server images containing SQL Server 2012 will not be impacted. • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2012_SP4_Enterprise-* • Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP4_Enterprise-* • Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP4_Express-* • Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP4_Standard-* • Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP4_Web-* • Windows_Server-2012-RTM-Japanese-64Bit-SQL_2012_SP4_Express-* • Windows_Server-2012-RTM-Japanese-64Bit-SQL_2012_SP4_Standard-*

Release	Changes
	• Windows_Server-2012-RTM-Japanese-64Bit-SQL_2012_SP4_Web-* • Windows_Server-2016-English-64Bit-SQL_2012_SP4_Enterprise-* • Windows_Server-2016-English-Full-SQL_2012_SP4_Standard-* For more information on Windows Server product lifecycles, please consult the following Microsoft documentation and AWS Microsoft FAQ: • https://docs.microsoft.com/en-us/lifecycle/products/microsoft-sql-server-2012 • https://aws.amazon.com/windows/faq/#eos-mhttps://aws.amazon.com/windows/faq/#eos-m
2022.05.25	**All AMIs** • Out-of-band updates applied to resolve issues resulting from May patches. For additional details, see https://docs.microsoft.com/en-us/windows/release-health/status-windows-10-20h2#2826msgdesc. Previous versions of Amazon-published Windows AMIs dated February 10, 2022 and earlier were made private.

Release	Changes

2022.05.11	**All AMIs**
	• Windows security updates current to May 10th, 2022
	• AWS Tools for Windows PowerShell version 3.15.1643
	• AWS PV version 8.4.2
	• AWS ENA version 2.4.0
	• SQL Server CUs installed:
	  • SQL_2019: CU 16
	  • SQL_2017: CU 29

2022.05.05	**New Windows AMIs** New Windows AMIs with support for NitroTPM and UEFI Secure Boot have been released. These images feature EC2Launch v2 as the default launch agent. They are available to launch on any instance type that supports NitroTPM and UEFI boot mode. • TPM-Windows_Server-2022-English-Core-Base-2022.05.05 • TPM-Windows_Server-2022-English-Full-Base-2022.05.05 • TPM-Windows_Server-2019-English-Core-Base-2022.05.05 • TPM-Windows_Server-2019-English-Full-Base-2022.05.05 • TPM-Windows_Server-2016-English-Core-Base-2022.05.05 • TPM-Windows_Server-2016-English-Full-Base-2022.05.05

Release	Changes
2022.04.13	**All AMIs** • Windows security updates current to April 12th, 2022 • AWS Tools for Windows PowerShell version 3.15.1620 Previous versions of Amazon-published Windows AMIs dated January 21st, 2022 and earlier were made private. After June 2022, we will no longer release updated versions of the following images that include SQL Server 2016 SP2. SQL Server SP3 AMIs are available and will continue to be updated and released monthly. • Windows_Server-2019-English-Full-SQL_2016_SP2_Web • Windows_Server-2019-English-Full-SQL_2016_SP2_Standard • Windows_Server-2019-English-Full-SQL_2016_SP2_Express • Windows_Server-2019-English-Full-SQL_2016_SP2_Enterprise • Windows_Server-2016-Korean-Full-SQL_2016_SP2_Standard • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Web • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Standard • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Express • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Enterprise • Windows_Server-2016-English-Full-SQL_2016_SP2_Web • Windows_Server-2016-English-Full-SQL_2016_SP2_Standard

Release	Changes
	• Windows_Server-2016-English-Full-SQL_2016_SP2_Express • Windows_Server-2016-English-Full-SQL_2016_SP2_Enterprise • Windows_Server-2016-English-Core-SQL_2016_SP2_Web • Windows_Server-2016-English-Core-SQL_2016_SP2_Standard • Windows_Server-2016-English-Core-SQL_2016_SP2_Express • Windows_Server-2016-English-Core-SQL_2016_SP2_Enterprise • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Web • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Standard • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Express • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Enterprise • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Web • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Standard • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Express • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Enterprise

Release	Changes
2022.03.09	**All AMIs** • Windows security updates current to March 8th, 2022 • AWS Tools for Windows PowerShell version 3.15.1583 • AWS ENA version 2.2.3 (reverted due to potential performance degradation on 6th generation EC2 instances) • EC2Config version 4.9.4556 • SSM version 3.1.1045.0 • SQL Server CUs installed: • SQL_2019: CU 15 Previous versions of Amazon-published Windows AMIs dated December 12th, 2021 and earlier were made private.

Release	Changes
2022.02.10	**All AMIs** • Windows security updates current to February 8th, 2022 • AWS Tools for Windows PowerShell version 3.15.1546 • cfn-init version 2.0.10 • EC2Config version 4.9.4536 • EC2Launch version 1.3.2003498 • EC2Launch v2 version 2.0.698 • SSM version 3.1.804.0 • SQL Server CUs installed: • SQL_2017: CU 28 Previous versions of Amazon-published Windows AMIs dated November 16th, 2021 and earlier were made private.
2022.01.19	**All AMIs** • Out-of-band updates applied to resolve issues resulting from January patches. For more details, see https://docs.microsoft.com/en-us/windows/release-health/windows-message-center#2777. Previous versions of Amazon-published Windows AMIs dated October 13th, 2021 and earlier were made private.

Release	Changes
2022.01.12	**All AMIs** • Windows security updates current to January 11th, 2022 • AWS Tools for Windows PowerShell version 3.15.1511 • AWS PV version 8.4.1 • SQL Server CUs installed: • SQL_2019: CU 14

Monthly AMI updates for 2021

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

Release	Changes

2021.12.15	**All AMIs**
	• Windows security updates current to December 14th, 2021
	• AWS Tools for Windows PowerShell version 3.15.1494
	• AWS NVMe version 1.4.0
	• SQL Server CUs installed:
	  • SQL_2017: CU 27
	  • SQL_2019: CU 13

Release	Changes
	Previous versions of Amazon-published Windows AMIs dated September 15th, 2021 and earlier were made private.
2021.11.16	**Windows Server 2022 and EC2LaunchV2-* AMIs** • EC2Launch v2 version 2.0.674 Windows Server 2004 reached End-of-support on December 14th, 2021. All public versions of the following images have been made private. Existing instances and custom images owned by your account that are based on Windows Server 2004 will not be impacted. • Windows_Server-2004-English-Core-Base • Windows_Server-2004-English-Core-ContainersLatest

Release	Changes
2021.11.10	**All AMIs** • Windows security updates current to November 9th, 2021 • AWS Tools for Windows PowerShell version 3.15.1451 • AWS ENA version 2.2.4 • SQL Server CUs installed: • SQL_2017: CU 26 **New Windows AMIs** • Windows_Server-2022-Japanese-Full-SQL_2019_Enterprise-2021.11.10 • Windows_Server-2022-Japanese-Full-SQL_2019_Standard-2021.11.10 • Windows_Server-2022-Japanese-Full-SQL_2019_Web-2021.11.10 • Windows_Server-2022-Japanese-Full-SQL_2017_Enterprise-2021.11.10 • Windows_Server-2022-Japanese-Full-SQL_2017_Standard-2021.11.10 • Windows_Server-2022-Japanese-Full-SQL_2017_Web-2021.11.10

Release	Changes
2021.10.13	**All AMIs** • Windows security updates current to October 12, 2021 • AWS Tools for Windows PowerShell version 3.15.1421 • SSM version 3.1.338.0 **Windows Server 2022 and EC2LaunchV2_Preview AMIs** • EC2Launch v2 version 2.0.651 **Windows Server 2012 RTM and R2 AMIs** • EC2Config version 4.9.4508 **New Windows AMIs** • Windows_Server-2022-English-Full-SQL_2019_Enterprise-2021.10.13 • Windows_Server-2022-English-Full-SQL_2019_Standard-2021.10.13 • Windows_Server-2022-English-Full-SQL_2019_Web-2021.10.13 • Windows_Server-2022-English-Full-SQL_2019_Express-2021.10.13 • Windows_Server-2022-English-Full-SQL_2017_Enterprise-2021.10.13 • Windows_Server-2022-English-Full-SQL_2017_Standard-2021.10.13 •

Release	Changes
	Windows_Server-2022-English-Full-SQL_2017_Web-2021.10.13 • Windows_Server-2022-English-Full-SQL_2017_Express-2021.10.13 **New EC2Launch v2 AMIs** The following AMIs with EC2Launch v2 long-term support are now available. The following AMIs include EC2Launch v2 as the default launch agent and will be updated with new versions each month. • EC2LaunchV2-Windows_Server-2019-English-Full-Base-2021.10.13 • EC2LaunchV2-Windows_Server-2019-English-Core-Base-2021.10.13 • EC2LaunchV2-Windows_Server-2019-English-Full-ContainersLatest-2021.10.13 • EC2LaunchV2-Windows_Server-2016-English-Full-Base-2021.10.13 • EC2LaunchV2-Windows_Server-2016-English-Core-Base-2021.10.13 • EC2LaunchV2-Windows_Server-2012_R2_RTM-English-Full-Base-2021.10.13 • EC2LaunchV2-Windows_Server-2012_RTM-English-Full-Base-2021.10.13 EC2LaunchV2_Preview AMIs are discontinued, and will not be updated with new versions. However, earlier versions will continue to be available until January 2022. Existing images and custom images based on EC2Launch V2_Preview AMIs will not be impacted, and you can continue to use them in your account. We recommend that you use the new EC2Launch v2 AMIs going forward to receive security and software updates.

Release	Changes
	Windows Server 2004 will reach End-of-support on December 14th, 2021. All public versions of the following images will be made private on December 14th, 2021. Existing instances and custom images owned by your account that are based on Windows Server 2004 will not be impacted. If you want to retain access to Windows Server 2004, create a custom image in your account prior to December 14th. • Windows_Server-2004-English-Core-Base • Windows_Server-2004-English-Core-ContainersLatest Previous versions of Amazon-published Windows AMIs dated July 14th, 2021 and earlier were made private.

Release	Changes

2021.09.15	**All AMIs**
	• Windows security updates current to September 14th, 2021
	• AWS Tools for Windows PowerShell version 3.15.1398
	• SSM version 3.1.282.0
	• SQL Server CUs installed:
	  • SQL_2019: CU12
	  • SQL_2017: CU 25
	
	Windows Server 2022 and EC2LaunchV2_Preview AMIs
	• EC2Launch v2 version 2.0.592
	
	Windows Server 2012 RTM and R2 AMIs
	• EC2Config version 4.9.4500
	
	Previous versions of Amazon-published Windows AMIs dated June 9th, 2021 and earlier were made private.

Release	Changes
2021.09.01	**New Windows AMIs** • Windows_Server-2022-English-Full-Base-2021.08.25 • Windows_Server-2022-English-Full-ContainersLatest-2021.08.25 • Windows_Server-2022-English-Core-Base-2021.08.25 • Windows_Server-2022-English-Core-ContainersLatest-2021.08.25 • Windows_Server-2022-Chinese_Simplified-Full-Base-2021.08.25 • Windows_Server-2022-Chinese_Traditional-Full-Base-2021.08.25 • Windows_Server-2022-Czech-Full-Base-2021.08.25 • Windows_Server-2022-Dutch-Full-Base-2021.08.25 • Windows_Server-2022-French-Full-Base-2021.08.25 • Windows_Server-2022-German-Full-Base-2021.08.25 • Windows_Server-2022-Hungarian-Full-Base-2021.08.25 • Windows_Server-2022-Italian-Full-Base-2021.08.25 • Windows_Server-2022-Japanese-Full-Base-2021.08.25 • Windows_Server-2022-Korean-Full-Base-2021.08.25 • Windows_Server-2022-Polish-Full-Base-2021.08.25 • Windows_Server-2022-Portuguese_Brazil-Full-Base-2021.08.25 • Windows_Server-2022-Portuguese_Portugal-Full-Base-2021.08.25

Release	Changes
	• Windows_Server-2022-Russian-Full-Base-2021.08.25 • Windows_Server-2022-Spanish-Full-Base-2021.08.25 • Windows_Server-2022-Swedish-Full-Base-2021.08.25 • Windows_Server-2022-Turkish-Full-Base-2021.08.25 Windows Server 2022 AMIs include EC2Launch v2 by default. For more information, see EC2Launch v2 overview. **EC2LaunchV2_Preview AMIs** • EC2Launch v2 version 2.0.592 Previous versions of Amazon-published Windows AMIs dated May 12th, 2021 and earlier were made private.

Release	Changes
2021.08.11	**All AMIs** • Windows security updates current to August 10th, 2021 • AWS Tools for Windows PowerShell version 3.15.13571 • EC2Launch version 1.3.2003411 • SSM version 3.0.1181.0 • SQL Server CUs installed: • SQL_2019: CU11 **EC2LaunchV2_Preview AMIs** • EC2Launch v2 version 2.0.548 Previous versions of Amazon-published Windows AMIs dated April 14th, 2021 and earlier were made private.

Release	Changes
2021.07.14	**All AMIs** • Windows security updates current to July 13th, 2021 • AWS Tools for Windows PowerShell version 3.15.1350 • EC2Launch version 1.3.2003364 • SQL Server CUs installed: • SQL_2017: CU24

Release	Changes
2021.07.07	**All AMIs** Out-of-band AMI release that applies the July out-of-band security update recently released by Microsoft as an additional mitigation to CVE-34527. **Note** `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\PointAndPrint` is not defined on Windows AMIs provided by AWS, which is the default state. For more information, see: • https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527 • https://support.microsoft.com/en-us/topic/kb5005010-restricting-installation-of-new-printer-drivers-after-applying-the-july-6-2021-updates-31b91c02-05bc-4ada-a7ea-183b129578a7 Previous versions of Amazon-published Windows AMIs dated March 10th, 2021 and earlier were made private.

Release	Changes
2021.06.09	**All AMIs** • Windows security updates current to June 8th, 2021 • AWS Tools for Windows PowerShell version 3.15.1326 • SSM version 3.0.1124.0 **Windows Server 2012RTM/2012 R2 AMIs** • EC2Config version 4.9.4419

Release	Changes
2021.05.12	**All AMIs** • Windows security updates current to May 11th, 2021 • AWS Tools for Windows PowerShell version 3.15.1302 • EC2Launch version 1.3.2003312 • SQL Server CUs installed: • SQL_2019: CU10 • Previous versions of Amazon-published Windows AMIs dated February 10th, 2021 and earlier were made private. **Windows Server 2012RTM/2012 R2 AMIs** • EC2Config version 4.9.4381 • SSM version 3.0.529.0 **NVIDIA GPU AMIs** • GRID version 462.31 • Tesla version 462.31 **Radeon GPU AMIs** • Radeon version 20.10.25.04

Release	Changes

Release	Changes

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

Release	Changes
	Windows_Server-1909-English-Core-ContainersLatest **EC2LaunchV2_Preview AMIs** • EC2Launch v2 version 2.0.285

Release	Changes

2021.03.11	**All AMIs**
	• Windows security updates current to March 9th, 2021
	• AWS Tools for Windows PowerShell version 3.15.1248
	• cfn-init version 2.0.5. This package includes Microsoft Visual C++ 2015-2019 Redistributable version 14.28.29910.0 as a dependency.
	• EC2Launch version 1.3.2003236
	• SSM Agent version 3.0.529.0
	• NVIDIA GRID version 461.33
	• SQL Server CUs installed:
	  • SQL 2016_SP2: CU16
	  • SQL 2019: CU9
	• KB4577586 update for the removal of Adobe Flash Player installed on all applicable images (Adobe Flash player is not enabled by default on all images).
	
	Note
	Amazon Root CAs have been added to the Trusted Root Certification Authorities certificate store on all AMIs. For more information, see https://www.amazontrust.com/repository/#rootcas.

Release	Changes
	Windows Server 2016 and 2019 AMIs • Updated from default .NET framework versions to version 4.8. **Windows Server 2012RTM/2012 R2 AMIs** • EC2Config version 4.9.4326 • SSM Agent version 3.0.431.0

Release	Changes
2021.02.10	**All AMIs** • Windows security updates current to February 9th, 2021 • AWS Tools for Windows PowerShell version 3.15.1224 • NVIDIA GRID version 461.09 Beginning in March 2021, Windows AMIs provided by AWS include Amazon Root CAs in the certificate store to minimize potential disruption from the upcoming S3 and CloudFront certificate migration, which is scheduled for March 23rd, 2021. For more information, see the following: • https://aws.amazon.com/blogs/security/how-to-prepare-for-aws-move-to-its-own-certificate-authority/ • https://forums.aws.amazon.com/ann.jspa?annID=7541 Additionally, AWS will apply "update for Removal of Adobe Flash Player" (KB4577586) to all Windows AMIs in March to remove the built-in Adobe Flash player, which ended support on December 31st, 2020. If your use case requires the built-in Adobe Flash player, we recommend creating a custom image based on AMIs with version 2021.02.10 or earlier. For more information on the End of Support of Adobe Flash Player, see: • https://blogs.windows.com/msedgedev/2020/09/04/update-adobe-flash-end-support/ • https://www.adobe.com/products/flashplayer/end-of-life.html

Release	Changes
	EC2LaunchV2_Preview AMIs • EC2Launch v2 version 2.0.207 **New Windows AMIs** • Windows_Server-2016-Japanese-Full-SQL_2019_Enterprise-2021.02.10 • Windows_Server-2016-Japanese-Full-SQL_2019_Standard-2021.02.10 • Windows_Server-2016-Japanese-Full-SQL_2019_Web-2021.02.10 • Windows_Server-2019-Japanese-Full-SQL_2019_Enterprise-2021.02.10 • Windows_Server-2019-Japanese-Full-SQL_2019_Standard-2021.02.10 • Windows_Server-2019-Japanese-Full-SQL_2019_Web-2021.02.10

Release	Changes
2021.01.13	**All AMIs** • Windows security updates current to January 12th, 2021 • AWS Tools for Windows PowerShell version 3.15.1204 • AWS ENA version 2.2.2 • EC2Launch v1 version 1.3.2003210 **Windows Server SAC/2019/2016 AMIs** • SSM Agent version 3.0.431.0

Monthly AMI updates for 2020

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

Release	Changes
2020.12.09	**All AMIs** • Windows security updates current to December 8th, 2020 • AWS Tools for Windows PowerShell version 3.15.1181 • All SQL Server Enterprise, Standard, and Web AMIs now include SQL Server installation media at `C:\SQLServerSetup` •

Release	Changes
	EC2Launch v1 version 1.3.2003189 • Previous versions of Amazon-published Windows AMIs dated September 9th, 2020 and earlier were made private. **Windows Server 2012/2012 R2 AMIs** • EC2Config version 4.9.4279 • SSM Agent version 2.3.871.0 **EC2LaunchV2_Preview AMIs** • EC2Launch v2 version 2.0.160

Release	Changes

2020.11.11	**All AMIs**
	• Windows security updates current to November 10th, 2020
	• AWS Tools for Windows PowerShell version 3.15.1160
	• SQL Server CUs installed:
	  • SQL 2016 SP2: CU15
	  • SQL 2017: CU22
	  • SQL 2019: CU8
	• SSM Agent version 2.3.1644.0
	• EC2Launch v2 Preview AMIs: EC2Launch version 2.0.153
	• Previous versions of Amazon-published Windows AMIs dated August 12th, 2020 and earlier were made private.
	
	New Windows AMIs
	• Windows_Server-20H2-English-Core-Base-2020.11.11
	• Windows_Server-20H2-English-Core-ContainersLatest-2020.11.11

Release	Changes
2020.10.14	**All AMIs** • Windows security updates current to October 13th, 2020 • AWS Tools for Windows PowerShell version 3.15.1140 • NVIDIA GRID version 452.39 • EC2Launch v2 Preview AMIs: EC2Launch version 2.0.146 • AWS ENA version 2.2.1 • cfn-init version 1.4.34 • Previous versions of Amazon-published Windows AMIs dated July 15th, 2020 and earlier were made private.

Release	Changes
2020.9.25	A new version of Amazon Machine Images with SQL Server 2019 dated 2020.09.25 has been released. This release includes the same software components as the previous release dated 2020.09.09 but does not include CU7 for SQL 2019, which has recently been removed from public availability by Microsoft due to a known issue with reliability of the database snapshot feature. For more information, please see the following Microsoft blog post: https://techcommunity.microsoft.com/t5/sql-server/cumulative-update-7-for-sql-server-2019-rtm-removed/ba-p/1629317. **New Windows AMIs** • Windows_Server-2016-English-Full-SQL_2019_Enterprise-2020.09.25 • Windows_Server-2016-English-Full-SQL_2019_Express-2020.09.25 • Windows_Server-2016-English-Full-SQL_2019_Standard-2020.09.25 • Windows_Server-2016-English-Full-SQL_2019_Web-2020.09.25 • Windows_Server-2019-English-Full-SQL_2019_Enterprise-2020.09.25 • Windows_Server-2019-English-Full-SQL_2019_Express-2020.09.25 • Windows_Server-2019-English-Full-SQL_2019_Standard-2020.09.25 • Windows_Server-2019-English-Full-SQL_2019_Web-2020.09.25 **EC2LaunchV2_Preview AMIs** • EC2LaunchV2_Preview-Windows_Server-2019-English-Full-SQL_2019_Express-2020.09.25

Release	Changes

2020.9.9	**All AMIs**
	• Windows security updates current to September 8th, 2020
	• AWS PV drivers version 8.3.4
	• AWS ENA version 2.2.0
	• AWS Tools for Windows PowerShell version 3.15.1110
	• SQL Server CUs installed
	  • SQL_2016_SP2: CU14
	  • SQL_2019: CU7
	• Previous versions of Amazon-published Windows AMIs dated June 10th, 2020 and earlier were made private.
	
	Windows Server 2016/2019/1809/1903/1909/2004 AMIs
	• EC2Launch version 1.3.2003155
	• SSM Agent version 2.3.1319.0
	
	EC2LaunchV2_Preview AMIs
	• EC2Launch v2 version 2.0.124

Release	Changes
2020.8.12	**All AMIs** • Windows security updates current to August 11th, 2020 • AWS Tools for Windows PowerShell version 3.15.1084 • G3 AMIs: NVIDIA GRID version 451.48 • EC2Launch v2 Preview AMIs: EC2Launch version 2.0.104 • SQL CUs installed • SQL_2019: CU6 • Previous versions of Amazon-published Windows AMIs dated May 13th, 2020 and earlier were made private.

Release	Changes

2020.7.15	**All AMIs**
	• Windows security updates current to July 14th, 2020
	• AWS Tools for Windows PowerShell version 3.15.1064
	• ENA version 2.1.5
	• SQL Server CUs installed
	  • SQL_2017: CU21
	  • SQL_2019: CU5
	• Previous versions of Amazon-published Windows AMIs dated April 15th, 2020 and earlier were made private.

Release	Changes
2020.7.01	A new version of Amazon Machine Images has been released. These images include EC2Launch v2 and serve as a functional preview of the new launch agent in advance of it being included by default on all Windows AMIs currently provided by AWS later this year. Note that some SSM documents and dependent services, such as EC2 Image Builder, may require updates to support EC2 Launch v2. These updates will follow in the coming weeks. These images are not recommended for use in production environments. You can read more about EC2Launch v2 at https://aws.amazon.com/about-aws/whats-new/2020/07/introducing-ec2-launch-v2-simplify-customizing-windows-instances/ and Configure a Windows instance using EC2Launch v2. All current Windows Server AMIs will continue to be provided without changes to the current launch agent, either EC2Config (Server 2012 RTM or 2012 R2) or EC2Launch v1 (Server 2016 or later), for the next several months. In the near future, all Windows Server AMIs currently provided by AWS will be migrated to use EC2Launch v2 by default as part of the monthly release. EC2LaunchV2_Preview AMIs will be updated monthly and remain available until this migration occurs. **New Windows AMIs** • EC2LaunchV2_Preview-Windows_Server-2004-English-Core-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2019-English-Full-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2019-English-Core-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2016-English-Full-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2016-English-Core-Base-2020.06.30

Release	Changes
	• EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Full-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Core-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2012_RTM-English-Full-Base-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2019-English-Full-SQL_2019_Express-2020.06.30 • EC2LaunchV2_Preview-Windows_Server-2016-English-Full-SQL_2017_Express-2020.06.30
2020.6.10	**All AMIs** • Windows security updates current to June 9th, 2020 • AWS Tools for Windows PowerShell version 3.15.1034 • cfn-init version 1.4.33 • SQL CU installed: SQL_2016_SP2: CU13

Release	Changes
2020.5.27	**New Windows AMIs** • Windows_Server-2004-English-Core-Base-2020.05.27 • Windows_Server-2004-English-Core-ContainersLatest-2020.05.27
2020.5.13	**All AMIs** • Windows security updates current to May 12th, 2020 • AWS Tools for Windows PowerShell version 3.15.1013 • EC2Launch version 1.3.2003150

Release	Changes

2020.4.15	**All AMIs**
	• Windows security updates current to April 14th, 2020
	• AWS Tools for Windows PowerShell version 3.15.998
	• EC2Config version 4.9.4222
	• EC2Launch version 1.3.2003040
	• SSM Agent version 2.3.842.0
	• SQL Server CUs installed:
	  • SQL_2017: CU 20
	  • SQL_2019: CU 4

2020.3.18	**Windows Server 2019 AMIs** Resolves an intermittent issue discovered in the 2020.3.11 release in which the Background Intelligent Transfer Service (BITS) may not start within the expected time after initial OS boot, potentially resulting in timeouts, BITS errors in the event log, or failures of cmdlets involving BITS invoked quickly after the initial boot. Other Windows Server AMIs are not affected by this issue, and their latest version remains 2020.03.11.

Release	Changes

2020.3.11	**All AMIs**
	• Windows security updates current to March 10th, 2020
	• AWS Tools for Windows PowerShell version 3.15.969
	• EC2Config version 4.9.4122
	• EC2Launch version 1.3.2002730
	• SSM Agent version 2.3.814.0
	• SQL Server CUs installed:
	  • SQL_2016_SP2: CU 12
	  • SQL_2017: CU 19
	  • SQL_2019: CU 2 not applied due to known issue with SQL Agent
	• Out of band security update (KB4551762) for server core 1909 and 1903 applied to mitigate CVE-2020-0796. Other Windows Server versions are not impacted by this issue. For details, see https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0796

Release	Changes

2020.2.12	**All AMIs**
	• Windows security updates current to February 11th, 2020
	• AWS Tools for Windows PowerShell version 3.15.945
	• Intel SRIOV driver updates
	  • 2019/1903/1909: version 2.1.185.0
	  • 2016/1809: version 2.1.186.0
	  • 2012 R2: version 1.2.199.0
	• SQL Server CUs installed:
	  • SQL_2019: CU 1
	  • SQL_2017: CU 18
	  • SQL_2016_SP2: CU 11
	
	Windows Server 2008 SP2 and Windows Server 2008 R2
	Windows Server 2008 SP2 and Window Server 2008 R2 reached End of Support (EOS) on 01/14/20 and will no longer receive regular security updates from Microsoft. AWS will no longer publish or distribute Windows Server 2008 SP2 or Windows Server 2008 R2 AMIs. Existing 2008 SP2/R2 instances and custom AMIs in your account are not impacted, and you can continue to use them after the EOS date.
	For more information about Microsoft End of Service on AWS, including upgrade and import options, as well as a full list of AMIs that are no longer

Release	Changes
	published as of 01/14/2020, see End of Support (EOS) for Microsoft Products.
2020.1.15	**All AMIs** • Microsoft security updates current to January 14, 2020 • AWS Tools for Windows PowerShell version 3.15.925 • ENA version 2.1.4 **Windows Server 2008 SP2 and Windows Server 2008 R2** Windows Server 2008 SP2 and Window Server 2008 R2 reached End of Support (EOS) on 01/14/20 and will no longer receive regular security updates from Microsoft. AWS will no longer publish or distribute Windows Server 2008 SP2 or Windows Server 2008 R2 AMIs. Existing 2008 SP2/R2 instances and custom AMIs in your account are not impacted, and you can continue to use them after the EOS date. For more information about Microsoft End of Service on AWS, including upgrade and import options, as well as a full list of AMIs that are no longer published as of 01/14/2020, see End of Support (EOS) for Microsoft Products.

Monthly AMI updates for 2019

For more information about Microsoft updates, see Description of Software Update Services and Windows Server Update Services changes in content for 2019.

Release	Changes
2019.12.16	

Release	Changes
	All AMIs • Microsoft security updates current to December 10, 2019 • AWS Tools for Windows PowerShell version 3.15.903 **Windows Server 2008 SP2 and Windows Server 2008 R2** Microsoft will end mainstream support for Windows Server 2008 SP2 and Windows Server 2008 R2 on January 14, 2020. On this date, AWS will no longer publish or distribute Windows Server 2008 SP2 or Windows Server 2008 R2 AMIs. Existing 2008 SP2/R2 instances and custom AMIs in your acc ount will not be impacted and you can continue to use them after the end-of-service (EOS) date. For more information about Microsoft EOS on AWS, including upgrade and import options, along with a full list of AMIS that will no longer be published or distributed on January 14, 2020, see End of Support (EOS) for Microsoft Products.

Release	Changes
2019.11.13	**All AMIs** • AWS Tools for Windows PowerShell version 3.15.876 • Windows security updates current to November 12th, 2019 • EC2 Config version 4.9.3865 • EC2 Launch version 1.3.2002240 • SSM Agent v2.3.722.0 Previous versions of AMIs have been marked private. **New Windows AMIs** • Windows_Server-1909-English-Core-Base-2019.11.13 • Windows_Server-1909-English-Core-ContainersLatest-2019.11.13 • Windows_Server-2016-English-Full-SQL_2019_Enterprise-2019.11.13 • Windows_Server-2016-English-Full-SQL_2019_Express-2019.11.13 • Windows_Server-2016-English-Full-SQL_2019_Standard-2019.11.13 • Windows_Server-2016-English-Full-SQL_2019_Web-2019.11.13 • Windows_Server-2019-English-Full-SQL_2019_Enterprise-2019.11.13 • Windows_Server-2019-English-Full-SQL_2019_Express-2019.11.13 • Windows_Server-2019-English-Full-SQL_2019_Standard-2019.11.13 •

Release	Changes
	Windows_Server-2019-English-Full-SQL_2019_Web-2019.11.13
2019.11.05	**New Windows AMIs** New SQL AMIs available: • Windows_Server-2016-English-Full-SQL_2019_Enterprise-2019.11.05 • Windows_Server-2016-English-Full-SQL_2019_Express-2019.11.05 • Windows_Server-2016-English-Full-SQL_2019_Standard-2019.11.05 • Windows_Server-2016-English-Full-SQL_2019_Web-2019.11.05 • Windows_Server-2019-English-Full-SQL_2019_Enterprise-2019.11.05 • Windows_Server-2019-English-Full-SQL_2019_Express-2019.11.05 • Windows_Server-2019-English-Full-SQL_2019_Standard-2019.11.05 • Windows_Server-2019-English-Full-SQL_2019_Web-2019.11.05

Release	Changes
2019.10.09	**All AMIs** • AWS Tools for Windows PowerShell version 3.15.846 • Windows security updates current to October 8th, 2019 • Windows Defender platform updates current and update block via registry removed. For details, see https://support.microsoft.com/en-us/help/4513240/sfc-incorrectly-flags-windows-defender-ps-files-as-corrupted **New Windows AMIs** New ECS-optimized AMI available: • Windows_Server-2019-English-Core-ECS_Optimized-2019.10.09
2019.09.12	**New Windows AMI** • amzn2-ami-hvm-2.0.20190618-x86_64-gp2-mono .NET Core 2.2, Mono 5.18, and PowerShell 6.2 pre-installed to run your .NET applications on Amazon Linux 2 with Long Term Support (LTS)

Release	Changes
2019.09.11	**All AMIs** • AWS PV driver version 8.3.2 • AWS NVMe driver version 1.3.2 • AWS Tools for Windows PowerShell version 3.15.826 • NLA enabled on all OS 2012 RTM to 2019 AMIs • Intel 82599 VF driver reverted to version 2.0.210.0 (Server 2016) or version 2.1.138.0 (Server 2019) due to customer reported issues. Engagement with Intel concerning these issues ongoing. • Windows security updates current to September 10th, 2019 • Windows Defender platform update blocked via registry due to SFC failures introduced by latest client. Will be reenabled when patch available. See https://support.microsoft.com/en-us/help/4513240/sfc-incorrectly-flags-windows-defender-ps-files-as-corrupted. Platform update block: HKLM:\SOFTWARE\Microsoft\Windows Defender\Miscellaneous Configuration\PreventPlatformUpdate type=DWORD, value=1 Previous versions of AMIs have been marked private. **New Windows AMIs** New STIG-compliant AMIs available: • Windows_Server-2012-R2-English-STIG-Full • Windows_Server-2012-R2-English-STIG-Core

Release	Changes
	• Windows_Server-2016-English-STIG-Full • Windows_Server-2016-English-STIG-Core • Windows_Server-2019-English-STIG-Full • Windows_Server-2019-English-STIG-Core **Windows Server 2008 R2 SP1** Includes the following updates, which are required for Microsoft Extended Security (ESU) updates. • KB4490628 • KB4474419 • KB4516655 **Windows Server 2008 SP2** Includes the following updates, which are required for Microsoft Extended Security (ESU) updates. • KB4493730 • KB4474419 • KB4517134

Release	Changes
	Note NLA is now enabled on all 2012 RTM, 2012 R2, and 2016 AMIs to increase default RDP security posture. NLA remains enabled on 2019 AMIs.
2019.08.16	**All AMIs** • Microsoft security updates current to August 13th, 2019. Includes KBs addressing CVE-2019-1181, CVE-2019-1182, CVE-2019-1222, and CVE-2019-1226. • EC2Config version 4.9.3519 • SSM Agent version 2.3.634.0 • AWS Tools for Windows PowerShell version 3.15.802 • Windows Defender platform update blocked via registry due to SFC failures introduced by update. Update will be re-enabled when new patch is available. **Note** Starting in September, NLA will be enabled on all 2012 RTM, 2012 R2, and 2016 AMIs to increase default RDP security posture.

Release	Changes
2019.07.19	**New Windows AMIs** • Windows_Server-2016-English-Full-ECS_Optimized-2019.07.19 • Windows_Server-2019-English-Full-ECS_Optimized-2019.07.19
2019.07.12	**All AMIs** • Microsoft security updates current to July 9th, 2019

Release	Changes

2019.06.12	**All AMIs**
	• Microsoft security updates current to June 11th, 2019
	• AWS SDK version 3.15.756
	• AWS PV driver version 8.2.7
	• AWS NVMe driver version 1.3.1
	• The following "P3" AMIs will be renamed as "Tesla" AMIs. These AMIs will support all GPU-backed AWS instances using the Tesla driver. P3 AMIs will no longer be updated after this release and will be removed as part of our regular cycle.
	  • Windows_Server-2012-R2_RTM-English-P3-2019.06.12 replaced with Windows_Server-2012-R2_RTM-English-Tesla-2019.06.12
	  • Windows_Server-2016-English-P3-2016.06.12 replaced with Windows_Server-2016-English-Tesla-2019.06.12
	
	New Windows AMIs
	• Windows_Server-2019-English-Tesla-2019.06.12
	
	Previous versions of AMIs have been marked private.

2019.05.21	**Windows Server, version 1903** • AMIs are now available

Release	Changes
2019.05.15	**All AMIs** • Microsoft security updates current to May 14th, 2019 • EC2Config version 4.9.3429 • SSM Agent version 2.3.542.0 • AWS SDK version 3.15.735
2019.04.26	**All AMIs** • Fixed AMIs for Windows Server 2019 with SQL to address edge cases where the first launch of an instance may result in Instance Impairment and Windows displays the message "Please wait for the User Profile Service".
2019.04.21	**All AMIs** • AWS PV Driver rollback to version 8.2.6 from version 8.3.0

Release	Changes
2019.04.10	**All AMIs** • Microsoft security updates current to April 9, 2019 • AWS SDK version 3.15.715 • AWS PV Driver version 8.3.0 • EC2Launch version 1.3.2001360 **New Windows AMIs** • Windows_Server-2016-English-Full-SQL_2012_SP4_Standard-2019.04.10 • Windows_Server-2016-English-Full-SQL_2014_SP3_Standard-2019.04.10 • Windows_Server-2016-English-Full-SQL_2014_SP3_Enterprise-2019.04.10

Release	Changes
2019.03.13	**All AMIs** • Microsoft security updates current to March 12, 2019 • AWS SDK version 3.15.693 • EC2Launch version 1.3.2001220 • NVIDIA Tesla driver version 412.29 for Deep Learning and P3 AMIs (https://nvidia.custhelp.com/app/answers/detail/a_id/4772) Previous versions of AMIs have been marked private
2019.02.13	**All AMIs** • Microsoft security updates current to February 12, 2019 • SSM Agent version 2.3.444.0 • AWS SDK version 3.15.666 • EC2Launch version 1.3.2001040 • EC2Config version 4.9.3289 • AWS PV driver 8.2.6 • EBS NVMe tool SQL 2014 with Service Pack 2 and SQL 2016 with Service Pack 1 will no longer be updated after this release.

Release	Changes
2019.02.09	**All AMIs** • Windows AMIs have been updated. New AMIs can be found with the following date versions: November "2018.11.29" December "2018.12.13" January "2019.02.09" Previous versions of AMIs have been marked private
2019.01.10	**All AMIs** • Microsoft security updates current to January 10, 2019 • SSM Agent version 2.3.344.0 • AWS SDK version 3.15.647 • EC2Launch version 1.3.2000930 • EC2Config version 4.9.3160 **All AMIs with SQL Server** • Latest cumulative updates

Monthly AMI updates for 2018

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

Release	Changes
2018.12.12	**All AMIs** • Microsoft security updates current to December 12, 2018 • SSM Agent version 2.3.274.0 • AWS SDK version 3.15.629 • EC2Launch version 1.3.2000760 **New Windows AMIs** • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2014_SP3_Standard-2018.12.12 • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2014_SP3_Express-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_SP3_Enterprise-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_SP3_Standard-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_SP3_Express-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_SP3_Web-2018.12.12

Release	Changes
	• Windows_Server-2012-RTM-Japanese-64Bit-SQL_2014_SP3_Express-2018.12.12 • Windows_Server-2012-RTM-Japanese-64Bit-SQL_2014_SP3_Standard-2018.12.12 • Windows_Server-2012-RTM-Japanese-64Bit-SQL_2014_SP3_Web-2018.12.12 • Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP3_Standard-2018.12.12 • Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP3_Express-2018.12.12 • Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP3_Web-2018.12.12 • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Web-2018.12.12 • Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Express-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Enterprise-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Express-2018.12.12 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP2_Web-2018.12.12

Release	Changes
	• Windows_Server-2012-R2_RTM-Japanese-64Bit-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2016-Korean-Full-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Enterprise-2018.12.12 • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Web-2018.12.12 • Windows_Server-2016-English-Full-SQL_2016_SP2_Web-2018.12.12 • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2016-English-Full-SQL_2016_SP2_Express-2018.12.12 • Windows_Server-2016-English-Full-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2016-English-Core-SQL_2016_SP2_Enterprise-2018.12.12 • Windows_Server-2016-English-Core-SQL_2016_SP2_Web-2018.12.12 • Windows_Server-2016-English-Core-SQL_2016_SP2_Express-2018.12.12 • Windows_Server-2016-English-Core-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2016-Japanese-Full-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2016-Korean-Full-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2019-Spanish-Full-Base-2018.12.12 •

Release	Changes
	Windows_Server-2019-Japanese-Full-Base-2018.12.12 • Windows_Server-2019-Portuguese_Portugal-Full-Base-2018.12.12 • Windows_Server-2019-Chinese_Traditional-Full-Base-2018.12.12 • Windows_Server-2019-Italian-Full-Base-2018.12.12 • Windows_Server-2019-Swedish-Full-Base-2018.12.12 • Windows_Server-2019-English-Core-Base-2018.12.12 • Windows_Server-2019-Hungarian-Full-Base-2018.12.12 • Windows_Server-2019-Polish-Full-Base-2018.12.12 • Windows_Server-2019-Turkish-Full-Base-2018.12.12 • Windows_Server-2019-Korean-Full-Base-2018.12.12 • Windows_Server-2019-Dutch-Full-Base-2018.12.12 • Windows_Server-2019-German-Full-Base-2018.12.12 • Windows_Server-2019-Russian-Full-Base-2018.12.12 • Windows_Server-2019-Czech-Full-Base-2018.12.12 • Windows_Server-2019-English-Full-Base-2018.12.12 • Windows_Server-2019-French-Full-Base-2018.12.12 • Windows_Server-2019-Portuguese_Brazil-Full-Base-2018.12.12 • Windows_Server-2019-Chinese_Simplified-Full-Base-2018.12.12 •

Release	Changes
	Windows_Server-2019-English-Full-HyperV-2018.12.12 • Windows_Server-2019-English-Full-ContainersLatest-2018.12.12 • Windows_Server-2019-English-Core-ContainersLatest-2018.12.12 • Windows_Server-2019-English-Full-SQL_2017_Enterprise-2018.12.12 • Windows_Server-2019-English-Full-SQL_2017_Standard-2018.12.12 • Windows_Server-2019-English-Full-SQL_2017_Web-2018.12.12 • Windows_Server-2019-English-Full-SQL_2017_Express-2018.12.12 • Windows_Server-2019-English-Full-SQL_2016_SP2_Enterprise-2018.12.12 • Windows_Server-2019-English-Full-SQL_2016_SP2_Standard-2018.12.12 • Windows_Server-2019-English-Full-SQL_2016_SP2_Web-2018.12.12 • Windows_Server-2019-English-Full-SQL_2016_SP2_Express-2018.12.12 **Updated Linux AMI** • amzn2-ami-hvm-2.0.20180622.1-x86_64-gp2-dotnetcore-2018.12.12

Release	Changes
2018.11.28	**All AMIs** • SSM Agent version 2.3.235.0 • Changes in all power schemes to set the display to never turn off
2018.11.20	**Windows_Server-2016-English-Deep-Learning** **Windows_Server-2016-English-Deep-Learning** • TensorFlow version 1.12 • MXNet version 1.3 • NVIDIA version 392.05

Release	Changes
2018.11.19	**All AMIs** • Microsoft security updates current to November 19, 2018 • AWS SDK version 3.15.602.0 • SSM Agent version 2.3.193.0 • EC2Config version 4.9.3067 • Intel Chipset INF configurations to support new instance types **Windows Server, version 1809** • AMIs are now available.

Release	Changes
2018.10.14	**All AMIs** • Microsoft security updates current to October 9, 2018 • AWS Tools for Windows PowerShell version 3.3.365.0 • CloudFormation version 1.4.31 • AWS PV Driver version 8.2.4 • AWS PCI Serial Driver version 1.0.0.0 (support for Windows 2008R2 and 2012 on Bare Metal instances • ENA Driver version 1.5.0 **Windows Server 2016 Datacenter and Standard Editions for Nano Server** Microsoft ended mainstream support for Windows Server 2016 Datacenter and Standard Editions for Nano Server installation options as of April 10, 2018.

Release	Changes
2018.09.15	**All AMIs** • Microsoft security updates current to September 12, 2018 • AWS Tools for Windows PowerShell version 3.3.343 • EC2Launch version 1.3.2000430 • AWS NVMe Driver version 1.3 0 • EC2 WinUtil Driver version 2.0.0 **Windows Server 2016 Base Nano** Access to all public versions of Windows_Server-2016-English-Nano-Base will be removed in September 2018. Additional information about Nano Server lifecycle, including details on launching Nano Server as a Container, can be found here: https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel.

Release	Changes
2018.08.15	**All AMIs** • Microsoft security updates current to August 14, 2018 • AWS Tools for Windows PowerShell version 3.3.335 • AMIs now default to use Amazon's NTP service at IP 169.254.169.123 for time synchronization. For more information, see Default network time protocol (NTP) settings for Amazon Windows AMIs . **Windows Server 2016 Base Nano** Access to all public versions of Windows_Server-2016-English-Nano-Base will be removed in September 2018. Additional information about Nano Server lifecycle, including details on launching Nano Server as a Container, can be found here: https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel.
2018.07.11	**All AMIs** • Microsoft security updates current to July 10, 2018 • EC2Config version 4.9.2756 • SSM Agent 2.2.800.0

Release	Changes
2018.06.22	**Windows Server 2008 R2** • Resolves an issue with the 2018.06.13 AMIs when changing an instance from a previous generation to a current generation (for example, M4 to M5).
2018.06.13	**All AMIs** • Microsoft security updates current to June 12, 2018 • EC2Config version 4.9.2688 • SSM Agent 2.2.619.0 • AWS Tools for Windows PowerShell 3.3.283.0 • AWS NVMe driver 1.2.0 • AWS PV driver 8.2.3

Release	Changes
2018.05.09	**All AMIs** • Microsoft security updates current to May 9, 2018 • EC2Config version 4.9.2644 • SSM Agent 2.2.493.0 • AWS Tools for Windows PowerShell 3.3.270.0 **Windows Server, version 1709 and Windows Server, version 1803** • AMIs are now available. For more information, see Windows Server version 1709 and 1803 AMIs for Amazon EC2.

Release	Changes
2018.04.11	**All AMIs** • Microsoft security updates current to April 10, 2018 • EC2Config version 4.9.2586 • SSM Agent 2.2.392.0 • AWS Tools for Windows PowerShell 3.3.256.0 • AWS CloudFormation templates 1.4.30 • Serial INF and Intel Chipset INF configurations to support new instance types **SQL Server 2017** • Cumulative update 5 (CU5) **SQL Server 2016 SP1** • Cumulative update 8 (CU8)

Release	Changes
2018.03.24	**All AMIs** • Microsoft security updates current to March 13, 2018 • EC2Config version 4.9.2565 • SSM Agent 2.2.355.0 • AWS Tools for Windows PowerShell 3.3.245.0 • AWS PV driver 8.2 • AWS ENA driver 1.2.3.0 • Amazon EC2 Hibernate Agent 1.0 (rollback from 2.1.0 in the 2018.03.16 AMI release) • AWS EC2WinUtilDriver 1.0.1 (for troubleshooting) **Windows Server 2016** • EC2Launch 1.3.2000080
2018.03.16	AWS has removed all Windows AMIs dated 2018.03.16 due to an issue with an unquoted path in the configuration for the Amazon EC2 Hibernate Agent.

Release	Changes
2018.03.06	**All AMIs** • AWS PV driver 8.2.1
2018.02.23	**All AMIs** • AWS PV driver 7.4.6 (rollback from 8.2 in the 2018.02.13 AMI release)

Release	Changes
2018.02.13	**All AMIs** • Microsoft security updates current to February 13, 2018 • EC2Config version 4.9.2400 • SSM Agent 2.2.160.0 • AWS Tools for Windows PowerShell 3.3.225.1 • AWS PV driver 8.2 • AWS ENA driver 1.2.3.0 • AWS NVMe driver 1.0.0.146 • Amazon EC2 HibernateAgent 1.0.0 **Windows Server 2016** • EC2Launch 1.3.740
2018.01.12	**All AMIs** • Microsoft security updates current to January 9, 2018

Release	Changes
2018.01.05	**All AMIs** • Microsoft security updates current to January 2018 • Registry settings to enable mitigations for the Spectre and Meltdown exploits • AWS Tools for Windows PowerShell 3.3.215 • EC2Config version 4.9.2262

Monthly AMI updates for 2017

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

Release	Changes
2017.12.13	**All AMIs** • Microsoft security updates current to December 12, 2017 • EC2Config version 4.9.2218 • AWS CloudFormation templates 1.4.27 • AWS NVMe driver 1.02 • SSM Agent 2.2.93.0 • AWS Tools for Windows PowerShell 3.3.201

Release	Changes
2017.11.29	**All AMIs** • Removed components for Volume Shadow Copy Service (VSS) included in 2017.11.18 and 2017.11.19 due to a compatibility issue with Windows Backup.
2017.11.19	**All AMIs** • EC2 Hibernate Agent 1.0 (supports hibernation for Spot Instances)

Release	Changes
2017.11.18	**All AMIs** • Microsoft security updates current to November 14, 2017 • EC2Config version 4.9.2218 • SSM Agent 2.2.64.0 • AWS Tools for Windows PowerShell 3.3.182 • Elastic Network Adapter (ENA) driver 1.08 (rollback from 1.2.2 in the 2017.10.13 AMI release) • Query for the latest Windows AMI using Systems Manager Parameter Store **Windows Server 2016** • EC2Launch 1.3.640

Release	Changes
2017.10.13	**All AMIs** • Microsoft security updates current to October 11, 2017 • EC2Config version 4.9.2188 • SSM Agent 2.2.30.0 • AWS CloudFormation templates 1.4.24 • Elastic Network Adapter (ENA) driver 1.2.2. (Windows Server 2008 R2 through Windows Server 2016)

Release	Changes
2017.10.04	**Microsoft SQL Server** Windows Server 2016 with Microsoft SQL Server 2017 AMIs are now public in all regions. • Windows_Server-2016-English-Full-SQL_2017_Enterprise-2017.10.04 • Windows_Server-2016-English-Full-SQL_2017_Standard-2017.10.04 • Windows_Server-2016-English-Full-SQL_2017_Web-2017.10.04 • Windows_Server-2016-English-Full-SQL_2017_Express-2017.10.04 Microsoft SQL Server 2017 supports the following features: • Machine Learning Services with Python (ML and AI) and R language support • Automatic database tuning • Clusterless Availability Groups • Runs on Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), and Ubuntu. For more information, see the following Microsoft article: Installation guidance for SQL Server on Linux. Not supported on Amazon Linux. • Windows-Linux cross-OS migrations • Resumable online index rebuild • Improved adaptive query processing • Graph data support

Release	Changes
2017.09.13	**All AMIs** • Microsoft security updates current to September 13, 2017 • EC2Config version 4.9.2106 • SSM Agent 2.0.952.0 • AWS Tools for Windows PowerShell 3.3.143 • AWS CloudFormation templates 1.4.21
2017.08.09	**All AMIs** • Microsoft security updates current to August 9, 2017 • EC2Config version 4.9.2016 • SSM Agent 2.0.879.0 **Windows Server 2012 R2** • Due to an internal error, these AMIs were released with an older version of AWS Tools for Windows PowerShell, 3.3.58.0.

Release	Changes
2017.07.13	**All AMIs** • Microsoft security updates current to July 13, 2017 • EC2Config version 4.9.1981 • SSM Agent 2.0.847.0 **Windows Server 2016** • Intel SRIOV Driver 2.0.210.0

Release	Changes
2017.06.14	**All AMIs** • Microsoft security updates current to June 14, 2017 • Updates for .NET Framework 4.7 installed from Windows Update • Microsoft updates to address the "privilege not held" error using the PowerShell Stop-Computer cmdlet. For more information, see Privilege not held error on the Microsoft site. • EC2Config version 4.9.1900 • SSM Agent 2.0.805.0 • AWS Tools for Windows PowerShell 3.3.99.0 • Internet Explorer 11 for the desktop is the default, instead of the immersive Internet Explorer **Windows Server 2016** • EC2Launch 1.3.610
2017.05.30	The Windows_Server-2008-SP2-English-32Bit-Base-2017.05.10 AMI was updated to the Windows_Server-2008-SP2-English-32Bit-Base-2017.05.30 AMI to resolve an issue with password generation.
2017.05.22	The Windows_Server-2016-English-Full-Base-2017.05.10 AMI was updated to the Windows_Server-2016-English-Full-Base-2017.05.22 AMI after some log cleaning.

Release	Changes
2017.05.10	**All AMIs** • Microsoft security updates current to May 9, 2017 • AWS PV Driver v7.4.6 • AWS Tools for Windows PowerShell 3.3.83.0 **Windows Server 2016** • SSM Agent 2.0.767

Release	Changes
2017.04.12	**All AMIs** • Microsoft security updates current to April 11, 2017 • AWS Tools for Windows PowerShell 3.3.71.0 • AWS CloudFormation templates 1.4.18 **Windows Server 2003 to Windows Server 2012** • EC2Config version 4.9.1775 • SSM Agent 2.0.761.0 **Windows Server 2016** • SSM Agent 2.0.730.0

Release	Changes
2017.03.15	**All AMIs** • Microsoft security updates current to March 14, 2017 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation templates **Windows Server 2003 to Windows Server 2012** • EC2Config version 4.7.1631 • SSM Agent 2.0.682.0 **Windows Server 2016** • SSM Agent 2.0.706.0 • EC2Launch v1.3.540
2017.02.21	Microsoft recently announced that they will not release monthly patches or security updates for the month of February. All February patches and security updates will be included in the March update. Amazon Web Services did not release updated Windows Server AMIs in February.

Release	Changes
2017.01.11	**All AMIs** • Microsoft security updates current to January 10, 2017 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation templates **Windows Server 2003 to Windows Server 2012** • EC2Config version 4.2.1442 • SSM Agent 2.0.599.0

Monthly AMI updates for 2016

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

Release	Changes
2016.12.14	**All AMIs** • Microsoft security updates current to December 13, 2016 • Current AWS Tools for Windows PowerShell **Windows Server 2003 to Windows Server 2012** •

Release	Changes
	Released EC2Config version 4.1.1396 • Elastic Network Adapter (ENA) driver 1.0.9.0 (Windows Server 2008 R2 only) **Windows Server 2016** New AMIs available in all regions: • Windows_Server-2016-English-Core-Base **Microsoft SQL Server** All Microsoft SQL Server AMIs with the latest service pack are now public in all regions. These new AMIs replace old SQL Service Pack AMIs going forward. • Windows_Server-2008-R2_SP1-English-64Bit-SQL_2012_SP3_*edition*-2016.12.14 • Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP3_*edition*-2016.12.14 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_SP2_*edition*-2016.12.14 • Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP2_*edition*-2016.12.14 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP1_*edition*-2016.12.14 • Windows_Server-2016-English-Full-SQL_2016_SP1_*edition*-2016.12.14

Release	Changes
	SQL Server 2016 SP1 is a major release. The following features, which were previously available in Enterprise edition only, are now enabled in Standard, Web, and Express editions with SQL Server 2016 SP1: • Row-level security • Dynamic Data Masking • Change Data Capture • Database snapshot • Column store • Partitioning • Compression • In Memory OLTP • Always Encrypted

Release	Changes
2016.11.23	**Windows Server 2003 to Windows Server 2012** • Released EC2Config version 4.1.1378 • The AMIs released this month, and going forward, use the EC2Config service to process boot-time configurations and SSM Agent to process AWS Systems Manager Run Command and Config requests. EC2Config no longer processes requests for Systems Manager Run Command and State Manager. The latest EC2Config installer installs SSM Agent side-by-side with the EC2Config service. For more information, see Configure a Windows instance using the EC2Config service (legacy).
2016.11.09	**All AMIs** • Microsoft security updates current to November 8 2016 • Released AWS PV driver, version 7.4.3.0 for Windows 2008 R2 and later • Current AWS Tools for Windows PowerShell

Release	Changes
2016.10.18	**All AMIs** • Microsoft security updates current to October 12, 2016 • Current AWS Tools for Windows PowerShell **Windows Server 2016** • Released AMIs for Windows Server 2016. These AMIs include significant changes. For example, they don't include the EC2Config service. For more information, see Changes in Windows Server 2016 and later AMIs.
2016.9.14	**All AMIs** • Microsoft security updates current to September 13, 2016 • Current AWS Tools for Windows PowerShell • Renamed AMI Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_**R3_SP2**_Standard to Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_**R2_SP3**_Standard
2016.8.26	All Windows Server 2008 R2 AMIs dated 2016.08.11 were updated to fix a known issue. New AMIs are dated 2016.08.25.

Release	Changes
2016.8.11	**All AMIs** • Ec2Config v3.19.1153 • Microsoft security updates current to August 10, 2016 • Enabled the registry key User32 exception handler hardening feature in Internet Explorer for MS15-124 **Windows Server 2008 R2, Windows Server 2012 RTM, and Windows Server 2012 R2** • Elastic Network Adapter (ENA) Driver 1.0.8.0 • ENA AMI property set to enabled • AWS PV Driver for Windows Server 2008 R2 was re-released this month because of a known issue. Windows Server 2008 R2 AMI's were removed in July because of this issue.
2016.8.2	All Windows Server 2008 R2 AMIs for July were removed and rolled back to AMIs dated 2016.06.15, because of an issue discovered in the AWS PV driver. The AWS PV driver issue has been fixed. The August AMI release will include Windows Server 2008 R2 AMIs with the fixed AWS PV driver and July/August Windows updates.

Release	Changes
2016.7.26	**All AMIs** • Ec2Config v3.18.1118 • 2016.07.13 AMIs were missing security patches. AMIs were re-patched. Additional processes were put in place to verify successful patch installat ions going forward.
2016.7.13	**All AMIs** • Microsoft security updates current to July 2016 • Current AWS Tools for Windows PowerShell • Updated AWS PV Driver 7.4.2.0 • AWS PV Driver for Windows Server 2008 R2

Release	Changes
2016.6.16	**All AMIs** • Microsoft security updates current to June 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.17.1032 **Microsoft SQL Server** • Released 10 AMIs that include 64-bit versions of Microsoft SQL Server 2016. If using the Amazon EC2 console, navigate to **Images**, **AMIs**, **Public Images**, and type **`Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_Standard`** in the search bar. For more information, see What's New in SQL Server 2016 on MSDN.

Release	Changes
2016.5.11	**All AMIs** • Microsoft security updates current to May 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.16.930 • MS15-011 Active Directory patch installed **Windows Server 2012 R2** • Intel SRIOV Driver 1.0.16.1
2016.4.13	**All AMIs** • Microsoft security updates current to April 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.15.880

Release	Changes
2016.3.9	**All AMIs** • Microsoft security updates current to March 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.14.786
2016.2.10	**All AMIs** • Microsoft security updates current to February 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.13.727
2016.1.25	**All AMIs** • Microsoft security updates current to January 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.12.649

Release	Changes
2016.1.5	**All AMIs** • Current AWS Tools for Windows PowerShell

Monthly AMI updates for 2015

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

Release	Changes
2015.12.15	**All AMIs** • Microsoft security updates current to December 2015 • Current AWS Tools for Windows PowerShell
2015.11.11	**All AMIs** • Microsoft security updates current to November 2015 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.11.521 • CFN Agent updated to latest version
2015.10.26	Corrected boot volume sizes of base AMIs to be 30GB instead of 35GB

Release	Changes
2015.10.14	**All AMIs** • Microsoft security updates current to October 2015 • EC2Config service version 3.10.442 • Current AWS Tools for Windows PowerShell • Updated SQL Service Packs to latest versions for all SQL variants • Removed old entries in Event Logs • AMI Names have been changed to reflect the latest service pack. For example, the latest AMI with Server 2012 and SQL 2014 Standard is named "Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP1_Standard-2015.10.26", not "Windows_Server-2012-RTM-English-64Bit-SQL_2014_RTM_Standard-2015.10.26".
2015.9.9	**All AMIs** • Microsoft security updates current to September 2015 • EC2Config service version 3.9.359 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts

Release	Changes
2015.8.18	**All AMIs** • Microsoft security updates current to August 2015 • EC2Config service version 3.8.294 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 and Windows Server 2012 R2** • AWS PV Driver 7.3.2
2015.7.21	**All AMIs** • Microsoft security updates current to July 2015 • EC2Config service version 3.7.308 • Current AWS Tools for Windows PowerShell • Modified AMI descriptions of SQL images for consistency

Release	Changes
2015.6.10	**All AMIs** • Microsoft security updates current to June 2015 • EC2Config service version 3.6.269 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.3.1
2015.5.13	**All AMIs** • Microsoft security updates current to May 2015 • EC2Config service version 3.5.228 • Current AWS Tools for Windows PowerShell

Release	Changes
2015.04.15	**All AMIs** • Microsoft security updates current to April 2015 • EC2Config service version 3.3.174 • Current AWS Tools for Windows PowerShell
2015.03.11	**All AMIs** • Microsoft security updates current to March 2015 • EC2Config service version 3.2.97 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.3.0

Release	Changes
2015.02.11	**All AMIs** • Microsoft security updates current to February 2015 • EC2Config service version 3.0.54 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2015.01.14	**All AMIs** • Microsoft security updates current to January 2015 • EC2Config service version 2.3.313 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts

Monthly AMI updates for 2014

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

Release	Changes
2014.12.10	**All AMIs** • Microsoft security updates current to December 2014

Release	Changes
	• EC2Config service version 2.2.12 • Current AWS Tools for Windows PowerShell
2014.11.19	**All AMIs** • Microsoft security updates current to November 2014 • EC2Config service version 2.2.11 • Current AWS Tools for Windows PowerShell
2014.10.15	**All AMIs** • Microsoft security updates current to October 2014 • EC2Config service version 2.2.10 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.2.4.1 (resolves the issues with Plug and Play Cleanup, which is now enabled by default)

Release	Changes
2014.09.10	**All AMIs** • Microsoft security updates current to September 2014 • EC2Config service version 2.2.8 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • Disable Plug and Play Cleanup (see Important information) • AWS PV Driver 7.2.2.1 (resolves issues with the uninstaller)
2014.08.13	**All AMIs** • Microsoft security updates current to August 2014 • EC2Config service version 2.2.7 • Current AWS Tools for Windows PowerShell **Only AMIs with Windows Server 2012 R2** • AWS PV Driver 7.2.2.1 (improves disk performance, resolves issues with reconnecting multiple network interfaces and lost network settings)

Release	Changes
2014.07.10	**All AMIs** • Microsoft security updates current to July 2014 • EC2Config service version 2.2.5 • Current AWS Tools for Windows PowerShell
2014.06.12	**All AMIs** • Microsoft security updates current to June 2014 • EC2Config service version 2.2.4 • Removed NVIDIA drivers (except for Windows Server 2012 R2 AMIs) • Current AWS Tools for Windows PowerShell
2014.05.14	**All AMIs** • Microsoft security updates current to May 2014 • EC2Config service version 2.2.2 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.4.0

Release	Changes
2014.04.09	**All AMIs** • Microsoft security updates current to April 2014 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2014.03.12	**All AMIs** • Microsoft security updates current to March 2014

Release	Changes
2014.02.12	**All AMIs** • Microsoft security updates current to February 2014 • EC2Config service version 2.2.1 • Current AWS Tools for Windows PowerShell • KB2634328 • Remove the BCDEdit `useplatformclock` value **Only AMIs with Microsoft SQL Server** • Microsoft SQL Server 2012 SP1 cumulative update package 8 • Microsoft SQL Server 2008 R2 cumulative update package 10

Monthly AMI updates for 2013

Release	Changes
2013.11.13	**All AMIs** • Microsoft security updates current to November 2013 • EC2Config service version 2.1.19 • Current AWS Tools for Windows PowerShell •

Release	Changes
	Configure NTP to synchronize the time once a day (the default is every seven days) **Only AMIs with Windows Server 2012** • Clean up the WinSXS folder using the following command: `dism /online /cleanup-image /StartComponentCleanup`
2013.09.11	**All AMIs** • Microsoft security updates current to September 2013 • EC2Config service version 2.1.18 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.15

Release	Changes
2013.07.10	**All AMIs** • Microsoft security updates current to July 2013 • EC2Config service version 2.1.16 • Expanded the root volume to 50 GB • Set the page file to 512 MB, expanding to 8 GB as needed • Current AWS Tools for Windows PowerShell
2013.06.12	**All AMIs** • Microsoft security updates current to June 2013 • Current AWS Tools for Windows PowerShell **Only AMIs with Microsoft SQL Server** • Microsoft SQL Server 2012 SP1 with cumulative update package 4

Release	Changes
2013.05.15	**All AMIs** • Microsoft security updates current to May 2013 • EC2Config service version 2.1.15 • All instance store volumes attached by default • Remote PowerShell enabled by default • Current AWS Tools for Windows PowerShell
2013.04.14	**All AMIs** • Microsoft security updates current to April 2013 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.14

Release	Changes
2013.03.14	**All AMIs** • Microsoft security updates current to March 2013 • EC2Config service version 2.1.14 • Citrix Agent with CPU heartbeat fix • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.11

Release	Changes
2013.02.22	**All AMIs** • Microsoft security updates current to February 2013 • KB2800213 • Windows PowerShell 3.0 upgrade • EC2Config service version 2.1.13 • Citrix Agent with time fix • Citrix PV drivers dated 2011.07.19 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.8 **Only AMIs with Microsoft SQL Server** • Microsoft SQL Server 2012 cumulative update package 5

Monthly AMI updates for 2012

Release	Changes
2012.12.12	**All AMIs** • Microsoft security updates current to December 2012 •

Release	Changes
	Set the ActiveTimeBias registry value to 0 • Disable IPv6 for the network adapter • EC2Config service version 2.1.9 • Add AWS Tools for Windows PowerShell and set the policy to allow import-module
2012.11.15	**All AMIs** • Microsoft security updates current to November 2012 • EC2Config service version 2.1.7
2012.10.10	**All AMIs** • Microsoft security updates current to October 2012
2012.08.15	**All AMIs** • Microsoft security updates current to August 2012 • EC2Config service version 2.1.2 • KB2545227

Release	Changes
2012.07.11	**All AMIs** • Microsoft security updates current to July 2012
2012.06.12	**All AMIs** • Microsoft security updates current to June 2012 • Set page file to 4 GB • Remove installed language packs • Set performance option to "Adjust for best performance" • Set the screen saver to no longer display the logon screen on resume • Remove previous RedHat driver versions using pnputil • Remove duplicate bootloaders and set bootstatuspolicy to ignoreall failures using bcdedit
2012.05.10	**All AMIs** • Microsoft security updates current to May 2012 • EC2Config service version 2.1.0

Release	Changes
2012.04.11	**All AMIs** • Microsoft security updates current to April 2012 • KB2582281 • Current version of EC2Config • System time in UTC instead of GMT
2012.03.13	**All AMIs** • Microsoft security updates current to March 2012
2012.02.24	**All AMIs** • Microsoft security updates current to February 2012 • Standardize AMI names and descriptions
2012.01.12	**All AMIs** • Microsoft security updates current to January 2012 • RedHat PV driver version 1.3.10

Monthly AMI updates for 2011 and earlier

Release	Changes
2011.09.11	**All AMIs** • Microsoft security updates current to September 2011
1.04	**All AMIs** • Current Microsoft security updates • Update network driver • Fix issue with instances in a VPC losing connectivity when changing the time zone of the instance
1.02	**All AMIs** • Current Microsoft security updates • Update network driver • Add support for licensing activation for instances in a VPC
1.01	**All AMIs** • Current Microsoft security updates •

Release	Changes
	Fix issue with password improperly generated while waiting for network availability
1.0	**All AMIs** • Initial release

Find a Windows 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`) or 64-bit (`x86_64`)
- The provider (for example, Amazon Web Services)
- Additional software (for example, SQL Server)

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.

If you want to find a Linux AMI, see Find a Linux AMI in the *Amazon EC2 User Guide for Linux Instances*.

Find a Windows AMI topics

- Find a Windows AMI using the Amazon EC2 console
- Find an AMI using the AWS Tools for Windows PowerShell

- Find an AMI using the AWS CLI
- Find the latest Windows AMI using Systems Manager
- Use a Systems Manager parameter to find an AMI

Find a Windows AMI using the Amazon EC2 console

You can find Windows AMIs using the Amazon EC2 console. You can select from the list of AMIs when you use the launch 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 Windows 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 Windows 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 Windows 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. 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.

Find an AMI using the AWS Tools for Windows PowerShell

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

Amazon EC2

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

Systems Manager Parameter Store

For information and examples, see Query for the Latest Windows AMI Using Systems Manager Parameter Store.

Find an AMI using the AWS CLI

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

Amazon EC2

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

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

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

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

> **Important**
>
> Omitting the `--owners` flag from the describe-images command returns all images for which you have launch permissions, regardless of ownership.

Systems Manager Parameter Store

For information and examples, see Query for the Latest Windows AMI Using Systems Manager Parameter Store.

Find the latest Windows AMI using Systems Manager

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

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-windows-latest`

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

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

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

Use a Systems Manager parameter to find an AMI

When you launch an instance using the EC2 launch 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 Windows AMI using a Systems Manager parameter (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose **Launch instance**.
4. 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 Windows AMI using a Systems Manager parameter (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose **Launch instance**.
4. Choose **Search by Systems Manager parameter** (at top right).
5. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.
6. Choose **Search**. The AMIs that match the AMI ID appear in the list.

7. Select the AMI from the list, and choose **Select**.

For more information about launching an instance from an AMI using the launch 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
- Best practices for shared Windows AMIs

If you're looking for information about other topics

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

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 (Tools for Windows PowerShell)
- Find a shared AMI (AWS CLI)

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 (Tools for Windows PowerShell)

Use the Get-EC2Image command (Tools for Windows PowerShell) to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

Example: List all public AMIs

The following command lists all public AMIs, including any public AMIs that you own.

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

Example: List AMIs with explicit launch permissions

The following command lists the AMIs for which you have explicit launch permissions. This list does not include any AMIs that you own.

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

Example: List AMIs owned by 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.

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

Example: List AMIs owned by an account

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

```
PS C:\> Get-EC2Image -Owner 123456789012
```

Example: Scope AMIs using a filter

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

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

Find a shared AMI (AWS CLI)

Use the describe-images command (AWS CLI) to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

Example: List all public AMIs

The following command lists all public AMIs, including any public AMIs that you own.

```
aws ec2 describe-images --executable-users all
```

Example: List AMIs with explicit launch permissions

The following command lists the AMIs for which you have explicit launch permissions. This list does not include any AMIs that you own.

```
aws ec2 describe-images --executable-users self
```

Example: List AMIs owned by 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"
```

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
- **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.

- **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**.

Tools for Windows PowerShell

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts), or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

1. Use the Edit-EC2ImageAttribute command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

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

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

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

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

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`.

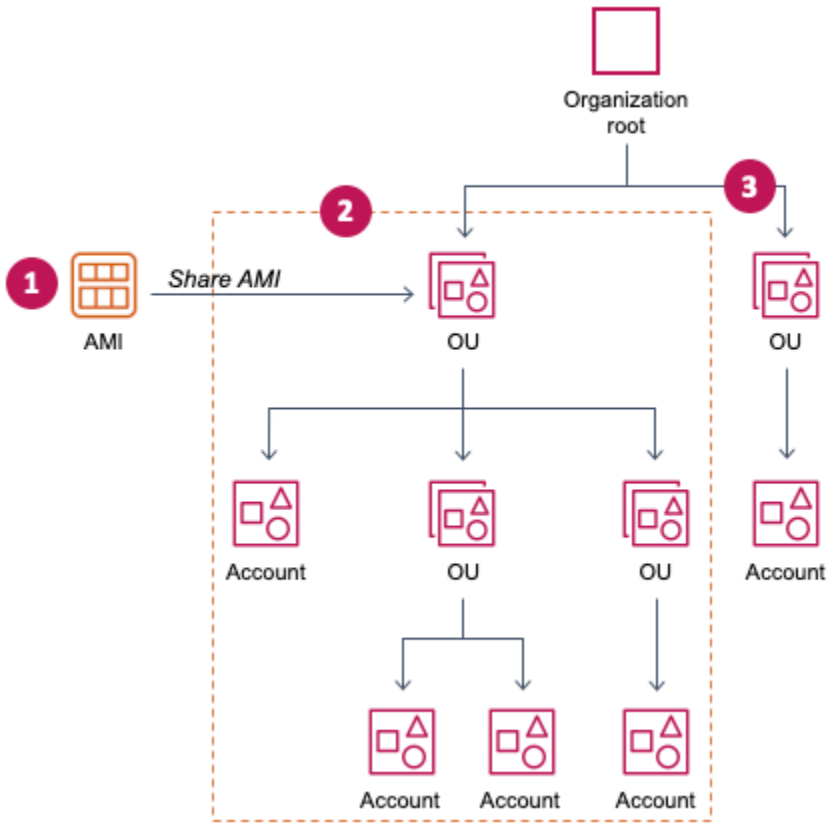
```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

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'' 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 (Tools for Windows PowerShell)

Use the Edit-EC2ImageAttribute command (Tools for Windows PowerShell) to share an AMI as shown in the following examples.

To share an AMI with an organization or an OU

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

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -
Attribute launchPermission -OperationType add -OrganizationArn
 "arn:aws:organizations::123456789012:organization/o-123example"
```

> **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.

To stop sharing an AMI with an organization or OU

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

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -
Attribute launchPermission -OperationType remove -OrganizationArn
 "arn:aws:organizations::123456789012:organization/o-123example"
```

To stop sharing an AMI with all organizations, OUs, and AWS accounts

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

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

Share an AMI (AWS CLI)

Use the modify-image-attribute command (AWS CLI) to share an AMI.

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 (Tools for Windows PowerShell)

Use the Edit-EC2ImageAttribute command (Tools for Windows PowerShell) to share an AMI as shown in the following examples.

To grant explicit launch permissions

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

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -Attribute
 launchPermission -OperationType 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.

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -Attribute
 launchPermission -OperationType 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.

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

Share an AMI (AWS CLI)

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

To grant explicit launch permissions

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

Best practices for shared Windows AMIs

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

- No list of security guidelines can be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.
- Develop a repeatable process for building, updating, and republishing AMIs.
- Build AMIs using the most up-to-date operating systems, packages, and software.
- Download and install the latest version of the `EC2Config` service. For more information about installing this service, see Install the latest version of EC2Config.
- Verify that `Ec2SetPassword`, `Ec2WindowsActivate` and `Ec2HandleUserData` are enabled.
- Verify that no guest accounts or Remote Desktop user accounts are present.
- Disable or remove unnecessary services and programs to reduce the attack surface of your AMI.
- Remove instance credentials, such as your key pair, from the AMI (if you saved them on the AMI). Store the credentials in a safe location.

- Ensure that the administrator password and passwords on any other accounts are set to an appropriate value for sharing. These passwords are available for anyone who launches your shared AMI.
- Test your AMI before you share it.

Paid AMIs

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

You can purchase AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users.

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS, including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the AWS Marketplace 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 Tools for Windows PowerShell

You can find a paid AMI using the following Get-EC2Image command.

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

The output for a paid AMI includes the product code.

```
ProductCodeId           ProductCodeType
-------------           ---------------
product_code            marketplace
```

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

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

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:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/product-codes
```

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

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 a custom Windows AMI
- Modify an AMI
- Copy an AMI
- Store and restore an AMI using S3
- Deprecate an AMI
- Disable an AMI
- Archive AMI snapshots
- Deregister your AMI
- Automate the EBS-backed AMI lifecycle

Create a custom Windows AMI

You can launch an instance from an existing Windows AMI, customize the instance, and then save this updated configuration as a custom AMI. Instances launched from this new custom AMI include the customizations that you made when you created the AMI.

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

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

Topics

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

How the creation of a custom AMI works

First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, ensure data integrity by stopping the instance before you create an AMI and then create the image. We automatically register the AMI for you.

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

Depending on the size of the volumes, it can take several minutes for the AMI-creation process to complete (sometimes up to 24 hours). You may find it more efficient to create snapshots of your volumes prior to creating your AMI. This way, only small, incremental snapshots need to be created when the AMI is created, and the process completes more quickly (the total time for snapshot creation remains the same).

After the process completes, you have a new AMI and snapshot created from the root volume of the instance. When you launch an instance using the new AMI, we create a new EBS volume for its root volume using the snapshot.

> **Note**
>
> A Windows AMI must be created from an Amazon EC2 instance. Creation of a Windows AMI from an EBS snapshot is currently not supported as it might cause issues with billing, performance, and general operation.

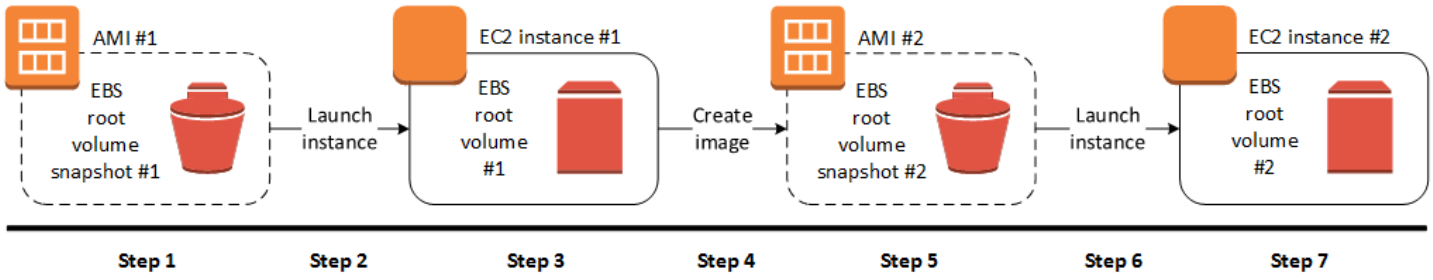
If you add instance store volumes or Amazon Elastic Block Store (Amazon EBS) volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. The instance store volumes specified in the block device mapping for the new instance are new and don't contain any data from the instance store volumes of the instance you used to create the AMI. The data on EBS volumes persists. For more information, see Block device mappings.

> **Note**
>
> When you create a new instance from a custom AMI, you should initialize both its root volume and any additional EBS storage before putting it into production. For more information, see Initialize Amazon EBS volumes.

Create a Windows AMI from a running instance

You can create an AMI using the AWS Management Console or the command line. The following diagram summarizes the process for creating an AMI from a running EC2 instance. Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI. The steps in the following diagram match the steps in the procedure below.

> **Note**
>
> If you already have a running Windows instance, you can go directly to step 5.



To create an AMI from an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Use the **Filter** options to scope the list of AMIs to the Windows AMIs that meet your needs. For example, to view the Windows AMIs provided by AWS, choose **Public images** from the drop-down list. Choose the Search bar, and then from the menu, choose **Owner alias**, then **=**, and then **amazon**. Choose **Source** from the menu and enter one of the following, depending on the version of Windows Server that you need:

   - **amazon/Windows_Server-2022**
   - **amazon/Windows_Server-2019**
   - **amazon/Windows_Server-2016**
   - **amazon/Windows_Server-2012**

   Add any other filters that you need. When you have chosen an AMI, select its check box.
4. Choose **Launch instance from AMI** (new console) or **Launch** (old console). Accept the default values as you step through the wizard. For more information, see Launch an instance using the new launch instance wizard. When the instance is ready, connect to it. For more information, see Connect to your Windows instance.
5. Once you connect to the instance, you can perform any of the following actions to customize it for your needs:

   - Install software and applications
   - Copy data
   - Reduce start time by deleting temporary files and defragmenting your hard drive
   - Attach additional EBS volumes
   - Create a new user account and add it to the Administrators group

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

- [Windows Server 2022 and later] Configure settings using EC2Launch v2. To generate a random password at launch time, configure the `setAdminAccount` task. For more information, see setAdminAccount.
- [Windows Server 2016 and 2019] Configure settings using EC2Launch. To generate a random password at launch time, use the `adminPasswordType` setting. For more information, see Configure EC2Launch.
- [Windows Server 2012 R2 and earlier] Configure settings using EC2Config. To generate a random password at launch time, enable the `Ec2SetPassword` plugin; otherwise, the current administrator password is used. For more information, see EC2Config settings files.

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

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

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

   By default, 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. For the **No reboot** setting, you can select the **Enable** check box to prevent Amazon EC2 from shutting down and rebooting the instance.

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

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

   - To change the size of the root volume, locate the **Root** volume in the **Type** column, and fill in the **Size** field.

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

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

8. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Clear your previous filters, and choose **Owned by me** from the drop-down list. Initially, the status is `pending`. After a few minutes, the status should change to `available`.

   (Optional) Choose **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, we use this snapshot to create its root device volume.

9. Launch an instance from your new AMI. For more information, see Launch an instance using the new launch instance wizard. The new running instance contains all of the customizations you applied in previous steps, and any additional customization you add when launching the instance, such as user data (scripts that run when the instance starts).

Create an AMI from an instance using the command line

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

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

Create a standardized Amazon Machine Image (AMI) using Sysprep

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

We recommend that you use EC2 Image Builder to automate the creation, management, and deployment of customized, secure, and up-to-date "golden" server images that are pre-installed and preconfigured with software and settings.

If you use Sysprep to create a standardized AMI, we recommend that you run Sysprep with EC2Launch v2. If you are still using the EC2Config (Windows Server 2012 R2 and earlier) or EC2Launch (Windows Server 2016 and 2019) agents, see the documentation for using Sysprep with EC2Config and EC2Launch below.

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

To troubleshoot Sysprep, see Troubleshoot Sysprep.

Contents

- Before you begin
- Use Sysprep with EC2Launch v2
- Use Sysprep with EC2Launch
- Use Sysprep with EC2Config

Before you begin

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

- Learn which server roles are supported for Sysprep.

Use Sysprep with EC2Launch v2

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

Sysprep with EC2Launch v2 topics

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

Sysprep phases

Sysprep runs through the following phases:

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

  > **Note**
  >
  > When you run Sysprep with the EC2Launch v2 service, the system prevents drivers from being removed because the `PersistAllDeviceInstalls` setting is set to true by default.

- **Specialize**: Plug and Play scans the computer and installs drivers for any detected devices. The tool generates OS requirements, like the computer name and SID. Optionally, you can run commands in this phase.
- **Out-of-Box Experience (OOBE)**: The system runs an abbreviated version of Windows Setup and asks you to enter information such as system language, time zone, and registered organization. When you run Sysprep with EC2Launch v2, the answer file automates this phase.

Sysprep actions

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

1. When you choose **Shutdown with Sysprep** in the **EC2Launch settings** dialog box, the system runs the `ec2launch sysprep` command.
2. EC2Launch v2 edits the content of the `unattend.xml` file by reading the registry value at `HKEY_USERS\.DEFAULT\Control Panel\International\LocaleName`. This file is located in the following directory: `C:\ProgramData\Amazon\EC2Launch\sysprep`.
3. The system run the `BeforeSysprep.cmd`. This command creates a registry key as follows:

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

   The registry key disables RDP connections until they are re-enabled. Disabling RDP connections is a necessary security measure because, during the first boot session after Sysprep has run, there is a short period of time where RDP allows connections and the Administrator password is blank.
4. The EC2Launch v2 service calls Sysprep by running the following command:

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

Generalize phase

- EC2Launch v2 removes image-specific information and configurations, such as the computer name and the SID. If the instance is a member of a domain, it is removed from the domain. The `unattend.xml` answer file includes the following settings that affect this phase:
  - **PersistAllDeviceInstalls**: This setting prevents Windows Setup from removing and reconfiguring devices, which speeds up the image preparation process because Amazon AMIs require certain drivers to run and re-detection of those drivers would take time.
  - **DoNotCleanUpNonPresentDevices**: This setting retains Plug and Play information for devices that are not currently present.
- Sysprep shuts down the OS as it prepares to create the AMI. The system either launches a new instance or starts the original instance.

Specialize phase

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

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

  CopyProfile replaces the default profile with the existing local administrator profile. All accounts that you log in to after running Sysprep receive a copy of that profile and its contents at first login.

  If you don't have specific user-profile customizations that you want to carry over to the new image, then change this setting to `False`. Sysprep will remove all user profiles (this saves time and disk space).
- **TimeZone**: The time zone is set to Coordinate Universal Time (UTC) by default.
- **Synchronous command with order 1**: The system runs the following command, which enables the administrator account and specifies the password requirement:

  net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /PASSWORDREQ:YES
- **Synchronous command with order 2**: The system scrambles the administrator password. This security measure is designed to prevent the instance from being accessible after Sysprep completes if you did not enable the `ec2setpassword` setting.

  C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator
- **Synchronous command with order 3**: The system runs the following command:

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

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

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

OOBE phase

1. The system specifies the following configurations using the EC2Launch v2 answer file:
   - `<InputLocale>en-US</InputLocale>`
   - `<SystemLocale>en-US</SystemLocale>`
   - `<UILanguage>en-US</UILanguage>`
   - `<UserLocale>en-US</UserLocale>`
   - `<HideEULAPage>true</HideEULAPage>`
   - `<HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>`
   - `<ProtectYourPC>3</ProtectYourPC>`
   - `<BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>`
   - `<TimeZone>UTC</TimeZone>`
   - `<RegisteredOrganization>Amazon.com</RegisteredOrganization>`
   - `<RegisteredOwner>EC2</RegisteredOwner>`

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

2. The system runs EC2Launch v2.

Post Sysprep

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

```
Windows sysprep configuration complete.
```

EC2Launch v2 then performs the following actions:

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

4. Executes all tasks in the `PostReady` stage.

For more information about EC2Launch v2 , see Configure a Windows instance using EC2Launch v2.

Run Sysprep with EC2Launch v2

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

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

When you are asked to confirm that you want to run Sysprep and shut down the instance, click **Yes**. EC2Launch v2 runs Sysprep. Next, you are logged off the instance, and the instance shuts down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from `Running` to `Stopping` to `Stopped`. At this point, it's safe to create an AMI from this instance.

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

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

Use Sysprep with EC2Launch

EC2Launch offers a default answer file and batch files for Sysprep that automate and secure the image-preparation process on your AMI. Modifying these files is optional. These files are located in the following directory by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep`.

> **Important**
>
> Do not use Sysprep to create an instance backup. Sysprep removes system-specific information. If you remove this information there might be unintended consequences for an instance backup.

Sysprep with EC2Launch topics

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

EC2Launch answer and batch files for Sysprep

The EC2Launch answer file and batch files for Sysprep include the following:

`Unattend.xml`

This is the default answer file. If you run `SysprepInstance.ps1` or choose **ShutdownWithSysprep** in the user interface, the system reads the setting from this file.

`BeforeSysprep.cmd`

Customize this batch file to run commands before EC2Launch runs Sysprep.

`SysprepSpecialize.cmd`

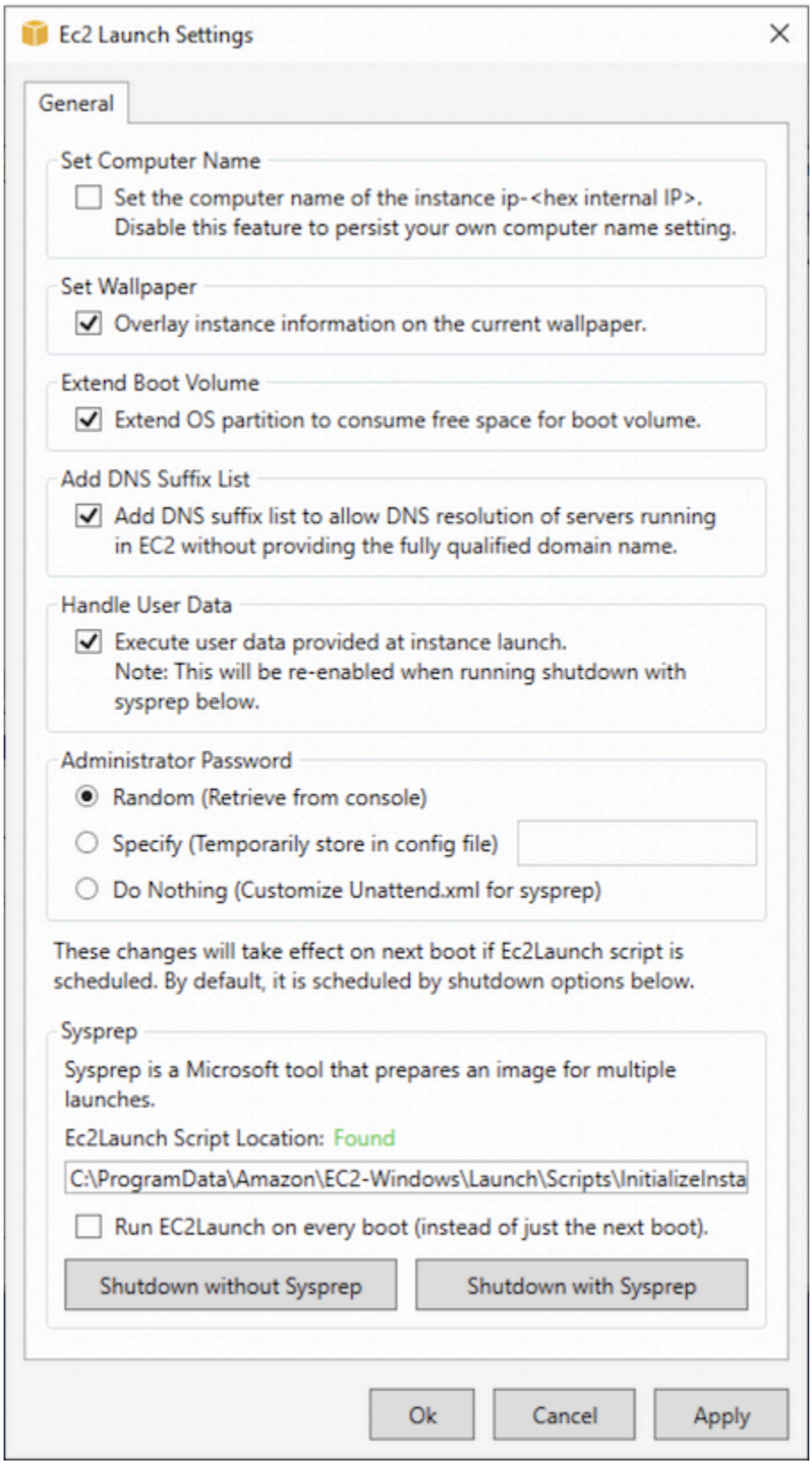
Customize this batch file to run commands during the Sysprep specialize phase.

Run Sysprep with EC2Launch

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

To run Sysprep using the EC2Launch Settings application

1. In the Amazon EC2 console, locate or create a Windows Server 2016 or later AMI.
2. Launch a Windows instance from the AMI.
3. Connect to your Windows instance and customize it.
4. Search for and run the **EC2LaunchSettings** application. It is located in the following directory by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Settings`.



5. Select or clear options as needed. These settings are stored in the `LaunchConfig.json` file.
6. For **Administrator Password**, do one of the following:

- Choose **Random**. EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.
- Choose **Specify** and type a password that meets the system requirements. The password is stored in `LaunchConfig.json` as clear text and is deleted after Sysprep sets the administrator password. If you shut down now, the password is set immediately. EC2Launch encrypts the password using the user's key.
- Choose **DoNothing** and specify a password in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.

7. Choose **Shutdown with Sysprep**.

To manually run Sysprep using EC2Launch

1. In the Amazon EC2 console locate or create a Windows Server 2016 or later Datacenter edition AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize the instance.
4. Specify settings in the `LaunchConfig.json` file. This file is located in the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory by default.

   For `adminPasswordType`, specify one of the following values:

   `Random`

   EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.

   `Specify`

   EC2Launch uses the password you specify in `adminPassword`. If the password does not meet the system requirements, EC2Lauch generates a random password instead. The password is stored in `LaunchConfig.json` as clear text and is deleted after Sysprep sets the administrator password. EC2Launch encrypts the password using the user's key.

`DoNothing`

EC2Launch uses the password you specify in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.

5. (Optional) Specify settings in `unattend.xml` and other configuration files. If plan to attend to the installation, then you don't need to make changes in these files. The files are located in the following directory by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep`.
6. In Windows PowerShell, run `./InitializeInstance.ps1 -Schedule`. The script is located in the following directory, by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts`. This script schedules the instance to initialize during the next boot. You must run this script before you run the `SysprepInstance.ps1` script in the next step.
7. In Windows PowerShell, run `./SysprepInstance.ps1`. The script is located in the following directory by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts`.

You are logged off the instance and the instance shuts down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from `Running` to `Stopping`, and then to `Stopped`. At this point, it is safe to create an AMI from this instance.

Update metadata/KMS routes for Server 2016 and later when launching a custom AMI

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

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

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

- To update the routes on a running instance that is experiencing Windows activation or communication with instance metadata failures, see "Unable to activate Windows".

Use Sysprep with EC2Config

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

Sysprep with EC2Config topics

- Sysprep phases
- Sysprep actions
- Post Sysprep
- Run Sysprep with the EC2Config service

Sysprep phases

Sysprep runs through the following phases:

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

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

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

Sysprep actions

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

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

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

   - **AutoSysprep**: Indicates whether to use Sysprep automatically. You do not need to change this value if you are running Sysprep from the EC2 Service Properties dialog box. The default value is `No`.
   - **SetRDPCertificate**: Sets a self-signed certificate for the Remote Desktop server. This enables you to securely use the Remote Desktop Protocol (RDP) to connect to the instance. Change the value to `Yes` if new instances should use a certificate. This setting is not used with Windows Server 2008 or Windows Server 2012 instances because these operating systems can generate their own certificates. The default value is `No`.
   - **SetPasswordAfterSysprep**: Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value to `No` if new instances should not be set to a random encrypted password. The default value is `Yes`.
   - **PreSysprepRunCmd**: The location of the command to run. The command is located in the following directory, by default: `C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd`
3. The system runs `BeforeSysprep.cmd`. This command creates a registry key as follows:

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

   The registry key disables RDP connections until they are re-enabled. Disabling RDP connections is a necessary security measure because, during the first boot session after Sysprep has run, there is a short period of time where RDP allows connections and the Administrator password is blank.
4. The EC2Config service calls Sysprep by running the following command:

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

Generalize phase

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

Specialize phase

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

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

  CopyProfile replaces the default profile with the existing local administrator profile. All accounts logged into after running Sysprep will receive a copy of that profile and its contents at first login.

  If you don’t have specific user-profile customizations that you want to carry over to the new image then change this setting to False. Sysprep will remove all user profiles; this saves time and disk space.
- **TimeZone**: The time zone is set to Coordinate Universal Time (UTC) by default.
- **Synchronous command with order 1**: The system runs the following command that enables the administrator account and specifies the password requirement.

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

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

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

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

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

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

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

OOBE phase

1. Using the EC2Config service answer file, the system specifies the following configurations:
   - <InputLocale>en-US</InputLocale>
   - <SystemLocale>en-US</SystemLocale>
   - <UILanguage>en-US</UILanguage>
   - <UserLocale>en-US</UserLocale>
   - <HideEULAPage>true</HideEULAPage>
   - <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
   - <NetworkLocation>Other</NetworkLocation>
   - <ProtectYourPC>3</ProtectYourPC>
   - <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>
   - <TimeZone>UTC</TimeZone>
   - <RegisteredOrganization>Amazon.com</RegisteredOrganization>
   - <RegisteredOwner>Amazon</RegisteredOwner>

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

EC2ConfigMonitorState: 0 Windows is being configured.
SysprepState=IMAGE_STATE_UNDEPLOYABLE

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

Post Sysprep

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

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

EC2Config then performs the following actions:

1. Reads the content of the config.xml file and lists all enabled plug-ins.
2. Executes all “Before Windows is ready” plug-ins at the same time.
   - Ec2SetPassword
   - Ec2SetComputerName
   - Ec2InitializeDrives
   - Ec2EventLog
   - Ec2ConfigureRDP
   - Ec2OutputRDPCert
   - Ec2SetDriveLetter
   - Ec2WindowsActivate

- Ec2DynamicBootVolumeSize

3. After it is finished, sends a “Windows is ready” message to the instance system logs.
4. Runs all “After Windows is ready” plug-ins at the same time.
   - Amazon CloudWatch Logs
   - UserData
   - AWS Systems Manager (Systems Manager)

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

Run Sysprep with the EC2Config service

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

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

   ```
   C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml
   ```

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

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

8. Choose **OK**.

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

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

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

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

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

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.

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.
- **Shared AWS Marketplace AMIs** – The `CopyImage` action is not supported for copying an AWS Marketplace AMI that was shared from another account. Instead, if you want to copy an AWS Marketplace AMI in another account, you must do the following: share the AWS Marketplace AMI with the other account and then, in the other account, launch an EC2 instance using the AWS Marketplace AMI. You can then create an AMI from the instance using the `CreateImage` action. The new AMI retains all the AWS Marketplace codes. Note that this process also applies to any AMIs that were directly or indirectly derived from an AWS Marketplace AMI. For more information about creating an AMI from an instance, see Create a custom Windows AMI.

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 a custom Windows 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.

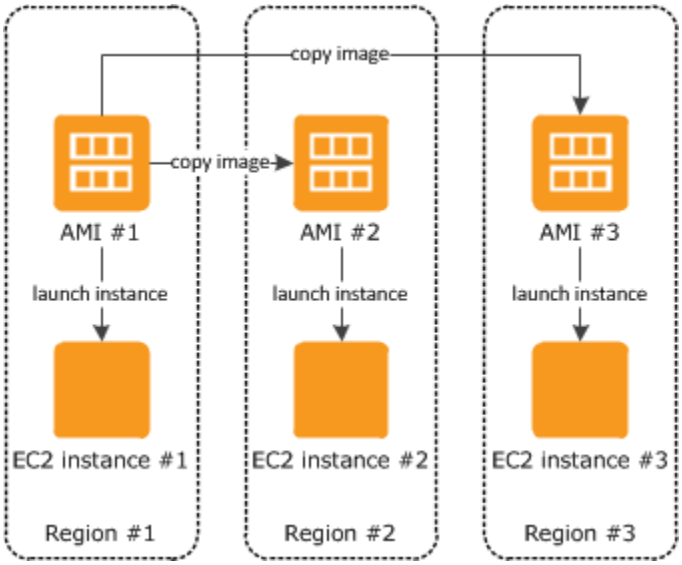
- deregister-image (AWS CLI)
- Unregister-EC2Image (AWS Tools for Windows PowerShell)

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.

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.

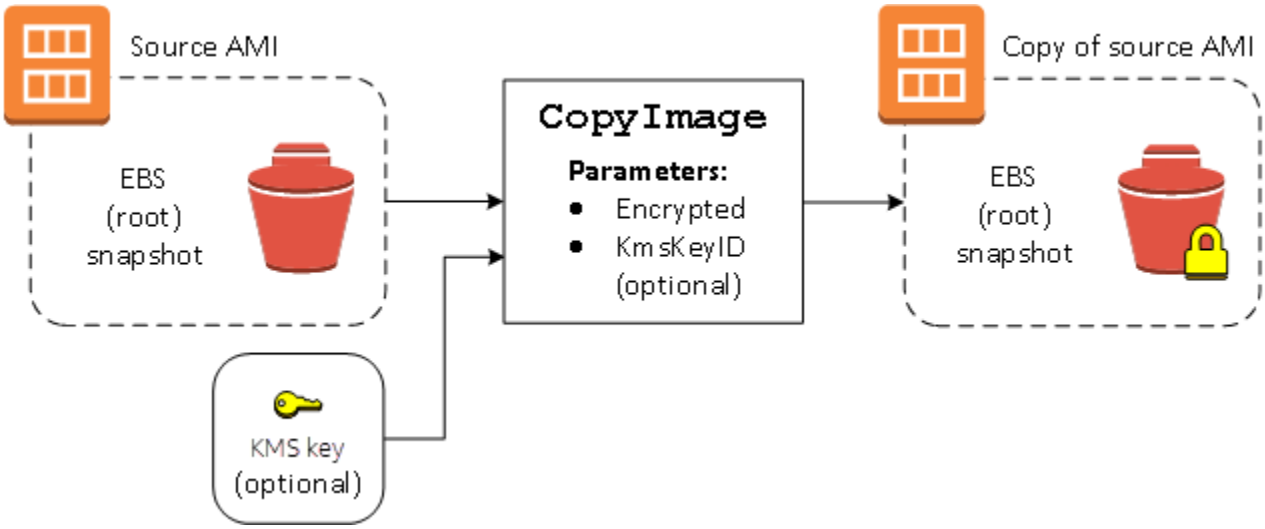
By default (i.e., without specifying encryption parameters), the backing snapshot of an AMI is copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot results in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, 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"
}
```

Last launched time

`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 Clean up your 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

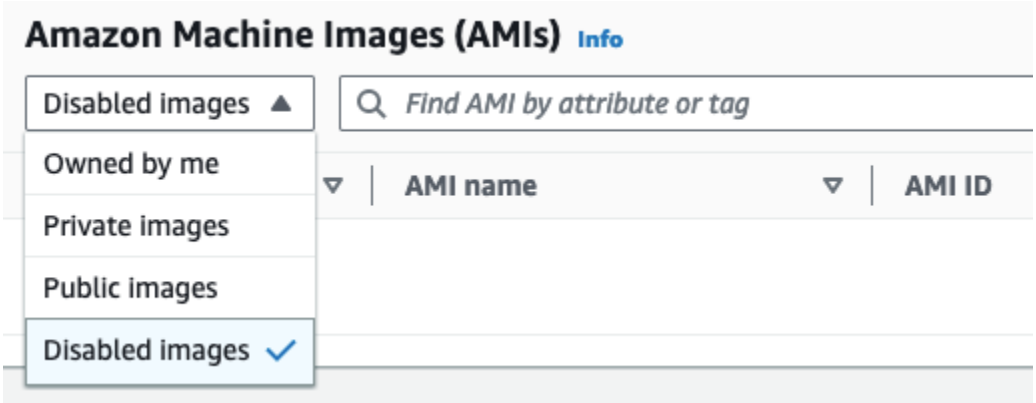
You can view disabled AMIs in the EC2 console and by using the AWS CLI.

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-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": 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 your AMI

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI or any snapshots created during the AMI creation process. You'll continue to incur usage costs for these instances and storage costs for the snapshot. Therefore, you should terminate any instances and delete any snapshots that you're finished with.

Contents

- Considerations
- Clean up your AMI
- Last launched time

Considerations

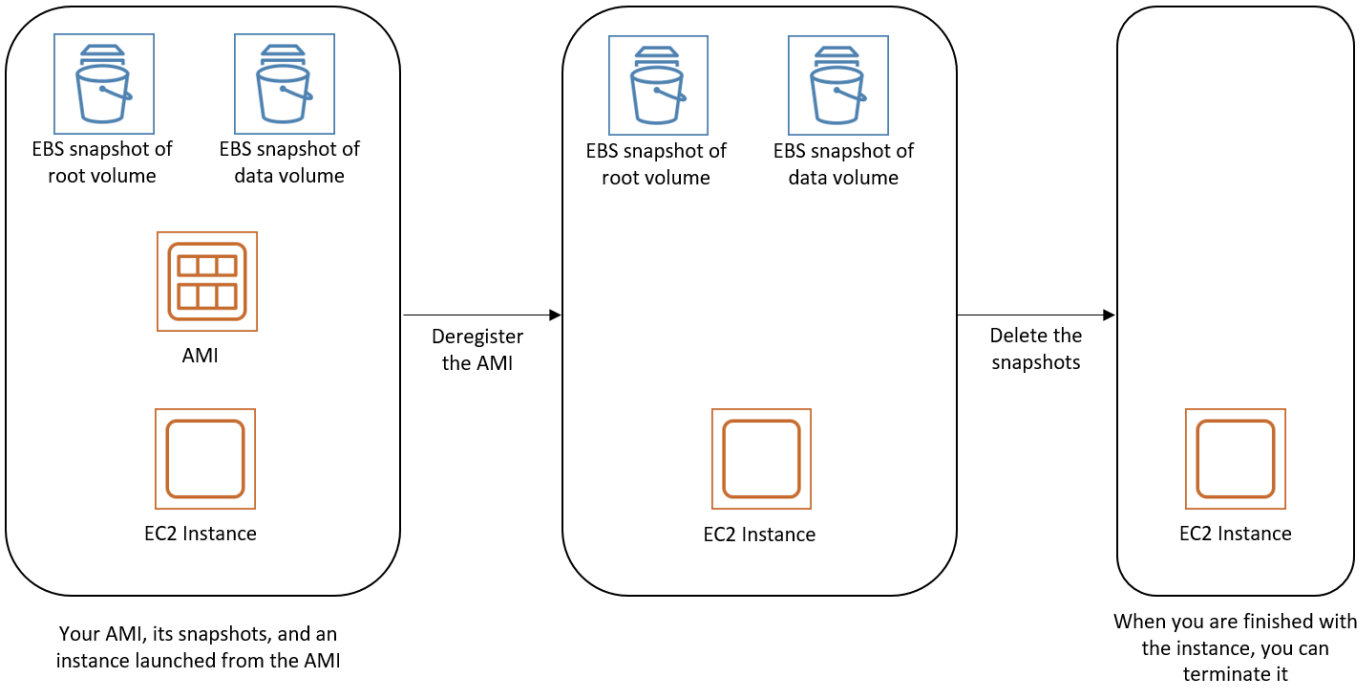
The following considerations apply to deregistering AMIs:

- You can't deregister an AMI that is not owned by your account.
- You can't deregister an AMI that is managed by the AWS Backup service using Amazon EC2. Instead, use AWS Backup to delete the corresponding recovery points in the backup vault. For more information, see Deleting backups in the *AWS Backup Developer Guide*.

Clean up your AMI

When you deregister an AMI, it doesn't affect the snapshot(s) that were created for the volume(s) of the instance during the AMI creation process. You'll continue to incur storage costs for the snapshots. Therefore, if you are finished with the snapshots, you should delete them.

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



You can use one of the following methods to clean up your AMI.

Console

To clean up your AMI

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. **Deregister the AMI**
   a. In the navigation pane, choose **AMIs**.
   b. From the filter bar, choose **Owned by me** to list your available AMIs or **Disabled images** to list your disabled AMIs.
   c. Select the AMI to deregister, and take note of its ID—this can help you find the snapshots to delete in the next step.

d. Choose **Actions**, **Deregister AMI**. When prompted for confirmation, choose **Deregister AMI**.

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

3. **Delete snapshots that are no longer needed**

   a. In the navigation pane, choose **Snapshots**.

   b. Select a snapshot to delete (look for the AMI ID from the prior step in the **Description** column).

   c. Choose **Actions**, **Delete snapshot**. When prompted for confirmation, choose **Delete**.

4. **(Optional) Terminate instances**

   If you are finished with an instance that you launched from the AMI, you can terminate it.

   a. In the navigation pane, choose **Instances**, and then select the instance to terminate.

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

AWS CLI

Follow these steps to clean up your AMI

1. **Deregister the AMI**

   Deregister the AMI using the deregister-image command:

```
aws ec2 deregister-image --image-id ami-12345678
```

2. **Delete snapshots that are no longer needed**

   Delete snapshots that are no longer needed by using the delete-snapshot command:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

3. **Terminate instances (Optional)**

   If you are finished with an instance that you launched from the AMI, you can terminate it by using the terminate-instances command:

```
aws ec2 terminate-instances --instance-ids i-12345678
```

PowerShell

Follow these steps to clean up your AMI

1. **Deregister the AMI**

   Deregister the AMI using the Unregister-EC2Image cmdlet:

```
Unregister-EC2Image -ImageId ami-12345678
```

2. **Delete snapshots that are no longer needed**

   Delete snapshots that are no longer needed by using the Remove-EC2Snapshot cmdlet:

```
Remove-EC2Snapshot -SnapshotId snap-12345678
```

3. **Terminate instances (Optional)**

   If you are finished with an instance that you launched from the AMI, you can terminate it by using the Remove-EC2Instance cmdlet:

```
Remove-EC2Instance -InstanceId i-12345678
```

Last launched time

`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 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",
}
```

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.

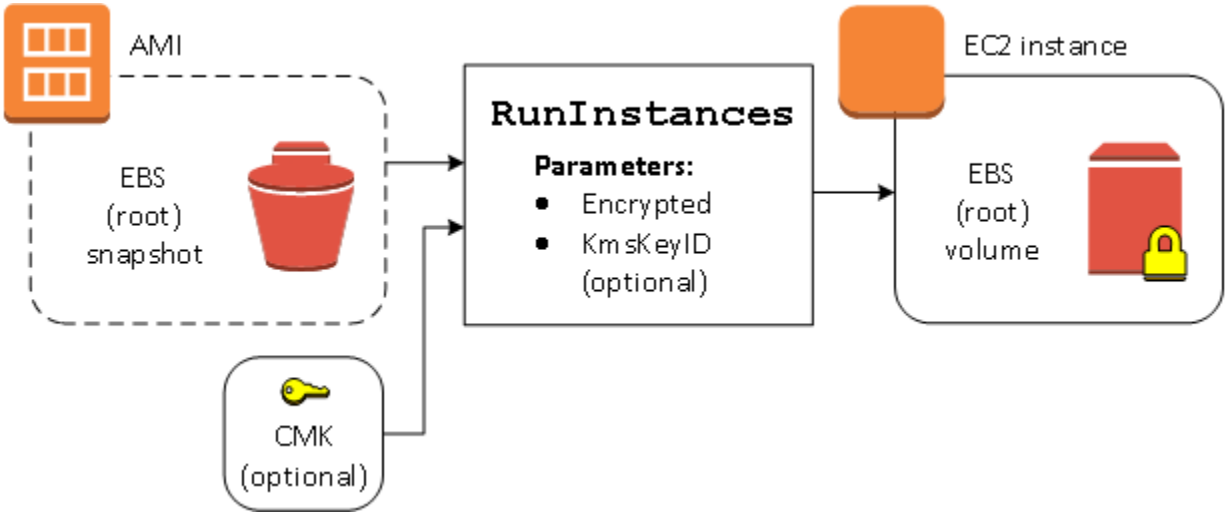
Submitting a `KmsKeyId` without also setting the `Encrypted` parameter results in an error.

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

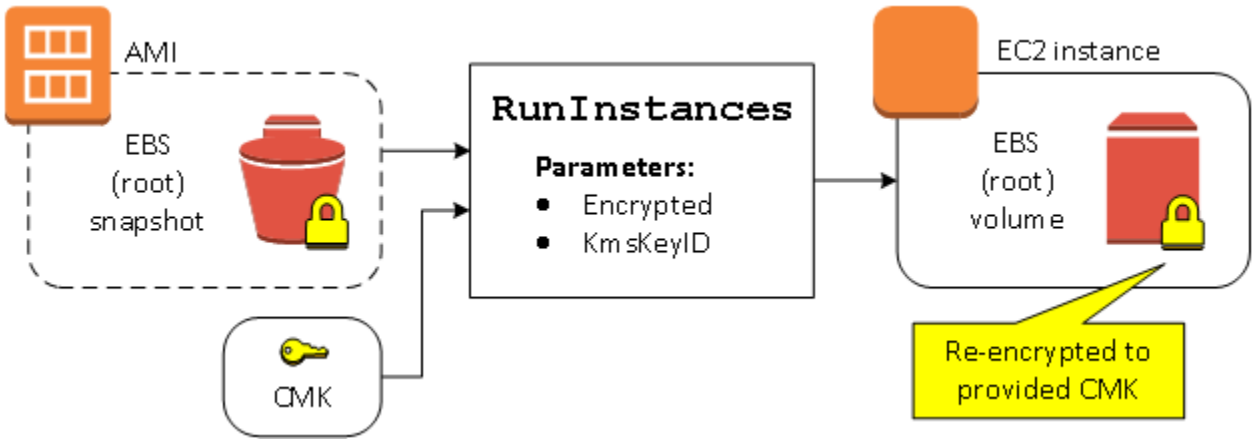
In this example, an AMI backed by an unencrypted snapshot is used to launch an EC2 instance with an encrypted EBS volume.



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

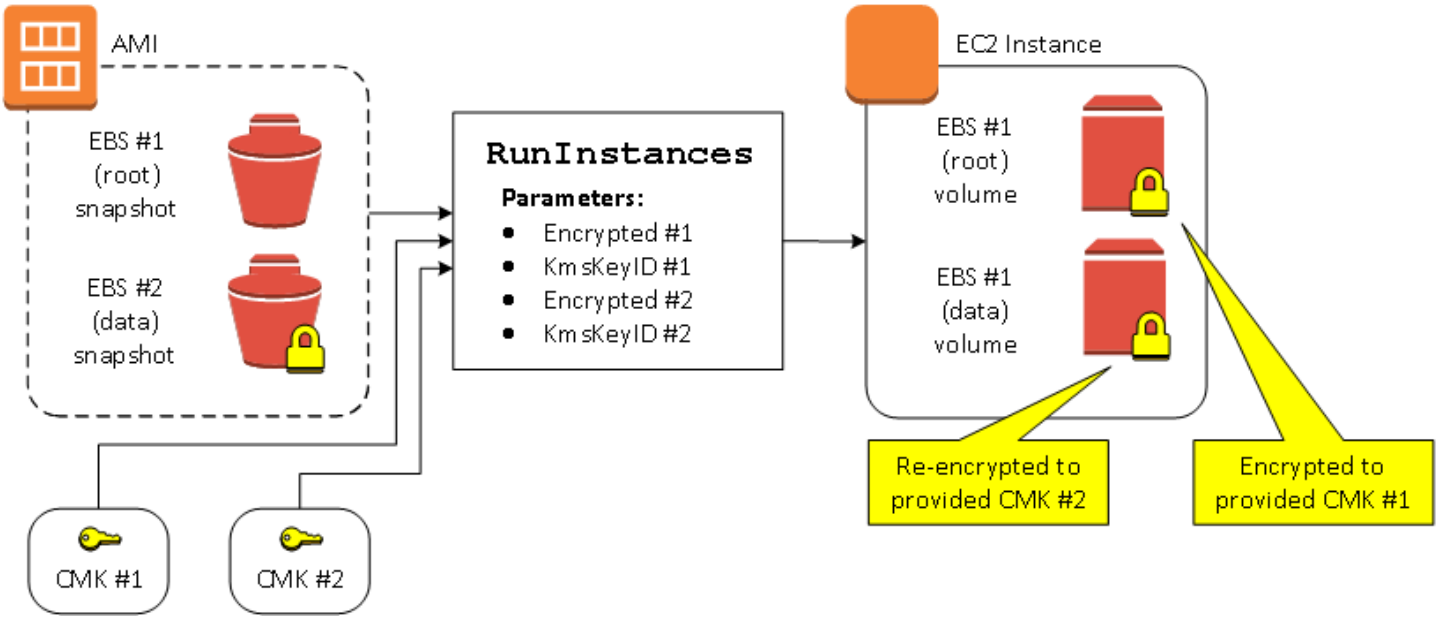
In this example, an AMI backed by an encrypted snapshot is used to launch an EC2 instance with an EBS volume encrypted by a new KMS key.



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.

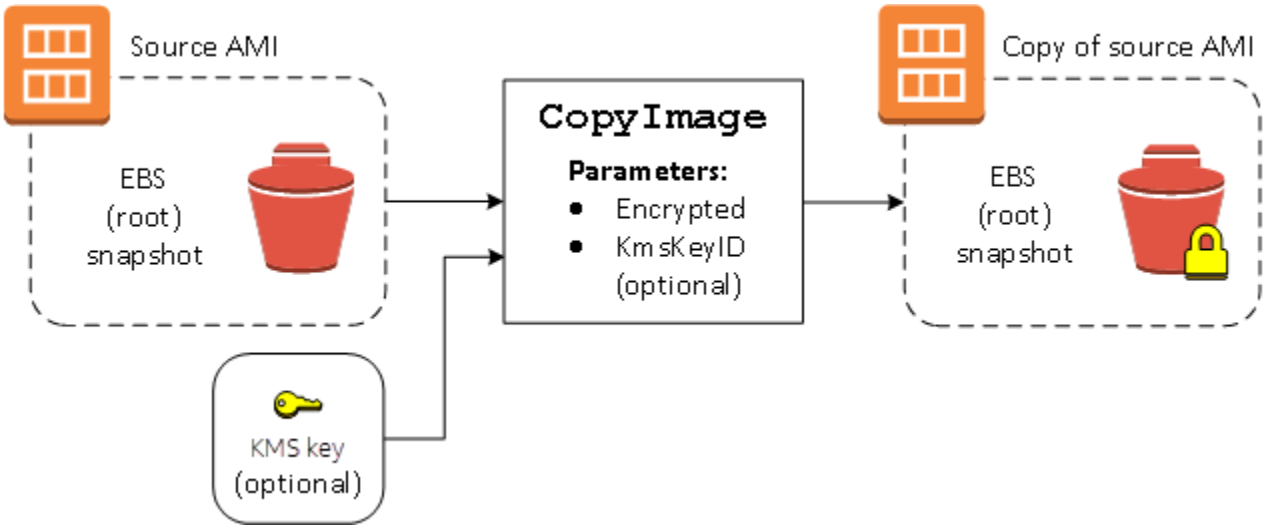
The following section provides an example of copying an AMI using non-default encryption parameters, resulting in a change of encryption state.

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		
CreateRestoreImageTask	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 **
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0

Platform details	Usage operation **
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

Platform details	Usage operation **
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise *	RunInstances:0102
Windows with SQL Server Standard *	RunInstances:0006
Windows with SQL Server Web *	RunInstances:0202

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

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

Finding AMI billing and usage details

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

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

- **Platform details**
- **Usage operation**
- **AMI ID**

Find AMI billing information (console)

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

Look up AMI billing information from the AMIs page

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

Look up AMI billing information from the Instances page

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

Find AMI billing information (AWS CLI)

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

To find the AMI ID

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

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

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

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

To find the AMI billing information

If you know the AMI ID, you can use the describe-images command to get the AMI platform and usage operation details.

```
$ 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`.

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

Quota name	Description	Default quota per Region
AMI sharing	The maximum number of entities (organizations, organizational units (OUs), and accounts) that an AMI can be shared with in a Region. Note that if you share an AMI with an organization or OU, the number of accounts in the organization or OU does not count towards the quota.	1,000

If you exceed your quotas and you want to create or share more AMIs, you can do the following:

- If you exceed your total AMIs or public AMIs quota, consider deregistering unused images.
- If you exceed your public AMIs quota, consider making one or more public AMIs private.
- If you exceed your AMI sharing quota, consider sharing your AMIs with an organization or OU instead of separate accounts.
- Request a quota increase for AMIs.

Request a quota increase for AMIs

If you need more than the default quota for AMIs, you can request a quota increase.

To request a quota increase for AMIs

1. Open the Service Quotas console at https://console.aws.amazon.com/servicequotas/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 the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

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

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

Amazon EC2 Windows instances

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

Instances and AMIs

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch *instances*, which are copies of the AMI running as virtual servers in the cloud.

Amazon publishes many AMIs that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMIs. Doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.

To improve launch time for Windows instances, you can optimize your AMI for faster launching, which creates a set of pre-provisioned snapshots to launch instances up to 65% faster. To learn more, see Configure Windows fast launch for your Windows Server AMI

You can launch different types of instances from a single AMI. An *instance type* determines the infrastructure that is used for your instance. Some instance types are intended for general purpose use, while others support optimizations for specific uses such as high performance processors for computing, enhanced memory for processing large data sets, and fast I/O for storage. Select an instance type that delivers the performance and size that you need for the applications or software that you plan to run on the instance. For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Your Windows instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

Your AWS account has a limit on the number of instances that you can have running. For more information about this limit, and how to request an increase, see How many instances can I run in Amazon EC2 in the Amazon EC2 General FAQ.

Differences between Windows Server and Windows instances

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

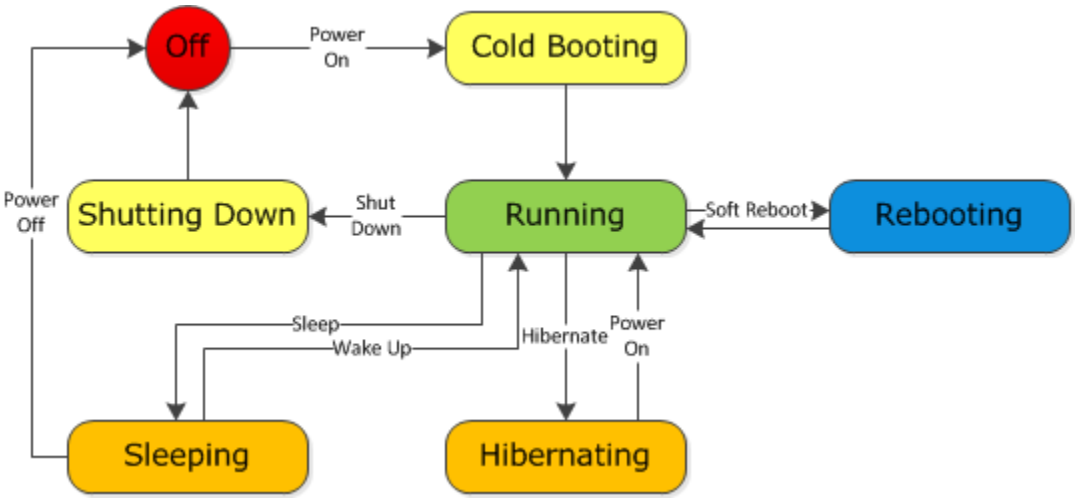
Before you begin launching Amazon EC2 Windows instances, you should be aware that the architecture of applications running on cloud servers can differ significantly from the architecture for traditional application models running on your hardware. Implementing applications on cloud servers requires a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

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

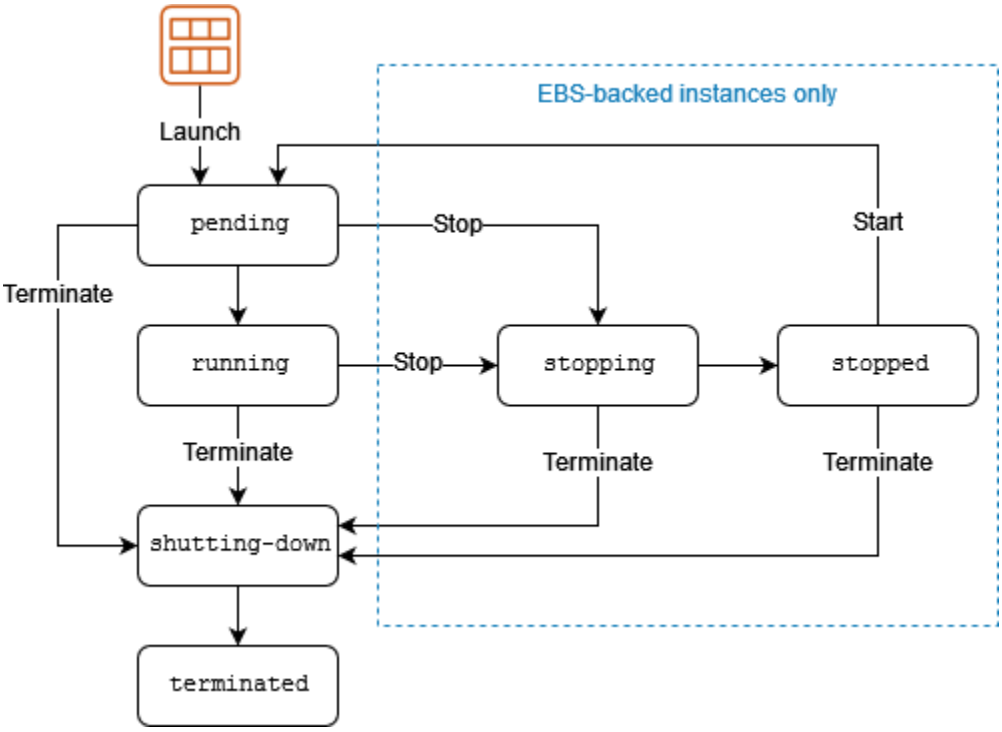
Windows Server	Amazon EC2 Windows Instance
Resources and capacity are physically limited.	Resources and capacity are scalable.
You pay for the infrastructure, even if you don't use it.	You pay for the usage of the infrastructure. We stop charging you for the instance as soon as you stop or terminate it.
Occupies physical space and must be maintained on a regular basis.	Doesn't occupy physical space and does not require regular maintenance.
Starts with push of the power button (known as *cold booting*).	Starts with the launch of the instance.
You can keep the server running until it is time to shut it down, or put it in a sleep or hibernation state (during which the server is powered down).	You can keep the server running, or stop and restart it (during which the instance is moved to a new host computer).
When you shut down the server, all resources remain intact and in the state they were in when you switched it off. Information you	When you terminate the instance, its infrastru cture is no longer available to you. You can't connect to or restart an instance after you'

Windows Server	Amazon EC2 Windows Instance
stored on the hard drives persists and can be accessed whenever it's needed. You can restore the server to the running state by powering it on.	ve terminated it. However, you can create an image from your instance while it's running, and launch new instances from the image at any time.

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



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



Design your applications to run on Windows instances

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

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

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

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

With Amazon EC2, designing for failure and outages is an integral and crucial part of the architecture. As with any scalable and redundant system, architecture of your system should

account for computing, network, and storage failures. You have to build mechanisms in your applications that can handle different kinds of failures. The key is to build a modular system with individual components that are not tightly coupled, can interact asynchronously, and treat one another as black boxes that are independently scalable. Thus, if one of your components fails or is busy, you can launch more instances of that component without breaking your current system.

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

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

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

Instance types

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see 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.

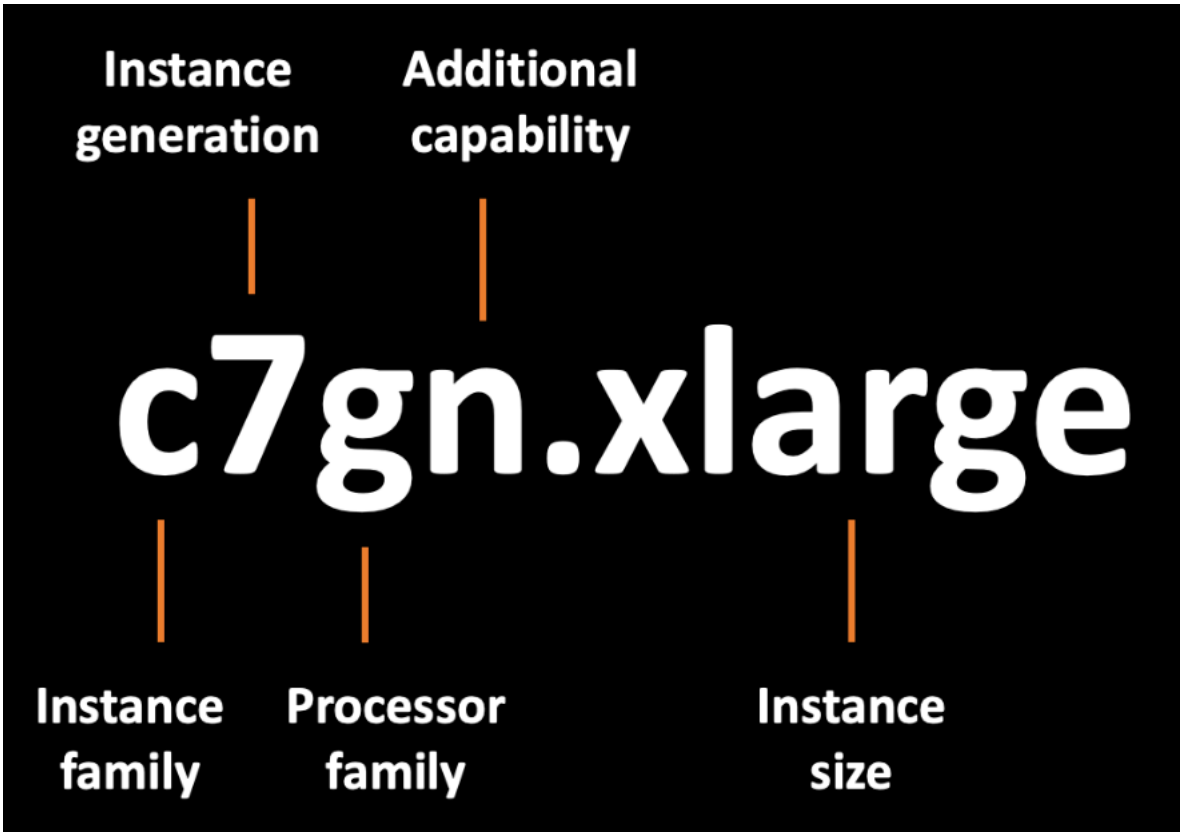
Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, 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

- Instance type naming convention
- Available instance types
- Hardware specifications
- Instances built on the Nitro System
- Networking and storage features
- Instance limits
- General purpose instances
- Compute optimized instances
- Memory optimized instances
- Storage optimized instances
- Windows accelerated computing instances
- High-performance computing instances
- Find an Amazon EC2 instance type
- Get recommendations for an instance type
- Change the instance type

Instance type naming convention

Amazon EC2 provides a variety of instance types so you can choose the type that best meets your requirements. Instance types are named based on their family, generation, processor family, additional capabilities, and size. The first position of the instance type name indicates the instance family, for example `c`. The second position indicates the instance generation, for example `7`. The third position indicates the processor family, for example `g`. The remaining letters before the period indicate additional capabilities, such as instance store volumes. After the period (`.`) is the instance size, such as `small` or `4xlarge`, or `metal` for bare metal instances.



Instance families

- **C** – Compute optimized
- **D** – Dense storage
- **F** – FPGA
- **G** – Graphics intensive
- **Hpc** – High performance computing
- **I** – Storage optimized
- **Im** – Storage optimized with a one to four ratio of vCPU to memory
- **Is** – Storage optimized with a one to six ratio of vCPU to memory
- **Inf** – AWS Inferentia
- **M** – General purpose
- **Mac** – macOS
- **P** – GPU accelerated
- **R** – Memory optimized
- **T** – Burstable performance

- **Trn** – AWS Trainium
- **U** – High memory
- **VT** – Video transcoding
- **X** – Memory intensive

Processor families

- **a** – AMD processors
- **g** – AWS Graviton processors
- **i** – Intel processors

Additional capabilities

- **b** – EBS optimized
- **d** – Instance store volumes
- **n** – Network and EBS optimized
- **e** – Extra storage or memory
- **z** – High performance
- **q** – Qualcomm inference accelerators
- **flex** – Flex instance

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.

> **Note**
>
> Previous generation instances are still fully supported and retain the same features and functionality. We encourage you to use the latest generation of instances to get the best performance.

To determine which instance types meet your requirements, such as supported Regions, compute resources, or storage resources, see Find an Amazon EC2 instance type.

Topics

- Current generation instances
- Previous generation instances

Current generation instances

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

Sixth generation and later Amazon EC2 instance types

- **General purpose:** M6a, M6i, M6id, M6idn, M6in, M7a, M7i, M7i-flex
- **Compute optimized:** C6a, C6i, C6id, C6in, C7a, C7i
- **Memory optimized:** R6a, R6i, R6id, R6idn, R6in, R7a, R7i, R7iz, X2idn, X2iedn
- **Storage optimized:** I4i
- **High-performance computing:** Hpc6a, Hpc6id, Hpc7a

Instances

- General purpose
- Compute optimized
- Memory optimized
- Storage optimized
- Accelerated computing
- High-performance computing

General purpose

Type	Sizes	
M5	m5.large|m5.xlarge |m5.2xlarge |m5.4xlarge | m5.8xlarge |m5.12xlarge |m5.16xlarge |m5.24xlarge | m5.metal	
M5a	m5a.large |m5a.xlarge |m5a.2xlarge |m5a.4xlarge | m5a.8xlarge |m5a.12xlarge |m5a.16xlarge |m5a.24xlarge	
M5ad	m5ad.large |m5ad.xlarge |m5ad.2xlarge |m5ad.4xlarge | m5ad.8xlarge |m5ad.12xlarge |m5ad.16xlarge |m5ad.24xlarge	
M5d	m5d.large |m5d.xlarge |m5d.2xlarge |m5d.4xlarge | m5d.8xlarge |m5d.12xlarge |m5d.16xlarge |m5d.24xlarge |m5d.metal	
M5dn	m5dn.large |m5dn.xlarge |m5dn.2xlarge |m5dn.4xlarge | m5dn.8xlarge |m5dn.12xlarge |m5dn.16xlarge |m5dn.24xlarge |m5dn.metal	
M5n	m5n.large |m5n.xlarge |m5n.2xlarge |m5n.4xlarge | m5n.8xlarge |m5n.12xlarge |m5n.16xlarge |m5n.24xlarge |m5n.metal	
M5zn	m5zn.large |m5zn.xlarge |m5zn.2xlarge |m5zn.3xlarge | m5zn.6xlarge |m5zn.12xlarge |m5zn.metal	
M6a	m6a.large |m6a.xlarge |m6a.2xlarge |m6a.4xlarge | m6a.8xlarge |m6a.12xlarge |m6a.16xlarge |m6a.24xlarge |m6a.32xlarge |m6a.48xlarge |m6a.metal	
M6i	m6i.large |m6i.xlarge |m6i.2xlarge |m6i.4xlarge | m6i.8xlarge |m6i.12xlarge |m6i.16xlarge |m6i.24xlarge |m6i.32xlarge |m6i.metal	

Type	Sizes
M6id	m6id.large |m6id.xlarge |m6id.2xlarge |m6id.4xlarge |m6id.8xlarge |m6id.12xlarge |m6id.16xlarge |m6id.24xlarge |m6id.32xlarge |m6id.metal
M6idn	m6idn.large |m6idn.xlarge |m6idn.2xlarge |m6idn.4xlarge |m6idn.8xlarge |m6idn.12xlarge |m6idn.16xlarge |m6idn.24xlarge |m6idn.32xlarge |m6idn.metal
M6in	m6in.large |m6in.xlarge |m6in.2xlarge |m6in.4xlarge |m6in.8xlarge |m6in.12xlarge |m6in.16xlarge |m6in.24xlarge |m6in.32xlarge |m6in.metal
M7a	m7a.medium |m7a.large |m7a.xlarge |m7a.2xlarge |m7a.4xlarge |m7a.8xlarge |m7a.12xlarge |m7a.16xlarge |m7a.24xlarge |m7a.32xlarge |m7a.48xlarge |m7a.metal-48xl
M7i	m7i.large |m7i.xlarge |m7i.2xlarge |m7i.4xlarge |m7i.8xlarge |m7i.12xlarge |m7i.16xlarge |m7i.24xlarge |m7i.48xlarge |m7i.metal-24xl |m7i.metal-48xl
M7i-flex	m7i-flex.large |m7i-flex.xlarge |m7i-flex.2xlarge |m7i-flex.4xlarge |m7i-flex.8xlarge
T2	t2.nano|t2.micro|t2.small|t2.medium |t2.large|t2.xlarge |t2.2xlarge
T3	t3.nano|t3.micro|t3.small|t3.medium |t3.large|t3.xlarge |t3.2xlarge
T3a	t3a.nano|t3a.micro |t3a.small |t3a.medium |t3a.large |t3a.xlarge |t3a.2xlarge

Compute optimized

Type	Sizes
C5	c5.large|c5.xlarge |c5.2xlarge |c5.4xlarge | c5.9xlarge |c5.12xlarge |c5.18xlarge |c5.24xlarge | c5.metal
C5a	c5a.large |c5a.xlarge |c5a.2xlarge |c5a.4xlarge | c5a.8xlarge |c5a.12xlarge |c5a.16xlarge |c5a.24xlarge
C5ad	c5ad.large |c5ad.xlarge |c5ad.2xlarge |c5ad.4xlarge | c5ad.8xlarge |c5ad.12xlarge |c5ad.16xlarge |c5ad.24xlarge
C5d	c5d.large |c5d.xlarge |c5d.2xlarge |c5d.4xlarge | c5d.9xlarge |c5d.12xlarge |c5d.18xlarge |c5d.24xlarge |c5d.metal
C5n	c5n.large |c5n.xlarge |c5n.2xlarge |c5n.4xlarge | c5n.9xlarge |c5n.18xlarge |c5n.metal
C6a	c6a.large |c6a.xlarge |c6a.2xlarge |c6a.4xlarge | c6a.8xlarge |c6a.12xlarge |c6a.16xlarge |c6a.24xlarge |c6a.32xlarge |c6a.48xlarge |c6a.metal
C6i	c6i.large |c6i.xlarge |c6i.2xlarge |c6i.4xlarge | c6i.8xlarge |c6i.12xlarge |c6i.16xlarge |c6i.24xlarge |c6i.32xlarge |c6i.metal
C6id	c6id.large |c6id.xlarge |c6id.2xlarge |c6id.4xlarge | c6id.8xlarge |c6id.12xlarge |c6id.16xlarge |c6id.24xlarge |c6id.32xlarge |c6id.metal
C6in	c6in.large |c6in.xlarge |c6in.2xlarge |c6in.4xlarge | c6in.8xlarge |c6in.12xlarge |c6in.16xlarge |c6in.24xlarge |c6in.32xlarge |c6in.metal

Type	Sizes
C7a	c7a.medium |c7a.large |c7a.xlarge |c7a.2xlarge | c7a.4xlarge |c7a.8xlarge |c7a.12xlarge |c7a.16xlarge |c7a.24xlarge |c7a.32xlarge |c7a.48xlarge |c7a.metal -48xl
C7i	c7i.large |c7i.xlarge |c7i.2xlarge |c7i.4xlarge | c7i.8xlarge |c7i.12xlarge |c7i.16xlarge |c7i.24xlarge |c7i.48xlarge |c7i.metal-24xl |c7i.metal-48xl

Memory optimized

Type	Sizes
R5	r5.large|r5.xlarge |r5.2xlarge |r5.4xlarge | r5.8xlarge |r5.12xlarge |r5.16xlarge |r5.24xlarge | r5.metal
R5a	r5a.large |r5a.xlarge |r5a.2xlarge |r5a.4xlarge | r5a.8xlarge |r5a.12xlarge |r5a.16xlarge |r5a.24xlarge
R5ad	r5ad.large |r5ad.xlarge |r5ad.2xlarge |r5ad.4xlarge | r5ad.8xlarge |r5ad.12xlarge |r5ad.16xlarge |r5ad.24xlarge
R5b	r5b.large |r5b.xlarge |r5b.2xlarge |r5b.4xlarge | r5b.8xlarge |r5b.12xlarge |r5b.16xlarge |r5b.24xlarge |r5b.metal
R5d	r5d.large |r5d.xlarge |r5d.2xlarge |r5d.4xlarge | r5d.8xlarge |r5d.12xlarge |r5d.16xlarge |r5d.24xlarge |r5d.metal

Type	Sizes
R5dn	r5dn.large |r5dn.xlarge |r5dn.2xlarge |r5dn.4xlarge |r5dn.8xlarge |r5dn.12xlarge |r5dn.16xlarge |r5dn.24xlarge |r5dn.metal
R5n	r5n.large |r5n.xlarge |r5n.2xlarge |r5n.4xlarge |r5n.8xlarge |r5n.12xlarge |r5n.16xlarge |r5n.24xlarge |r5n.metal
R6a	r6a.large |r6a.xlarge |r6a.2xlarge |r6a.4xlarge |r6a.8xlarge |r6a.12xlarge |r6a.16xlarge |r6a.24xlarge |r6a.32xlarge |r6a.48xlarge |r6a.metal
R6i	r6i.large |r6i.xlarge |r6i.2xlarge |r6i.4xlarge |r6i.8xlarge |r6i.12xlarge |r6i.16xlarge |r6i.24xlarge |r6i.32xlarge |r6i.metal
R6idn	r6idn.large |r6idn.xlarge |r6idn.2xlarge |r6idn.4xlarge |r6idn.8xlarge |r6idn.12xlarge |r6idn.16xlarge |r6idn.24xlarge |r6idn.32xlarge |r6idn.metal
R6in	r6in.large |r6in.xlarge |r6in.2xlarge |r6in.4xlarge |r6in.8xlarge |r6in.12xlarge |r6in.16xlarge |r6in.24xlarge |r6in.32xlarge |r6in.metal
R6id	r6id.large |r6id.xlarge |r6id.2xlarge |r6id.4xlarge |r6id.8xlarge |r6id.12xlarge |r6id.16xlarge |r6id.24xlarge |r6id.32xlarge |r6id.metal
R7a	r7a.medium |r7a.large |r7a.xlarge |r7a.2xlarge |r7a.4xlarge |r7a.8xlarge |r7a.12xlarge |r7a.16xlarge |r7a.24xlarge |r7a.32xlarge |r7a.48xlarge |r7a.metal-48xl
R7i	r7i.large |r7i.xlarge |r7i.2xlarge |r7i.4xlarge |r7i.8xlarge |r7i.12xlarge |r7i.16xlarge |r7i.24xlarge |r7i.48xlarge |r7i.metal-24xl |r7i.metal-48xl

Type	Sizes
R7iz	r7iz.large |r7iz.xlarge |r7iz.2xlarge |r7iz.4xlarge | r7iz.8xlarge |r7iz.12xlarge |r7iz.16xlarge |r7iz.32xl arge |r7iz.metal-16xl |r7iz.metal-32xl
U-3tb1	u-3tb1.56xlarge
U-6tb1	u-6tb1.56xlarge |u-6tb1.112xlarge |u-6tb1.metal
U-9tb1	u-9tb1.112xlarge |u-9tb1.metal
U-12tb1	u-12tb1.112xlarge |u-12tb1.metal
U-18tb1	u-18tb1.112xlarge |u-18tb1.metal
U-24tb1	u-24tb1.112xlarge |u-24tb1.metal
X1	x1.16xlarge |x1.32xlarge
X2idn	x2idn.16xlarge |x2idn.24xlarge |x2idn.32xlarge | x2idn.metal
X2iedn	x2iedn.xlarge |x2iedn.2xlarge |x2iedn.4xlarge | x2iedn.8xlarge |x2iedn.16xlarge |x2iedn.24xlarge | x2iedn.32xlarge |x2iedn.metal
X2iezn	x2iezn.2xlarge |x2iezn.4xlarge |x2iezn.6xlarge | x2iezn.8xlarge |x2iezn.12xlarge |x2iezn.metal
X1e	x1e.xlarge |x1e.2xlarge |x1e.4xlarge |x1e.8xlarge | x1e.16xlarge |x1e.32xlarge
z1d	z1d.large |z1d.xlarge |z1d.2xlarge |z1d.3xlarge | z1d.6xlarge |z1d.12xlarge |z1d.metal

Storage optimized

Type	Sizes
D2	d2.xlarge |d2.2xlarge |d2.4xlarge |d2.8xlarge
D3	d3.xlarge |d3.2xlarge |d3.4xlarge |d3.8xlarge
D3en	d3en.xlarge |d3en.2xlarge |d3en.4xlarge |d3en.6xlarge |d3en.8xlarge |d3en.12xlarge
H1	h1.2xlarge |h1.4xlarge |h1.8xlarge |h1.16xlarge
I3	i3.large|i3.xlarge |i3.2xlarge |i3.4xlarge |i3.8xlarge |i3.16xlarge |i3.metal
I3en	i3en.large |i3en.xlarge |i3en.2xlarge |i3en.3xlarge |i3en.6xlarge |i3en.12xlarge |i3en.24xlarge |i3en.metal
I4i	i4i.large |i4i.xlarge |i4i.2xlarge |i4i.4xlarge |i4i.8xlarge |i4i.12xlarge |i4i.16xlarge |i4i.24xlarge |i4i.32xlarge |i4i.metal

Accelerated computing

Type	Sizes
G3	g3.4xlarge |g3.8xlarge |g3.16xlarge
G4ad	g4ad.xlarge |g4ad.2xlarge |g4ad.4xlarge |g4ad.8xlarge |g4ad.16xlarge
G4dn	g4dn.xlarge |g4dn.2xlarge |g4dn.4xlarge |g4dn.8xlarge |g4dn.12xlarge |g4dn.16xlarge |g4dn.metal
G5	g5.xlarge |g5.2xlarge |g5.4xlarge |g5.8xlarge |g5.12xlarge |g5.16xlarge |g5.24xlarge |g5.48xlarge

Type	Sizes
P2	p2.xlarge |p2.8xlarge |p2.16xlarge
P3	p3.2xlarge |p3.8xlarge |p3.16xlarge
P3dn	p3dn.24xlarge

High-performance computing

Type	Sizes
Hpc6id	hpc6id.32xlarge
Hpc7a	hpc7a.12xlarge |hpc7a.24xlarge |hpc7a.48xlarge | hpc7a.96xlarge

Previous generation instances

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

Type	Sizes
C1	c1.medium |c1.xlarge
C3	c3.large|c3.xlarge |c3.2xlarge |c3.4xlarge |c3.8xlarge
C4	c4.large|c4.xlarge |c4.2xlarge |c4.4xlarge |c4.8xlarge
G2	g2.2xlarge |g2.8xlarge
I2	i2.xlarge |i2.2xlarge |i2.4xlarge |i2.8xlarge
M1	m1.small|m1.medium |m1.large|m1.xlarge

Type	Sizes
M2	`m2.xlarge` |`m2.2xlarge` |`m2.4xlarge`
M3	`m3.medium` |`m3.large`|`m3.xlarge` |`m3.2xlarge`
M4	`m4.large`|`m4.xlarge` |`m4.2xlarge` |`m4.4xlarge` |`m4.10xlarge` | `m4.16xlarge`
R3	`r3.large`|`r3.xlarge` |`r3.2xlarge` |`r3.4xlarge` |`r3.8xlarge`
R4	`r4.large`|`r4.xlarge` |`r4.2xlarge` |`r4.4xlarge` |`r4.8xlarge` | `r4.16xlarge`
T1	`t1.micro`

Hardware specifications

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance second, it's convenient and inexpensive to test multiple instance types before making a decision. If your needs change, even after you make a decision, you can change the instance type later. For more information, see Change the instance type.

Processor features

Intel processor features

Amazon EC2 instances that run on Intel processors may include the following features. Not all of the following processor features are supported by all instance types. For detailed information about which features are available for each instance type, see Amazon EC2 Instance Types.

- **Intel AES New Instructions (AES-NI)** — Intel AES-NI encryption instruction set improves upon the original Advanced Encryption Standard (AES) algorithm to provide faster data protection and greater security. All current generation EC2 instances support this processor feature.

- **Intel Advanced Vector Extensions (Intel AVX, Intel AVX2, and Intel AVX-512)** — Intel AVX and Intel AVX2 are 256-bit, and Intel AVX-512 is a 512-bit instruction set extension designed for applications that are Floating Point (FP) intensive. Intel AVX instructions improve performance for applications like image and audio/video processing, scientific simulations, financial analytics, and 3D modeling and analysis. These features are only available on instances launched with HVM AMIs.
- **Intel Turbo Boost Technology** — Intel Turbo Boost Technology processors automatically run cores faster than the base operating frequency.
- **Intel Deep Learning Boost (Intel DL Boost)** — Accelerates AI deep learning use cases. The 2nd Gen Intel Xeon Scalable processors extend Intel AVX-512 with a new Vector Neural Network Instruction (VNNI/INT8) that significantly increases deep learning inference performance over previous generation Intel Xeon Scalable processors (with FP32) for image recognition/segmentation, object detection, speech recognition, language translation, recommendation systems, reinforcement learning, and more. VNNI may not be compatible with all Linux distributions.

  The following instances support VNNI: `M5n`, `R5n`, `M5dn`, `M5zn`, `R5b`, `R5dn`, `D3`, `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.

Instances built on the Nitro System

The Nitro System is a collection of hardware and software components built by AWS that enable high performance, high availability, and high security. For more information, see AWS Nitro System.

The Nitro System provides bare metal capabilities that eliminate virtualization overhead and support workloads that require full access to host hardware. Bare metal instances are well suited for the following:

- Workloads that require access to low-level hardware features (for example, Intel VT) that are not available or fully supported in virtualized environments
- Applications that require a non-virtualized environment for licensing or support

Nitro components

The following components are part of the Nitro System:

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

Virtualized instances

The following virtualized instances are built on the Nitro System:

- **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 | U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | U-24tb1 | X2idn | X2iedn | X2iezn | z1d
- **Storage optimized:** D3 | D3en | I3en | I4i
- **Accelerated computing:** G4ad | G4dn | G5 | P3dn
- **High-performance computing:** Hpc6id | Hpc7a

Bare metal instances

The following bare metal instances are built on the Nitro System:

- **General purpose:** `m5.metal` | `m5d.metal` | `m5dn.metal` | `m5n.metal` | `m5zn.metal` | `m6a.metal` | `m6i.metal` | `m6id.metal` | `m6idn.metal` | `m6in.metal` | `m7a.metal-48xl` | `m7i.metal-24xl` | `m7i.metal-48xl`
- **Compute optimized:** `c5.metal` | `c5d.metal` | `c5n.metal` | `c6a.metal` | `c6i.metal` | `c6id.metal` | `c6in.metal` | `c7a.metal-48xl` | `c7i.metal-24xl` | `c7i.metal-48xl`
- **Memory optimized:** `r5.metal` | `r5b.metal` | `r5d.metal` | `r5dn.metal` | `r5n.metal` | `r6a.metal` | `r6i.metal` | `r6idn.metal` | `r6in.metal` | `r6id.metal` | `r7a.metal-48xl` | `r7i.metal-24xl` | `r7i.metal-48xl` | `r7iz.metal-16xl` | `r7iz.metal-32xl` | `u-6tb1.metal` | `u-9tb1.metal` | `u-12tb1.metal` | `u-18tb1.metal` | `u-24tb1.metal` | `x2idn.metal` | `x2iedn.metal` | `x2iezn.metal` | `z1d.metal`
- **Storage optimized:** `i3.metal` | `i3en.metal` | `i4i.metal`
- **Accelerated computing:** `g4dn.metal`

Learn more

For more information, see the following videos:

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

Networking and storage features

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

Networking features

- IPv6 is supported on all instance types, except C1, M1, M2, M3, and T1.
- To maximize the networking and bandwidth performance of your instance type, you can do the following:
  - Launch supported instance types into a cluster placement group to optimize your instances for high performance computing (HPC) applications. Instances in a common cluster placement group can benefit from high-bandwidth, low-latency networking. For more information, see Placement groups.

  - Enable enhanced networking for supported current generation instance types to get significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see Enhanced networking on Windows.
- Current generation instance types that are enabled for enhanced networking have the following networking performance attributes:
  - Traffic within the same Region over private IPv4 or IPv6 can support 5 Gbps for single-flow traffic and up to 25 Gbps for multi-flow traffic (depending on the instance type).
  - Traffic to and from Amazon S3 buckets within the same Region over the public IP address space or through a VPC endpoint can use all available instance aggregate bandwidth.
- The maximum transmission unit (MTU) supported varies across instance types. All Amazon EC2 instance types support standard Ethernet V2 1500 MTU frames. All current generation instances support 9001 MTU, or jumbo frames, and some previous generation instances support them as well. For more information, see Network maximum transmission unit (MTU) for your EC2 instance.

Storage features

- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instance types that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. Some instance types support NVMe instance store volumes. Some instance types support NVMe EBS volumes. For more information, see Amazon EBS and NVMe and NVMe SSD volumes.
- To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as EBS–optimized instances. Some instance types are EBS–optimized by default. For more information, see Amazon EBS–optimized instances.

Summary of networking and storage features

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

Instances

- General purpose
- Compute optimized
- Memory optimized

- Storage optimized
- Accelerated computing
- High-performance computing
- Previous generation instance types

General purpose

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
M5	Yes	Yes	No	Yes	ENA
M5a	Yes	Yes	No	Yes	ENA
M5ad	No	Yes	NVMe	Yes	ENA
M5d	No	Yes	NVMe	Yes	ENA
M5dn	No	Yes	NVMe	Yes	ENA | EFA
M5n	Yes	Yes	No	Yes	ENA | EFA
M5zn	Yes	Yes	No	Yes	ENA | EFA
M6a	Yes	Yes	No	Yes	ENA | EFA
M6i	Yes	Yes	No	Yes	ENA | EFA
M6id	No	Yes	NVMe	Yes	ENA | EFA
M6idn	No	Yes	NVMe	Yes	ENA | EFA
M6in	Yes	Yes	No	Yes	ENA | EFA
M7a	Yes	Yes	No	Yes	ENA | EFA
M7i	Yes	Yes	No	Yes	ENA | EFA
M7i-flex	Yes	Yes	No	Yes	ENA

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
T2	Yes	No	No	Yes	Not supported
T3	Yes	Yes	No	Yes	ENA
T3a	Yes	Yes	No	Yes	ENA

Compute optimized

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
C5	Yes	Yes	No	Yes	ENA
C5a	Yes	Yes	No	Yes	ENA
C5ad	No	Yes	NVMe	Yes	ENA
C5d	No	Yes	NVMe	Yes	ENA
C5n	Yes	Yes	No	Yes	ENA | EFA
C6a	Yes	Yes	No	Yes	ENA | EFA
C6i	Yes	Yes	No	Yes	ENA | EFA
C6id	No	Yes	NVMe	Yes	ENA | EFA
C6in	Yes	Yes	No	Yes	ENA | EFA
C7a	Yes	Yes	No	Yes	ENA | EFA
C7i	Yes	Yes	No	Yes	ENA | EFA

Memory optimized

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
R5	Yes	Yes	No	Yes	ENA
R5a	Yes	Yes	No	Yes	ENA
R5ad	No	Yes	NVMe	Yes	ENA
R5b	Yes	Yes	No	Yes	ENA
R5d	No	Yes	NVMe	Yes	ENA
R5dn	No	Yes	NVMe	Yes	ENA | EFA
R5n	Yes	Yes	No	Yes	ENA | EFA
R6a	Yes	Yes	No	Yes	ENA | EFA
R6i	Yes	Yes	No	Yes	ENA | EFA
R6idn	No	Yes	NVMe	Yes	ENA | EFA
R6in	Yes	Yes	No	Yes	ENA | EFA
R6id	No	Yes	NVMe	Yes	ENA | EFA
R7a	Yes	Yes	No	Yes	ENA | EFA
R7i	Yes	Yes	No	Yes	ENA | EFA
R7iz	Yes	Yes	No	Yes	ENA | EFA
U-3tb1	Yes	Yes	No	Yes	ENA
U-6tb1	Yes	Yes	No	Yes	ENA
U-9tb1	Yes	Yes	No	Yes	ENA
U-12tb1	Yes	Yes	No	Yes	ENA

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
U-18tb1	Yes	Yes	No	Yes	ENA
U-24tb1	Yes	Yes	No	Yes	ENA
X1	No	No	SSD	Yes	ENA
X2idn	No	Yes	NVMe	Yes	ENA | EFA
X2iedn	No	Yes	NVMe	Yes	ENA | EFA
X2iezn	Yes	Yes	No	Yes	ENA | EFA
X1e	No	No	SSD	Yes	ENA
z1d	No	Yes	NVMe	Yes	ENA

Storage optimized

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
D2	No	No	HDD	Yes	Intel 82599 VF
D3	No	Yes	NVMe	Yes	ENA
D3en	No	Yes	NVMe	Yes	ENA
H1	No	No	HDD	Yes	ENA
I3	No	Yes	NVMe	Yes	ENA
I3en	No	Yes	NVMe	Yes	ENA | EFA
I4i	No	Yes	NVMe	Yes	ENA | EFA

Accelerated computing

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
G3	Yes	No	No	Yes	ENA
G4ad	No	Yes	NVMe	Yes	ENA
G4dn	No	Yes	NVMe	Yes	ENA | EFA
G5	No	Yes	NVMe	Yes	ENA | EFA
P2	Yes	No	No	Yes	ENA
P3	Yes	No	No	Yes	ENA
P3dn	No	Yes	NVMe	Yes	ENA | EFA

High-performance computing

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
Hpc6id	No	Yes	NVMe	Yes	ENA | EFA
Hpc7a	Yes	Yes	No	Yes	ENA | EFA

Previous generation instance types

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

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
C1	No	No	HDD	Yes	Not supported

Instance type	EBS only	NVME EBS	Instance store	Placement group	Enhanced networking
C3	No	No	SSD	Yes	Intel 82599 VF
C4	Yes	No	No	Yes	Intel 82599 VF
G2	No	No	SSD	Yes	Not supported
I2	No	No	SSD	Yes	Intel 82599 VF
M1	No	No	HDD	Yes	Not supported
M2	No	No	HDD	Yes	Not supported
M3	No	No	SSD	Yes	Not supported
M4	Yes	No	No	Yes	ENA
R3	No	No	SSD	Yes	Intel 82599 VF
R4	Yes	No	No	Yes	ENA
T1	Yes	No	No	Yes	Not supported

> **Note**
>
> `m4.16xlarge` instances support enhanced networking using an Elastic Network Adapter (ENA). All other M4 instance types support enhanced networking using an Intel 82599 VF interface.

Instance limits

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

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

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

General purpose instances

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

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

M5 and M5a instances

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

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

For more information, see Amazon EC2 M5 Instances.

Bare metal instances, such as `m5.metal`, `m5n.metal`, and `m5zn.metal` provide your applications with direct access to physical resources of the host server, such as processors and memory.

M5zn

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

- Gaming
- High performance computing
- Simulation modeling

For more information, see Amazon EC2 M5 Instances.

M6i and M6id instances

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

- Application servers and web servers
- Microservices
- High performance computing
- App development
- Small and midsize databases
- Caching fleets

Bare metal instances such as `m6i.metal` provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 M6i Instances.

M6in and M6idn instances

These instances are well suited for network-intensive workloads such as the following:

- High-performance file systems
- Distributed web scale in-memory caches
- Caching fleets

- Real-time big data analytics
- Telco applications such as 5G User Plane Function (UPF)

For more information, see Amazon EC2 M6i Instances.

M7i instances

M7i instances expand the seventh generation Amazon EC2 instance portfolio to include x86-based options. These instances are powered by custom 4th generation Intel Xeon Scalable Processors (Sapphire Rapids) that are exclusive to AWS. They deliver up to 15 percent improvement in price-performance compared to M6i instances. M7i instances are a good fit for running general-purpose workloads, such as web-servers, application servers, micro-services, and small data stores.

For more information, see Amazon EC2 M7i instances.

M7i-flex instances

A majority of general-purpose workloads benefit from the latest generation performance but do not fully utilize the compute resources. Amazon EC2 flex instances are an ideal first choice to run such workloads. Flex instances are cost-optimized variants of Amazon EC2 instances that provide the easiest way for you to get price performance benefits and lower prices for a majority of common workloads.

M7i-flex instances are the first Amazon EC2 flex instances that are based on the 4th generation Intel Xeon Scalable processors (Sapphire Rapids), and can save costs when compared to equivalent M instances. M7i-flex instances offer up to 32 vCPUs and up to 128 GiB of memory to deliver a balance of compute, memory, and networking resources. These instances are well suited for general purpose workloads such as the following:

- Application servers and web servers
- Microservices
- Virtual desktops
- App development
- Databases
- Mobile applications
- Batch workloads

For more information, see Amazon EC2 M7i and M7i-flex instances.

M7g and M7gd instances

These instances are powered by AWS Graviton3 processors and deliver balanced compute, memory, and networking for a broad range a general purpose workloads. They are well suited for the following:

- Application servers
- Microservices
- Gaming servers
- Midsize data stores
- Caching fleets

For more information, see Amazon EC2 M7g instances.

M7a instances

M7a instances are built on the AWS Nitro System and are ideal for applications that benefit from high performance and high throughput, low latency networking, such as application servers, microservices, gaming servers, mid-size data stores, application development environments, and caching fleets.

For more information, see Amazon EC2 M7a Instances.

T2, T3, and T3a instances

These instances provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. An Unlimited instance can sustain high CPU performance for any period of time whenever required. For more information, see Burstable performance instances. They are well-suited for the following:

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

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

Contents

- Hardware specifications
- Instance performance
- Network performance
- Amazon EBS I/O performance
- SSD-based instance store volume I/O performance
- Release notes
- Burstable performance instances

Hardware specifications

The following is a summary of the hardware specifications for general purpose instances. A virtual central processing unit (vCPU) represents a portion of the physical CPU assigned to a virtual machine (VM). For x86 instances, there are two vCPUs per core. For Graviton instances, there is one vCPU per core.

Instance type	Default vCPUs	Memory (GiB)
m1.small	1	1.70
m1.medium	1	3.70
m1.large	2	7.50
m1.xlarge	4	15.00
m2.xlarge	2	17.10
m2.2xlarge	4	34.20
m2.4xlarge	8	68.40
m3.medium	1	3.75
m3.large	2	7.50
m3.xlarge	4	15.00
m3.2xlarge	8	30.00

Instance type	Default vCPUs	Memory (GiB)
m4.large	2	8.00
m4.xlarge	4	16.00
m4.2xlarge	8	32.00
m4.4xlarge	16	64.00
m4.10xlarge	40	160.00
m4.16xlarge	64	256.00
m5.large	2	8.00
m5.xlarge	4	16.00
m5.2xlarge	8	32.00
m5.4xlarge	16	64.00
m5.8xlarge	32	128.00
m5.12xlarge	48	192.00
m5.16xlarge	64	256.00
m5.24xlarge	96	384.00
m5.metal	96	384.00
m5a.large	2	8.00
m5a.xlarge	4	16.00
m5a.2xlarge	8	32.00
m5a.4xlarge	16	64.00
m5a.8xlarge	32	128.00

Instance type	Default vCPUs	Memory (GiB)
m5a.12xlarge	48	192.00
m5a.16xlarge	64	256.00
m5a.24xlarge	96	384.00
m5ad.large	2	8.00
m5ad.xlarge	4	16.00
m5ad.2xlarge	8	32.00
m5ad.4xlarge	16	64.00
m5ad.8xlarge	32	128.00
m5ad.12xlarge	48	192.00
m5ad.16xlarge	64	256.00
m5ad.24xlarge	96	384.00
m5d.large	2	8.00
m5d.xlarge	4	16.00
m5d.2xlarge	8	32.00
m5d.4xlarge	16	64.00
m5d.8xlarge	32	128.00
m5d.12xlarge	48	192.00
m5d.16xlarge	64	256.00
m5d.24xlarge	96	384.00
m5d.metal	96	384.00

Instance type	Default vCPUs	Memory (GiB)
m5dn.large	2	8.00
m5dn.xlarge	4	16.00
m5dn.2xlarge	8	32.00
m5dn.4xlarge	16	64.00
m5dn.8xlarge	32	128.00
m5dn.12xlarge	48	192.00
m5dn.16xlarge	64	256.00
m5dn.24xlarge	96	384.00
m5dn.metal	96	384.00
m5n.large	2	8.00
m5n.xlarge	4	16.00
m5n.2xlarge	8	32.00
m5n.4xlarge	16	64.00
m5n.8xlarge	32	128.00
m5n.12xlarge	48	192.00
m5n.16xlarge	64	256.00
m5n.24xlarge	96	384.00
m5n.metal	96	384.00
m5zn.large	2	8.00
m5zn.xlarge	4	16.00

Instance type	Default vCPUs	Memory (GiB)
m5zn.2xlarge	8	32.00
m5zn.3xlarge	12	48.00
m5zn.6xlarge	24	96.00
m5zn.12xlarge	48	192.00
m5zn.metal	48	192.00
m6a.large	2	8.00
m6a.xlarge	4	16.00
m6a.2xlarge	8	32.00
m6a.4xlarge	16	64.00
m6a.8xlarge	32	128.00
m6a.12xlarge	48	192.00
m6a.16xlarge	64	256.00
m6a.24xlarge	96	384.00
m6a.32xlarge	128	512.00
m6a.48xlarge	192	768.00
m6a.metal	192	768.00
m6i.large	2	8.00
m6i.xlarge	4	16.00
m6i.2xlarge	8	32.00
m6i.4xlarge	16	64.00

Instance type	Default vCPUs	Memory (GiB)
m6i.8xlarge	32	128.00
m6i.12xlarge	48	192.00
m6i.16xlarge	64	256.00
m6i.24xlarge	96	384.00
m6i.32xlarge	128	512.00
m6i.metal	128	512.00
m6id.large	2	8.00
m6id.xlarge	4	16.00
m6id.2xlarge	8	32.00
m6id.4xlarge	16	64.00
m6id.8xlarge	32	128.00
m6id.12xlarge	48	192.00
m6id.16xlarge	64	256.00
m6id.24xlarge	96	384.00
m6id.32xlarge	128	512.00
m6id.metal	128	512.00
m6idn.large	2	8.00
m6idn.xlarge	4	16.00
m6idn.2xlarge	8	32.00
m6idn.4xlarge	16	64.00

Instance type	Default vCPUs	Memory (GiB)
m6idn.8xlarge	32	128.00
m6idn.12xlarge	48	192.00
m6idn.16xlarge	64	256.00
m6idn.24xlarge	96	384.00
m6idn.32xlarge	128	512.00
m6idn.metal	128	512.00
m6in.large	2	8.00
m6in.xlarge	4	16.00
m6in.2xlarge	8	32.00
m6in.4xlarge	16	64.00
m6in.8xlarge	32	128.00
m6in.12xlarge	48	192.00
m6in.16xlarge	64	256.00
m6in.24xlarge	96	384.00
m6in.32xlarge	128	512.00
m6in.metal	128	512.00
m7a.medium	1	4.00
m7a.large	2	8.00
m7a.xlarge	4	16.00
m7a.2xlarge	8	32.00

Instance type	Default vCPUs	Memory (GiB)
m7a.4xlarge	16	64.00
m7a.8xlarge	32	128.00
m7a.12xlarge	48	192.00
m7a.16xlarge	64	256.00
m7a.24xlarge	96	384.00
m7a.32xlarge	128	512.00
m7a.48xlarge	192	768.00
m7a.metal-48xl	192	768.00
m7i.large	2	8.00
m7i.xlarge	4	16.00
m7i.2xlarge	8	32.00
m7i.4xlarge	16	64.00
m7i.8xlarge	32	128.00
m7i.12xlarge	48	192.00
m7i.16xlarge	64	256.00
m7i.24xlarge	96	384.00
m7i.48xlarge	192	768.00
m7i.metal-24xl	96	384.00
m7i.metal-48xl	192	768.00
m7i-flex.large	2	8.00

Instance type	Default vCPUs	Memory (GiB)
m7i-flex.xlarge	4	16.00
m7i-flex.2xlarge	8	32.00
m7i-flex.4xlarge	16	64.00
m7i-flex.8xlarge	32	128.00
t1.micro	1	0.61
t2.nano	1	0.50
t2.micro	1	1.00
t2.small	1	2.00
t2.medium	2	4.00
t2.large	2	8.00
t2.xlarge	4	16.00
t2.2xlarge	8	32.00
t3.nano	2	0.50
t3.micro	2	1.00
t3.small	2	2.00
t3.medium	2	4.00
t3.large	2	8.00
t3.xlarge	4	16.00
t3.2xlarge	8	32.00
t3a.nano	2	0.50

Instance type	Default vCPUs	Memory (GiB)
t3a.micro	2	1.00
t3a.small	2	2.00
t3a.medium	2	4.00
t3a.large	2	8.00
t3a.xlarge	4	16.00
t3a.2xlarge	8	32.00

The general purpose instances use the following processors.

AMD processors

- **AMD EPYC 7000 series processors (AMD EPYC 7571)**: M5a, M5ad, T3a
- **3rd generation AMD EPYC processors (AMD EPYC 7R13)**: M6a
- **4th generation AMD EPYC processors (AMD EPYC 9R14)**: M7a

Intel processors

- **Intel Xeon Scalable processors (Haswell E5-2676 v3 or Broadwell E5-2686 v4)**: M4, T2
- **Intel Xeon Scalable processors (Skylake 8175M or Cascade Lake 8259CL)**: M5, M5d, T3
- **2nd generation Intel Xeon Scalable processors (Cascade Lake 8259CL)**: M5n
- **2nd generation Intel Xeon Scalable processors (Cascade Lake 8252C)**: M5zn
- **3rd generation Intel Xeon Scalable processors (Ice Lake 8375C)**: M6i, M6id
- **4th generation Intel Xeon Scalable processors (Sapphire Rapids 8488C)**: M7i, M7i-flex

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Instance performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some general purpose instances are EBS-optimized by default at no additional cost. For more information, see Amazon EBS–optimized instances.

Flex instance performance

M7i-flex instances offer a balance of compute, memory, and network resources, and they provide the most cost-effective way to run a broad spectrum of general purpose applications. M7i-flex instances provide reliable CPU resources to deliver a baseline CPU performance of 40 percent, which is designed to meet the compute requirements for a majority of general purpose workloads. For times when workloads need more performance, M7i-flex instances provide the ability to exceed baseline CPU and deliver up to 100 percent CPU for 95 percent of the time over a 24-hour window.

M7i-flex instances running at a high CPU utilization that is consistently above the baseline for long periods of time might see a gradual reduction in the maximum burst CPU throughput.

Network performance

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

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

> **Note**
>
> Instance types indicated with a † have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth.

Instance type	Network performance	Enhanced networking features
m1.small	Low	Not supported
m1.medium	Moderate	Not supported
m1.large	Moderate	Not supported
m1.xlarge	High	Not supported
m2.xlarge	Moderate	Not supported
m2.2xlarge	Moderate	Not supported
m2.4xlarge	High	Not supported
m3.medium	Moderate	Not supported
m3.large	Moderate	Not supported
m3.xlarge	High	Not supported
m3.2xlarge	High	Not supported
m4.large	Moderate	Intel 82599 VF
m4.xlarge	High	Intel 82599 VF
m4.2xlarge	High	Intel 82599 VF
m4.4xlarge	High	Intel 82599 VF
m4.10xlarge	10 Gigabit	Intel 82599 VF
m4.16xlarge	25 Gigabit	ENA
m5.large †	Up to 10 Gigabit	ENA
m5.xlarge †	Up to 10 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
m5.2xlarge †	Up to 10 Gigabit	ENA
m5.4xlarge †	Up to 10 Gigabit	ENA
m5.8xlarge	10 Gigabit	ENA
m5.12xlarge	12 Gigabit	ENA
m5.16xlarge	20 Gigabit	ENA
m5.24xlarge	25 Gigabit	ENA
m5.metal	25 Gigabit	ENA
m5a.large †	Up to 10 Gigabit	ENA
m5a.xlarge †	Up to 10 Gigabit	ENA
m5a.2xlarge †	Up to 10 Gigabit	ENA
m5a.4xlarge †	Up to 10 Gigabit	ENA
m5a.8xlarge †	Up to 10 Gigabit	ENA
m5a.12xlarge	10 Gigabit	ENA
m5a.16xlarge	12 Gigabit	ENA
m5a.24xlarge	20 Gigabit	ENA
m5ad.large †	Up to 10 Gigabit	ENA
m5ad.xlarge †	Up to 10 Gigabit	ENA
m5ad.2xlarge †	Up to 10 Gigabit	ENA
m5ad.4xlarge †	Up to 10 Gigabit	ENA
m5ad.8xlarge †	Up to 10 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
m5ad.12xlarge	10 Gigabit	ENA
m5ad.16xlarge	12 Gigabit	ENA
m5ad.24xlarge	20 Gigabit	ENA
m5d.large †	Up to 10 Gigabit	ENA
m5d.xlarge †	Up to 10 Gigabit	ENA
m5d.2xlarge †	Up to 10 Gigabit	ENA
m5d.4xlarge †	Up to 10 Gigabit	ENA
m5d.8xlarge	10 Gigabit	ENA
m5d.12xlarge	12 Gigabit	ENA
m5d.16xlarge	20 Gigabit	ENA
m5d.24xlarge	25 Gigabit	ENA
m5d.metal	25 Gigabit	ENA
m5dn.large †	Up to 25 Gigabit	ENA
m5dn.xlarge †	Up to 25 Gigabit	ENA
m5dn.2xlarge †	Up to 25 Gigabit	ENA
m5dn.4xlarge †	Up to 25 Gigabit	ENA
m5dn.8xlarge	25 Gigabit	ENA
m5dn.12xlarge	50 Gigabit	ENA
m5dn.16xlarge	75 Gigabit	ENA
m5dn.24xlarge	100 Gigabit	ENA | EFA

Instance type	Network performance	Enhanced networking features
m5dn.metal	100 Gigabit	ENA | EFA
m5n.large †	Up to 25 Gigabit	ENA
m5n.xlarge †	Up to 25 Gigabit	ENA
m5n.2xlarge †	Up to 25 Gigabit	ENA
m5n.4xlarge †	Up to 25 Gigabit	ENA
m5n.8xlarge	25 Gigabit	ENA
m5n.12xlarge	50 Gigabit	ENA
m5n.16xlarge	75 Gigabit	ENA
m5n.24xlarge	100 Gigabit	ENA | EFA
m5n.metal	100 Gigabit	ENA | EFA
m5zn.large †	Up to 25 Gigabit	ENA
m5zn.xlarge †	Up to 25 Gigabit	ENA
m5zn.2xlarge †	Up to 25 Gigabit	ENA
m5zn.3xlarge †	Up to 25 Gigabit	ENA
m5zn.6xlarge	50 Gigabit	ENA
m5zn.12xlarge	100 Gigabit	ENA | EFA
m5zn.metal	100 Gigabit	ENA | EFA
m6a.large †	Up to 12.5 Gigabit	ENA
m6a.xlarge †	Up to 12.5 Gigabit	ENA
m6a.2xlarge †	Up to 12.5 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
m6a.4xlarge †	Up to 12.5 Gigabit	ENA
m6a.8xlarge	12.5 Gigabit	ENA
m6a.12xlarge	18.75 Gigabit	ENA
m6a.16xlarge	25 Gigabit	ENA
m6a.24xlarge	37.5 Gigabit	ENA
m6a.32xlarge	50 Gigabit	ENA
m6a.48xlarge	50 Gigabit	ENA | EFA
m6a.metal	50 Gigabit	ENA | EFA
m6i.large †	Up to 12.5 Gigabit	ENA
m6i.xlarge †	Up to 12.5 Gigabit	ENA
m6i.2xlarge †	Up to 12.5 Gigabit	ENA
m6i.4xlarge †	Up to 12.5 Gigabit	ENA
m6i.8xlarge	12.5 Gigabit	ENA
m6i.12xlarge	18.75 Gigabit	ENA
m6i.16xlarge	25 Gigabit	ENA
m6i.24xlarge	37.5 Gigabit	ENA
m6i.32xlarge	50 Gigabit	ENA | EFA
m6i.metal	50 Gigabit	ENA | EFA
m6id.large †	Up to 12.5 Gigabit	ENA
m6id.xlarge †	Up to 12.5 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
m6id.2xlarge †	Up to 12.5 Gigabit	ENA
m6id.4xlarge †	Up to 12.5 Gigabit	ENA
m6id.8xlarge	12.5 Gigabit	ENA
m6id.12xlarge	18.75 Gigabit	ENA
m6id.16xlarge	25 Gigabit	ENA
m6id.24xlarge	37.5 Gigabit	ENA
m6id.32xlarge	50 Gigabit	ENA | EFA
m6id.metal	50 Gigabit	ENA | EFA
m6idn.large †	Up to 25 Gigabit	ENA
m6idn.xlarge †	Up to 30 Gigabit	ENA
m6idn.2xlarge †	Up to 40 Gigabit	ENA
m6idn.4xlarge †	Up to 50 Gigabit	ENA
m6idn.8xlarge	50 Gigabit	ENA
m6idn.12xlarge	75 Gigabit	ENA
m6idn.16xlarge	100 Gigabit	ENA
m6idn.24xlarge	150 Gigabit	ENA
m6idn.32xlarge	200 Gigabit	ENA | EFA
m6idn.metal	200 Gigabit	ENA | EFA
m6in.large †	Up to 25 Gigabit	ENA
m6in.xlarge †	Up to 30 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
m6in.2xlarge †	Up to 40 Gigabit	ENA
m6in.4xlarge †	Up to 50 Gigabit	ENA
m6in.8xlarge	50 Gigabit	ENA
m6in.12xlarge	75 Gigabit	ENA
m6in.16xlarge	100 Gigabit	ENA
m6in.24xlarge	150 Gigabit	ENA
m6in.32xlarge	200 Gigabit	ENA | EFA
m6in.metal	200 Gigabit	ENA | EFA
m7a.medium †	Up to 12.5 Gigabit	ENA
m7a.large †	Up to 12.5 Gigabit	ENA
m7a.xlarge †	Up to 12.5 Gigabit	ENA
m7a.2xlarge †	Up to 12.5 Gigabit	ENA
m7a.4xlarge †	Up to 12.5 Gigabit	ENA
m7a.8xlarge	12.5 Gigabit	ENA
m7a.12xlarge	18.75 Gigabit	ENA
m7a.16xlarge	25 Gigabit	ENA
m7a.24xlarge	37.5 Gigabit	ENA
m7a.32xlarge	50 Gigabit	ENA
m7a.48xlarge	50 Gigabit	ENA | EFA
m7a.metal-48xl	50 Gigabit	ENA | EFA

Instance type	Network performance	Enhanced networking features
m7i.large †	Up to 12.5 Gigabit	ENA
m7i.xlarge †	Up to 12.5 Gigabit	ENA
m7i.2xlarge †	Up to 12.5 Gigabit	ENA
m7i.4xlarge †	Up to 12.5 Gigabit	ENA
m7i.8xlarge	12.5 Gigabit	ENA
m7i.12xlarge	18.75 Gigabit	ENA
m7i.16xlarge	25 Gigabit	ENA
m7i.24xlarge	37.5 Gigabit	ENA
m7i.48xlarge	50 Gigabit	ENA | EFA
m7i.metal-24xl	37.5 Gigabit	ENA
m7i.metal-48xl	50 Gigabit	ENA | EFA
m7i-flex.large †	Up to 12.5 Gigabit	ENA
m7i-flex.xlarge †	Up to 12.5 Gigabit	ENA
m7i-flex.2xlarge †	Up to 12.5 Gigabit	ENA
m7i-flex.4xlarge †	Up to 12.5 Gigabit	ENA
m7i-flex.8xlarge †	Up to 12.5 Gigabit	ENA
t1.micro	Very Low	Not supported
t2.nano	Low to Moderate	Not supported
t2.micro	Low to Moderate	Not supported
t2.small	Low to Moderate	Not supported

Instance type	Network performance	Enhanced networking features
t2.medium	Low to Moderate	Not supported
t2.large	Low to Moderate	Not supported
t2.xlarge	Moderate	Not supported
t2.2xlarge	Moderate	Not supported
t3.nano †	Up to 5 Gigabit	ENA
t3.micro †	Up to 5 Gigabit	ENA
t3.small †	Up to 5 Gigabit	ENA
t3.medium †	Up to 5 Gigabit	ENA
t3.large †	Up to 5 Gigabit	ENA
t3.xlarge †	Up to 5 Gigabit	ENA
t3.2xlarge †	Up to 5 Gigabit	ENA
t3a.nano †	Up to 5 Gigabit	ENA
t3a.micro †	Up to 5 Gigabit	ENA
t3a.small †	Up to 5 Gigabit	ENA
t3a.medium †	Up to 5 Gigabit	ENA
t3a.large †	Up to 5 Gigabit	ENA
t3a.xlarge †	Up to 5 Gigabit	ENA
t3a.2xlarge †	Up to 5 Gigabit	ENA

For `32xlarge` and `metal` instance types that support 200 Gbps, at least 2 ENIs, each attached to a different network card, are required on the instance to achieve 200 Gbps throughput. Each ENI attached to a network card can achieve a max of 170 Gbps.

The following table shows the baseline and burst bandwidth for instance types that use the network I/O credit mechanism to burst beyond their baseline bandwidth.

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
`m5.large`	0.75	10.0
`m5.xlarge`	1.25	10.0
`m5.2xlarge`	2.5	10.0
`m5.4xlarge`	5.0	10.0
`m5a.large`	0.75	10.0
`m5a.xlarge`	1.25	10.0
`m5a.2xlarge`	2.5	10.0
`m5a.4xlarge`	5.0	10.0
`m5a.8xlarge`	7.5	10.0
`m5ad.large`	0.75	10.0
`m5ad.xlarge`	1.25	10.0
`m5ad.2xlarge`	2.5	10.0
`m5ad.4xlarge`	5.0	10.0
`m5ad.8xlarge`	7.5	10.0
`m5d.large`	0.75	10.0
`m5d.xlarge`	1.25	10.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
m5d.2xlarge	2.5	10.0
m5d.4xlarge	5.0	10.0
m5dn.large	2.1	25.0
m5dn.xlarge	4.1	25.0
m5dn.2xlarge	8.125	25.0
m5dn.4xlarge	16.25	25.0
m5n.large	2.1	25.0
m5n.xlarge	4.1	25.0
m5n.2xlarge	8.125	25.0
m5n.4xlarge	16.25	25.0
m5zn.large	3.0	25.0
m5zn.xlarge	5.0	25.0
m5zn.2xlarge	10.0	25.0
m5zn.3xlarge	15.0	25.0
m6a.large	0.781	12.5
m6a.xlarge	1.562	12.5
m6a.2xlarge	3.125	12.5
m6a.4xlarge	6.25	12.5
m6i.large	0.781	12.5
m6i.xlarge	1.562	12.5

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
m6i.2xlarge	3.125	12.5
m6i.4xlarge	6.25	12.5
m6id.large	0.781	12.5
m6id.xlarge	1.562	12.5
m6id.2xlarge	3.125	12.5
m6id.4xlarge	6.25	12.5
m6idn.large	3.125	25.0
m6idn.xlarge	6.25	30.0
m6idn.2xlarge	12.5	40.0
m6idn.4xlarge	25.0	50.0
m6in.large	3.125	25.0
m6in.xlarge	6.25	30.0
m6in.2xlarge	12.5	40.0
m6in.4xlarge	25.0	50.0
m7a.medium	0.39	12.5
m7a.large	0.781	12.5
m7a.xlarge	1.562	12.5
m7a.2xlarge	3.125	12.5
m7a.4xlarge	6.25	12.5
m7i.large	0.781	12.5

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
m7i.xlarge	1.562	12.5
m7i.2xlarge	3.125	12.5
m7i.4xlarge	6.25	12.5
m7i-flex.large	0.39	12.5
m7i-flex.xlarge	0.781	12.5
m7i-flex.2xlarge	1.562	12.5
m7i-flex.4xlarge	3.125	12.5
m7i-flex.8xlarge	6.25	12.5
t3.nano	0.032	5.0
t3.micro	0.064	5.0
t3.small	0.128	5.0
t3.medium	0.256	5.0
t3.large	0.512	5.0
t3.xlarge	1.024	5.0
t3.2xlarge	2.048	5.0
t3a.nano	0.032	5.0
t3a.micro	0.064	5.0
t3a.small	0.128	5.0
t3a.medium	0.256	5.0
t3a.large	0.512	5.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
t3a.xlarge	1.024	5.0
t3a.2xlarge	2.048	5.0

Amazon EBS I/O performance

Amazon EBS optimized instances use an optimized configuration stack and provide additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

For more information, see Amazon EBS–optimized instances.

SSD-based instance store volume I/O performance

If you use all the SSD-based instance store volumes available to your instance, you can get up to the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
m5ad.large	30000	15000
m5ad.xlarge	59000	29000
m5ad.2xlarge	117000	57000
m5ad.4xlarge	234000	114000
m5ad.8xlarge	466666	233334
m5ad.12xlarge	700000	340000
m5ad.16xlarge	933332	466668
m5ad.24xlarge	1400000	680000
m5d.large	30000	15000

Instance Size	100% Random Read IOPS	Write IOPS
m5d.xlarge	59000	29000
m5d.2xlarge	117000	57000
m5d.4xlarge	234000	114000
m5d.8xlarge	466666	233334
m5d.12xlarge	700000	340000
m5d.16xlarge	933332	466668
m5d.24xlarge	1400000	680000
m5d.metal	1400000	680000
m5dn.large	29000	14500
m5dn.xlarge	58000	29000
m5dn.2xlarge	116000	58000
m5dn.4xlarge	232000	116000
m5dn.8xlarge	464000	232000
m5dn.12xlarge	700000	350000
m5dn.16xlarge	930000	465000
m5dn.24xlarge	1400000	700000
m5dn.metal	1400000	700000
m6id.large	33542	16771
m6id.xlarge	67083	33542
m6id.2xlarge	134167	67084

Instance Size	100% Random Read IOPS	Write IOPS
m6id.4xlarge	268333	134167
m6id.8xlarge	536666	268334
m6id.12xlarge	804998	402500
m6id.16xlarge	1073332	536668
m6id.24xlarge	1609996	805000
m6id.32xlarge	2146664	1073336
m6id.metal	2146664	1073336
m6idn.large	33542	16771
m6idn.xlarge	67083	33542
m6idn.2xlarge	134167	67084
m6idn.4xlarge	268333	134167
m6idn.8xlarge	536666	268334
m6idn.12xlarge	804998	402500
m6idn.16xlarge	1073332	536668
m6idn.24xlarge	1609996	805000
m6idn.32xlarge	2146664	1073336
m6idn.metal	2146664	1073336

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

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

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

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support.

Release notes

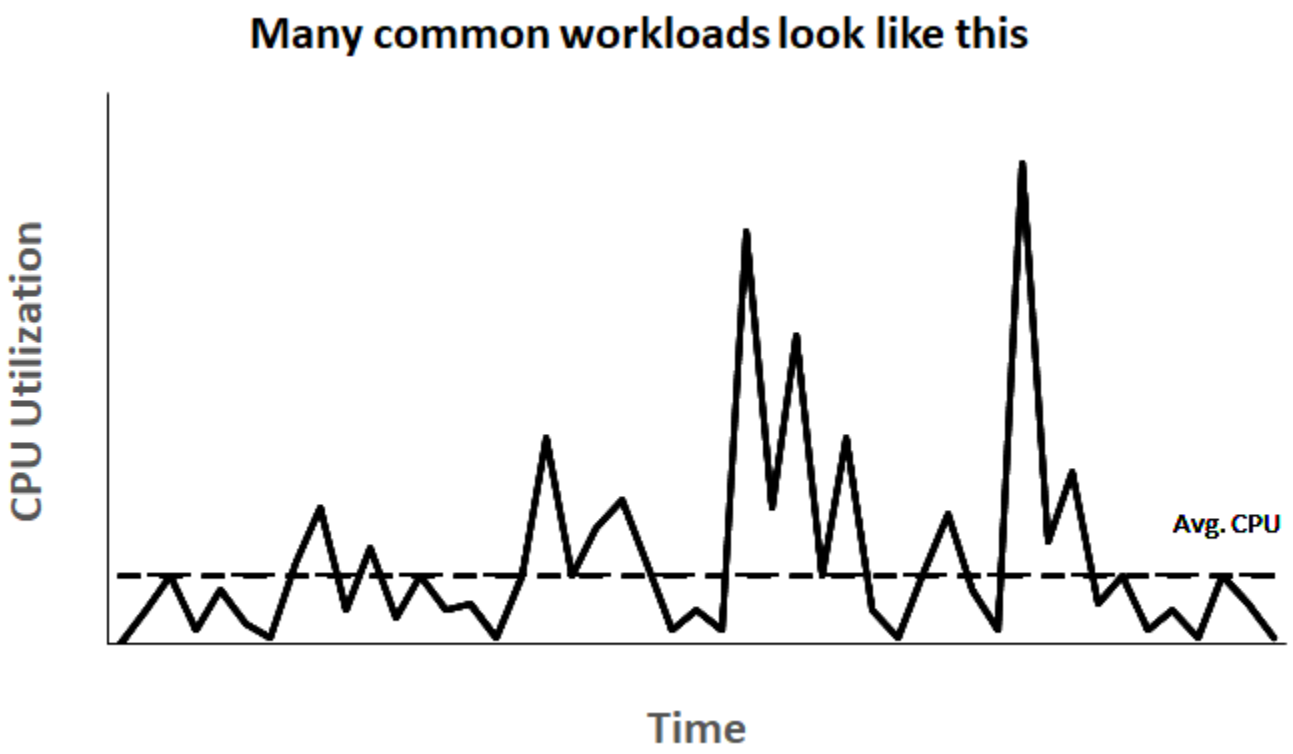
- `m6a.metal` instances do not support Windows Server 2016 and earlier.
- `m7a.metal-48xl` instances do not support Windows Server 2019 AMIs released before July 2023, and Windows Server 2016 AMIs.
- Instances built on the Nitro System, M4, `t2.large` and larger, `t3.large` and larger, and `t3a.large` and larger require 64-bit HVM AMIs. They have high-memory, and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- Instances built on the Nitro System have the following requirements:
  - NVMe drivers must be installed
  - Elastic Network Adapter (ENA) drivers must be installed

  The current AWS Windows AMIs meet these requirements.
- To get the best performance from your M6i instances, ensure that they have ENA driver version 2.2.3 or later. Using an ENA driver earlier than version 2.0.0 with these instances causes network interface attachment failures. The following AMIs have a compatible ENA driver.

  - AWS Windows AMI from May 2021 or later
- 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.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

Burstable performance instances

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



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 T3a and T3 instance types, and the previous generation T2 instance types.

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

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

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

Type	Description	Processor family
T2	Previous generation burstable instances	Intel Xeon processors

For information about instance pricing and additional specifications, see Amazon EC2 Pricing and Amazon EC2 Instance Types. For information about burstable network performance, see Amazon EC2 instance network bandwidth.

If your 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*.
- For additional requirements, see Release notes.

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: Baseline utilization % = (number of credits earned/number of vCPUs)/60 minutes.

For the baseline utilization of each burstable performance instance type, see the credit table.

Earned credits

Credits earned continuously by an instance when it is running.

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

Example:

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

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

Spent or used credits

Credits used continuously by an instance when it is running.

CPU credits spent per minute = Number of vCPUs * CPU utilization * 1 minute

Accrued credits

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

Example:

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

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

Credit accrual limit

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

Example:

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

Launch credits

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

Surplus credits

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

Standard mode

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

Unlimited mode

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

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

Type	Type of CPU credits supported	Credit configuration modes	Accrued CPU credits lifespan between instance starts and stops
Latest generation			
T4g	Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)	Standard, Unlimited (default)	7 days (credits persist for 7 days after an instance stops)
T3a	Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)	Standard, Unlimited (default)	7 days (credits persist for 7 days after an instance stops)
T3	Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)	Standard, Unlimited (default)	7 days (credits persist for 7 days after an instance stops)
Previous generation			
T2	Earned credits, Accrued credits, Spent credits, Launch credits (Standard mode only), Surplus credits (Unlimited mode only)	Standard (default), Unlimited	0 days (credits are lost when an instance stops)

> **Note**
>
> Unlimited mode is not supported for T3 instances that are launched on a Dedicated Host.

Earn CPU credits

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

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

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

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

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

* 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.

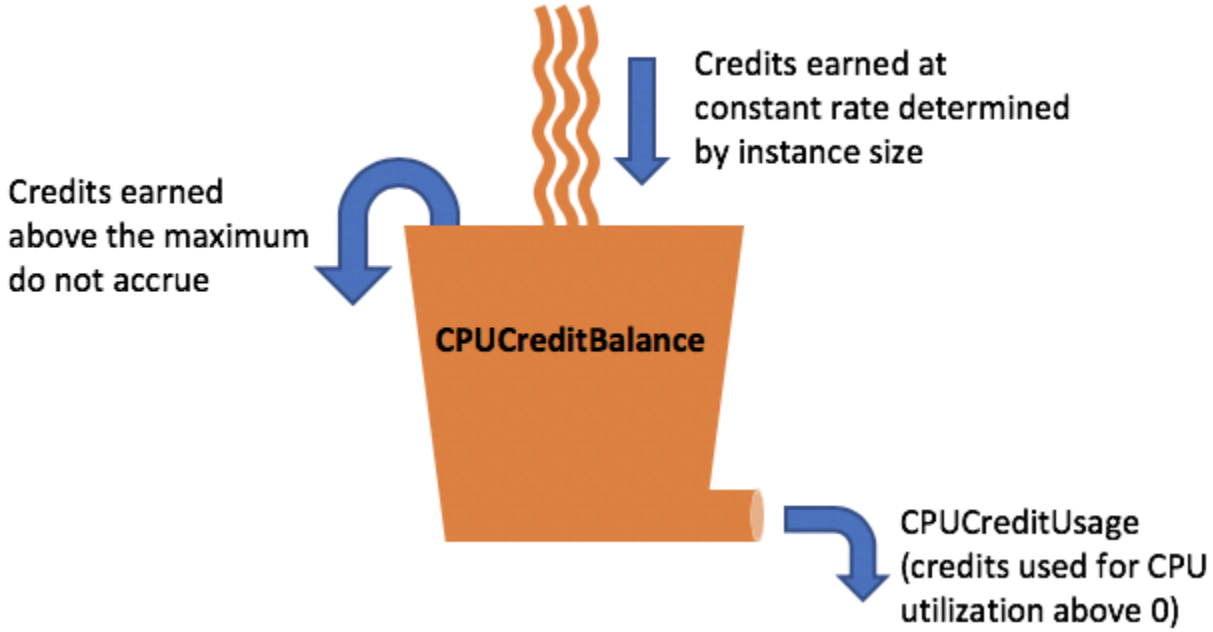
*** Each vCPU is a thread of either an Intel Xeon core or an AMD EPYC core, except for T2 instances.

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.

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 T3a and T3, the CPU credit balance persists for seven days after an instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

For more information, see `CPUCreditBalance` in the CloudWatch metrics table.

Baseline utilization

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

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

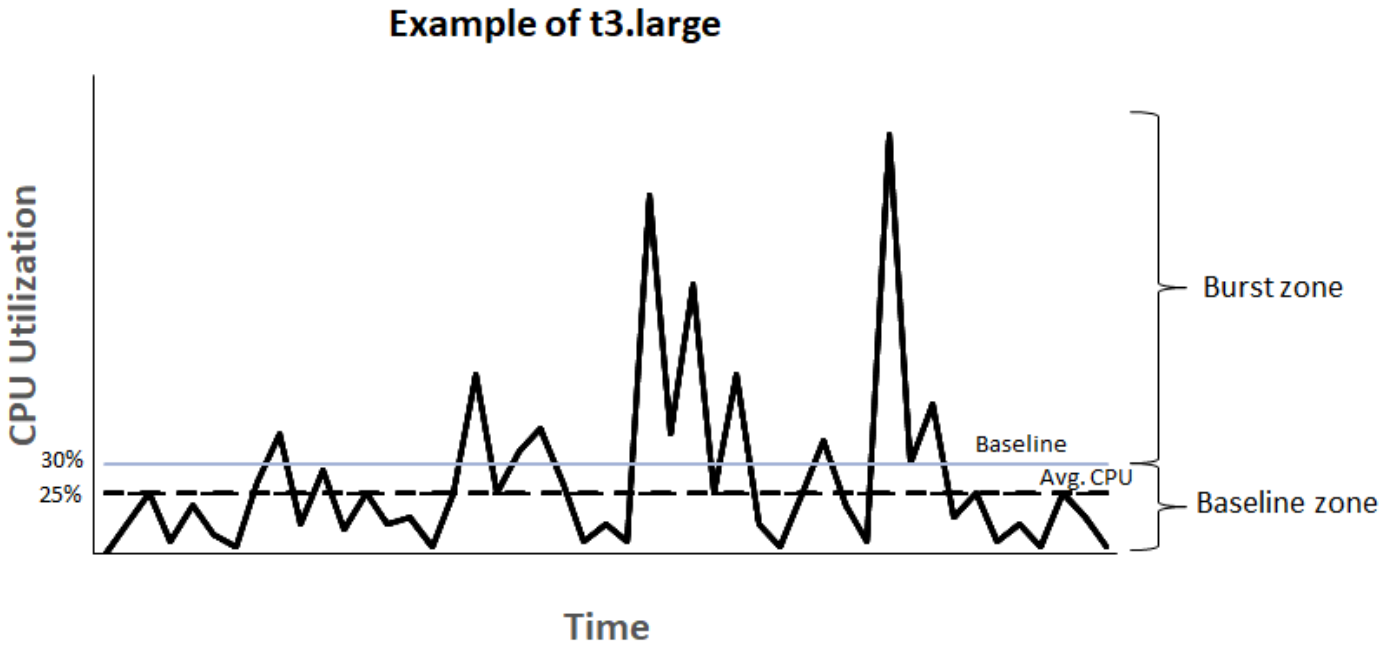
```
(number of credits earned/number of vCPUs)/60 minutes = % baseline
utilization
```

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

```
(6 credits earned/2 vCPUs)/60 minutes = 5% baseline utilization
```

A `t3.large` instance, with 2 vCPUs, earns 36 credits per hour, resulting in a baseline utilization of 30% (`(36/2)/60`).

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



Unlimited mode for burstable performance instances

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

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

If 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.

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

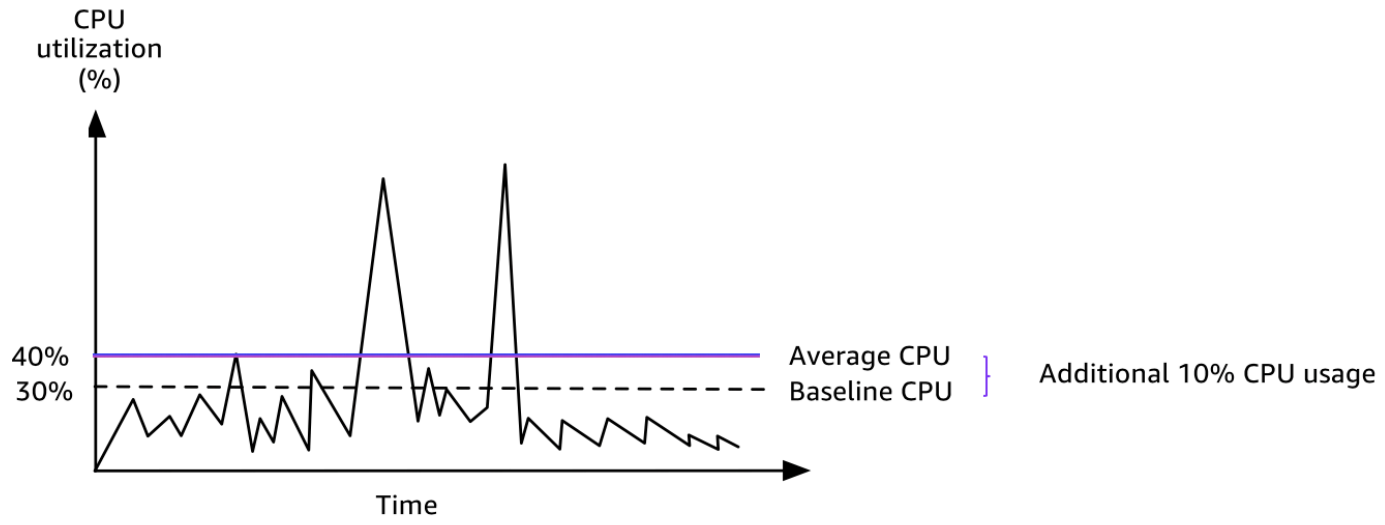
Unlimited mode concepts

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.

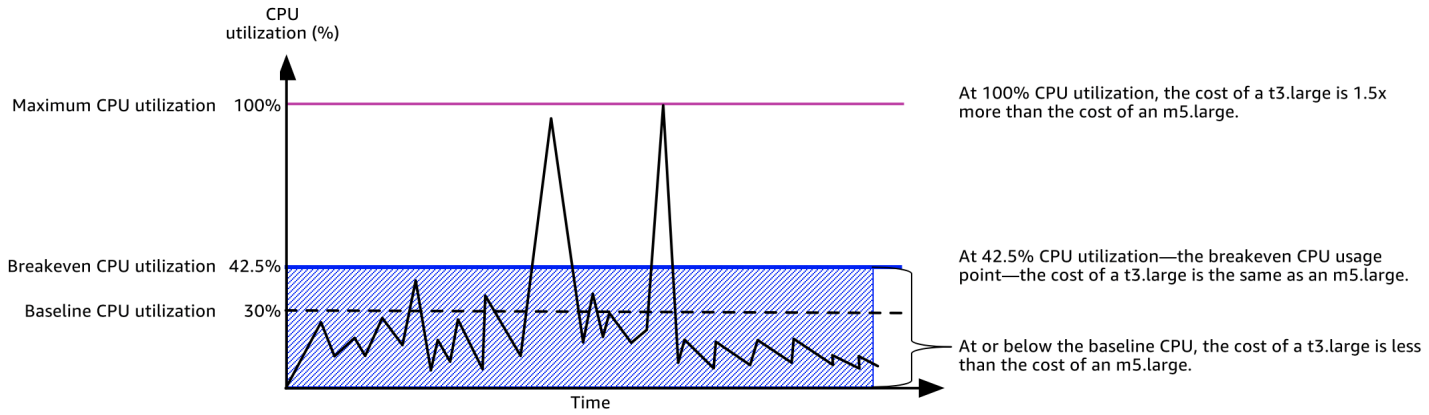
When to use unlimited mode versus fixed CPU

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	Additional burst minutes available per vCPU	Additional CPU % available	Breakeven CPU %
A	B	C	D	E = D - C	F	G	H = G / 60	I = E / H	J = (I / 60) / B	K = F + J
t3.large	2	$0.0835	$0.096	$0.0125	30%	$0.05	$0.00083	15	12.5%	42.5%

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

The table provides the following information:

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

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

T3 instance type	Breakeven CPU usage (in %) for T3 compared to M5
`t3.large`	42.5%
`t3.xlarge`	52.5%
`t3.2xlarge`	52.5%

Surplus credits can incur charges

If the average CPU utilization of an instance is at or below the baseline, the instance incurs no additional charges. Because an instance earns a maximum number of credits in a 24-hour period

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

However, if CPU utilization stays above the baseline, the instance cannot earn enough credits to pay down the surplus credits that it has spent. The surplus credits that are not paid down are charged at a flat additional rate per vCPU-hour. For information about the rate, see T2/T3/T4g Unlimited Mode Pricing.

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

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

Spent surplus credits are tracked by the CloudWatch metric `CPUSurplusCreditBalance`. Surplus credits that are charged are tracked by the CloudWatch metric `CPUSurplusCreditsCharged`. For more information, see Additional CloudWatch metrics for burstable performance instances.

No launch credits for T2 Unlimited instances

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

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

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 `CPUSurplusCreditBalance` value is immediately charged.

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

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

Monitor credit usage

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

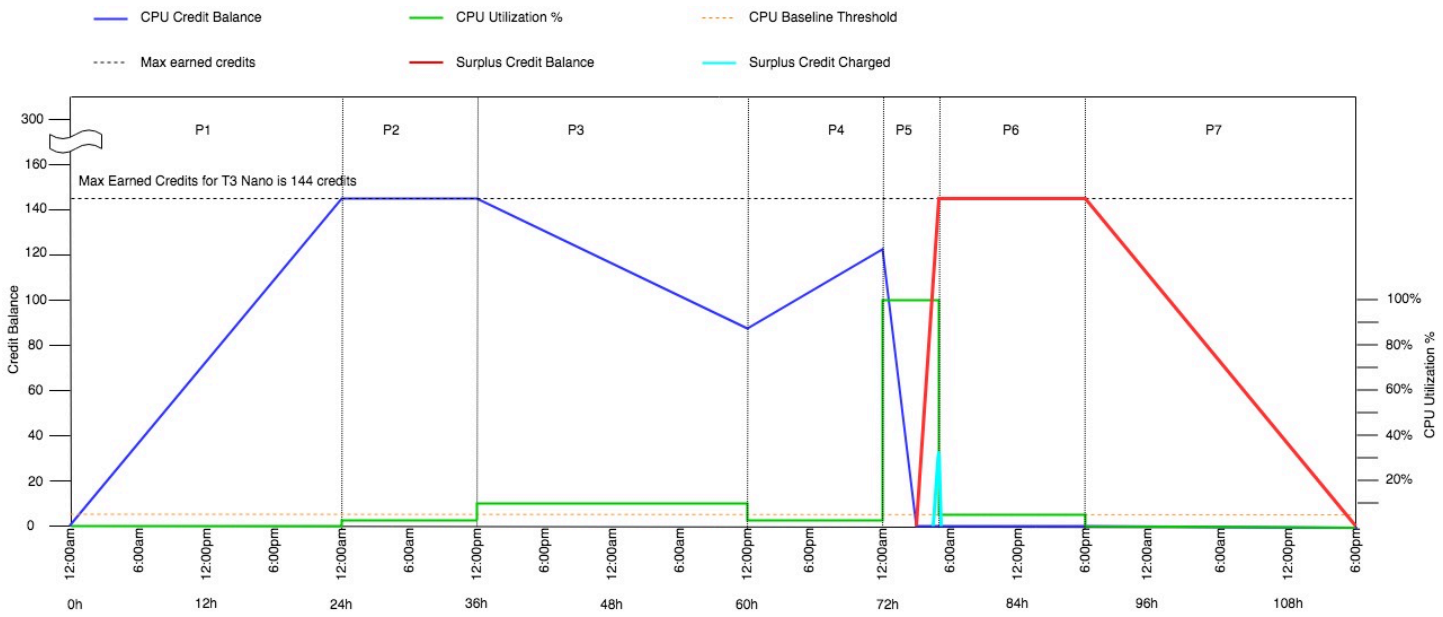
P4 – For the next 12 hours, CPU utilization decreases to 2.5% (below the baseline), which requires a spend of 36 credits. In the same time, the instance earns 72 credits. The instance earns more credits than it spends, and the `CPUCreditBalance` value increases to 122 credits.

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:

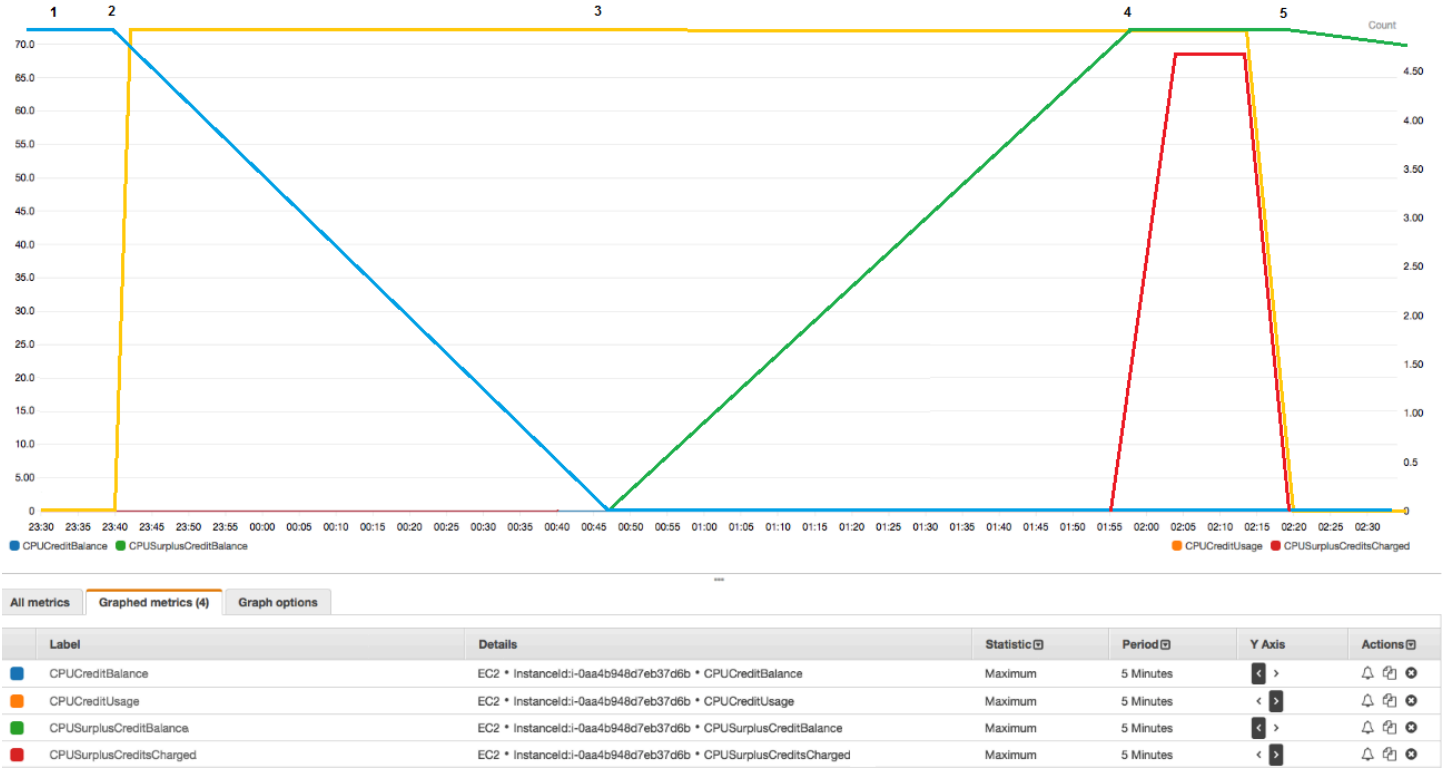
1 – In the first 10 minutes, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value remains at its maximum of 72.

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 `CPUSurplusCreditBalance` value reaches 72 CPU credits. This is equal to the maximum a `t2.nano` instance can earn in a 24-hour period. Any surplus credits spent thereafter cannot be offset by earned credits within the 24-hour period, which results in a small additional charge at the end of the hour.

5 – The instance continues to spend surplus credits until around 02:20. At this time, CPU utilization falls below the baseline, and the instance starts to earn credits at 3 credits per hour (or 0.25 credits every 5 minutes), which it uses to pay down the `CPUSurplusCreditBalance`. After the `CPUSurplusCreditBalance` value reduces to 0, the instance starts to accrue earned credits in its `CPUCreditBalance` at 0.25 credits every 5 minutes.



Calculating the bill

Surplus credits cost $0.096 per vCPU-hour. The instance spent approximately 25 surplus credits between 01:55 and 02:20, which is equivalent to 0.42 vCPU-hours.

Additional charges for this instance are 0.42 vCPU-hours x $0.096/vCPU-hour = $0.04032, rounded to $0.04.

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

Amazon Elastic Compute Cloud running Windows		
$0.0081 per On Demand Windows t2.nano Instance Hour	720.000 Hrs	$5.83

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.096 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.04

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

Standard mode for burstable performance instances

A burstable performance instance configured as `standard` is suited to workloads with an average CPU utilization that is consistently below the baseline CPU utilization of the instance. To burst above the baseline, the instance spends credits that it has accrued in its CPU credit balance. If the instance is running low on accrued credits, CPU utilization is gradually lowered to the baseline level, so that the instance does not experience a sharp performance drop-off when its accrued CPU credit balance is depleted. For more information, see Key concepts 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 T3a and T3 Standard instances, the CPU credit balance persists for seven days after the instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

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

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

Launch credits

T2 Standard instances get 30 launch credits per vCPU at launch or start, and T1 Standard instances get 15 launch credits. For example, a `t2.micro` instance has one vCPU and gets 30 launch credits, while a `t2.xlarge` instance has four vCPUs and gets 120 launch credits. Launch credits are designed to provide a good startup experience to allow instances to burst immediately after launch before they have accrued earned credits.

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

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

Instance type	Launch credits	vCPUs
`t1.micro`	15	1
`t2.nano`	30	1
`t2.micro`	30	1
`t2.small`	30	1
`t2.medium`	60	2
`t2.large`	60	2
`t2.xlarge`	120	4
`t2.2xlarge`	240	8

Launch credit limits

There is a limit to the number of times T2 Standard instances can receive launch credits. The default limit is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. For example, the limit is reached when one instance is stopped and started 100 times within a 24-hour period, or when 100 instances are launched within a 24-hour period, or other combinations that equate to 100 starts. New accounts may have a lower limit, which increases over time based on your usage.

> **Tip**
>
> To ensure that your workloads always get the performance they need, switch to Unlimited mode for burstable performance instances or consider using a larger instance size.

Differences between launch credits and earned credits

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

	Launch credits	Earned credits
Credit earn rate	T2 Standard instances get 30 launch credits per vCPU at launch or start. If a T2 instance is switched from `unlimited` to `standard`, it does not get launch credits at the time of switching.	Each T2 instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. For more information about the number of CPU credits earned per instance size, see the credit table.
Credit earn limit	The limit for receiving launch credits is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. New accounts may have a lower limit, which increases over time based on your usage.	A T2 instance cannot accrue more credits than the CPU credit balance limit. If the CPU credit balance has reached its limit, any credits that are earned after the limit is reached are discarded. Launch credits do not count towards the limit. For more information about the CPU credit

	Launch credits	Earned credits
		balance limit for each T2 instance size, see the credit table.
Credit use	Launch credits are spent first, before earned credits.	Earned credits are spent only after all launch credits are spent.
Credit expiration	When a T2 Standard instance is running, launch credits do not expire. When a T2 Standard instance stops or is switched to T2 Unlimited, all launch credits are lost.	When a T2 instance is running, earned credits that have accrued do not expire. When the T2 instance stops, all accrued earned credits are lost.

The number of accrued launch credits and accrued earned credits is tracked by the CloudWatch metric `CPUCreditBalance`. For more information, see `CPUCreditBalance` in the CloudWatch metrics table.

Standard mode examples

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

Examples

- Example 1: Explain credit use with T3 Standard
- Example 2: Explain credit use with T2 Standard

Example 1: Explain credit use with T3 Standard

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

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

You might launch a T3 Standard instance and use it immediately. Or, you might launch a T3 Standard instance and leave it idle for a few days before running applications on it. Whether an instance is used or remains idle determines if credits are spent or accrued. If an instance remains

idle for 24 hours from the time it is launched, the credit balance reaches it limit, which is the maximum number of earned credits that can be accrued.

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

The following workflow references the numbered points on the graph:

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

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

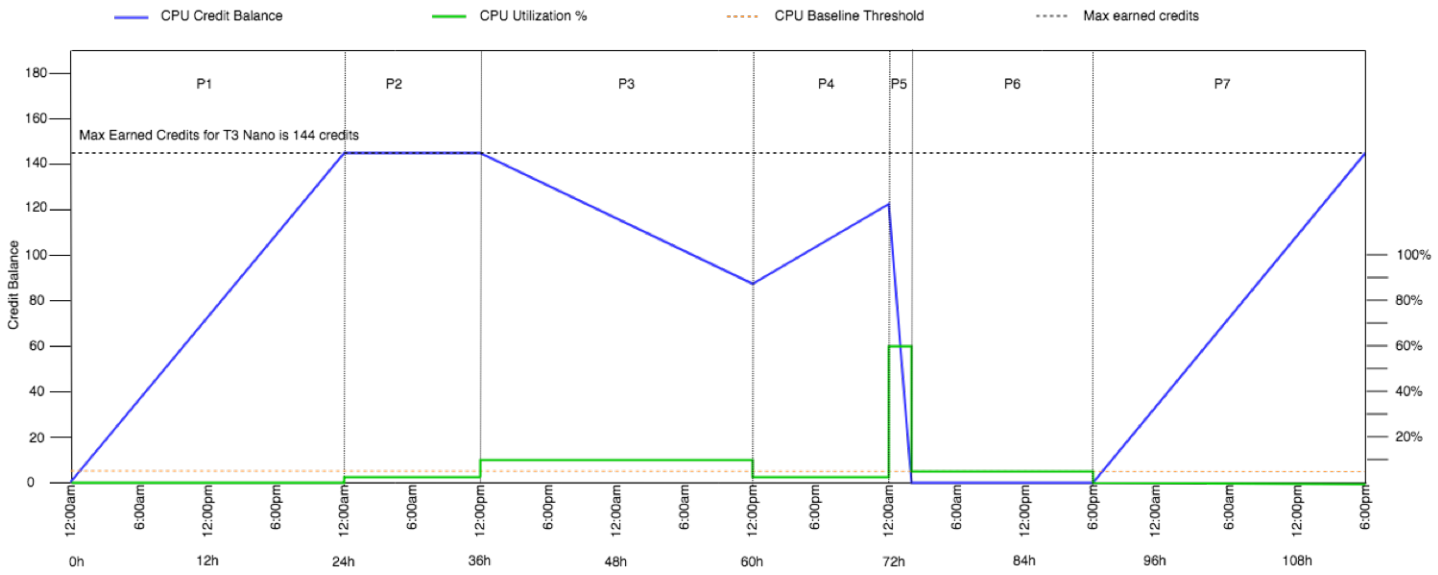
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

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

P5 – For the next two hours, the instance bursts at 60% CPU utilization, and depletes its entire `CPUCreditBalance` value of 122 credits. At the end of this period, with the `CPUCreditBalance` at zero, CPU utilization is forced to drop to the baseline utilization level of 5%. At the baseline, the instance earns as many credits as it spends.

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

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



Example 2: Explain credit use with T2 Standard

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

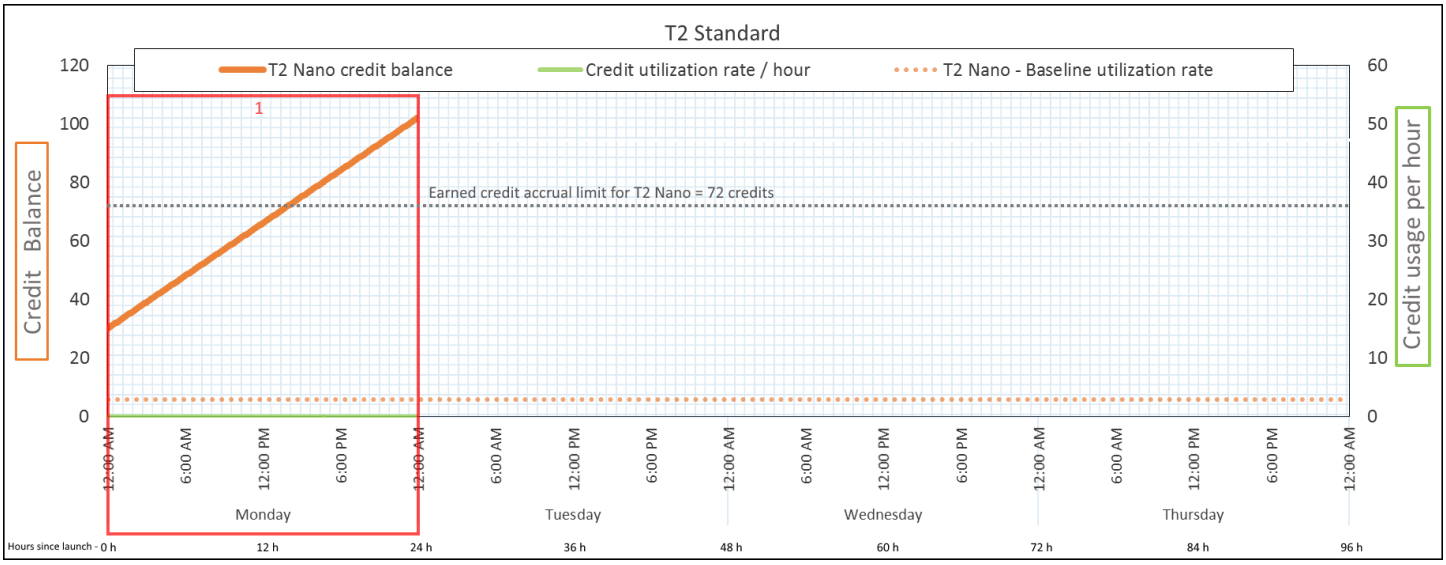
A `t2.nano` instance gets 30 launch credits when it is launched, and earns 72 credits every 24 hours. Its credit balance limit is 72 earned credits; launch credits do not count towards the limit. After the limit is reached, new credits that are earned are discarded. For more information about the number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

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

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

Period 1: 1 – 24 hours

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



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

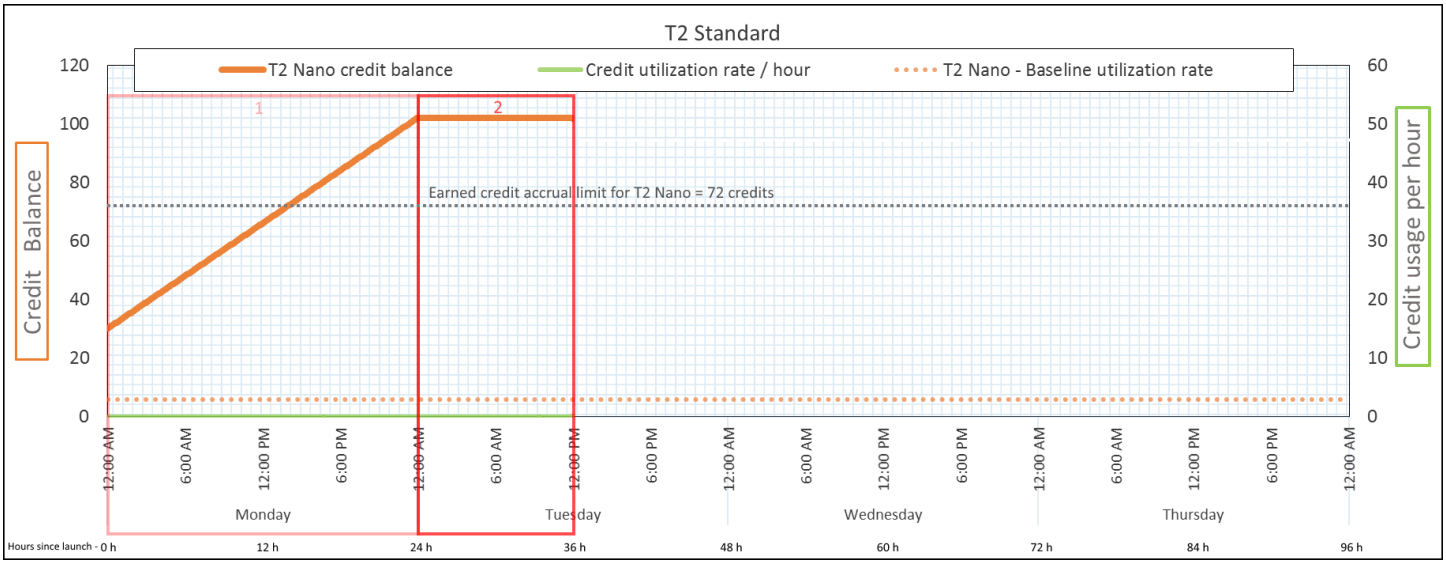
Conclusion

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

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

Period 2: 25 – 36 hours

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



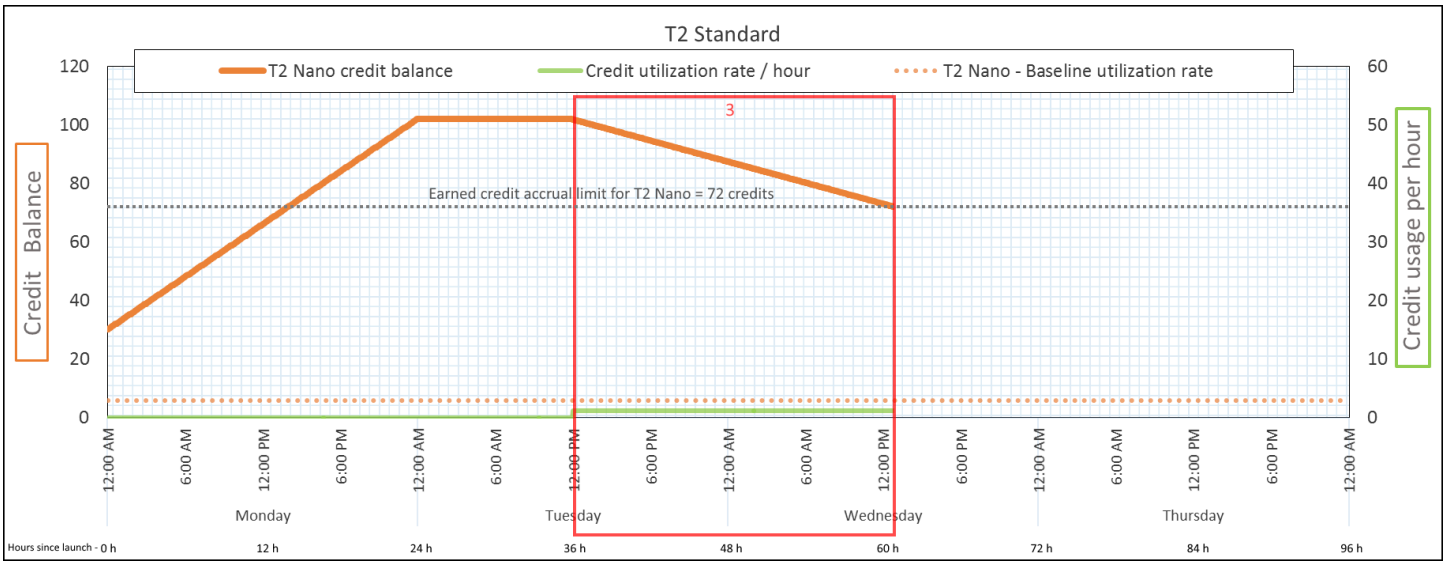
Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours (3 credits per hour)
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	102 credits (30 launch credits + 72 earned credits)—balance is unchanged

Conclusion

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

Period 3: 37 – 61 hours

For the next 25 hours, the instance uses 2% CPU, which requires 30 credits. In the same period, it earns 75 credits, but the credit balance decreases. The balance decreases because the accrued *launch* credits are spent first, while newly earned credits are discarded because the credit balance is already at its limit of 72 earned credits.



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

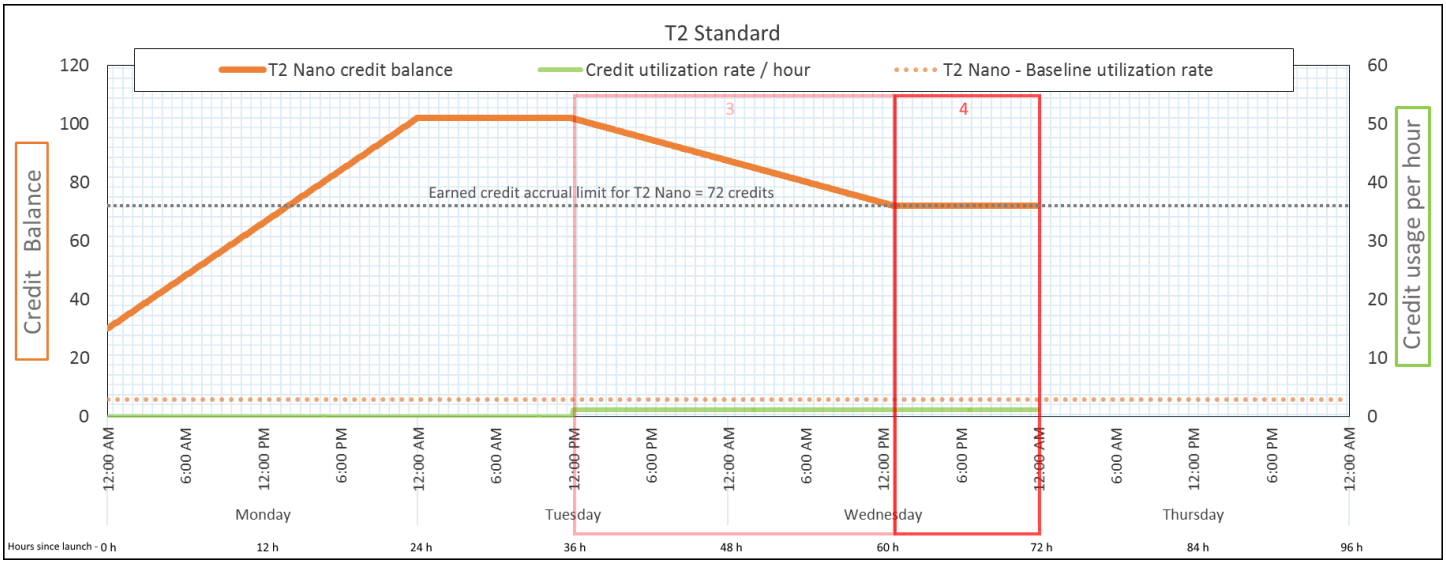
Conclusion

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

Period 4: 62 – 72 hours

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

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



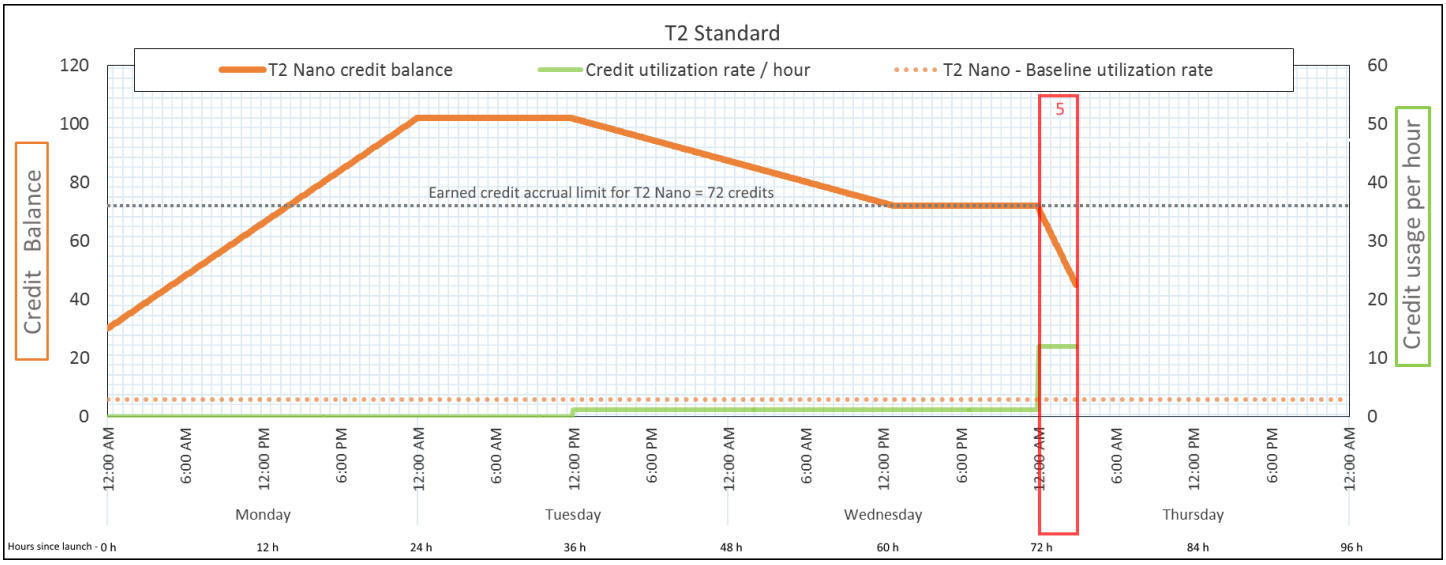
Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—13.2 credits over 11 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	43.2 credits per 24 hours (60% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned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

Conclusion

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

Period 5: 73 – 75 hours

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



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

Conclusion

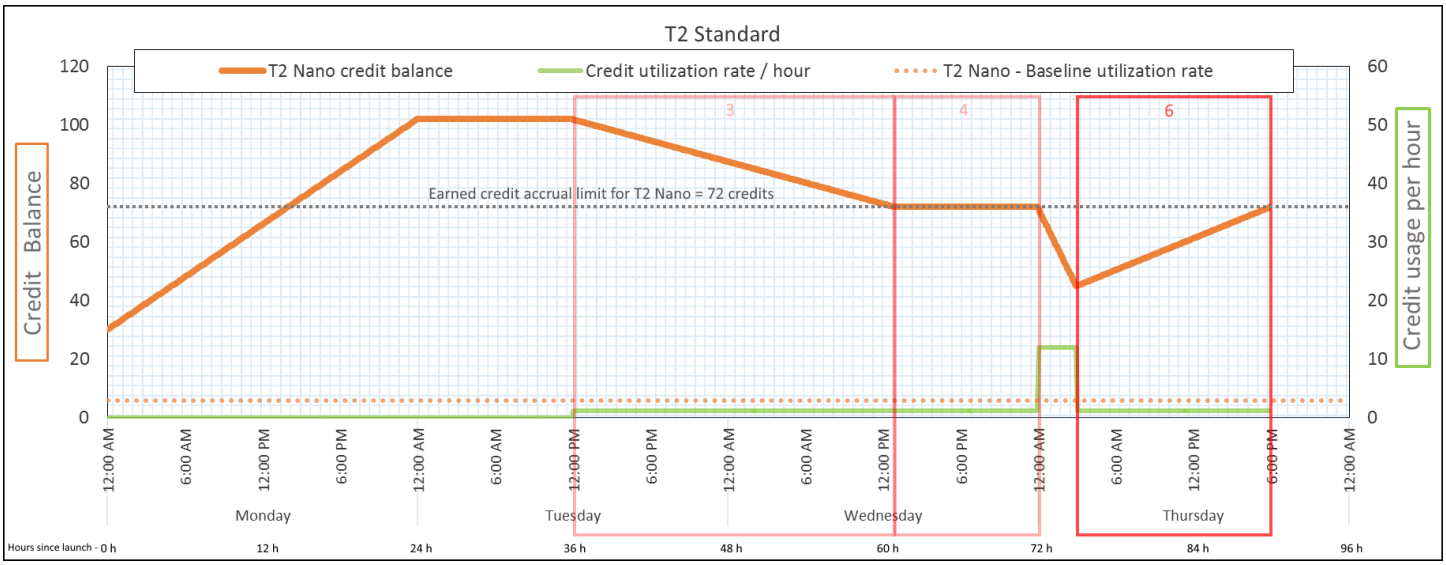
If an instance spends more credits than it earns, its credit balance decreases.

Period 6: 76 – 90 hours

For the next 15 hours, the instance uses 2% CPU, which requires 18 credits. This is the same CPU utilization as in Periods 3 and 4. However, the balance increases in this period, whereas it decreased in Period 3 and plateaued in Period 4.

In Period 3, the accrued launch credits were spent, and any earned credits that exceeded the credit limit were discarded, resulting in a decrease in the credit balance. In Period 4, the instance spent fewer credits than it earned. Any earned credits that exceeded the limit were discarded, so the balance plateaued at its maximum of 72 credits.

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



Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—18 credits over 15 hours
Credit Earn Rate	72 credits per 24 hours (45 credits over 15 hours)
Credit Discard Rate	0 credits per 24 hours

Credit Balance	72 credits (balance increases at a rate of 43.2 credits per 24 hours—change rate = spend rate 28.8/24 + earn rate 72/24)

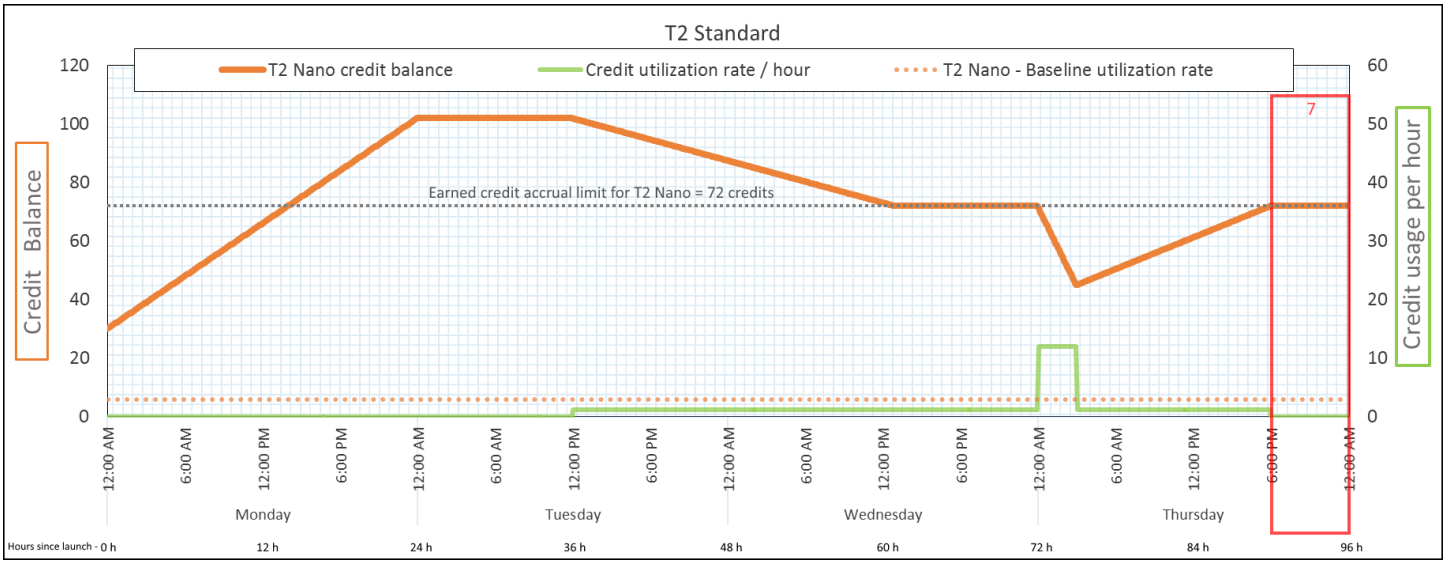
Conclusion

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

Period 7: 91 – 96 hours

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

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



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

Conclusion

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

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*:

- 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 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 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 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
	utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes). CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the `Sum` statistic instead of the `Average` statistic. Units: Credits (vCPU-minutes)
`CPUCreditBalance`	The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the `CPUCreditBalance` also includes the number of launch credits that have been accrued. Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits do not count towards the limit. The credits in the `CPUCreditBalance` are available for the instance to spend to burst beyond its baseline CPU utilization. When an instance is running, credits in the `CPUCreditBalance` do not expire. When a T3a or T3 instance stops, the `CPUCreditBalance` value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the `CPUCreditBalance` value does not persist, and all accrued credits are lost. CPU credit metrics are available at a five-minute frequency only. Units: Credits (vCPU-minutes)

Metric	Description
`CPUSurplusCreditBalance`	The number of surplus credits that have been spent by an `unlimited` instance when its `CPUCreditBalance` value is zero. The `CPUSurplusCreditBalance` value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge. Units: Credits (vCPU-minutes)
`CPUSurplusCreditsCharged`	The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge. Spent surplus credits are charged when any of the following occurs: • The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. • The instance is stopped or terminated. • The instance is switched from `unlimited` to `standard`. Units: Credits (vCPU-minutes)

Calculate CPU credit usage

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

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

Calculate CPU credit usage for Standard instances

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

Mathematically, this is captured by the following equation:

Example

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

The size of the instance determines the number of credits that the instance can earn per hour and the number of earned credits that it can accrue in the credit balance. For information about the number of credits earned per hour, and the credit balance limit for each instance size, see the credit table.

Example

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

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

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

Example

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

Calculate CPU credit usage for Unlimited instances

When a burstable performance instance needs to burst above the baseline, it always spends accrued credits before spending surplus credits. When it depletes its accrued CPU credit balance, it can spend surplus credits to burst CPU for as long as it needs. When CPU utilization falls below the baseline, surplus credits are always paid down before the instance accrues earned credits.

We use the term `Adjusted balance` in the following equations to reflect the activity that occurs in this five-minute interval. We use this value to arrive at the values for the `CPUCreditBalance` and `CPUSurplusCreditBalance` CloudWatch metrics.

Example

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

A value of `0` for `Adjusted balance` indicates that the instance spent all its earned credits for bursting, and no surplus credits were spent. As a result, both `CPUCreditBalance` and `CPUSurplusCreditBalance` are set to `0`.

A positive `Adjusted balance` value indicates that the instance accrued earned credits, and previous surplus credits, if any, were paid down. As a result, the `Adjusted balance` value is assigned to `CPUCreditBalance`, and the `CPUSurplusCreditBalance` is set to `0`. The instance size determines the maximum number of credits that it can accrue.

Example

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

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

Example

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

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

Example

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

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

Compute optimized instances

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Compute optimized instances are ideal for compute-bound applications that benefit from high-performance processors.

C5 and C5n instances

These instances are well suited for the following:

- Batch processing workloads
- Media transcoding
- High-performance web servers
- High-performance computing (HPC)
- Scientific modeling
- Dedicated gaming servers and ad serving engines

- Machine learning inference and other compute-intensive applications

Bare metal instances, such as `c5.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 C5 Instances.

C6i and C6id instances

These instances are ideal for running advanced, compute-intensive workloads, such as the following:

- High-performance computing (HPC)
- Batch processing
- Ad serving
- Video encoding
- Distributed analytics
- Highly scalable multiplayer gaming

C6in instances

These instances are well suited for compute-intensive workloads such as the following:

- Distributed computing applications
- Network virtual appliances
- Data analytics
- High Performance Computing (HPC)
- CPU-based AI/ML

For more information, see Amazon EC2 C6i Instances.

C7a instances

These instances are powered by 4th generation AMD EPYC processors and are ideal for running advanced, compute-intensive workloads, such as the following:

- High-performance computing (HPC)

- Batch processing
- Ad serving
- Video encoding
- Gaming servers
- Scientific modeling
- Distributed analytics

For more information, see Amazon EC2 C7a instances.

C7g and C7gd instances

These instances are powered by AWS Graviton3 processors and are ideal for running advanced, compute-intensive workloads, such as the following:

- High-performance computing (HPC)
- Batch processing
- Ad serving
- Video encoding
- Gaming servers
- Scientific modeling
- Distributed analytics

For more information, see Amazon EC2 C7g instances.

C7gn instances

Featuring the new AWS Nitro Cards, C7gn instances deliver the highest network bandwidth, and the best packet-processing performance for Graviton-based Amazon EC2 instances. C7gn instances offer up to 200 Gbps network bandwidth and up to 50 percent higher packet-processing performance compared to previous generation C6gn instances. C7gn instances are ideal for network-intensive workloads, including:

- Network virtual appliance workloads
- Data-intensive workloads, such as data analytics
- CPU-based artificial intelligence and machine learning (AI/ML) inference workloads

For more information, see Amazon EC2 C7g instances.

C7i instances

C7i instances are ideal for running compute intensive workloads, such as batch processing, machine learning, high end gaming, ad serving, and video encoding.

For more information, see Amazon EC2 C7i instances.

Contents

- Hardware specifications
- Instance performance
- Network performance
- Amazon EBS I/O performance
- SSD-based instance store volume I/O performance
- Release notes

Hardware specifications

The following is a summary of the hardware specifications for compute optimized instances. A virtual central processing unit (vCPU) represents a portion of the physical CPU assigned to a virtual machine (VM). For x86 instances, there are two vCPUs per core. For Graviton instances, there is one vCPU per core.

Instance type	Default vCPUs	Memory (GiB)
c1.medium	2	1.70
c1.xlarge	8	7.00
c3.large	2	3.75
c3.xlarge	4	7.50
c3.2xlarge	8	15.00
c3.4xlarge	16	30.00
c3.8xlarge	32	60.00

Instance type	Default vCPUs	Memory (GiB)
c4.large	2	3.75
c4.xlarge	4	7.50
c4.2xlarge	8	15.00
c4.4xlarge	16	30.00
c4.8xlarge	36	60.00
c5.large	2	4.00
c5.xlarge	4	8.00
c5.2xlarge	8	16.00
c5.4xlarge	16	32.00
c5.9xlarge	36	72.00
c5.12xlarge	48	96.00
c5.18xlarge	72	144.00
c5.24xlarge	96	192.00
c5.metal	96	192.00
c5a.large	2	4.00
c5a.xlarge	4	8.00
c5a.2xlarge	8	16.00
c5a.4xlarge	16	32.00
c5a.8xlarge	32	64.00
c5a.12xlarge	48	96.00

Instance type	Default vCPUs	Memory (GiB)
c5a.16xlarge	64	128.00
c5a.24xlarge	96	192.00
c5ad.large	2	4.00
c5ad.xlarge	4	8.00
c5ad.2xlarge	8	16.00
c5ad.4xlarge	16	32.00
c5ad.8xlarge	32	64.00
c5ad.12xlarge	48	96.00
c5ad.16xlarge	64	128.00
c5ad.24xlarge	96	192.00
c5d.large	2	4.00
c5d.xlarge	4	8.00
c5d.2xlarge	8	16.00
c5d.4xlarge	16	32.00
c5d.9xlarge	36	72.00
c5d.12xlarge	48	96.00
c5d.18xlarge	72	144.00
c5d.24xlarge	96	192.00
c5d.metal	96	192.00
c5n.large	2	5.25

Instance type	Default vCPUs	Memory (GiB)
c5n.xlarge	4	10.50
c5n.2xlarge	8	21.00
c5n.4xlarge	16	42.00
c5n.9xlarge	36	96.00
c5n.18xlarge	72	192.00
c5n.metal	72	192.00
c6a.large	2	4.00
c6a.xlarge	4	8.00
c6a.2xlarge	8	16.00
c6a.4xlarge	16	32.00
c6a.8xlarge	32	64.00
c6a.12xlarge	48	96.00
c6a.16xlarge	64	128.00
c6a.24xlarge	96	192.00
c6a.32xlarge	128	256.00
c6a.48xlarge	192	384.00
c6a.metal	192	384.00
c6i.large	2	4.00
c6i.xlarge	4	8.00
c6i.2xlarge	8	16.00

Instance type	Default vCPUs	Memory (GiB)
c6i.4xlarge	16	32.00
c6i.8xlarge	32	64.00
c6i.12xlarge	48	96.00
c6i.16xlarge	64	128.00
c6i.24xlarge	96	192.00
c6i.32xlarge	128	256.00
c6i.metal	128	256.00
c6id.large	2	4.00
c6id.xlarge	4	8.00
c6id.2xlarge	8	16.00
c6id.4xlarge	16	32.00
c6id.8xlarge	32	64.00
c6id.12xlarge	48	96.00
c6id.16xlarge	64	128.00
c6id.24xlarge	96	192.00
c6id.32xlarge	128	256.00
c6id.metal	128	256.00
c6in.large	2	4.00
c6in.xlarge	4	8.00
c6in.2xlarge	8	16.00

Instance type	Default vCPUs	Memory (GiB)
c6in.4xlarge	16	32.00
c6in.8xlarge	32	64.00
c6in.12xlarge	48	96.00
c6in.16xlarge	64	128.00
c6in.24xlarge	96	192.00
c6in.32xlarge	128	256.00
c6in.metal	128	256.00
c7a.medium	1	2.00
c7a.large	2	4.00
c7a.xlarge	4	8.00
c7a.2xlarge	8	16.00
c7a.4xlarge	16	32.00
c7a.8xlarge	32	64.00
c7a.12xlarge	48	96.00
c7a.16xlarge	64	128.00
c7a.24xlarge	96	192.00
c7a.32xlarge	128	256.00
c7a.48xlarge	192	384.00
c7a.metal-48xl	192	384.00
c7i.large	2	4.00

Instance type	Default vCPUs	Memory (GiB)
c7i.xlarge	4	8.00
c7i.2xlarge	8	16.00
c7i.4xlarge	16	32.00
c7i.8xlarge	32	64.00
c7i.12xlarge	48	96.00
c7i.16xlarge	64	128.00
c7i.24xlarge	96	192.00
c7i.48xlarge	192	384.00
c7i.metal-24xl	96	192.00
c7i.metal-48xl	192	384.00

The compute optimized instances use the following processors.

AMD processors

- **2nd generation AMD EPYC processors (AMD EPYC 7R32)**: C5a, C5ad
- **3rd generation AMD EPYC processors (AMD EPYC 7R13)**: C6a
- **4th generation AMD EPYC processors (AMD EPYC 9R14)**: C7a

Intel processors

- **Intel Xeon Scalable processors (Haswell E5-2666 v3)**: C4
- **Intel Xeon Scalable processors (Skylake 8124)**: C5n
- **Intel Xeon Scalable processors (Skylake 8124M or Cascade Lake 8223CL)**: Smaller C5 and C5d
- **2nd generation Intel Xeon Scalable processors (Cascade Lake 8275CL)**: Larger C5 and C5d
- **3rd generation Intel Xeon Scalable processors (Ice Lake 8375C)**: C6i, C6id
- **4th generation Intel Xeon Scalable processors (Sapphire Rapids 8488C)**: C7i

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Instance performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some compute optimized instances are EBS-optimized by default at no additional cost. For more information, see Amazon EBS–optimized instances.

Network performance

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

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

> **Note**
>
> Instance types indicated with a † have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth.

Instance type	Network performance	Enhanced networking features
`c1.medium`	Moderate	Not supported
`c1.xlarge`	High	Not supported
`c3.large`	Moderate	Intel 82599 VF
`c3.xlarge`	Moderate	Intel 82599 VF
`c3.2xlarge`	High	Intel 82599 VF

Instance type	Network performance	Enhanced networking features
c3.4xlarge	High	Intel 82599 VF
c3.8xlarge	10 Gigabit	Intel 82599 VF
c4.large	Moderate	Intel 82599 VF
c4.xlarge	High	Intel 82599 VF
c4.2xlarge	High	Intel 82599 VF
c4.4xlarge	High	Intel 82599 VF
c4.8xlarge	10 Gigabit	Intel 82599 VF
c5.large †	Up to 10 Gigabit	ENA
c5.xlarge †	Up to 10 Gigabit	ENA
c5.2xlarge †	Up to 10 Gigabit	ENA
c5.4xlarge †	Up to 10 Gigabit	ENA
c5.9xlarge	12 Gigabit	ENA
c5.12xlarge	12 Gigabit	ENA
c5.18xlarge	25 Gigabit	ENA
c5.24xlarge	25 Gigabit	ENA
c5.metal	25 Gigabit	ENA
c5a.large †	Up to 10 Gigabit	ENA
c5a.xlarge †	Up to 10 Gigabit	ENA
c5a.2xlarge †	Up to 10 Gigabit	ENA
c5a.4xlarge †	Up to 10 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
c5a.8xlarge	10 Gigabit	ENA
c5a.12xlarge	12 Gigabit	ENA
c5a.16xlarge	20 Gigabit	ENA
c5a.24xlarge	20 Gigabit	ENA
c5ad.large †	Up to 10 Gigabit	ENA
c5ad.xlarge †	Up to 10 Gigabit	ENA
c5ad.2xlarge †	Up to 10 Gigabit	ENA
c5ad.4xlarge †	Up to 10 Gigabit	ENA
c5ad.8xlarge	10 Gigabit	ENA
c5ad.12xlarge	12 Gigabit	ENA
c5ad.16xlarge	20 Gigabit	ENA
c5ad.24xlarge	20 Gigabit	ENA
c5d.large †	Up to 10 Gigabit	ENA
c5d.xlarge †	Up to 10 Gigabit	ENA
c5d.2xlarge †	Up to 10 Gigabit	ENA
c5d.4xlarge †	Up to 10 Gigabit	ENA
c5d.9xlarge	12 Gigabit	ENA
c5d.12xlarge	12 Gigabit	ENA
c5d.18xlarge	25 Gigabit	ENA
c5d.24xlarge	25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
c5d.metal	25 Gigabit	ENA
c5n.large †	Up to 25 Gigabit	ENA
c5n.xlarge †	Up to 25 Gigabit	ENA
c5n.2xlarge †	Up to 25 Gigabit	ENA
c5n.4xlarge †	Up to 25 Gigabit	ENA
c5n.9xlarge	50 Gigabit	ENA | EFA
c5n.18xlarge	100 Gigabit	ENA | EFA
c5n.metal	100 Gigabit	ENA | EFA
c6a.large †	Up to 12.5 Gigabit	ENA
c6a.xlarge †	Up to 12.5 Gigabit	ENA
c6a.2xlarge †	Up to 12.5 Gigabit	ENA
c6a.4xlarge †	Up to 12.5 Gigabit	ENA
c6a.8xlarge	12.5 Gigabit	ENA
c6a.12xlarge	18.75 Gigabit	ENA
c6a.16xlarge	25 Gigabit	ENA
c6a.24xlarge	37.5 Gigabit	ENA
c6a.32xlarge	50 Gigabit	ENA
c6a.48xlarge	50 Gigabit	ENA | EFA
c6a.metal	50 Gigabit	ENA | EFA
c6i.large †	Up to 12.5 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
c6i.xlarge †	Up to 12.5 Gigabit	ENA
c6i.2xlarge †	Up to 12.5 Gigabit	ENA
c6i.4xlarge †	Up to 12.5 Gigabit	ENA
c6i.8xlarge	12.5 Gigabit	ENA
c6i.12xlarge	18.75 Gigabit	ENA
c6i.16xlarge	25 Gigabit	ENA
c6i.24xlarge	37.5 Gigabit	ENA
c6i.32xlarge	50 Gigabit	ENA | EFA
c6i.metal	50 Gigabit	ENA | EFA
c6id.large †	Up to 12.5 Gigabit	ENA
c6id.xlarge †	Up to 12.5 Gigabit	ENA
c6id.2xlarge †	Up to 12.5 Gigabit	ENA
c6id.4xlarge †	Up to 12.5 Gigabit	ENA
c6id.8xlarge	12.5 Gigabit	ENA
c6id.12xlarge	18.75 Gigabit	ENA
c6id.16xlarge	25 Gigabit	ENA
c6id.24xlarge	37.5 Gigabit	ENA
c6id.32xlarge	50 Gigabit	ENA | EFA
c6id.metal	50 Gigabit	ENA | EFA
c6in.large †	Up to 25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
c6in.xlarge †	Up to 30 Gigabit	ENA
c6in.2xlarge †	Up to 40 Gigabit	ENA
c6in.4xlarge †	Up to 50 Gigabit	ENA
c6in.8xlarge	50 Gigabit	ENA
c6in.12xlarge	75 Gigabit	ENA
c6in.16xlarge	100 Gigabit	ENA
c6in.24xlarge	150 Gigabit	ENA
c6in.32xlarge	200 Gigabit	ENA | EFA
c6in.metal	200 Gigabit	ENA | EFA
c7a.medium †	Up to 12.5 Gigabit	ENA
c7a.large †	Up to 12.5 Gigabit	ENA
c7a.xlarge †	Up to 12.5 Gigabit	ENA
c7a.2xlarge †	Up to 12.5 Gigabit	ENA
c7a.4xlarge †	Up to 12.5 Gigabit	ENA
c7a.8xlarge	12.5 Gigabit	ENA
c7a.12xlarge	18.75 Gigabit	ENA
c7a.16xlarge	25 Gigabit	ENA
c7a.24xlarge	37.5 Gigabit	ENA
c7a.32xlarge	50 Gigabit	ENA
c7a.48xlarge	50 Gigabit	ENA | EFA

Instance type	Network performance	Enhanced networking features
c7a.metal-48xl	50 Gigabit	ENA | EFA
c7i.large †	Up to 12.5 Gigabit	ENA
c7i.xlarge †	Up to 12.5 Gigabit	ENA
c7i.2xlarge †	Up to 12.5 Gigabit	ENA
c7i.4xlarge †	Up to 12.5 Gigabit	ENA
c7i.8xlarge	12.5 Gigabit	ENA
c7i.12xlarge	18.75 Gigabit	ENA
c7i.16xlarge	25 Gigabit	ENA
c7i.24xlarge	37.5 Gigabit	ENA
c7i.48xlarge	50 Gigabit	ENA | EFA
c7i.metal-24xl	37.5 Gigabit	ENA
c7i.metal-48xl	50 Gigabit	ENA | EFA

For `32xlarge` and `metal` instance types that support 200 Gbps, at least 2 ENIs, each attached to a different network card, are required on the instance to achieve 200 Gbps throughput. Each ENI attached to a network card can achieve a max of 170 Gbps.

The following table shows the baseline and burst bandwidth for instance types that use the network I/O credit mechanism to burst beyond their baseline bandwidth.

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
c5.large	0.75	10.0
c5.xlarge	1.25	10.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
c5.2xlarge	2.5	10.0
c5.4xlarge	5.0	10.0
c5a.large	0.75	10.0
c5a.xlarge	1.25	10.0
c5a.2xlarge	2.5	10.0
c5a.4xlarge	5.0	10.0
c5ad.large	0.75	10.0
c5ad.xlarge	1.25	10.0
c5ad.2xlarge	2.5	10.0
c5ad.4xlarge	5.0	10.0
c5d.large	0.75	10.0
c5d.xlarge	1.25	10.0
c5d.2xlarge	2.5	10.0
c5d.4xlarge	5.0	10.0
c5n.large	3.0	25.0
c5n.xlarge	5.0	25.0
c5n.2xlarge	10.0	25.0
c5n.4xlarge	15.0	25.0
c6a.large	0.781	12.5
c6a.xlarge	1.562	12.5

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
c6a.2xlarge	3.125	12.5
c6a.4xlarge	6.25	12.5
c6i.large	0.781	12.5
c6i.xlarge	1.562	12.5
c6i.2xlarge	3.125	12.5
c6i.4xlarge	6.25	12.5
c6id.large	0.781	12.5
c6id.xlarge	1.562	12.5
c6id.2xlarge	3.125	12.5
c6id.4xlarge	6.25	12.5
c6in.large	3.125	25.0
c6in.xlarge	6.25	30.0
c6in.2xlarge	12.5	40.0
c6in.4xlarge	25.0	50.0
c7a.medium	0.39	12.5
c7a.large	0.781	12.5
c7a.xlarge	1.562	12.5
c7a.2xlarge	3.125	12.5
c7a.4xlarge	6.25	12.5
c7i.large	0.781	12.5

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
c7i.xlarge	1.562	12.5
c7i.2xlarge	3.125	12.5
c7i.4xlarge	6.25	12.5

Amazon EBS I/O performance

Amazon EBS optimized instances use an optimized configuration stack and provide additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

For more information, see Amazon EBS–optimized instances.

SSD-based instance store volume I/O performance

If you use all the SSD-based instance store volumes available to your instance, you can get up to the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
c5ad.large	16283	7105
c5ad.xlarge	32566	14211
c5ad.2xlarge	65132	28421
c5ad.4xlarge	130262	56842
c5ad.8xlarge	260526	113684
c5ad.12xlarge	412500	180000
c5ad.16xlarge	521052	227368
c5ad.24xlarge	825000	360000

Instance Size	100% Random Read IOPS	Write IOPS
c5d.large	20000	9000
c5d.xlarge	40000	18000
c5d.2xlarge	80000	37000
c5d.4xlarge	175000	75000
c5d.9xlarge	350000	170000
c5d.12xlarge	700000	340000
c5d.18xlarge	700000	340000
c5d.24xlarge	1400000	680000
c5d.metal	1400000	680000
c6id.large	33542	16771
c6id.xlarge	67083	33542
c6id.2xlarge	134167	67084
c6id.4xlarge	268333	134167
c6id.8xlarge	536666	268334
c6id.12xlarge	804998	402500
c6id.16xlarge	1073332	536668
c6id.24xlarge	1609996	805000
c6id.32xlarge	2146664	1073336
c6id.metal	2146664	1073336

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

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

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support.

Release notes

- `c6a.metal` instances do not support Windows Server 2016 and earlier.
- `c7a.metal-48xl` instances do not support Windows Server 2019 AMIs released before July 2023, and Windows Server 2016 AMIs.
- C4 instances and instances built on the Nitro System require 64-bit EBS-backed HVM AMIs. They have high-memory and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- Instances built on the Nitro System have the following requirements:
  - NVMe drivers must be installed
  - Elastic Network Adapter (ENA) drivers must be installed

The current AWS Windows AMIs meet these requirements.

- To get the best performance from your C6i instances, ensure that they have ENA driver version 2.2.3 or later. Using an ENA driver earlier than version 2.0.0 with these instances causes network interface attachment failures. The following AMIs have a compatible ENA driver.
  - AWS Windows AMI from May 2021 or later
- 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.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

Memory optimized instances

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

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

R5, R5a, R5b, and R5n instances

These instances are well suited for the following:

- High-performance, including relational MySQL and NoSQL, for example MongoDB and Cassandra databases.
- Distributed web scale cache stores that provide in-memory caching of key-value type data, for example Memcached and Redis
- In-memory databases using optimized data storage formats and analytics for business intelligence; for example, SAP HANA
- Applications performing real-time processing of big unstructured data, using Hadoop and Spark clusters
- High-performance computing (HPC) and Electronic Design Automation (EDA) applications.

Bare metal instances, such as `r5.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 R5 Instances.

R6a instances

These instances are ideal for running memory-intensive workloads, such as the following:

- High-performance databases, both relational and NoSQL
- Distributed web scale in-memory caches, such as Memcached and Redis
- Real-time big data analytics, such as Hadoop and Spark clusters

R6i and R6id instances

These instances are ideal for running memory-intensive workloads, such as the following:

- High-performance databases, relational and NoSQL
- In-memory databases, for example SAP HANA
- Distributed web scale in-memory caches, for example Memcached and Redis
- Real-time big data analytics, including Hadoop and Spark clusters

R6in and R6idn instances

These instances are well suited for network-intensive workloads such as the following:

- High-performance relational, MySQL and NoSQL. For example, MongoDB and Cassandra databases
- Distributed web scale cache stores that provide in-memory caching of key-value type data, including Memcached and Redis
- In-memory databases using optimized data storage formats and analytics for business intelligence, for example SAP HANA
- Real-time big data analytics for financial services, for example Hadoop and Spark clusters

For more information, see Amazon EC2 R6i Instances.

R7a instances

These instances are powered by 4th generation AMD EPYC processors and are ideal for running memory-intensive workloads, such as the following:

- High-performance databases, both relational and NoSQL
- Distributed web scale in-memory caches, such as Memcached and Redis
- Real-time big data analytics, such as Hadoop and Spark clusters

R7i instances

R7i instances are powered by 4th generation Intel Xeon processors and are ideal for running memory-intensive workloads, such as the following:

- High-performance databases
- Distributed web scale in-memory caches
- In-memory databases, such as SAP HANA
- Real-time big-data analytics, such as Hadoop and Spark clusters

R7iz instances

R7iz are high-frequency and high memory instances that are powered by 4th generation Intel Xeon processors. They provide sustained all-core turbo frequency of up to 3.9 GHz, up to 1024 GB of system memory, up to 50 Gbps network bandwidth, and up to 40 Gbps of dedicated Amazon EBS bandwidth.

R7iz instances are well suited to workloads that require a combination of high compute and high memory such as the following:

- Electronic design automation
- Relational databases
- Data analytics simulations

For more information, see Amazon EC2 R7iz instances.

R7g and R7gd instances

These instances are powered by AWS Graviton3 processors and are ideal for running memory-intensive workloads, such as the following:

- Open-source databases; for example, MySQL, MariaDB, and PostgreSQL
- In-memory caches; for example, Memcached, Redis, and KeyDB

For more information, see Amazon EC2 R7g instances.

High memory (u-*) instances

These instances offer 3 TiB, 6 TiB, 9 TiB, 12 TiB, 18 TiB, and 24 TiB of memory per instance. They are designed to run large in-memory databases, including production deployments of the SAP HANA in-memory database.

For more information, see Amazon EC2 High Memory Instances and Storage Configuration for SAP HANA. For information about supported operating systems, see Migrating SAP HANA on AWS to an EC2 High Memory Instance.

X1 instances

These instances are well suited for the following:

- In-memory databases such as SAP HANA, including SAP-certified support for Business Suite S/4HANA, Business Suite on HANA (SoH), Business Warehouse on HANA (BW), and Data Mart Solutions on HANA. For more information, see SAP HANA on the AWS Cloud.
- Big-data processing engines such as Apache Spark or Presto.
- High-performance computing (HPC) applications.

For more information, see Amazon EC2 X1 Instances.

X1e instances

These instances are well suited for the following:

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

For more information, see Amazon EC2 X1e Instances.

X2idn, X2iedn, and X2iezn instances

These instances are well suited for the following:

- In-memory databases, such as Redis and Memcached.
- Relational databases, such as MySQL and PostGreSQL.
- Electronic design automation (EDA) workloads, such as physical verification and layout tools.
- Memory-intensive workloads, such as real-time analytics and real-time caching servers.

For more information, see Amazon EC2 X2i Instances.

z1d instances

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

- Electronic Design Automation (EDA)
- Relational database workloads

`z1d.metal` instances provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 z1d Instances.

Contents

- Hardware specifications

- Memory performance
- Instance performance
- Network performance
- Amazon EBS I/O performance
- SSD-based instance store volume I/O performance
- High availability and reliability (X1)
- Support for vCPUs
- Release notes

Hardware specifications

The following is a summary of the hardware specifications for memory optimized instances. A virtual central processing unit (vCPU) represents a portion of the physical CPU assigned to a virtual machine (VM). For x86 instances, there are two vCPUs per core. For Graviton instances, there is one vCPU per core.

Instance type	Default vCPUs	Memory (GiB)
r3.large	2	15.00
r3.xlarge	4	30.50
r3.2xlarge	8	61.00
r3.4xlarge	16	122.00
r3.8xlarge	32	244.00
r4.large	2	15.25
r4.xlarge	4	30.50
r4.2xlarge	8	61.00
r4.4xlarge	16	122.00
r4.8xlarge	32	244.00

Instance type	Default vCPUs	Memory (GiB)
r4.16xlarge	64	488.00
r5.large	2	16.00
r5.xlarge	4	32.00
r5.2xlarge	8	64.00
r5.4xlarge	16	128.00
r5.8xlarge	32	256.00
r5.12xlarge	48	384.00
r5.16xlarge	64	512.00
r5.24xlarge	96	768.00
r5.metal	96	768.00
r5a.large	2	16.00
r5a.xlarge	4	32.00
r5a.2xlarge	8	64.00
r5a.4xlarge	16	128.00
r5a.8xlarge	32	256.00
r5a.12xlarge	48	384.00
r5a.16xlarge	64	512.00
r5a.24xlarge	96	768.00
r5ad.large	2	16.00
r5ad.xlarge	4	32.00

Instance type	Default vCPUs	Memory (GiB)
r5ad.2xlarge	8	64.00
r5ad.4xlarge	16	128.00
r5ad.8xlarge	32	256.00
r5ad.12xlarge	48	384.00
r5ad.16xlarge	64	512.00
r5ad.24xlarge	96	768.00
r5b.large	2	16.00
r5b.xlarge	4	32.00
r5b.2xlarge	8	64.00
r5b.4xlarge	16	128.00
r5b.8xlarge	32	256.00
r5b.12xlarge	48	384.00
r5b.16xlarge	64	512.00
r5b.24xlarge	96	768.00
r5b.metal	96	768.00
r5d.large	2	16.00
r5d.xlarge	4	32.00
r5d.2xlarge	8	64.00
r5d.4xlarge	16	128.00
r5d.8xlarge	32	256.00

Instance type	Default vCPUs	Memory (GiB)
r5d.12xlarge	48	384.00
r5d.16xlarge	64	512.00
r5d.24xlarge	96	768.00
r5d.metal	96	768.00
r5dn.large	2	16.00
r5dn.xlarge	4	32.00
r5dn.2xlarge	8	64.00
r5dn.4xlarge	16	128.00
r5dn.8xlarge	32	256.00
r5dn.12xlarge	48	384.00
r5dn.16xlarge	64	512.00
r5dn.24xlarge	96	768.00
r5dn.metal	96	768.00
r5n.large	2	16.00
r5n.xlarge	4	32.00
r5n.2xlarge	8	64.00
r5n.4xlarge	16	128.00
r5n.8xlarge	32	256.00
r5n.12xlarge	48	384.00
r5n.16xlarge	64	512.00

Instance type	Default vCPUs	Memory (GiB)
r5n.24xlarge	96	768.00
r5n.metal	96	768.00
r6a.large	2	16.00
r6a.xlarge	4	32.00
r6a.2xlarge	8	64.00
r6a.4xlarge	16	128.00
r6a.8xlarge	32	256.00
r6a.12xlarge	48	384.00
r6a.16xlarge	64	512.00
r6a.24xlarge	96	768.00
r6a.32xlarge	128	1024.00
r6a.48xlarge	192	1536.00
r6a.metal	192	1536.00
r6i.large	2	16.00
r6i.xlarge	4	32.00
r6i.2xlarge	8	64.00
r6i.4xlarge	16	128.00
r6i.8xlarge	32	256.00
r6i.12xlarge	48	384.00
r6i.16xlarge	64	512.00

Instance type	Default vCPUs	Memory (GiB)
r6i.24xlarge	96	768.00
r6i.32xlarge	128	1024.00
r6i.metal	128	1024.00
r6idn.large	2	16.00
r6idn.xlarge	4	32.00
r6idn.2xlarge	8	64.00
r6idn.4xlarge	16	128.00
r6idn.8xlarge	32	256.00
r6idn.12xlarge	48	384.00
r6idn.16xlarge	64	512.00
r6idn.24xlarge	96	768.00
r6idn.32xlarge	128	1024.00
r6idn.metal	128	1024.00
r6in.large	2	16.00
r6in.xlarge	4	32.00
r6in.2xlarge	8	64.00
r6in.4xlarge	16	128.00
r6in.8xlarge	32	256.00
r6in.12xlarge	48	384.00
r6in.16xlarge	64	512.00

Instance type	Default vCPUs	Memory (GiB)
r6in.24xlarge	96	768.00
r6in.32xlarge	128	1024.00
r6in.metal	128	1024.00
r6id.large	2	16.00
r6id.xlarge	4	32.00
r6id.2xlarge	8	64.00
r6id.4xlarge	16	128.00
r6id.8xlarge	32	256.00
r6id.12xlarge	48	384.00
r6id.16xlarge	64	512.00
r6id.24xlarge	96	768.00
r6id.32xlarge	128	1024.00
r6id.metal	128	1024.00
r7a.medium	1	8.00
r7a.large	2	16.00
r7a.xlarge	4	32.00
r7a.2xlarge	8	64.00
r7a.4xlarge	16	128.00
r7a.8xlarge	32	256.00
r7a.12xlarge	48	384.00

Instance type	Default vCPUs	Memory (GiB)
r7a.16xlarge	64	512.00
r7a.24xlarge	96	768.00
r7a.32xlarge	128	1024.00
r7a.48xlarge	192	1536.00
r7a.metal-48xl	192	1536.00
r7i.large	2	16.00
r7i.xlarge	4	32.00
r7i.2xlarge	8	64.00
r7i.4xlarge	16	128.00
r7i.8xlarge	32	256.00
r7i.12xlarge	48	384.00
r7i.16xlarge	64	512.00
r7i.24xlarge	96	768.00
r7i.48xlarge	192	1536.00
r7i.metal-24xl	96	768.00
r7i.metal-48xl	192	1536.00
r7iz.large	2	16.00
r7iz.xlarge	4	32.00
r7iz.2xlarge	8	64.00
r7iz.4xlarge	16	128.00

Instance type	Default vCPUs	Memory (GiB)
r7iz.8xlarge	32	256.00
r7iz.12xlarge	48	384.00
r7iz.16xlarge	64	512.00
r7iz.32xlarge	128	1024.00
r7iz.metal-16xl	64	512.00
r7iz.metal-32xl	128	1024.00
u-3tb1.56xlarge	224	3072.00
u-6tb1.56xlarge	224	6144.00
u-6tb1.112xlarge	448	6144.00
u-6tb1.metal	448	6144.00
u-9tb1.112xlarge	448	9216.00
u-9tb1.metal	448	9216.00
u-12tb1.112xlarge	448	12288.00
u-12tb1.metal	448	12288.00
u-18tb1.112xlarge	448	18432.00
u-18tb1.metal	448	18432.00
u-24tb1.112xlarge	448	24576.00
u-24tb1.metal	448	24576.00
x1.16xlarge	64	976.00
x1.32xlarge	128	1952.00

Instance type	Default vCPUs	Memory (GiB)
x2idn.16xlarge	64	1024.00
x2idn.24xlarge	96	1536.00
x2idn.32xlarge	128	2048.00
x2idn.metal	128	2048.00
x2iedn.xlarge	4	128.00
x2iedn.2xlarge	8	256.00
x2iedn.4xlarge	16	512.00
x2iedn.8xlarge	32	1024.00
x2iedn.16xlarge	64	2048.00
x2iedn.24xlarge	96	3072.00
x2iedn.32xlarge	128	4096.00
x2iedn.metal	128	4096.00
x2iezn.2xlarge	8	256.00
x2iezn.4xlarge	16	512.00
x2iezn.6xlarge	24	768.00
x2iezn.8xlarge	32	1024.00
x2iezn.12xlarge	48	1536.00
x2iezn.metal	48	1536.00
x1e.xlarge	4	122.00
x1e.2xlarge	8	244.00

Instance type	Default vCPUs	Memory (GiB)
x1e.4xlarge	16	488.00
x1e.8xlarge	32	976.00
x1e.16xlarge	64	1952.00
x1e.32xlarge	128	3904.00
z1d.large	2	16.00
z1d.xlarge	4	32.00
z1d.2xlarge	8	64.00
z1d.3xlarge	12	96.00
z1d.6xlarge	24	192.00
z1d.12xlarge	48	384.00
z1d.metal	48	384.00

* Each logical processor is a hyperthread on 224 cores.

The memory optimized instances use the following processors.

AMD processors

- **AMD EPYC 7000 series processors (AMD EPYC 7571)**: R5a, R5ad
- **3rd generation AMD EPYC processors (AMD EPYC 7R13)**: R6a
- **4th generation AMD EPYC processors (AMD EPYC 9R14)**: R7a

Intel processors

- **Intel Xeon Scalable processors (Haswell E7-8880 v3)**: X1, X1e
- **Intel Xeon Scalable processors (Broadwell E5-2686 v4)**: R4
- **Intel Xeon Scalable processors (Skylake 8151)**: z1d

- **Intel Xeon Scalable processors (Skylake 8175M or Cascade Lake 8259CL)**: R5, R5d
- **2nd generation Intel Xeon Scalable processors (Cascade Lake 8259CL)**: R5b, R5n
- **2nd generation Intel Xeon Scalable processors (Cascade Lake 8252C)**: X2iezn
- **3rd generation Intel Xeon Scalable processors (Ice Lake 8375C)**: R6i, R6id, X2idn, X2iedn
- **4th generation Intel Xeon Scalable processors (Sapphire Rapids 8488C)**: R7i
- **4th generation Intel Xeon Scalable processors (Sapphire Rapids 6455B)**: R7iz

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Memory performance

X1 instances include Intel Scalable Memory Buffers, providing 300 GiB/s of sustainable memory-read bandwidth and 140 GiB/s of sustainable memory-write bandwidth.

For more information about how much RAM can be enabled for memory optimized instances, see Hardware specifications.

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

Instance performance

Memory optimized instances enable increased cryptographic performance through the latest Intel AES-NI feature and support Advanced Vector Extensions 2 (Intel AVX2) processor instructions to expand most integer commands to 256 bits.

Network performance

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

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

> **Note**
>
> Instance types indicated with a † have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth.

Instance type	Network performance	Enhanced networking features
r3.large	Moderate	Intel 82599 VF
r3.xlarge	Moderate	Intel 82599 VF
r3.2xlarge	High	Intel 82599 VF
r3.4xlarge	High	Intel 82599 VF
r3.8xlarge	10 Gigabit	Intel 82599 VF
r4.large †	Up to 10 Gigabit	ENA
r4.xlarge †	Up to 10 Gigabit	ENA
r4.2xlarge †	Up to 10 Gigabit	ENA
r4.4xlarge †	Up to 10 Gigabit	ENA
r4.8xlarge	10 Gigabit	ENA
r4.16xlarge	25 Gigabit	ENA
r5.large †	Up to 10 Gigabit	ENA
r5.xlarge †	Up to 10 Gigabit	ENA
r5.2xlarge †	Up to 10 Gigabit	ENA
r5.4xlarge †	Up to 10 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r5.8xlarge	10 Gigabit	ENA
r5.12xlarge	12 Gigabit	ENA
r5.16xlarge	20 Gigabit	ENA
r5.24xlarge	25 Gigabit	ENA
r5.metal	25 Gigabit	ENA
r5a.large †	Up to 10 Gigabit	ENA
r5a.xlarge †	Up to 10 Gigabit	ENA
r5a.2xlarge †	Up to 10 Gigabit	ENA
r5a.4xlarge †	Up to 10 Gigabit	ENA
r5a.8xlarge †	Up to 10 Gigabit	ENA
r5a.12xlarge	10 Gigabit	ENA
r5a.16xlarge	12 Gigabit	ENA
r5a.24xlarge	20 Gigabit	ENA
r5ad.large †	Up to 10 Gigabit	ENA
r5ad.xlarge †	Up to 10 Gigabit	ENA
r5ad.2xlarge †	Up to 10 Gigabit	ENA
r5ad.4xlarge †	Up to 10 Gigabit	ENA
r5ad.8xlarge †	Up to 10 Gigabit	ENA
r5ad.12xlarge	10 Gigabit	ENA
r5ad.16xlarge	12 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r5ad.24xlarge	20 Gigabit	ENA
r5b.large †	Up to 10 Gigabit	ENA
r5b.xlarge †	Up to 10 Gigabit	ENA
r5b.2xlarge †	Up to 10 Gigabit	ENA
r5b.4xlarge †	Up to 10 Gigabit	ENA
r5b.8xlarge	10 Gigabit	ENA
r5b.12xlarge	12 Gigabit	ENA
r5b.16xlarge	20 Gigabit	ENA
r5b.24xlarge	25 Gigabit	ENA
r5b.metal	25 Gigabit	ENA
r5d.large †	Up to 10 Gigabit	ENA
r5d.xlarge †	Up to 10 Gigabit	ENA
r5d.2xlarge †	Up to 10 Gigabit	ENA
r5d.4xlarge †	Up to 10 Gigabit	ENA
r5d.8xlarge	10 Gigabit	ENA
r5d.12xlarge	12 Gigabit	ENA
r5d.16xlarge	20 Gigabit	ENA
r5d.24xlarge	25 Gigabit	ENA
r5d.metal	25 Gigabit	ENA
r5dn.large †	Up to 25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r5dn.xlarge †	Up to 25 Gigabit	ENA
r5dn.2xlarge †	Up to 25 Gigabit	ENA
r5dn.4xlarge †	Up to 25 Gigabit	ENA
r5dn.8xlarge	25 Gigabit	ENA
r5dn.12xlarge	50 Gigabit	ENA
r5dn.16xlarge	75 Gigabit	ENA
r5dn.24xlarge	100 Gigabit	ENA | EFA
r5dn.metal	100 Gigabit	ENA | EFA
r5n.large †	Up to 25 Gigabit	ENA
r5n.xlarge †	Up to 25 Gigabit	ENA
r5n.2xlarge †	Up to 25 Gigabit	ENA
r5n.4xlarge †	Up to 25 Gigabit	ENA
r5n.8xlarge	25 Gigabit	ENA
r5n.12xlarge	50 Gigabit	ENA
r5n.16xlarge	75 Gigabit	ENA
r5n.24xlarge	100 Gigabit	ENA | EFA
r5n.metal	100 Gigabit	ENA | EFA
r6a.large †	Up to 12.5 Gigabit	ENA
r6a.xlarge †	Up to 12.5 Gigabit	ENA
r6a.2xlarge †	Up to 12.5 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r6a.4xlarge †	Up to 12.5 Gigabit	ENA
r6a.8xlarge	12.5 Gigabit	ENA
r6a.12xlarge	18.75 Gigabit	ENA
r6a.16xlarge	25 Gigabit	ENA
r6a.24xlarge	37.5 Gigabit	ENA
r6a.32xlarge	50 Gigabit	ENA
r6a.48xlarge	50 Gigabit	ENA | EFA
r6a.metal	50 Gigabit	ENA | EFA
r6i.large †	Up to 12.5 Gigabit	ENA
r6i.xlarge †	Up to 12.5 Gigabit	ENA
r6i.2xlarge †	Up to 12.5 Gigabit	ENA
r6i.4xlarge †	Up to 12.5 Gigabit	ENA
r6i.8xlarge	12.5 Gigabit	ENA
r6i.12xlarge	18.75 Gigabit	ENA
r6i.16xlarge	25 Gigabit	ENA
r6i.24xlarge	37.5 Gigabit	ENA
r6i.32xlarge	50 Gigabit	ENA | EFA
r6i.metal	50 Gigabit	ENA | EFA
r6idn.large †	Up to 25 Gigabit	ENA
r6idn.xlarge †	Up to 30 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r6idn.2xlarge †	Up to 40 Gigabit	ENA
r6idn.4xlarge †	Up to 50 Gigabit	ENA
r6idn.8xlarge	50 Gigabit	ENA
r6idn.12xlarge	75 Gigabit	ENA
r6idn.16xlarge	100 Gigabit	ENA
r6idn.24xlarge	150 Gigabit	ENA
r6idn.32xlarge	200 Gigabit	ENA | EFA
r6idn.metal	200 Gigabit	ENA | EFA
r6in.large †	Up to 25 Gigabit	ENA
r6in.xlarge †	Up to 30 Gigabit	ENA
r6in.2xlarge †	Up to 40 Gigabit	ENA
r6in.4xlarge †	Up to 50 Gigabit	ENA
r6in.8xlarge	50 Gigabit	ENA
r6in.12xlarge	75 Gigabit	ENA
r6in.16xlarge	100 Gigabit	ENA
r6in.24xlarge	150 Gigabit	ENA
r6in.32xlarge	200 Gigabit	ENA | EFA
r6in.metal	200 Gigabit	ENA | EFA
r6id.large †	Up to 12.5 Gigabit	ENA
r6id.xlarge †	Up to 12.5 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r6id.2xlarge †	Up to 12.5 Gigabit	ENA
r6id.4xlarge †	Up to 12.5 Gigabit	ENA
r6id.8xlarge	12.5 Gigabit	ENA
r6id.12xlarge	18.75 Gigabit	ENA
r6id.16xlarge	25 Gigabit	ENA
r6id.24xlarge	37.5 Gigabit	ENA
r6id.32xlarge	50 Gigabit	ENA | EFA
r6id.metal	50 Gigabit	ENA | EFA
r7a.medium †	Up to 12.5 Gigabit	ENA
r7a.large †	Up to 12.5 Gigabit	ENA
r7a.xlarge †	Up to 12.5 Gigabit	ENA
r7a.2xlarge †	Up to 12.5 Gigabit	ENA
r7a.4xlarge †	Up to 12.5 Gigabit	ENA
r7a.8xlarge	12.5 Gigabit	ENA
r7a.12xlarge	18.75 Gigabit	ENA
r7a.16xlarge	25 Gigabit	ENA
r7a.24xlarge	37.5 Gigabit	ENA
r7a.32xlarge	50 Gigabit	ENA
r7a.48xlarge	50 Gigabit	ENA | EFA
r7a.metal-48xl	50 Gigabit	ENA | EFA

Instance type	Network performance	Enhanced networking features
r7i.large †	Up to 12.5 Gigabit	ENA
r7i.xlarge †	Up to 12.5 Gigabit	ENA
r7i.2xlarge †	Up to 12.5 Gigabit	ENA
r7i.4xlarge †	Up to 12.5 Gigabit	ENA
r7i.8xlarge	12.5 Gigabit	ENA
r7i.12xlarge	18.75 Gigabit	ENA
r7i.16xlarge	25 Gigabit	ENA
r7i.24xlarge	37.5 Gigabit	ENA
r7i.48xlarge	50 Gigabit	ENA | EFA
r7i.metal-24xl	37.5 Gigabit	ENA
r7i.metal-48xl	50 Gigabit	ENA | EFA
r7iz.large †	Up to 12.5 Gigabit	ENA
r7iz.xlarge †	Up to 12.5 Gigabit	ENA
r7iz.2xlarge †	Up to 12.5 Gigabit	ENA
r7iz.4xlarge †	Up to 12.5 Gigabit	ENA
r7iz.8xlarge	12.5 Gigabit	ENA
r7iz.12xlarge	25 Gigabit	ENA
r7iz.16xlarge	25 Gigabit	ENA
r7iz.32xlarge	50 Gigabit	ENA | EFA
r7iz.metal-16xl	25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
r7iz.metal-32xl	50 Gigabit	ENA | EFA
u-3tb1.56xlarge	50 Gigabit	ENA
u-6tb1.56xlarge	100 Gigabit	ENA
u-6tb1.112xlarge	100 Gigabit	ENA
u-6tb1.metal	100	ENA
u-9tb1.112xlarge	100 Gigabit	ENA
u-9tb1.metal	100	ENA
u-12tb1.112xlarge	100 Gigabit	ENA
u-12tb1.metal	100	ENA
u-18tb1.112xlarge	100 Gigabit	ENA
u-18tb1.metal	100 Gigabit	ENA
u-24tb1.112xlarge	100 Gigabit	ENA
u-24tb1.metal	100 Gigabit	ENA
x1.16xlarge	10 Gigabit	ENA
x1.32xlarge	25 Gigabit	ENA
x2idn.16xlarge	50 Gigabit	ENA
x2idn.24xlarge	75 Gigabit	ENA
x2idn.32xlarge	100 Gigabit	ENA | EFA
x2idn.metal	100 Gigabit	ENA | EFA
x2iedn.xlarge †	Up to 25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
x2iedn.2xlarge †	Up to 25 Gigabit	ENA
x2iedn.4xlarge †	Up to 25 Gigabit	ENA
x2iedn.8xlarge	25 Gigabit	ENA
x2iedn.16xlarge	50 Gigabit	ENA
x2iedn.24xlarge	75 Gigabit	ENA
x2iedn.32xlarge	100 Gigabit	ENA | EFA
x2iedn.metal	100 Gigabit	ENA | EFA
x2iezn.2xlarge †	Up to 25 Gigabit	ENA
x2iezn.4xlarge †	Up to 25 Gigabit	ENA
x2iezn.6xlarge	50 Gigabit	ENA
x2iezn.8xlarge	75 Gigabit	ENA
x2iezn.12xlarge	100 Gigabit	ENA | EFA
x2iezn.metal	100 Gigabit	ENA | EFA
x1e.xlarge †	Up to 10 Gigabit	ENA
x1e.2xlarge †	Up to 10 Gigabit	ENA
x1e.4xlarge †	Up to 10 Gigabit	ENA
x1e.8xlarge †	Up to 10 Gigabit	ENA
x1e.16xlarge	10 Gigabit	ENA
x1e.32xlarge	25 Gigabit	ENA
z1d.large †	Up to 10 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
z1d.xlarge †	Up to 10 Gigabit	ENA
z1d.2xlarge †	Up to 10 Gigabit	ENA
z1d.3xlarge †	Up to 10 Gigabit	ENA
z1d.6xlarge	12 Gigabit	ENA
z1d.12xlarge	25 Gigabit	ENA
z1d.metal	25 Gigabit	ENA

For 32xlarge and metal instance types that support 200 Gbps, at least 2 ENIs, each attached to a different network card, are required on the instance to achieve 200 Gbps throughput. Each ENI attached to a network card can achieve a max of 170 Gbps.

u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal instances launched after March 12, 2020 provide network performance of 100 Gbps. u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal instances launched before March 12, 2020 mightonly provide network performance of 25 Gbps. To ensure that instances launched before March 12, 2020 have a network performance of 100 Gbps, contact your account team to upgrade your instance at no additional cost.

The following table shows the baseline and burst bandwidth for instance types that use the network I/O credit mechanism to burst beyond their baseline bandwidth.

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
r4.large	0.75	10.0
r4.xlarge	1.25	10.0
r4.2xlarge	2.5	10.0
r4.4xlarge	5.0	10.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
r5.large	0.75	10.0
r5.xlarge	1.25	10.0
r5.2xlarge	2.5	10.0
r5.4xlarge	5.0	10.0
r5a.large	0.75	10.0
r5a.xlarge	1.25	10.0
r5a.2xlarge	2.5	10.0
r5a.4xlarge	5.0	10.0
r5a.8xlarge	7.5	10.0
r5ad.large	0.75	10.0
r5ad.xlarge	1.25	10.0
r5ad.2xlarge	2.5	10.0
r5ad.4xlarge	5.0	10.0
r5ad.8xlarge	7.5	10.0
r5b.large	0.75	10.0
r5b.xlarge	1.25	10.0
r5b.2xlarge	2.5	10.0
r5b.4xlarge	5.0	10.0
r5d.large	0.75	10.0
r5d.xlarge	1.25	10.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
r5d.2xlarge	2.5	10.0
r5d.4xlarge	5.0	10.0
r5dn.large	2.1	25.0
r5dn.xlarge	4.1	25.0
r5dn.2xlarge	8.125	25.0
r5dn.4xlarge	16.25	25.0
r5n.large	2.1	25.0
r5n.xlarge	4.1	25.0
r5n.2xlarge	8.125	25.0
r5n.4xlarge	16.25	25.0
r6a.large	0.781	12.5
r6a.xlarge	1.562	12.5
r6a.2xlarge	3.125	12.5
r6a.4xlarge	6.25	12.5
r6i.large	0.781	12.5
r6i.xlarge	1.562	12.5
r6i.2xlarge	3.125	12.5
r6i.4xlarge	6.25	12.5
r6idn.large	3.125	25.0
r6idn.xlarge	6.25	30.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
r6idn.2xlarge	12.5	40.0
r6idn.4xlarge	25.0	50.0
r6in.large	3.125	25.0
r6in.xlarge	6.25	30.0
r6in.2xlarge	12.5	40.0
r6in.4xlarge	25.0	50.0
r6id.large	0.781	12.5
r6id.xlarge	1.562	12.5
r6id.2xlarge	3.125	12.5
r6id.4xlarge	6.25	12.5
r7a.medium	0.39	12.5
r7a.large	0.781	12.5
r7a.xlarge	1.562	12.5
r7a.2xlarge	3.125	12.5
r7a.4xlarge	6.25	12.5
r7i.large	0.781	12.5
r7i.xlarge	1.562	12.5
r7i.2xlarge	3.125	12.5
r7i.4xlarge	6.25	12.5
r7iz.large	0.781	12.5

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
r7iz.xlarge	1.562	12.5
r7iz.2xlarge	3.125	12.5
r7iz.4xlarge	6.25	12.5
x2iedn.xlarge	1.875	25.0
x2iedn.2xlarge	5.0	25.0
x2iedn.4xlarge	12.5	25.0
x2iezn.2xlarge	12.5	25.0
x2iezn.4xlarge	15.0	25.0
x1e.xlarge	0.625	10.0
x1e.2xlarge	1.25	10.0
x1e.4xlarge	2.5	10.0
x1e.8xlarge	5.0	10.0
z1d.large	0.75	10.0
z1d.xlarge	1.25	10.0
z1d.2xlarge	2.5	10.0
z1d.3xlarge	5.0	10.0

Amazon EBS I/O performance

Amazon EBS optimized instances use an optimized configuration stack and provide additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

For more information, see Amazon EBS–optimized instances.

SSD-based instance store volume I/O performance

If you use all the SSD-based instance store volumes available to your instance, you can get up to the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
`r5ad.large`	30000	15000
`r5ad.xlarge`	59000	29000
`r5ad.2xlarge`	117000	57000
`r5ad.4xlarge`	234000	114000
`r5ad.8xlarge`	466666	233334
`r5ad.12xlarge`	700000	340000
`r5ad.16xlarge`	933332	466668
`r5ad.24xlarge`	1400000	680000
`r5d.large`	30000	15000
`r5d.xlarge`	59000	29000
`r5d.2xlarge`	117000	57000
`r5d.4xlarge`	234000	114000
`r5d.8xlarge`	466666	233334
`r5d.12xlarge`	700000	340000
`r5d.16xlarge`	933332	466668
`r5d.24xlarge`	1400000	680000

Instance Size	100% Random Read IOPS	Write IOPS
r5d.metal	1400000	680000
r5dn.large	29000	14500
r5dn.xlarge	58000	29000
r5dn.2xlarge	116000	58000
r5dn.4xlarge	232000	116000
r5dn.8xlarge	464000	232000
r5dn.12xlarge	700000	350000
r5dn.16xlarge	930000	465000
r5dn.24xlarge	1400000	700000
r5dn.metal	1400000	700000
r6idn.large	33542	16771
r6idn.xlarge	67083	33542
r6idn.2xlarge	134167	67084
r6idn.4xlarge	268333	134167
r6idn.8xlarge	536666	268334
r6idn.12xlarge	804998	402500
r6idn.16xlarge	1073332	536668
r6idn.24xlarge	1609996	805000
r6idn.32xlarge	2146664	1073336
r6idn.metal	2146664	1073336

Instance Size	100% Random Read IOPS	Write IOPS
r6id.large	33542	16771
r6id.xlarge	67083	33542
r6id.2xlarge	134167	67084
r6id.4xlarge	268333	134167
r6id.8xlarge	536666	268334
r6id.12xlarge	804998	402500
r6id.16xlarge	1073332	536668
r6id.24xlarge	1609996	805000
r6id.32xlarge	2146664	1073336
r6id.metal	2146664	1073336
x2idn.16xlarge	430000	180000
x2idn.24xlarge	645000	270000
x2idn.32xlarge	860000	360000
x2idn.metal	860000	360000
x2iedn.xlarge	26875	11250
x2iedn.2xlarge	53750	22500
x2iedn.4xlarge	107500	45000
x2iedn.8xlarge	215000	90000
x2iedn.16xlarge	430000	180000
x2iedn.24xlarge	645000	270000

Instance Size	100% Random Read IOPS	Write IOPS
`x2iedn.32xlarge`	860000	360000
`x2iedn.metal`	860000	360000
`z1d.large`	30000	15000
`z1d.xlarge`	59000	29000
`z1d.2xlarge`	117000	57000
`z1d.3xlarge`	175000	75000
`z1d.6xlarge`	350000	170000
`z1d.12xlarge`	700000	340000
`z1d.metal`	700000	340000

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

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

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support.

High availability and reliability (X1)

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

In addition, you can implement high availability (HA) and disaster recovery (DR) solutions to meet recovery point objective (RPO), recovery time objective (RTO), and cost requirements by leveraging Amazon CloudFormation and Recover your instance.

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

Support for vCPUs

Memory optimized instances provide a high number of vCPUs, which can cause launch issues with operating systems that have a lower vCPU limit. We strongly recommend that you use the latest AMIs when you launch memory optimized instances.

The following AMIs support launching memory optimized instances:

- Amazon Linux 2 (HVM)
- Amazon Linux AMI 2016.03 (HVM) or later
- Ubuntu Server 14.04 LTS (HVM)
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 SP1 (HVM)
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012

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

Release notes

- `r6a.metal` instances do not support Windows Server 2016 and earlier.
- `r7a.metal-48xl` instances support only Windows Server 2019 and Windows Server 2022 AMIs. For Windows Server 2019, you must use an AMI with a release date of July 2023 or later.
- Instances built on the Nitro System have the following requirements:
  - NVMe drivers must be installed
  - Elastic Network Adapter (ENA) drivers must be installed

  The current AWS Windows AMIs meet these requirements.
- To get the best performance from your R6i instances, ensure that they have ENA driver version 2.2.3 or later. Using an ENA driver earlier than version 2.0.0 with these instances causes network interface attachment failures. The following AMIs have a compatible ENA driver.
  - AWS Windows AMI from May 2021 or later
- 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.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- You can't launch X1 instances using a Windows Server 2008 SP2 64-bit AMI, except for `x1.16xlarge` instances.
- You can't launch X1e instances using a Windows Server 2008 SP2 64-bit AMI.
- With earlier versions of the Windows Server 2008 R2 64-bit AMI, you can't launch `r4.large` and `r4.4xlarge` instances. If you experience this issue, update to the latest version of this AMI.

- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

Storage optimized instances

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. For more information, including the technology used, see the Amazon EC2 Instance Type Details page.

D2 instances

These instances are well suited for the following:

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

D3 and D3en instances

These instances offer scale out of instance storage and are well suited for the following:

- Distributed file systems for Hadoop workloads
- File storage workloads such as GPFC and BeeFS
- Large data lakes for HPC workloads

H1 instances

These instances are well suited for the following:

- Data-intensive workloads such as MapReduce and distributed file systems
- Applications requiring sequential access to large amounts of data on direct-attached instance storage
- Applications that require high-throughput access to large quantities of data

I3 and I3en instances

These instances are well suited for the following:

- High frequency online transaction processing (OLTP) systems
- Relational databases
- NoSQL databases
- Cache for in-memory databases (for example, Redis)
- Data warehousing applications
- Distributed file systems

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

For more information, see Amazon EC2 I3 Instances.

I4i instances

These instances are well suited for I/O intensive workloads that require small to medium sized data sets on local storage, such as transactional databases and NoSQL databases.

For more information, see Amazon EC2 I4i Instances.

Contents

- Hardware specifications
- Instance performance
- Network performance
- Amazon EBS I/O performance
- SSD-based instance store volume I/O performance
- Release notes

Hardware specifications

The following is a summary of the hardware specifications for storage optimized instances. A virtual central processing unit (vCPU) represents a portion of the physical CPU assigned to a virtual machine (VM). For x86 instances, there are two vCPUs per core. For Graviton instances, there is one vCPU per core.

Instance type	Default vCPUs	Memory (GiB)
d2.xlarge	4	30.50
d2.2xlarge	8	61.00
d2.4xlarge	16	122.00
d2.8xlarge	36	244.00
d3.xlarge	4	32.00
d3.2xlarge	8	64.00
d3.4xlarge	16	128.00
d3.8xlarge	32	256.00
d3en.xlarge	4	16.00
d3en.2xlarge	8	32.00
d3en.4xlarge	16	64.00
d3en.6xlarge	24	96.00
d3en.8xlarge	32	128.00
d3en.12xlarge	48	192.00
h1.2xlarge	8	32.00
h1.4xlarge	16	64.00

Instance type	Default vCPUs	Memory (GiB)
h1.8xlarge	32	128.00
h1.16xlarge	64	256.00
i2.xlarge	4	30.50
i2.2xlarge	8	61.00
i2.4xlarge	16	122.00
i2.8xlarge	32	244.00
i3.large	2	15.25
i3.xlarge	4	30.50
i3.2xlarge	8	61.00
i3.4xlarge	16	122.00
i3.8xlarge	32	244.00
i3.16xlarge	64	488.00
i3.metal	72	512.00
i3en.large	2	16.00
i3en.xlarge	4	32.00
i3en.2xlarge	8	64.00
i3en.3xlarge	12	96.00
i3en.6xlarge	24	192.00
i3en.12xlarge	48	384.00
i3en.24xlarge	96	768.00

Instance type	Default vCPUs	Memory (GiB)
i3en.metal	96	768.00
i4i.large	2	16.00
i4i.xlarge	4	32.00
i4i.2xlarge	8	64.00
i4i.4xlarge	16	128.00
i4i.8xlarge	32	256.00
i4i.12xlarge	48	384.00
i4i.16xlarge	64	512.00
i4i.24xlarge	96	768.00
i4i.32xlarge	128	1024.00
i4i.metal	128	1024.00

The storage optimized instances use the following processors.

Intel processors

- **Intel Xeon Scalable processors (Haswell E5-2676 v3)**: D2
- **Intel Xeon Scalable processors (Broadwell E5-2686 v4)**: H1, I3
- **Intel Xeon Scalable processors (Skylake 8175M or Cascade Lake 8259CL)**: I3en
- **2nd generation Intel Xeon Scalable processors (Cascade Lake 8259CL)**: D3, D3en
- **3rd generation Intel Xeon Scalable processors (Ice Lake 8375C)**: I4i

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Instance performance

For instances with NVMe instance store volumes, be sure to use the AWS NVMe driver. For more information, see AWS NVMe drivers for Windows instances.

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some storage optimized instances are EBS-optimized by default at no additional cost. For more information, see Amazon EBS–optimized instances.

Network performance

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

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

> **Note**
>
> Instance types indicated with a † have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth.

Instance type	Network performance	Enhanced networking features
`d2.xlarge`	Moderate	Intel 82599 VF
`d2.2xlarge`	High	Intel 82599 VF
`d2.4xlarge`	High	Intel 82599 VF
`d2.8xlarge`	10 Gigabit	Intel 82599 VF
`d3.xlarge` †	Up to 15 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
d3.2xlarge †	Up to 15 Gigabit	ENA
d3.4xlarge †	Up to 15 Gigabit	ENA
d3.8xlarge	25 Gigabit	ENA
d3en.xlarge †	Up to 25 Gigabit	ENA
d3en.2xlarge †	Up to 25 Gigabit	ENA
d3en.4xlarge	25 Gigabit	ENA
d3en.6xlarge	40 Gigabit	ENA
d3en.8xlarge	50 Gigabit	ENA
d3en.12xlarge	75 Gigabit	ENA
h1.2xlarge †	Up to 10 Gigabit	ENA
h1.4xlarge †	Up to 10 Gigabit	ENA
h1.8xlarge	10 Gigabit	ENA
h1.16xlarge	25 Gigabit	ENA
i2.xlarge	Moderate	Intel 82599 VF
i2.2xlarge	High	Intel 82599 VF
i2.4xlarge	High	Intel 82599 VF
i2.8xlarge	10 Gigabit	Intel 82599 VF
i3.large †	Up to 10 Gigabit	ENA
i3.xlarge †	Up to 10 Gigabit	ENA
i3.2xlarge †	Up to 10 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
i3.4xlarge †	Up to 10 Gigabit	ENA
i3.8xlarge	10 Gigabit	ENA
i3.16xlarge	25 Gigabit	ENA
i3.metal	25 Gigabit	ENA
i3en.large †	Up to 25 Gigabit	ENA
i3en.xlarge †	Up to 25 Gigabit	ENA
i3en.2xlarge †	Up to 25 Gigabit	ENA
i3en.3xlarge †	Up to 25 Gigabit	ENA
i3en.6xlarge	25 Gigabit	ENA
i3en.12xlarge	50 Gigabit	ENA | EFA
i3en.24xlarge	100 Gigabit	ENA | EFA
i3en.metal	100 Gigabit	ENA | EFA
i4i.large †	Up to 10 Gigabit	ENA
i4i.xlarge †	Up to 10 Gigabit	ENA
i4i.2xlarge †	Up to 12 Gigabit	ENA
i4i.4xlarge †	Up to 25 Gigabit	ENA
i4i.8xlarge	18.75 Gigabit	ENA
i4i.12xlarge	28.12 Gigabit	ENA
i4i.16xlarge	37.5 Gigabit	ENA
i4i.24xlarge	56.25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
i4i.32xlarge	75 Gigabit	ENA | EFA
i4i.metal	75 Gigabit	ENA | EFA

The following table shows the baseline and burst bandwidth for instance types that use the network I/O credit mechanism to burst beyond their baseline bandwidth.

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
d3.xlarge	3.0	15.0
d3.2xlarge	6.0	15.0
d3.4xlarge	12.5	15.0
d3en.xlarge	6.0	25.0
d3en.2xlarge	12.5	25.0
h1.2xlarge	2.5	10.0
h1.4xlarge	5.0	10.0
i3.large	0.75	10.0
i3.xlarge	1.25	10.0
i3.2xlarge	2.5	10.0
i3.4xlarge	5.0	10.0
i3en.large	2.1	25.0
i3en.xlarge	4.2	25.0
i3en.2xlarge	8.4	25.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
i3en.3xlarge	12.5	25.0
i4i.large	0.781	10.0
i4i.xlarge	1.875	10.0
i4i.2xlarge	4.687	12.0
i4i.4xlarge	9.375	25.0

Amazon EBS I/O performance

Amazon EBS optimized instances use an optimized configuration stack and provide additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

For more information, see Amazon EBS–optimized instances.

SSD-based instance store volume I/O performance

The primary data storage for D2, D3, and D3en instances is HDD instance store volumes. The primary data storage for I3 and I3en instances is non-volatile memory express (NVMe) SSD instance store volumes.

Instance store volumes persist only for the life of the instance. When you stop, hibernate, or terminate an instance, the applications and data in its instance store volumes are erased. We recommend that you regularly back up or replicate important data in your instance store volumes. For more information, see Amazon EC2 instance store and SSD instance store volumes.

If you use all the SSD-based instance store volumes available to your instance, you can get up to the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
i3.large	103125	35000

Instance Size	100% Random Read IOPS	Write IOPS
i3.xlarge	206250	70000
i3.2xlarge	412500	180000
i3.4xlarge	825000	360000
i3.8xlarge	1650000	720000
i3.16xlarge	3300000	1440000
i3.metal	3300000	1440000
i3en.large	42500	32500
i3en.xlarge	85000	65000
i3en.2xlarge	170000	130000
i3en.3xlarge	250000	200000
i3en.6xlarge	500000	400000
i3en.12xlarge	1000000	800000
i3en.24xlarge	2000000	1600000
i3en.metal	2000000	1600000
i4i.large	50000	27500
i4i.xlarge	100000	55000
i4i.2xlarge	200000	110000
i4i.4xlarge	400000	220000
i4i.8xlarge	800000	440000
i4i.12xlarge	1200000	660000

Instance Size	100% Random Read IOPS	Write IOPS
i4i.16xlarge	1600000	880000
i4i.24xlarge	2400000	1320000
i4i.32xlarge	3200000	1760000
i4i.metal	3200000	1760000

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

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

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support.

For a comparison of the volume size across all instance types that support instance store volumes, see the Available instance store volumes table.

Release notes

- `r7a.metal-48xl` instances do not support Windows Server 2019 AMIs released before July 2023, and Windows Server 2016 AMIs.
- Instances built on the Nitro System have the following requirements:
  - NVMe drivers must be installed
  - Elastic Network Adapter (ENA) drivers must be installed

  The current AWS Windows AMIs meet these requirements.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- The `d3.8xlarge` and `d3en.12xlarge` instances support a maximum of three attachments, including the root volume. If you exceed the attachment limit when you add a network interface or EBS volume, this causes attachment issues on your instance.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

Windows accelerated computing instances

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Accelerated computing instances use hardware accelerators, or co-processors, to perform some functions, such as floating point number calculations, graphics processing, or data pattern

matching, more efficiently than is possible in software running on CPUs. These instances enable more parallelism for higher throughput on compute-intensive workloads.

If you require high processing capability, you'll benefit from using accelerated computing instances, which provide access to hardware-based compute accelerators such as Graphics Processing Units (GPUs).

Contents

- GPU instances
- Hardware specifications
- Instance performance
- Network performance
- Amazon EBS I/O performance
- SSD-based instance store volume I/O performance
- Release notes
- Install NVIDIA drivers on Windows instances
- Install AMD drivers on Windows instances
- Activate NVIDIA GRID Virtual Applications
- Optimize GPU settings

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.

G5 instances

G5 instances use NVIDIA A10G GPUs and provide high performance for graphics-intensive applications such as remote workstations, video rendering, and cloud gaming, and deep learning models for applications such as natural language processing, computer vision, and recommendation engines. These instances feature up to 8 NVIDIA A10G GPUs, second generation

AMD EPYC processors, up to 100 Gbps of network bandwidth, and up to 7.6 TB of local NVMe SSD storage.

For more information, see Amazon EC2 G5 Instances.

G4ad and G4dn instances

G4ad instances use AMD Radeon Pro V520 GPUs and 2nd generation AMD EPYC processors, and are well-suited for graphics applications such as remote graphics workstations, game streaming, and rendering that leverage industry-standard APIs such as OpenGL, DirectX, and Vulkan. They provide up to 4 AMD Radeon Pro V520 GPUs, 64 vCPUs, 25 Gbps networking, and 2.4 TB local NVMe-based SSD storage.

G4dn instances use NVIDIA Tesla GPUs and provide a cost-effective, high-performance platform for general purpose GPU computing using the CUDA or machine learning frameworks along with graphics applications using DirectX or OpenGL. These instances provide high- bandwidth networking, powerful half and single-precision floating-point capabilities, along with INT8 and INT4 precisions. Each GPU has 16 GiB of GDDR6 memory, making G4dn instances well-suited for machine learning inference, video transcoding, and graphics applications like remote graphics workstations and game streaming in the cloud.

For more information, see Amazon EC2 G4 Instances.

G4dn instances support NVIDIA GRID Virtual Workstation. For more information, see NVIDIA Marketplace offerings.

G3 instances

These instances use NVIDIA Tesla M60 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. G3 instances also provide NVIDIA GRID Virtual Workstation features, such as support for four monitors with resolutions up to 4096x2160, and NVIDIA GRID Virtual Applications. G3 instances are well-suited for applications such as 3D visualizations, graphics-intensive remote workstations, 3D rendering, video encoding, virtual reality, and other server-side graphics workloads requiring massively parallel processing power.

For more information, see Amazon EC2 G3 Instances.

G3 instances support NVIDIA GRID Virtual Workstation and NVIDIA GRID Virtual Applications. To activate either of these features, see Activate NVIDIA GRID Virtual Applications.

G2 instances

These instances use NVIDIA GRID K520 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. NVIDIA GRID GPUs also support NVIDIA's fast capture and encode API operations. Example applications include video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side graphics workloads.

P4de instances offer NVIDIA 80GB-A100s GPUs

P3 instances

These instances use NVIDIA Tesla V100 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models or through a machine learning framework. P3 instances provide high-bandwidth networking, powerful half, single, and double-precision floating-point capabilities, and up to 32 GiB of memory per GPU, which makes them ideal for deep learning, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads. Tesla V100 GPUs do not support graphics mode.

For more information, see Amazon EC2 P3 Instances.

P3 instances support NVIDIA NVLink peer to peer transfers. For more information, see NVIDIA NVLink.

P2 instances

P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. P2 instances provide high-bandwidth networking, powerful single and double precision floating-point capabilities, and 12 GiB of memory per GPU, which makes them ideal for deep learning, graph databases, high-performance databases, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads.

P2 instances support NVIDIA GPUDirect peer to peer transfers. For more information, see NVIDIA GPUDirect.

Hardware specifications

The following is a summary of the hardware specifications for accelerated computing instances. A virtual central processing unit (vCPU) represents a portion of the physical CPU assigned to a virtual

machine (VM). For x86 instances, there are two vCPUs per core. For Graviton instances, there is one vCPU per core.

Instance type	Default vCPUs	Memory (GiB)	Accelerators
g2.2xlarge	8	15.00	1 GPUs
g2.8xlarge	32	60.00	4 GPUs
g3.4xlarge	16	122.00	1 GPUs
g3.8xlarge	32	244.00	2 GPUs
g3.16xlarge	64	488.00	4 GPUs
g4ad.xlarge	4	16.00	1 GPUs
g4ad.2xlarge	8	32.00	1 GPUs
g4ad.4xlarge	16	64.00	1 GPUs
g4ad.8xlarge	32	128.00	2 GPUs
g4ad.16xlarge	64	256.00	4 GPUs
g4dn.xlarge	4	16.00	1 GPUs
g4dn.2xlarge	8	32.00	1 GPUs
g4dn.4xlarge	16	64.00	1 GPUs
g4dn.8xlarge	32	128.00	1 GPUs
g4dn.12xlarge	48	192.00	4 GPUs
g4dn.16xlarge	64	256.00	1 GPUs
g4dn.metal	96	384.00	8 GPUs
g5.xlarge	4	16.00	1 GPUs
g5.2xlarge	8	32.00	1 GPUs

Instance type	Default vCPUs	Memory (GiB)	Accelerators
g5.4xlarge	16	64.00	1 GPUs
g5.8xlarge	32	128.00	1 GPUs
g5.12xlarge	48	192.00	4 GPUs
g5.16xlarge	64	256.00	1 GPUs
g5.24xlarge	96	384.00	4 GPUs
g5.48xlarge	192	768.00	8 GPUs
p2.xlarge	4	61.00	1 GPUs
p2.8xlarge	32	488.00	8 GPUs
p2.16xlarge	64	732.00	16 GPUs
p3.2xlarge	8	61.00	1 GPUs
p3.8xlarge	32	244.00	4 GPUs
p3.16xlarge	64	488.00	8 GPUs
p3dn.24xlarge	96	768.00	8 GPUs

The accelerated computing instances use the following processors.

AMD processors

- **2nd generation AMD EPYC processors (AMD EPYC 7R32)**: G4ad, G5

Intel processors

- **Intel Xeon Scalable processors (Broadwell E5-2686 v4)**: G3, P2, P3
- **Intel Xeon Scalable processors (Skylake 8175)**: P3dn
- **2nd generation Intel Xeon Scalable processors (Cascade Lake P-8259CL)**: G4dn

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Instance performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some accelerated computing instances are EBS-optimized by default at no additional cost. For more information, see Amazon EBS–optimized instances.

Network performance

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

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

> **Note**
>
> Instance types indicated with a † have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth.

Instance type	Network performance	Enhanced networking features
`g2.2xlarge`	Moderate	Not supported
`g2.8xlarge`	High	Not supported
`g3.4xlarge` †	Up to 10 Gigabit	ENA
`g3.8xlarge`	10 Gigabit	ENA
`g3.16xlarge`	25 Gigabit	ENA

Instance type	Network performance	Enhanced networking features
g4ad.xlarge †	Up to 10 Gigabit	ENA
g4ad.2xlarge †	Up to 10 Gigabit	ENA
g4ad.4xlarge †	Up to 10 Gigabit	ENA
g4ad.8xlarge	15 Gigabit	ENA
g4ad.16xlarge	25 Gigabit	ENA
g4dn.xlarge †	Up to 25 Gigabit	ENA
g4dn.2xlarge †	Up to 25 Gigabit	ENA
g4dn.4xlarge †	Up to 25 Gigabit	ENA
g4dn.8xlarge	50 Gigabit	ENA | EFA
g4dn.12xlarge	50 Gigabit	ENA | EFA
g4dn.16xlarge	50 Gigabit	ENA | EFA
g4dn.metal	100 Gigabit	ENA | EFA
g5.xlarge †	Up to 10 Gigabit	ENA
g5.2xlarge †	Up to 10 Gigabit	ENA
g5.4xlarge †	Up to 25 Gigabit	ENA
g5.8xlarge	25 Gigabit	ENA | EFA
g5.12xlarge	40 Gigabit	ENA | EFA
g5.16xlarge	25 Gigabit	ENA | EFA
g5.24xlarge	50 Gigabit	ENA | EFA
g5.48xlarge	100 Gigabit	ENA | EFA

Instance type	Network performance	Enhanced networking features
p2.xlarge	High	ENA
p2.8xlarge	10 Gigabit	ENA
p2.16xlarge	25 Gigabit	ENA
p3.2xlarge †	Up to 10 Gigabit	ENA
p3.8xlarge	10 Gigabit	ENA
p3.16xlarge	25 Gigabit	ENA
p3dn.24xlarge	100 Gigabit	ENA | EFA

The following table shows the baseline and burst bandwidth for instance types that use the network I/O credit mechanism to burst beyond their baseline bandwidth.

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
g3.4xlarge	5	10
g4ad.xlarge	2.0	10.0
g4ad.2xlarge	4.167	10.0
g4ad.4xlarge	8.333	10.0
g4dn.xlarge	5.0	25.0
g4dn.2xlarge	10.0	25.0
g4dn.4xlarge	20.0	25.0
g5.xlarge	2.5	10.0
g5.2xlarge	5.0	10.0

Instance type	Baseline bandwidth (Gbps)	Burst bandwidth (Gbps)
g5.4xlarge	10.0	25.0

Amazon EBS I/O performance

Amazon EBS optimized instances use an optimized configuration stack and provide additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

For more information, see Amazon EBS–optimized instances.

SSD-based instance store volume I/O performance

If you use all the SSD-based instance store volumes available to your instance, you can get up to the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
g4ad.xlarge	10417	8333
g4ad.2xlarge	20833	16667
g4ad.4xlarge	41667	33333
g4ad.8xlarge	83333	66667
g4ad.16xlarge	166666	133332
g4dn.xlarge	42500	32500
g4dn.2xlarge	42500	32500
g4dn.4xlarge	85000	65000
g4dn.8xlarge	250000	200000
g4dn.12xlarge	250000	200000

Instance Size	100% Random Read IOPS	Write IOPS
g4dn.16xlarge	250000	200000
g4dn.metal	500000	400000
g5.xlarge	40625	20313
g5.2xlarge	40625	20313
g5.4xlarge	125000	62500
g5.8xlarge	250000	125000
g5.12xlarge	312500	156250
g5.16xlarge	250000	125000
g5.24xlarge	312500	156250
g5.48xlarge	625000	312500
p3dn.24xlarge	700000	340000

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

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

unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support.

Release notes

- You must launch the instance using an HVM AMI.
- Instances built on the Nitro System have the following requirements:
  - NVMe drivers must be installed
  - Elastic Network Adapter (ENA) drivers must be installed

  The current AWS Windows AMIs meet these requirements.
- GPU-based instances can't access the GPU unless the NVIDIA drivers are installed. For more information, see Install NVIDIA drivers on Windows instances.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- There is a limit of 100 AFIs per Region.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.
- If you launch a multi-GPU instance with a Windows AMI that was created on a single-GPU instance, Windows does not automatically install the NVIDIA driver for all GPUs. You must authorize the driver installation for the new GPU hardware. You can correct this manually in the Device Manager by opening the **Other** device category (the inactive GPUs do not appear under

Display Adapters). For each inactive GPU, open the context (right-click) menu, choose **Update Driver Software**, and then choose the default **Automatic Update** option.

- When using Microsoft Remote Desktop Protocol (RDP), GPUs that use the WDDM driver model are replaced with a non-accelerated Remote Desktop display driver. We recommend that you use a different remote access tool to access your GPU, such as Teradici Cloud Access Software, NICE Desktop Cloud Visualization (DCV), or VNC. You can also use one of the GPU AMIs from the AWS Marketplace because they provide remote access tools that support 3D acceleration.

Install NVIDIA drivers on Windows instances

An instance with an attached NVIDIA GPU, such as a P3 or G4dn instance, must have the appropriate NVIDIA driver installed. Depending on the instance type, you can either download a public NVIDIA driver, download a driver from Amazon S3 that is available only to AWS customers, or use an AMI with the driver pre-installed.

To install AMD drivers on a Linux instance with an attached AMD GPU, such as a G4ad instance, see Install AMD drivers instead. 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 (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 drivers

- OpenCL
- NVIDIA CUDA and related libraries (for example, cuDNN, TensorRT, nvJPEG, and cuBLAS)
- NVENC for video encoding and NVDEC for video decoding

Supported APIs for GRID and gaming drivers

- DirectX, Direct2D, DirectX Video Acceleration, DirectX Raytracing
- OpenCL, OpenGL, and Vulkan
- NVIDIA CUDA and related libraries (for example, cuDNN, TensorRT, nvJPEG, and cuBLAS)

- NVENC for video encoding and NVDEC for video decoding

Available drivers by instance type

The following table summarizes the supported NVIDIA drivers for each GPU instance type.

Instance type	Tesla driver	GRID driver	Gaming driver
G2	Yes	No	No
G3	Yes	Yes	No
G4dn	Yes	Yes	Yes
G5	Yes	Yes	Yes
P2	Yes	No	No
P3	Yes	No	No

[1] This Tesla driver also supports optimized graphics applications specific to the ARM64 platform

Installation options

Use one of the following options to get the NVIDIA drivers required for your GPU instance.

Options

- Option 1: AMIs with the NVIDIA drivers installed
- Option 2: Public NVIDIA drivers
- Option 3: GRID drivers (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

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 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.

To download a public NVIDIA driver

Log on to your Windows instance and download the 64-bit NVIDIA driver appropriate for the instance type from http://www.nvidia.com/Download/Find.aspx. For **Product Type**, **Product Series**, and **Product**, use the options in the following table.

Instance	Product Type	Product Series	Product
G2	GRID	GRID Series	GRID K520
G3	Tesla	M-Class	M60
G4dn	Tesla	T-Series	T4
G5 [1]	Tesla	A-Series	A10
P2	Tesla	K-Series	K80
P3	Tesla	V-Series	V100
P5 [3]	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] P5 instances require driver version 530 or later.

To install the NVIDIA driver on Windows

1. Open the folder where you downloaded the driver and launch the installation file. Follow the instructions to install the driver and reboot your instance as required.
2. Disable the display adapter named **Microsoft Basic Display Adapter** that is marked with a warning icon using Device Manager. Install these Windows features: **Media Foundation** and **Quality Windows Audio Video Experience**.

   > **Important**
   >
   > Don't disable the display adapter named **Microsoft Remote Display Adapter**. If **Microsoft Remote Display Adapter** is disabled your connection might be interrupted and attempts to connect to the instance after it has rebooted might fail.

3. Check Device Manager to verify that the GPU is working correctly.
4. To achieve the best performance from your GPU, complete the optimization steps in Optimize GPU settings.

Option 3: GRID drivers (G5, G4dn, and G3 instances)

These downloads are available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA 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.

> **Important**
>
> **Windows server 2016** is not supported in NVIDIA GRID driver release 14.2 or later.
> **Windows server 2019** is not supported in NVIDIA GRID driver release 17.0 or later.

Considerations

- 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 GRID driver works.

- G3 instances require AWS provided DNS resolution for GRID licensing to work.
- IMDSv2 is only supported with NVIDIA driver version 14.0 or greater.

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.

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.

Prerequisites

- 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*.
- G3 instances require AWS provided DNS resolution for GRID licensing to work.
- IMDSv2 is only supported with NVIDIA driver version 495.x or greater.

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
\GridSwCert.txt"
```

- For version 445.87:

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2020_04.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

- For earlier versions:

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2019_09.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

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 Windows instance.
2. Open the NVIDIA website and select the version of CUDA that you need.
3. For **Installer Type**, select **exe (local)** and then choose **Download**.
4. Using your browser, run the downloaded install file. Follow the instructions to install the CUDA toolkit. You might be required to reboot the instance.

Install AMD drivers on Windows instances

An instance with an attached AMD GPU, such as a G4ad instance, must have the appropriate AMD driver installed. Depending on your requirements, you can either use an AMI with the driver preinstalled or download a driver from Amazon S3.

To install NVIDIA drivers on an instance with an attached NVIDIA GPU, such as a G4dn instance, see Install NVIDIA drivers instead. To install AMD drivers on a Linux instance, see Install AMD drivers on a Linux instance.

Contents

- AMD Radeon Pro Software for Enterprise Driver
- AMIs with the AMD driver installed
- AMD driver download

AMD Radeon Pro Software for Enterprise Driver

The AMD Radeon Pro Software for Enterprise Driver is built to deliver support for professional-grade graphics use cases. Using the driver, you can configure your instances with two 4K displays per GPU.

Supported APIs

- OpenGL, OpenCL
- Vulkan
- DirectX 9 and later
- AMD Advanced Media Framework
- Microsoft Hardware Media Foundation Transform

AMIs with the AMD driver installed

AWS offers different Amazon Machine Images (AMI) that come with the AMD drivers installed. Open Marketplace offerings with the AMD driver.

AMD driver download

If you aren't using an AMI with the AMD driver installed, you can download the AMD driver and install it on your instance. The AMD driver is only supported for Windows Server 2016 and Windows Server 2019 operating systems.

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 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 from Amazon S3 to your desktop using the following PowerShell commands.

```
$Bucket = "ec2-amd-windows-drivers"
$KeyPrefix = "latest" # use "archives" for Windows Server 2016
$LocalPath = "$home\Desktop\AMD"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
    $LocalFileName = $Object.Key
    if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
        $LocalFilePath = Join-Path $LocalPath $LocalFileName
```

```
        Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile
 $LocalFilePath -Region us-east-1
    }
}
```

4. Unzip the downloaded driver file and run the installer using the following PowerShell commands.

```
Expand-Archive $LocalFilePath -DestinationPath "$home\Desktop\AMD\$KeyPrefix" -
Verbose
```

Now, check the content of the new directory. The directory name can be retrieved using the `Get-ChildItem` PowerShell command.

```
Get-ChildItem "$home\Desktop\AMD\$KeyPrefix"
```

The output should be similar to the following:

```
Directory: C:\Users\Administrator\Desktop\AMD\latest

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       10/13/2021  12:52 AM                210414a-365562C-Retail_End_User.2
```

Install the drivers:

```
pnputil /add-driver $home\Desktop\AMD\$KeyPrefix\*.inf /install /subdirs
```

5. Follow the instructions to install the driver and reboot your instance as required.
6. To verify that the GPU is working properly, check Device Manager. You should see "AMD Radeon Pro V520 MxGPU" listed as a display adapter.
7. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol, NICE DCV.

Activate NVIDIA GRID Virtual Applications

To activate the GRID Virtual Applications on G3, G4dn, and G5 instances (NVIDIA GRID Virtual Workstation is enabled by default), you must define the product type for the driver in the registry.

To activate GRID Virtual Applications on Windows instances

1. Run **regedit.exe** to open the registry editor.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.
3. Open the context (right-click) menu on the right pane and choose **New**, **DWORD**.
4. For **Name**, enter **FeatureType** and type `Enter`.
5. Open the context (right-click) menu on **FeatureType** and choose **Modify**.
6. For **Value data**, enter `0` for NVIDIA GRID Virtual Applications and choose **OK**.
7. Open the context (right-click) menu on the right pane and choose **New**, **DWORD**.
8. For **Name**, enter **IgnoreSP** and type `Enter`.
9. Open the context (right-click) menu on **IgnoreSP** and choose **Modify**.
10. For **Value data**, type 1 and choose **OK**.
11. Close the registry editor.

Optimize GPU settings

There are several GPU setting optimizations that you can perform to achieve the best performance on NVIDIA GPU instances. With some of these instance types, the NVIDIA driver uses an autoboost feature, which varies the GPU clock speeds. By disabling autoboost and setting the GPU clock speeds to their maximum frequency, you can consistently achieve the maximum performance with your GPU instances.

The following steps are for optimizing GPU settings on a Windows instance. For Linux instances, see Optimize GPU settings in the *Amazon EC2 User Guide for Linux Instances*.

To optimize GPU settings

1. Open a PowerShell window and navigate to the NVIDIA installation folder.

```
cd "C:\Windows\System32\DriverStore\FileRepository\nv_dispswi.inf_*\"
```

2. [G2, G3, and P2 instances only] Disable the autoboost feature for all GPUs on the instance.

```
.\nvidia-smi --auto-boost-default=0
```

3. Set all GPU clock speeds to their maximum frequency. Use the memory and graphics clock speeds specified in the following commands.

   Some versions of the NVIDIA driver do not support setting the application clock speed, and display the error `"Setting applications clocks is not supported for GPU..."`, which you can ignore.

   - G3 instances:

     ```
     .\nvidia-smi -ac "2505,1177"
     ```

   - G4dn instances:

     ```
     .\nvidia-smi -ac "5001,1590"
     ```

   - G5 instances:

     ```
     .\nvidia-smi -ac "6250,1710"
     ```

   - P2 instances:

     ```
     .\nvidia-smi -ac "2505,875"
     ```

   - P3 and P3dn instances:

     ```
     .\nvidia-smi -ac "877,1530"
     ```

High-performance computing instances

> **Note**
>
> For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

High-performance computing instances are purpose built to offer the best price performance for running HPC workloads at scale on AWS. These instances are ideal for applications that benefit from high-performance processors, such as large, complex simulations and deep learning workloads.

Hpc6id instances

These instances are ideal for running high performance computing (HPC) workloads, such as the following:

- Seismic and Reservoir
- Crash Simulation
- Finite Element Analysis

Hpc7g instances

Hpc7g instances are powered by AWS Graviton3E processors and the next generation AWS Nitro Card. They deliver 200 Gbps advanced networking with Elastic Fabric Adapter (EFA) for low latency and high network performance.

Hpc7g instances are ideal for running compute-intensive HPC applications that benefit from compute and memory bandwidth performance and low network latency, such as computational fluid dynamics, molecular dynamics, and weather simulations.

For more information, see Amazon EC2 Hpc7g instances.

Hpc7a instances

These instances feature up to 300 Gbps networking bandwidth with Elastic Fabric Adapter (EFA) for low latency and high network performance with Message Passing Interface (MPI). They feature up to 192 CPU cores with up to 768 GB of system memory.

Hpc7a instances are ideal for running compute-intensive HPC applications, such as computational fluid dynamics, molecular dynamics, and weather simulations that benefit from large core counts per instance and low network latency.

For more information, see Amazon EC2 Hpc7a instances.

Contents

- Hardware specifications
- Instance performance
- Network performance
- Amazon EBS I/O performance

- SSD-based instance store volume I/O performance
- Release notes

Hardware specifications

The following is a summary of the hardware specifications for high-performance computing instances. A virtual central processing unit (vCPU) represents a portion of the physical CPU assigned to a virtual machine (VM). For x86 instances, there are two vCPUs per core. For Graviton instances, there is one vCPU per core.

Instance type	Default vCPUs	Memory (GiB)
hpc6id.32xlarge	64	1024.00
hpc7a.12xlarge	24	768.00
hpc7a.24xlarge	48	768.00
hpc7a.48xlarge	96	768.00
hpc7a.96xlarge	192	768.00

The high-performance computing instances use the following processors.

AMD processors

- **3rd generation AMD EPYC processors (AMD EPYC 7R13)**: Hpc6a
- **4th generation AMD EPYC processors (AMD EPYC 9R14)**: Hpc7a

For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.

Instance performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some compute optimized instances are EBS-optimized by default at no additional cost. For more information, see Amazon EBS–optimized instances.

Network performance

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

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

> **Note**
>
> Instance types indicated with a † have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth.

Instance type	Network performance	Enhanced networking features
`hpc6id.32xlarge`	200 Gigabit	ENA | EFA
`hpc7a.12xlarge`	300 Gigabit	ENA | EFA
`hpc7a.24xlarge`	300 Gigabit	ENA | EFA
`hpc7a.48xlarge`	300 Gigabit	ENA | EFA
`hpc7a.96xlarge`	300 Gigabit	ENA | EFA

For `32xlarge` and `metal` instance types that support 200 Gbps, at least 2 ENIs, each attached to a different network card, are required on the instance to achieve 200 Gbps throughput. Each ENI attached to a network card can achieve a max of 170 Gbps.

Amazon EBS I/O performance

Amazon EBS optimized instances use an optimized configuration stack and provide additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your

Amazon EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

For more information, see Amazon EBS–optimized instances.

SSD-based instance store volume I/O performance

If you use all the SSD-based instance store volumes available to your instance, you can get up to the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
`hpc6id.32xlarge`	2146664	1073336

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

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

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support.

Release notes

- Instances built on the Nitro System have the following requirements:
  - NVMe drivers must be installed
  - Elastic Network Adapter (ENA) drivers must be installed

  The current AWS Windows AMIs meet these requirements.
- 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.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

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 (For AWS Builder Use) 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 in the *Amazon Q (For AWS Builder Use) 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, the Amazon Q AI assistant 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 (For AWS Builder Use) User Guide.

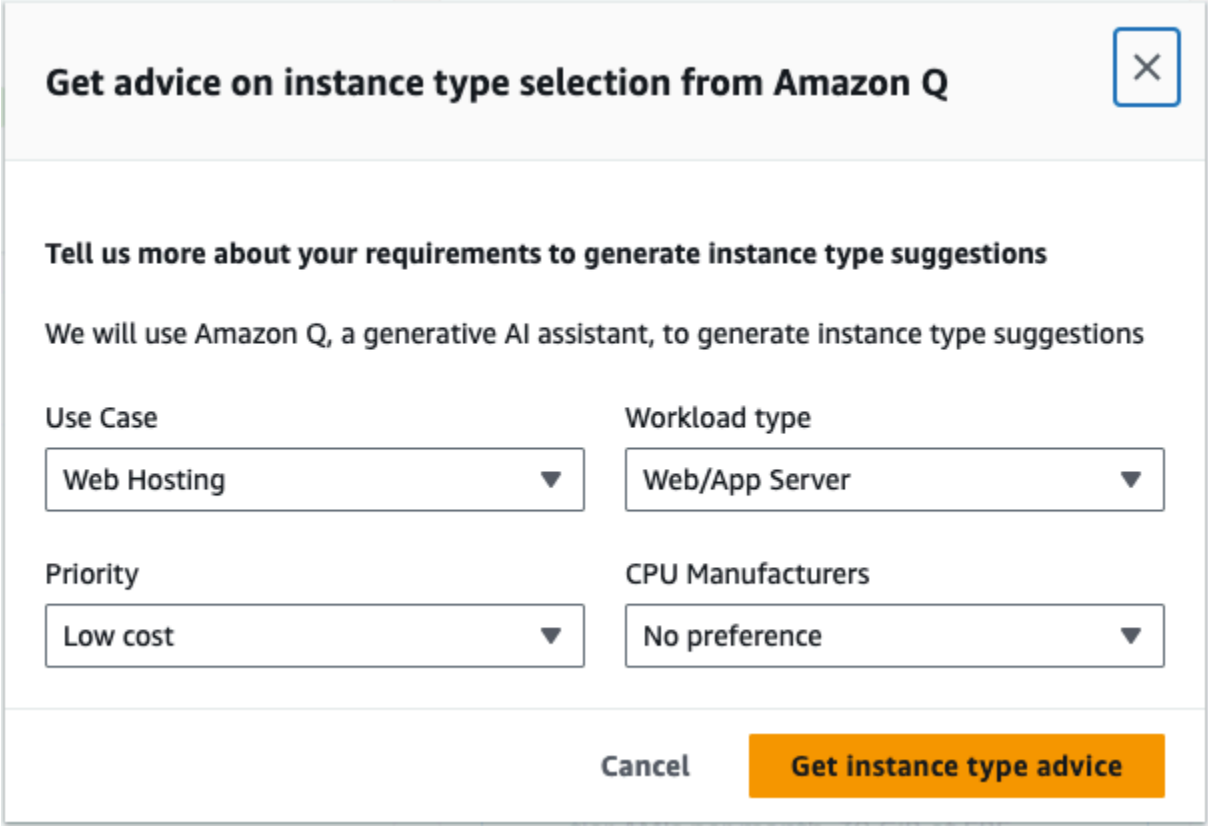
The **Get advice** link appears on the following EC2 console pages:

- 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.



c. Choose **Get instance type advice**.

   The Amazon Q AI assistant 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:

- 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 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.

Compatibility	Use these instructions
Compatible	Change the instance type of an existing instance
Not compatible	Change the instance type by launching a new instance

Considerations for compatible instance types

Consider the following when changing the instance type of an existing instance:

- We recommend that you update the AWS PV driver package before changing the instance type. For more information, see Upgrade PV drivers on Windows instances.
- 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 your instance.
- 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 instance

Use the following instructions to change the instance type of an 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. (Optional) If you configured your Windows instance to use static IP addressing and you change from an instance type that doesn't support enhanced networking to an instance type that does support enhanced networking, you might get a warning about a potential IP address conflict when you reconfigure static IP addressing. To prevent this, enable DHCP on the network interface for your instance before you change the instance type. From your instance, open the **Network and Sharing Center**, open **Internet Protocol Version 4 (TCP/IPv4) Properties** for the network interface, and choose **Obtain an IP address automatically**. Change the instance type and reconfigure static IP addressing on the network interface.
3. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
4. In the navigation pane, choose **Instances**.
5. Select the instance and choose **Instance state**, **Stop instance**. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
6. With the instance still selected, choose **Actions**, **Instance settings**, **Change instance type**. This option is grayed out if the instance state is not `stopped`.
7. On the **Change instance type** page, do the following:

   a. For **Instance type**, select the instance type that you want.

      If the instance type is not in the list, then it's not compatible with the configuration of your instance. Instead, use the following instructions: 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.

8. To start the instance, select the instance and choose **Instance state**, **Start instance**. It can take a few minutes for the instance to enter the `running` state. If your instance won't start, see Troubleshoot changing the instance type.
9. If your instance is running Windows Server 2016 or Windows Server 2019 with EC2Launch v1, connect to your Windows instance and run the following EC2Launch PowerShell script to configure the instance after the instance type is changed. For more information, see Configure a Windows instance using EC2Launch.

> **Important**
>
> The administrator password will reset when you enable the initialize instance EC2 Launch script. You can modify the configuration file to disable the administrator password reset by specifying it in the settings for the initialization tasks. For steps on how to disable password reset, see Configure initialization tasks.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

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 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.
   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 Linux instance types, see Compatibility for changing the instance type in the *User Guide for Linux Instances*.

> **Tip**
>
> For additional guidance on migrating compatible Windows instances from a Xen instance type to a Nitro instance type, see Migrate to latest generation instance types.

Compatibility is determined in the following ways:

Architecture

AMIs are specific to the architecture of the processor, so you must select an instance type with the same processor architecture as the current instance type. For example:

- If the current instance type has a processor based on the Arm architecture, you are limited to the instance types that support a processor based on the Arm architecture, such as C6g and M6g.

- The following instance types are the only instance types that support 32-bit AMIs: `t2.nano`, `t2.micro`, `t2.small`, `t2.medium`, `c3.large`, `t1.micro`, `m1.small`, `m1.medium`, and `c1.medium`. If you are changing the instance type of a 32-bit instance, you are limited to these instance types.

Network adapters

If you switch from a driver for one network adapter to another, the network adapter settings are reset when the operating system creates the new adapter. To reconfigure the settings, you might need access to a local account with administrator permissions. The following are examples of moving from one network adapter to another:

- AWS PV (T2 instances) to Intel 82599 VF (M4 instances)
- Intel 82599 VF (most M4 instances) to ENA (M5 instances)
- ENA (M5 instances) to high-bandwidth ENA (M5n instances)

Enhanced networking

Instance types that support enhanced networking require the necessary drivers installed. For example, instances based on the 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 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`).

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.

AMI

For information about the AMIs required by instance types that support enhanced networking and NVMe, see the Release Notes in the following documentation:

- General purpose instances
- Compute optimized instances
- Memory optimized instances
- Storage optimized instances

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: 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.

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 programatically 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)
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
m5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5.4xlarge [2]	4750		593.75		18750	
m5.8xlarge [2]	6800		850.0		30000	
m5.12xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5.16xlarge [2]	13600		1700.0		60000	
m5.24xlarge [2]	19000		2375.0		80000	
m5.metal [2]	19000		2375.0		80000	
m5a.large [1]	650	2880	81.25	360.00	3600	16000
m5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5a.4xlarge [2]	2880		360.0		16000	
m5a.8xlarge [2]	4750		593.75		20000	
m5a.12xlarge [2]	6780		847.5		30000	
m5a.16xlarge [2]	9500		1187.5		40000	
m5a.24xlarge [2]	13750		1718.75		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5ad.large [1]	650	2880	81.25	360.00	3600	16000
m5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5ad.4xlarge [2]	2880		360.0		16000	
m5ad.8xlarge [2]	4750		593.75		20000	
m5ad.12xlarge [2]	6780		847.5		30000	
m5ad.16xlarge [2]	9500		1187.5		40000	
m5ad.24xlarge [2]	13750		1718.75		60000	
m5d.large [1]	650	4750	81.25	593.75	3600	18750
m5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5d.4xlarge [2]	4750		593.75		18750	
m5d.8xlarge [2]	6800		850.0		30000	
m5d.12xlarge [2]	9500		1187.5		40000	
m5d.16xlarge [2]	13600		1700.0		60000	
m5d.24xlarge [2]	19000		2375.0		80000	
m5d.metal [2]	19000		2375.0		80000	
m5dn.large [1]	650	4750	81.25	593.75	3600	18750
m5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5dn.4xlarge [2]	4750		593.75		18750	
m5dn.8xlarge [2]	6800		850.0		30000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5dn.12xlarge [2]	9500		1187.5		40000	
m5dn.16xlarge [2]	13600		1700.0		60000	
m5dn.24xlarge [2]	19000		2375.0		80000	
m5dn.metal [2]	19000		2375.0		80000	
m5n.large [1]	650	4750	81.25	593.75	3600	18750
m5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5n.4xlarge [2]	4750		593.75		18750	
m5n.8xlarge [2]	6800		850.0		30000	
m5n.12xlarge [2]	9500		1187.5		40000	
m5n.16xlarge [2]	13600		1700.0		60000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5n.24xlarge [2]	19000		2375.0		80000	
m5n.metal [2]	19000		2375.0		80000	
m5zn.large [1]	800	3170	100.00	396.25	3333	13333
m5zn.xlarge [1]	1564	3170	195.50	396.25	6667	13333
m5zn.2xlarge [2]	3170		396.25		13333	
m5zn.3xlarge [2]	4750		593.75		20000	
m5zn.6xlarge [2]	9500		1187.5		40000	
m5zn.12xlarge [2]	19000		2375.0		80000	
m5zn.metal [2]	19000		2375.0		80000	
m6a.large [1]	650	10000	81.25	1250.00	3600	40000
m6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6a.8xlarge [2]	10000		1250.0		40000	
m6a.12xlarge [2]	15000		1875.0		60000	
m6a.16xlarge [2]	20000		2500.0		80000	
m6a.24xlarge [2]	30000		3750.0		120000	
m6a.32xlarge [2]	40000		5000.0		160000	
m6a.48xlarge [2]	40000		5000.0		240000	
m6a.metal [2]	40000		5000.0		240000	
m6i.large [1]	650	10000	81.25	1250.00	3600	40000
m6i.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)
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
m6id.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)
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.metal [2]	40000		5000.0		160000	
m6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
m6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000

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.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
m6in.8xlarge [2]	25000		3125.0		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.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.metal [2]	100000		12500.0		400000	
m7a.medium [1]	325	10000	40.62	1250.00	2500	40000
m7a.large [1]	650	10000	81.25	1250.00	3600	40000
m7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7a.8xlarge [2]	10000		1250.0		40000	

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.12xlarge [2]	15000		1875.0		60000	
m7a.16xlarge [2]	20000		2500.0		80000	
m7a.24xlarge [2]	30000		3750.0		120000	
m7a.32xlarge [2]	40000		5000.0		160000	
m7a.48xlarge [2]	40000		5000.0		240000	
m7a.metal-48xl [2]	40000		5000.0		240000	
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	

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.12xlarge [2]	15000		1875.0		60000	
m7i.16xlarge [2]	20000		2500.0		80000	
m7i.24xlarge [2]	30000		3750.0		120000	
m7i.48xlarge [2]	40000		5000.0		240000	
m7i.metal-24xl [2]	30000		3750.0		120000	
m7i.metal-48xl [2]	40000		5000.0		240000	
m7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
m7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
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

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
t3.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
t3a.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	

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)
c5.9xlarge [2]	9500		1187.5		40000	
c5.12xlarge [2]	9500		1187.5		40000	
c5.18xlarge [2]	19000		2375.0		80000	
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	

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)
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
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

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.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5d.4xlarge [2]	4750		593.75		20000	
c5d.9xlarge [2]	9500		1187.5		40000	
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	

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)
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
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	

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.32xlarge [2]	40000		5000.0		160000	
c6a.48xlarge [2]	40000		5000.0		240000	
c6a.metal [2]	40000		5000.0		240000	
c6i.large [1]	650	10000	81.25	1250.00	3600	40000
c6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6i.8xlarge [2]	10000		1250.0		40000	
c6i.12xlarge [2]	15000		1875.0		60000	
c6i.16xlarge [2]	20000		2500.0		80000	
c6i.24xlarge [2]	30000		3750.0		120000	
c6i.32xlarge [2]	40000		5000.0		160000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6i.metal [2]	40000		5000.0		160000	
c6id.large [1]	650	10000	81.25	1250.00	3600	40000
c6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6id.8xlarge [2]	10000		1250.0		40000	
c6id.12xlarge [2]	15000		1875.0		60000	
c6id.16xlarge [2]	20000		2500.0		80000	
c6id.24xlarge [2]	30000		3750.0		120000	
c6id.32xlarge [2]	40000		5000.0		160000	
c6id.metal [2]	40000		5000.0		160000	
c6in.large [1]	1562	25000	195.31	3125.00	6250	100000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
c6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
c6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
c6in.8xlarge [2]	25000		3125.0		100000	
c6in.12xlarge [2]	37500		4687.5		150000	
c6in.16xlarge [2]	50000		6250.0		200000	
c6in.24xlarge [2]	75000		9375.0		300000	
c6in.32xlarge [2]	100000		12500.0		400000	
c6in.metal [2]	100000		12500.0		400000	
c7a.medium [1]	325	10000	40.62	1250.00	2500	40000
c7a.large [1]	650	10000	81.25	1250.00	3600	40000

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
c7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7a.8xlarge [2]	10000		1250.0		40000	
c7a.12xlarge [2]	15000		1875.0		60000	
c7a.16xlarge [2]	20000		2500.0		80000	
c7a.24xlarge [2]	30000		3750.0		120000	
c7a.32xlarge [2]	40000		5000.0		160000	
c7a.48xlarge [2]	40000		5000.0		240000	
c7a.metal-48xl [2]	40000		5000.0		240000	
c7i.large [1]	650	10000	81.25	1250.00	3600	40000
c7i.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)
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	
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	
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	

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)
r5.12xlarge [2]	9500		1187.5		40000	
r5.16xlarge [2]	13600		1700.0		60000	
r5.24xlarge [2]	19000		2375.0		80000	
r5.metal [2]	19000		2375.0		80000	
r5a.large [1]	650	2880	81.25	360.00	3600	16000
r5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5a.4xlarge [2]	2880		360.0		16000	
r5a.8xlarge [2]	4750		593.75		20000	
r5a.12xlarge [2]	6780		847.5		30000	
r5a.16xlarge [2]	9500		1187.5		40000	
r5a.24xlarge [2]	13570		1696.25		60000	

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.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
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

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.4xlarge [2]	10000		1250.0		43333	
r5b.8xlarge [2]	20000		2500.0		86667	
r5b.12xlarge [2]	30000		3750.0		130000	
r5b.16xlarge [2]	40000		5000.0		173333	
r5b.24xlarge [2]	60000		7500.0		260000	
r5b.metal [2]	60000		7500.0		260000	
r5d.large [1]	650	4750	81.25	593.75	3600	18750
r5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5d.4xlarge [2]	4750		593.75		18750	
r5d.8xlarge [2]	6800		850.0		30000	
r5d.12xlarge [2]	9500		1187.5		40000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5d.16xlarge [2]	13600		1700.0		60000	
r5d.24xlarge [2]	19000		2375.0		80000	
r5d.metal [2]	19000		2375.0		80000	
r5dn.large [1]	650	4750	81.25	593.75	3600	18750
r5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5dn.4xlarge [2]	4750		593.75		18750	
r5dn.8xlarge [2]	6800		850.0		30000	
r5dn.12xlarge [2]	9500		1187.5		40000	
r5dn.16xlarge [2]	13600		1700.0		60000	
r5dn.24xlarge [2]	19000		2375.0		80000	

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.metal [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
r5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5n.4xlarge [2]	4750		593.75		18750	
r5n.8xlarge [2]	6800		850.0		30000	
r5n.12xlarge [2]	9500		1187.5		40000	
r5n.16xlarge [2]	13600		1700.0		60000	
r5n.24xlarge [2]	19000		2375.0		80000	
r5n.metal [2]	19000		2375.0		80000	
r6a.large [1]	650	10000	81.25	1250.00	3600	40000
r6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

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.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		1250.0		40000	
r6a.12xlarge [2]	15000		1875.0		60000	
r6a.16xlarge [2]	20000		2500.0		80000	
r6a.24xlarge [2]	30000		3750.0		120000	
r6a.32xlarge [2]	40000		5000.0		160000	
r6a.48xlarge [2]	40000		5000.0		240000	
r6a.metal [2]	40000		5000.0		240000	
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

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)
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
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

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.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
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	

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.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
r6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6id.8xlarge [2]	10000		1250.0		40000	
r6id.12xlarge [2]	15000		1875.0		60000	

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.16xlarge [2]	20000		2500.0		80000	
r6id.24xlarge [2]	30000		3750.0		120000	
r6id.32xlarge [2]	40000		5000.0		160000	
r6id.metal [2]	40000		5000.0		160000	
r7a.medium [1]	325	10000	40.62	1250.00	2500	40000
r7a.large [1]	650	10000	81.25	1250.00	3600	40000
r7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7a.8xlarge [2]	10000		1250.0		40000	
r7a.12xlarge [2]	15000		1875.0		60000	

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.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	
r7i.large [1]	650	10000	81.25	1250.00	3600	40000
r7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7i.8xlarge [2]	10000		1250.0		40000	
r7i.12xlarge [2]	15000		1875.0		60000	
r7i.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7i.24xlarge [2]	30000		3750.0		120000	
r7i.48xlarge [2]	40000		5000.0		240000	
r7i.metal-24xl [2]	30000		3750.0		120000	
r7i.metal-48xl [2]	40000		5000.0		240000	
r7iz.large [1]	792	10000	99.00	1250.00	3600	40000
r7iz.xlarge [1]	1584	10000	198.00	1250.00	6667	40000
r7iz.2xlarge [1]	3168	10000	396.00	1250.00	13333	40000
r7iz.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7iz.8xlarge [2]	10000		1250.0		40000	
r7iz.12xlarge [2]	19000		2375.0		76000	
r7iz.16xlarge [2]	20000		2500.0		80000	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
r7iz.32xlarge [2]	40000		5000.0		160000	
r7iz.metal-16xl [2]	20000		2500.0		80000	
r7iz.metal-32xl [2]	40000		5000.0		160000	
u-3tb1.56xlarge [2]	19000		2375.0		80000	
u-6tb1.56xlarge [2]	38000		4750.0		160000	
u-6tb1.112xlarge [2]	38000		4750.0		160000	
u-6tb1.metal [2]	38000		4750.0		160000	
u-9tb1.112xlarge [2]	38000		4750.0		160000	
u-9tb1.metal [2]	38000		4750.0		160000	
u-12tb1.112xlarge [2]	38000		4750.0		160000	
u-12tb1.metal [2]	38000		4750.0		160000	

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-18tb1.112xlarge [2]	38000		4750.0		160000	
u-18tb1.metal [2]	38000		4750.0		160000	
u-24tb1.112xlarge [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	
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

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.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000
x2iedn.4xlarge [1]	10000	20000	1250.00	2500.00	32500	65000
x2iedn.8xlarge [2]	20000		2500.0		65000	
x2iedn.16xlarge [2]	40000		5000.0		130000	
x2iedn.24xlarge [2]	60000		7500.0		195000	
x2iedn.32xlarge [2]	80000		10000.0		260000	
x2iedn.metal [2]	80000		10000.0		260000	
x2iezn.2xlarge [2]	3170		396.25		13333	
x2iezn.4xlarge [2]	4750		593.75		20000	
x2iezn.6xlarge [2]	9500		1187.5		40000	
x2iezn.8xlarge [2]	12000		1500.0		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)
x2iezn.12xlarge [2]	19000		2375.0		80000	
x2iezn.metal [2]	19000		2375.0		80000	
x1e.xlarge [2]	500		62.5		3700	
x1e.2xlarge [2]	1000		125.0		7400	
x1e.4xlarge [2]	1750		218.75		10000	
x1e.8xlarge [2]	3500		437.5		20000	
x1e.16xlarge [2]	7000		875.0		40000	
x1e.32xlarge [2]	14000		1750.0		80000	
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	

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.3xlarge [2]	4750		593.75		20000	
z1d.6xlarge [2]	9500		1187.5		40000	
z1d.12xlarge [2]	19000		2375.0		80000	
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	

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)
d2.2xlarge [2]	1000		125.0		8000	
d2.4xlarge [2]	2000		250.0		16000	
d2.8xlarge [2]	4000		500.0		32000	
d3.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3.4xlarge [2]	2800		350.0		15000	
d3.8xlarge [2]	5000		625.0		30000	
d3en.xlarge [1]	850	2800	106.25	350.00	5000	15000
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	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
d3en.8xlarge [2]	5000		625.0		30000	
d3en.12xlarge [2]	7000		875.0		40000	
h1.2xlarge [2]	1750		218.75		12000	
h1.4xlarge [2]	3500		437.5		20000	
h1.8xlarge [2]	7000		875.0		40000	
h1.16xlarge [2]	14000		1750.0		80000	
i3.large [2]	425		53.125		3000	
i3.xlarge [2]	850		106.25		6000	
i3.2xlarge [2]	1700		212.5		12000	
i3.4xlarge [2]	3500		437.5		16000	
i3.8xlarge [2]	7000		875.0		32500	
i3.16xlarge [2]	14000		1750.0		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)
i3.metal [2]	19000		2375.0		80000	
i3en.large [1]	576	4750	72.10	593.75	3000	20000
i3en.xlarge [1]	1153	4750	144.20	593.75	6000	20000
i3en.2xlarge [1]	2307	4750	288.39	593.75	12000	20000
i3en.3xlarge [1]	3800	4750	475.00	593.75	15000	20000
i3en.6xlarge [2]	4750		593.75		20000	
i3en.12xlarge [2]	9500		1187.5		40000	
i3en.24xlarge [2]	19000		2375.0		80000	
i3en.metal [2]	19000		2375.0		80000	
i4i.large [1]	625	10000	78.12	1250.00	2500	40000
i4i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000

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.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	

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)
g3.4xlarge [2]	3500		437.5		20000	
g3.8xlarge [2]	7000		875.0		40000	
g3.16xlarge [2]	14000		1750.0		80000	
g4ad.xlarge [1]	400	3170	50.00	396.25	1700	13333
g4ad.2xlarge [1]	800	3170	100.00	396.25	3400	13333
g4ad.4xlarge [1]	1580	3170	197.50	396.25	6700	13333
g4ad.8xlarge [2]	3170		396.25		13333	
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	

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.8xlarge [2]	9500		1187.5		40000	
g4dn.12xlarge [2]	9500		1187.5		40000	
g4dn.16xlarge [2]	9500		1187.5		40000	
g4dn.metal [2]	19000		2375.0		80000	
g5.xlarge [1]	700	3500	87.50	437.50	3000	15000
g5.2xlarge [1]	850	3500	106.25	437.50	3500	15000
g5.4xlarge [2]	4750		593.75		20000	
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	

Instance size	Baseline bandwidth (Mbps)	Maximum bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Maximum throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)	Maximum IOPS (16 KiB I/O)
g5.48xlarge [2]	19000		2375.0		80000	
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	

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)
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

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 programatically 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
g2.2xlarge	1000	125.0	8000
i2.xlarge	500	62.5	4000
i2.2xlarge	1000	125.0	8000
i2.4xlarge	2000	250.0	16000
m1.large	500	62.5	4000
m1.xlarge	1000	125.0	8000
m2.2xlarge	500	62.5	4000
m2.4xlarge	1000	125.0	8000
m3.xlarge	500	62.5	4000
m3.2xlarge	1000	125.0	8000
r3.xlarge	500	62.5	4000
r3.2xlarge	1000	125.0	8000
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, 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.MaximumIo
\"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.MaximumIo
\"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 Windows instance
- Stop and start your instance
- Hibernate your Amazon EC2 instance
- Reboot your instance
- Instance retirement
- Terminate your instance
- Recover your instance
- Configure your Windows instance
- Identify EC2 Windows 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

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.

> **Note**
>
> Savings Plans also offer significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. With Savings Plans, you make a commitment to a consistent usage amount, measured in USD per hour. This provides you with the flexibility to use the instance configurations that best meet your needs and continue to save money, instead of making a commitment to a specific instance configuration. For more information, see the AWS Savings Plans User Guide.

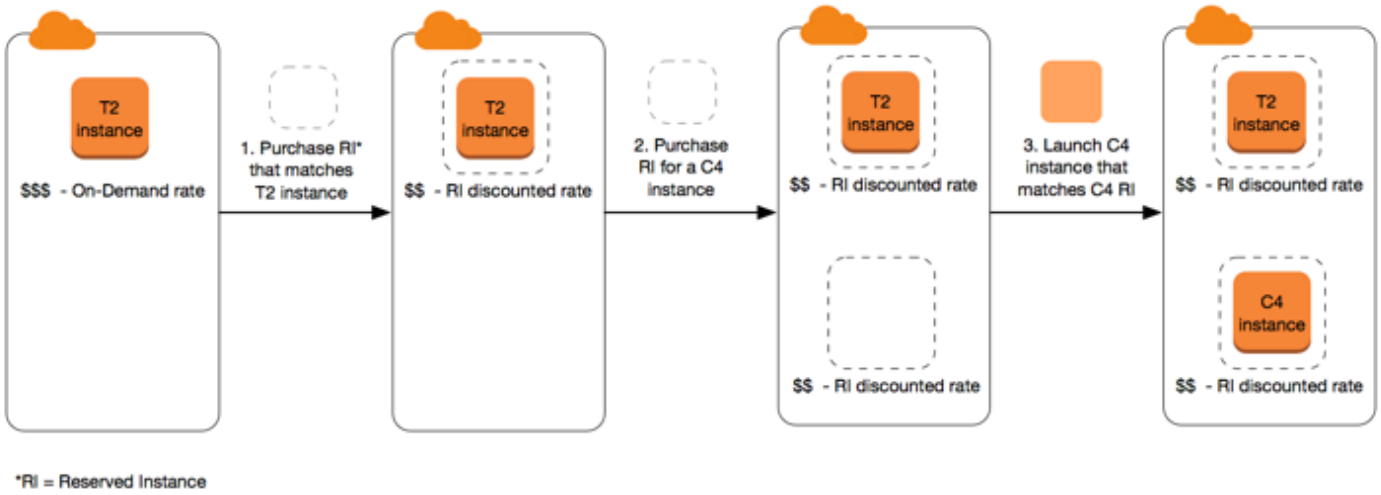
Reserved Instances topics

- Reserved Instance overview
- 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.



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 *not* reserve capacity.	A zonal Reserved Instance reserves capacity in the specified Availability Zone.
Availability Zone flexibility	The Reserved Instance discount applies to instance usage in any Availability Zone in the specified Region.	No Availability Zone flexibility—the Reserved Instance discount applies to instance usage in the specified Availability Zone only.
Instance size flexibility	The Reserved Instance discount applies to instance usage within the instance family, regardless of size. Only supported on Amazon Linux/Unix Reserved	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
	Instances with default tenancy. For more information, see Instance size flexibility determined by normalization factor.	
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

	Standard Reserved Instance	Convertible Reserved Instance
		new attributes, including instance family, instance type, platform, scope, or tenancy. For more information, 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

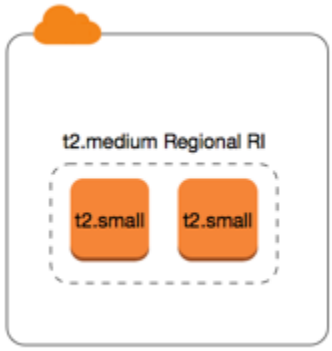
Instance size flexibility is determined by the normalization factor of the instance size. The discount applies either fully or partially to running instances of the same instance family, depending on the instance size of the reservation, in any Availability Zone in the Region. The only attributes that 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.

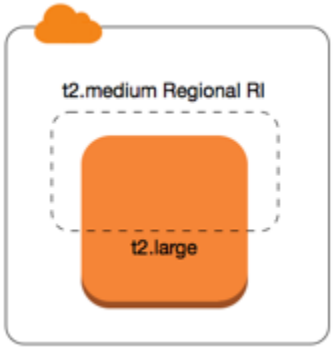
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64

Instance size	Normalization factor
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
112xlarge	896

For example, a `t2.medium` instance has a normalization factor of 2. If you purchase a `t2.medium` default tenancy Amazon Linux/Unix Reserved Instance in the US East (N. Virginia) and you have two running `t2.small` instances in your account in that Region, the billing benefit is applied in full to both instances.



Or, if you have one `t2.large` instance running in your account in the US East (N. Virginia) Region, the billing benefit is applied to 50% of the usage of the instance.



The normalization factor is also applied when modifying Reserved Instances. For more information, see Modify Reserved Instances.

Normalization factor for bare metal instances

Instance size flexibility also applies to bare metal instances within the instance family. If you have regional Amazon Linux/Unix Reserved Instances with shared tenancy on bare metal instances, you can benefit from the Reserved Instance savings within the same instance family. The opposite is also true: if you have regional Amazon Linux/Unix Reserved Instances with shared tenancy on instances in the same family as a bare metal instance, you can benefit from the Reserved Instance savings on the bare metal instance.

The `metal` instance size does not have a single normalization factor. A bare metal instance has the same normalization factor as the equivalent virtualized instance size within the same instance family. For example, an `i3.metal` instance has the same normalization factor as an `i3.16xlarge` instance.

Instance size	Normalization factor
`m5zn.metal` | `z1d.metal`	96
`i3.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

Instance size	Normalization factor
`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.

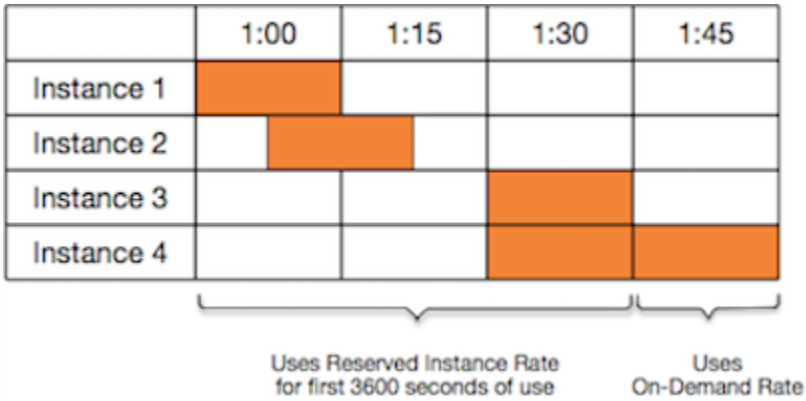
A Reserved Instance billing benefit can be applied to a running instance on a per-second basis.

A Reserved Instance billing benefit can apply to a maximum of 3600 seconds (one hour) of instance usage per clock-hour. You can run multiple instances concurrently, but can only receive the benefit of the Reserved Instance discount for a total of 3600 seconds per clock-hour; instance usage that exceeds 3600 seconds in a clock-hour is billed at the On-Demand rate.

For example, if you purchase one `m4.xlarge` Reserved Instance and run four `m4.xlarge` instances concurrently for one hour, one instance is charged at one hour of Reserved Instance usage and the other three instances are charged at three hours of On-Demand usage.

However, if you purchase one `m4.xlarge` Reserved Instance and run four `m4.xlarge` instances for 15 minutes (900 seconds) each within the same hour, the total running time for the instances is one hour, which results in one hour of Reserved Instance usage and 0 hours of On-Demand usage.

If multiple eligible instances are running concurrently, the Reserved Instance billing benefit is applied to all the instances at the same time up to a maximum of 3600 seconds in a clock-hour; thereafter, On-Demand rates apply.



Cost Explorer on the Billing and Cost Management console enables you to analyze the savings against running On-Demand Instances. The Reserved Instances FAQ includes an example of a list value calculation.

If you close your AWS account, On-Demand billing for your resources stops. However, if you have any Reserved Instances in your account, you continue to receive a bill for these until they expire.

Viewing your bill

You can find out about the charges and fees to your account by viewing the AWS Billing and Cost Management console.

- The **Dashboard** displays a spend summary for your account.
- On the **Bills** page, under **Details** expand the **Elastic Compute Cloud** section and the Region to get billing information about your Reserved Instances.

You can view the charges online, or you can download a CSV file.

You can also track your Reserved Instance utilization using the AWS Cost and Usage Report. For more information, see Reserved Instances under Cost and Usage Report in the *AWS Billing 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.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

For example, for a 1-year Partial Upfront `t2.small` Reserved Instance, assume the upfront price is $60.00 and the hourly rate is $0.007. This provides a list value of $121.32.

```
121.32 = 60.00 + (0.007 * 8760)
```

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. If, at the time of purchase, there are offerings similar to your choice but at a lower price, AWS sells you the offerings at the lower price.

Before you confirm your purchase, review the details of the Reserved Instance that you plan to buy, and make sure that all the parameters are accurate. After you purchase a Reserved Instance (either from a third-party seller in the Reserved Instance Marketplace or from AWS), you cannot cancel your purchase.

> **Note**
>
> To purchase and modify Reserved Instances, ensure that your 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 Windows platforms for Reserved Instances:

- Windows
- Windows with SQL Server Standard
- Windows with SQL Server Web
- Windows with SQL Server Enterprise

When you purchase a Reserved Instance, you must choose an offering for a *platform* that represents the operating system for your instance.

- For Windows with SQL Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web, you must choose offerings for those specific platforms.
- For all other Windows versions, choose an offering for the **Windows** platform.

> **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 Linux, see Choosing a platform in the *Amazon EC2 User Guide for Linux Instances*.

Queue your purchase

By default, when you purchase a Reserved Instance, the purchase is made immediately. Alternatively, you can queue your purchases for a future date and time. For example, you can queue a purchase for around the time that an existing Reserved Instance expires. This can help you ensure that you have uninterrupted coverage.

You can queue purchases for regional Reserved Instances, but not zonal Reserved Instances or Reserved Instances from other sellers. You can queue a purchase up to three years in advance. On the scheduled date and time, the purchase is made using the default payment method. After the payment is successful, the billing benefit is applied.

You can view your queued purchases in the Amazon EC2 console. The status of a queued purchase is **queued**. You can cancel a queued purchase any time before its scheduled time. For details, see Cancel a queued purchase.

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 \
    --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, 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 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.

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
- Submit modification requests
- Troubleshoot modification requests

Requirements and restrictions for modification

You can modify these attributes as follows.

Modifiable attribute	Supported platforms	Limitations and considera tions
Change **Availability Zones** within the same Region	Linux and Windows	-
Change the **scope** from Avai lability Zone to Region and vice versa	Linux and Windows	A zonal Reserved Instance is scoped to an Availability Zone and reserves capacity in that Availability Zone. If you change the scope from Availability Zone to Region (in other words, from zonal to regional), you lose the capacity reservation benefit. A regional Reserved Instance is scoped to a Region. Your Reserved Instance discount can apply to instances running in any Availabil ity Zone in that Region.

Modifiable attribute	Supported platforms	Limitations and considera tions
		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). For more information, see How Reserved Instances are applied.
Change the **instance size** within the same instance family and generation	Linux/UNIX only Instance size flexibility is not available for Reserved Instances on the other platforms, which include Linux with SQL Server Standard, Linux with SQL Server Web, Linux with SQL Server Enterprise, Red Hat Enterprise Linux, SUSE Linux, Windows, Windows with SQL Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web.	The reservation must use default tenancy. Some instance families are not supported, because there are no other sizes available . For more information, see Support for modifying instance sizes in the *Amazon EC2 User Guide for Linux Instances*.

Requirements

Amazon EC2 processes your modification request if there is sufficient capacity for your new configuration (if applicable), and if the following conditions are met:

- The Reserved Instance cannot be modified before or at the same time that you purchase it
- The Reserved Instance must be active
- There cannot be a pending modification request
- The Reserved Instance is not listed in the Reserved Instance Marketplace
- The original Reserved Instances are all Standard Reserved Instances or all Convertible Reserved Instances, not some of each type
- The original Reserved Instances must expire within the same hour, if they are Standard Reserved Instances
- The Reserved Instance is not a G4, G4ad, G4dn, G5, G5g, Inf1, or Inf2 instance.

Submit modification requests

Before you modify your Reserved Instances, ensure that you have read the applicable restrictions.

New console

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Actions**, **Modify Reserved Instances**.

   > **Note**
   >
   > If your Reserved Instances are not in the active state or cannot be modified, **Modify Reserved Instances** is disabled.

3. The first entry in the modification table displays attributes of the selected Reserved Instances, and at least one target configuration beneath it. The **Units** column displays the total instance size footprint. Choose **Add** for each new configuration to add. Modify the attributes as needed for each configuration.

- **Scope**: Choose whether the configuration applies to an Availability Zone or to the whole Region.
- **Availability Zone**: Choose the required Availability Zone. Not applicable for regional Reserved Instances.
- **Count**: Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose **Add**, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.

4. Choose **Continue**.
5. To confirm your modification choices when you finish specifying your target configurations, choose **Submit modifications**.
6. You can determine the status of your modification request by looking at the **State** column in the Reserved Instances screen. The following are the possible states.

   - **active *(pending modification)*** — Transition state for original Reserved Instances
   - **retired *(pending modification)*** — Transition state for original Reserved Instances while new Reserved Instances are being created
   - **retired** — Reserved Instances successfully modified and replaced
   - **active** — One of the following:
     - New Reserved Instances created from a successful modification request
     - Original Reserved Instances after a failed modification request

Old console

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Actions**, **Modify Reserved Instances**.

> **Note**
>
> If your Reserved Instances are not in the active state or cannot be modified, **Modify Reserved Instances** is disabled.

3. The first entry in the modification table displays attributes of selected Reserved Instances, and at least one target configuration beneath it. The **Units** column displays the total instance size footprint. Choose **Add** for each new configuration to add. Modify the attributes as needed for each configuration, and then choose **Continue**:
   - **Scope**: Choose whether the configuration applies to an Availability Zone or to the whole Region.
   - **Availability Zone**: Choose the required Availability Zone. Not applicable for regional Reserved Instances.
   - **Count**: Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose **Add**, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.
4. To confirm your modification choices when you finish specifying your target configurations, choose **Submit Modifications**.
5. You can determine the status of your modification request by looking at the **State** column in the Reserved Instances screen. The following are the possible states.
   - **active *(pending modification)*** — Transition state for original Reserved Instances
   - **retired *(pending modification)*** — Transition state for original Reserved Instances while new Reserved Instances are being created
   - **retired** — Reserved Instances successfully modified and replaced
   - **active** — One of the following:
     - New Reserved Instances created from a successful modification request
     - Original Reserved Instances after a failed modification request

To modify your Reserved Instances using the command line

1. To modify your Reserved Instances, you can use one of the following commands:

- modify-reserved-instances (AWS CLI)
- Edit-EC2ReservedInstance (AWS Tools for Windows PowerShell)

2. To get the status of your modification request (`processing`, `fulfilled`, or `failed`), use one of the following commands:
   - describe-reserved-instances-modifications (AWS CLI)
   - Get-EC2ReservedInstancesModification (AWS Tools for Windows PowerShell)

Troubleshoot modification requests

If the target configuration settings that you requested were unique, you receive a message that your request is being processed. At this point, Amazon EC2 has only determined that the parameters of your modification request are valid. Your modification request can still fail during processing due to unavailable capacity.

In some situations, you might get a message indicating incomplete or failed modification requests instead of a confirmation. Use the information in such messages as a starting point for resubmitting another modification request. Ensure that you have read the applicable restrictions 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 whole Convertible Reserved Instances + 1 additional Convertible Reserved
 Instance
```

The fourth Convertible Reserved Instance has the same end date as the other three. If you are exchanging Partial or All Upfront Convertible Reserved Instances, you pay the true-up cost for the fourth reservation. If the remaining upfront cost of your Convertible Reserved Instances is $500, and the new reservation would normally cost $600 on a prorated basis, you are charged $100.

```
$600 prorated upfront cost of new reservations - $500 remaining upfront cost of old
 reservations = $100 difference
```

Merge Convertible Reserved Instances

If you merge two or more Convertible Reserved Instances, the term of the new Convertible Reserved Instance must be the same as the old Convertible Reserved Instances, or the highest of the Convertible Reserved Instances. The expiration date for the new Convertible Reserved Instance is the expiration date that's furthest in the future.

For example, you have the following Convertible Reserved Instances in your account:

Reserved Instance ID	Term	Expiration date
aaaa1111	1-year	2018-12-31
bbbb2222	1-year	2018-07-31
cccc3333	3-year	2018-06-30
dddd4444	3-year	2019-12-31

- You can merge `aaaa1111` and bbbb2222 and exchange them for a 1-year Convertible Reserved Instance. You cannot exchange them for a 3-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2018-12-31.
- You can merge bbbb2222 and `cccc3333` and exchange them for a 3-year Convertible Reserved Instance. You cannot exchange them for a 1-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2018-07-31.
- You can merge `cccc3333` and dddd4444 and exchange them for a 3-year Convertible Reserved Instance. You cannot exchange them for a 1-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2019-12-31.

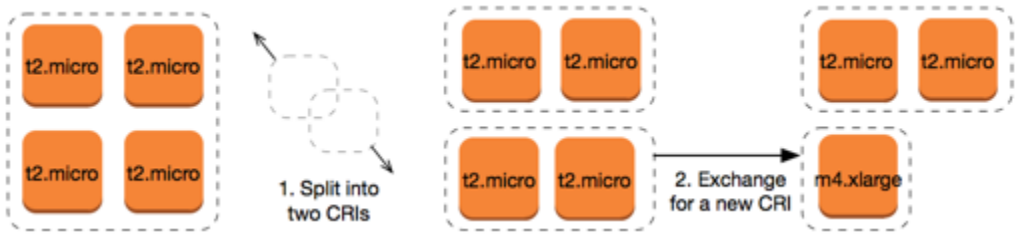
Exchange a portion of a Convertible Reserved Instance

You can use the modification process to split your Convertible Reserved Instance into smaller reservations, and then exchange one or more of the new reservations for a new Convertible Reserved Instance. The following examples demonstrate how you can do this.

Example Example: Convertible Reserved Instance with multiple instances

In this example, you have a `t2.micro` Convertible Reserved Instance with four instances in the reservation. To exchange two `t2.micro` instances for an `m4.xlarge` instance:

1. Modify the `t2.micro` Convertible Reserved Instance by splitting it into two `t2.micro` Convertible Reserved Instances with two instances each.
2. Exchange one of the new `t2.micro` Convertible Reserved Instances for an `m4.xlarge` Convertible Reserved Instance.



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.

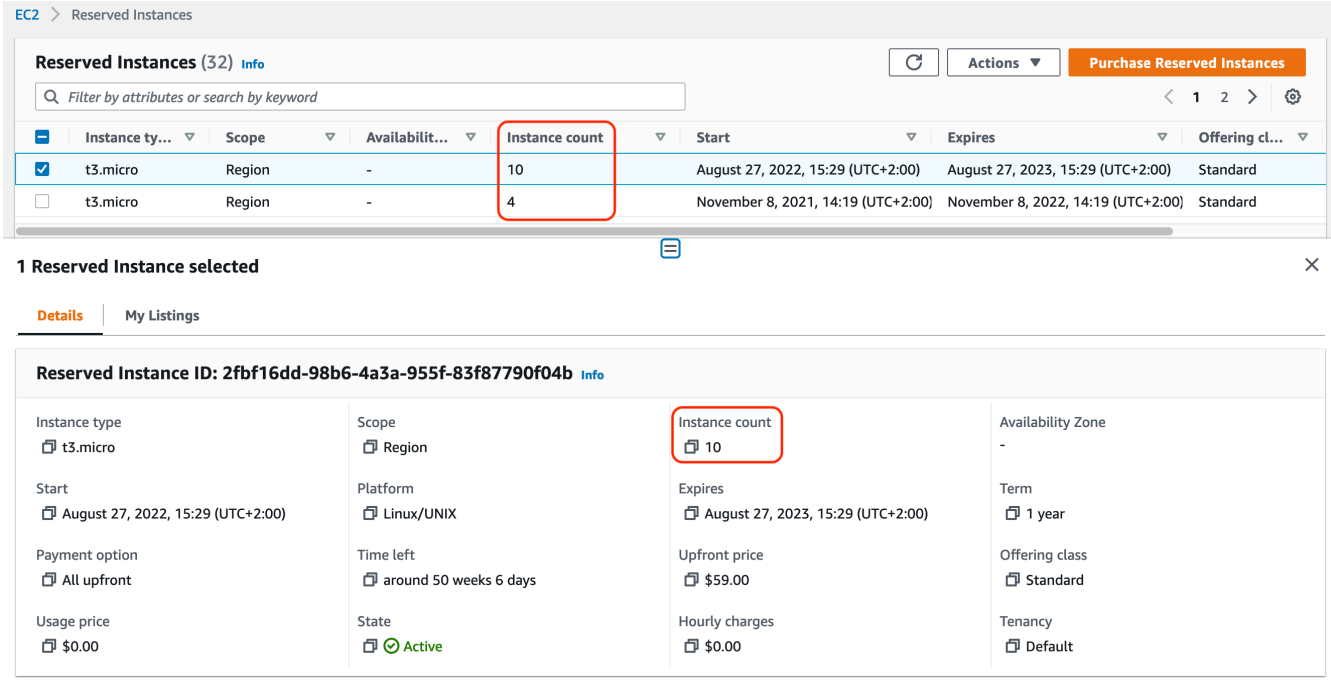
You can view how many Reserved Instances you have purchased by using the Amazon EC2 or the AWS CLI.

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.



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 Windows 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	• 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?
	You want to automate the lifecycle management through a configurable API.	scaling (adding more instances) between specified minimum and maximum limits.	
CreateFleet	• You need multiple instances with either a single configuration or a mixed configuration. • You want to self-manage your instance lifecycle. • If you don't need auto scaling, we recommend that you use an `instant` type fleet.	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`.	Yes – in `instant` mode if you don't need auto scaling

API	When to use?	Use case	Should I use this API?
RunInstances	• You're already using the RunInstances API to launch On-Demand Instances , and you simply want to chan ge to launching Spot Instances by changing a single parameter. • You do not need multiple instances with different instance types.	Launch a specified number of instances using an AMI and one instance type.	No – because RunInstances does not allow mixed instance types in a single request

API	When to use?	Use case	Should I use this API?
RequestSpotFleet	• We strongly discourage using the RequestSpotFleet API because it is a legacy API with no planned investment. • If you want to manage your instance lifecycle, use the CreateFleet API. • If you don't want to manage your instance lifecycle, use the CreateAutoScalingGroup API.	DO NOT USE. RequestSpotFleet is legacy API with no planned investment.	No
RequestSpotInstances	• We strongly discourage using the RequestSpotInstances API because it is a legacy API with no planned investment.	DO NOT USE. RequestSpotInstances is legacy API with no planned investment.	No

How Spot Instances work

To launch a Spot Instance, either you create a *Spot Instance request*, or Amazon EC2 creates a Spot Instance request on your behalf. The Spot Instance launches when the Spot Instance request is fulfilled.

You can launch a Spot Instance using several different services. For more information, see Getting Started with Amazon EC2 Spot Instances. In this user guide, we describe the following ways to launch a Spot Instance using EC2:

- You can create a Spot Instance request by using the launch instance wizard in the Amazon EC2 console or the run-instances 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.

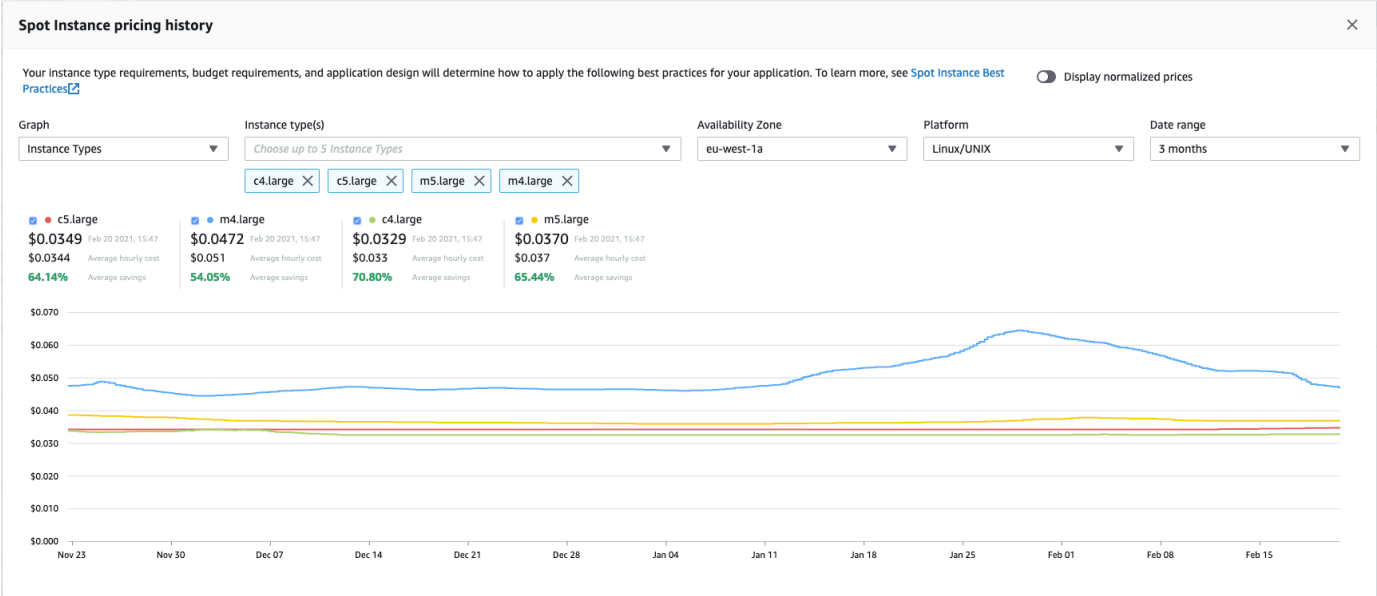
To view the current Spot prices

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

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.

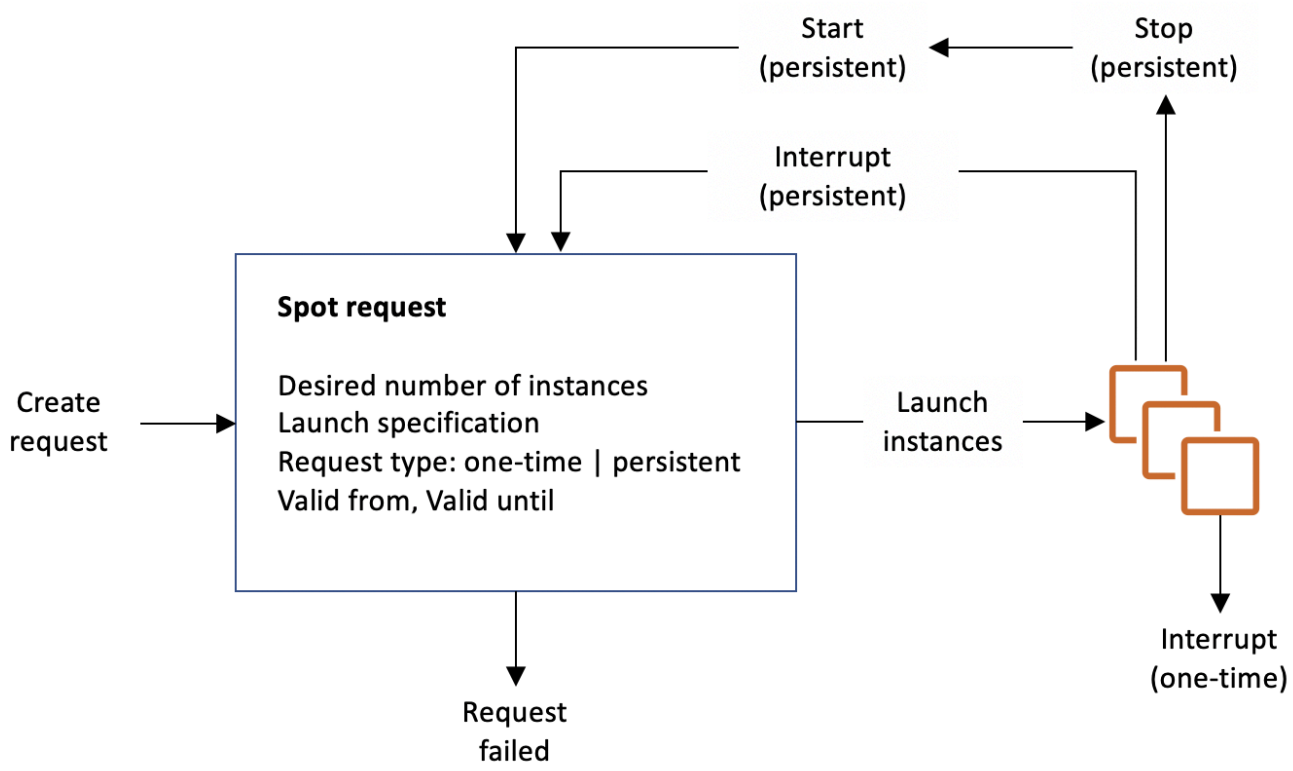
To view the savings information for all running Spot Instances (console)

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

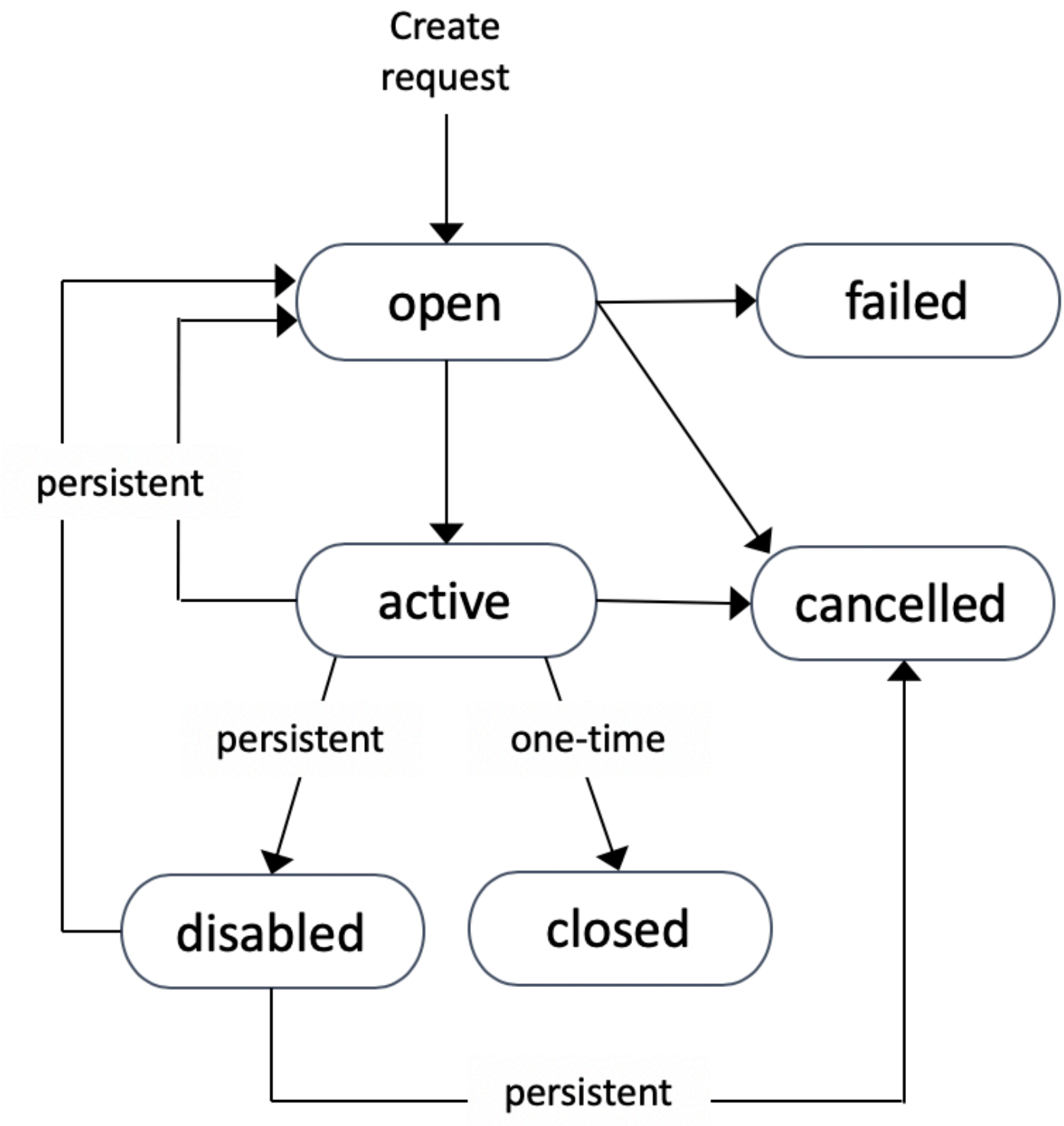
- Spot Instance request states
- Specify a tenancy for your Spot Instances
- Service-linked role for Spot Instance requests
- Create a Spot Instance request
- Find running Spot Instances
- Tag Spot Instance requests
- Cancel a Spot Instance request
- Stop a Spot Instance
- Start a Spot Instance
- Terminate 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 launched.
- 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 Windows 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.

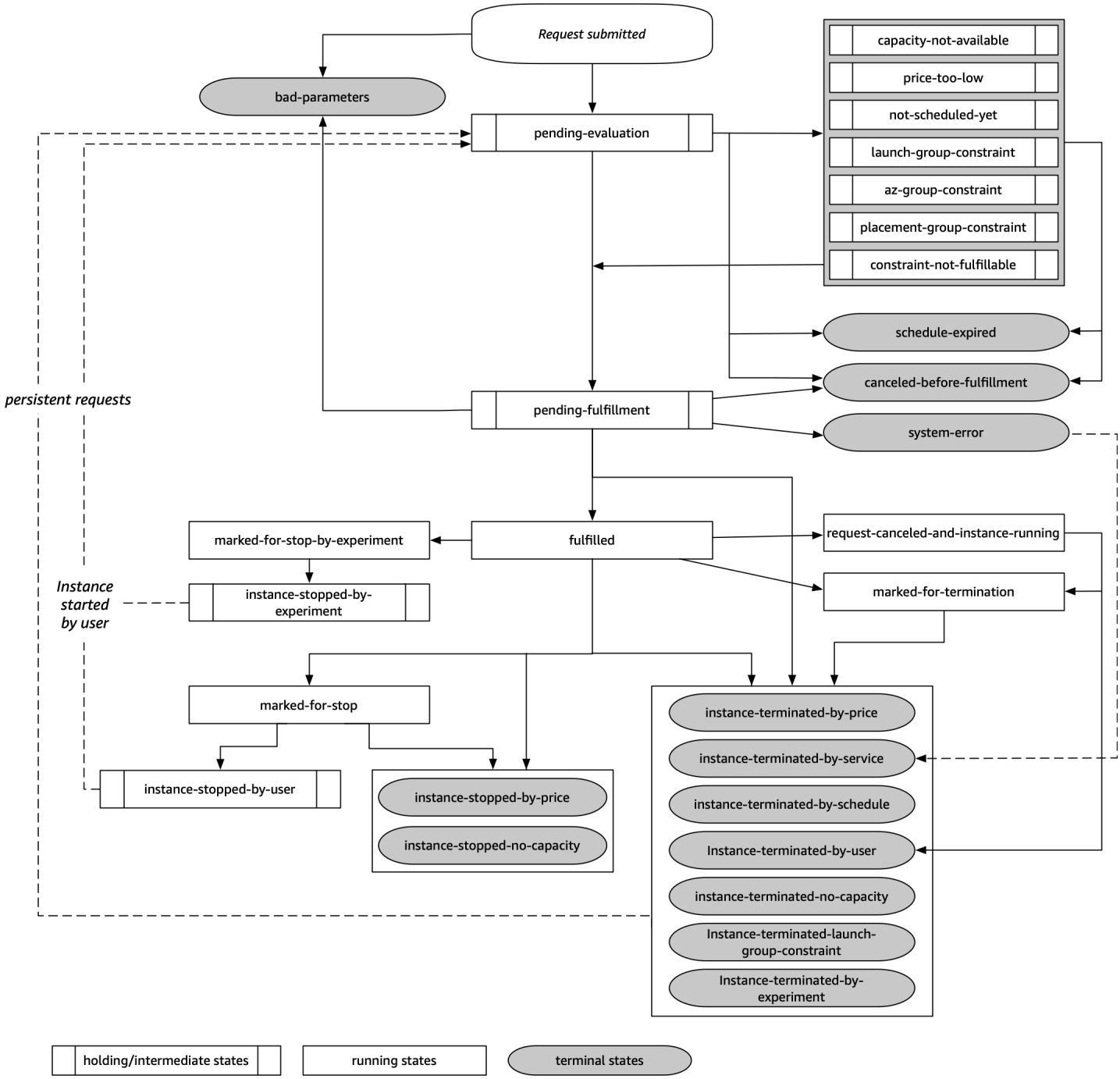
Contents

- 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
canceled-before-fulfillment [1]	cancelled	Not applicable
bad-parameters	failed	Not applicable
system-error	closed	Not applicable

[1] If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met, your Spot request goes into the `pending-fulfillment` state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was canceled by the user before a Spot Instance was launched. It might also be because an unexpected system error occurred.

Status code	Request state	Instance state
`pending-fulfillment`	`open`	Not applicable

Fulfilled

When all the specifications for your Spot Instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot Instances, which can take a few minutes. If a Spot Instance is hibernated or stopped when interrupted, it remains in this state until the request can be fulfilled again or the request is canceled.

Status code	Request state	Instance state
`fulfilled`	`active`	`pending` → `running`
`fulfilled`	`active`	`stopped` → `running`

If you stop a Spot Instance, your Spot request goes into the `marked-for-stop` or `instance-stopped-by-user` state until the Spot Instance can be started again or the request is cancelled.

Status code	Request state	Instance state
`marked-for-stop`	`active`	`stopping`
`instance-stopped-by-user` [1]	`disabled` or `cancelled` [2]	`stopped`

[1] A Spot Instance goes into the `instance-stopped-by-user` state if you stop the instance or run the shutdown command from the instance. After you've stopped the instance, you can start it again. On restart, the Spot Instance request returns to the `pending-evaluation` state and then Amazon EC2 launches a new Spot Instance when the constraints are met.

[2] The Spot request state is `disabled` if you stop the Spot Instance but do not cancel the request. The request state is `cancelled` if your Spot Instance is stopped and the request expires.

Fulfilled-terminal

Your Spot Instances continue to run as long as there is available capacity for your instance type, and you don't terminate the instance. If Amazon EC2 must terminate your Spot Instances, the Spot request goes into a terminal state. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot Instances.

Status code	Request state	Instance state
`request-canceled-and-instance-running`	`cancelled`	`running`
`marked-for-stop`	`active`	`running`
`marked-for-termination`	`active`	`running`
`instance-stopped-by-price`	`disabled`	`stopped`
`instance-stopped-by-user`	`disabled`	`stopped`
`instance-stopped-no-capacity`	`disabled`	`stopped`
`instance-terminated-by-price`	`closed` (one-time), `open` (persistent)	`terminated`
`instance-terminated-by-schedule`	`closed`	`terminated`
`instance-terminated-by-service`	`cancelled`	`terminated`
`instance-terminated-by-user`	`closed` or `cancelled` [1]	`terminated`
`instance-terminated-no-capacity`	`closed` (one-time), `open` (persistent)	`running` †

Status code	Request state	Instance state
`instance-terminated-no-capacity`	`closed` (one-time), `open` (persistent)	`terminated`
`instance-terminated-launch-group-constraint`	`closed` (one-time), `open` (persistent)	`terminated`

[1] The request state is `closed` if you terminate the instance but do not cancel the request. The request state is `cancelled` if you terminate the instance and cancel the request. Even if you terminate a Spot Instance before you cancel its request, there might be a delay before Amazon EC2 detects that your Spot Instance was terminated. In this case, the request state can either be `closed` or `cancelled`.

† When Amazon EC2 interrupts a Spot Instance if it needs the capacity back *and* the instance is configured to *terminate* on interruption, the status is immediately set to `instance-terminated-no-capacity` (it is not set to `marked-for-termination`). However, the instance remains in the `running` state for 2 minutes to reflect the 2-minute period when the instance receives the Spot Instance interruption notice. After 2 minutes, the instance state is set to `terminated`.

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.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/events/
recommendations/rebalance
```

The following is example output, which indicates the time, in UTC, that the rebalance recommendation signal was emitted for the Spot Instance.

```
{"noticeTime": "2020-10-27T08:22:00Z"}
```

If the signal has not been emitted for the instance, `events/recommendations/rebalance` is not present and you receive an HTTP 404 error when you try to retrieve it.

Services that use the rebalance recommendation signal

Amazon EC2 Auto Scaling, EC2 Fleet, and Spot Fleet use the rebalance recommendation signal to make it easy for you to maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running instance receives the two-minute Spot Instance interruption notice. You can have these services monitor and respond proactively to changes affecting the availability of your Spot Instances. For more information, see the following:

- Use Capacity Rebalancing to handle Amazon EC2 Spot interruptions in the *Amazon EC2 Auto Scaling User Guide*
- 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 aws:ec2:send-spot-instance-interruptions	Each supported Region: 5	Yes	The maximum number of Spot Instances that aws:ec2:send-spot-instance-interruptions can target when you identify targets using tags, per experiment.

You can request a quota increase. For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

To view all the quotas for AWS FIS, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **AWS Fault Injection Service**. You can also view all the quotas for AWS Fault Injection Service in the *AWS Fault Injection Service User Guide*.

Spot Instance interruption notices

A *Spot Instance interruption notice* is a warning that is issued two minutes before Amazon EC2 stops or terminates your Spot Instance. If you specify hibernation as the interruption behavior, you receive an interruption notice, but you do not receive a two-minute warning because the hibernation process begins immediately.

The best way for you to gracefully handle Spot Instance interruptions is to architect your application to be fault-tolerant. To accomplish this, you can take advantage of Spot Instance interruption notices. We recommend that you check for these interruption notices every 5 seconds.

The interruption notices are made available as 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 `instance-action` as follows.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/spot/instance-
action
```

The `instance-action` item specifies the action and the approximate time, in UTC, when the action will occur.

The following example 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, the `termination-time` item is present in your instance metadata. Otherwise, it is not present. You can retrieve `termination-time` as follows.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/spot/
termination-time
```

The `termination-time` item specifies the approximate time in UTC when the instance receives the shutdown signal. The following is example output.

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance, or if you terminated the Spot Instance yourself, the `termination-time` item is either not present 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`
6. 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

1. Open the CloudTrail console at https://console.aws.amazon.com/cloudtrail/.
2. In the navigation pane, choose **Event history**.
3. In the filter drop-down, choose **Event name**, and then in the filter field to the right, enter **BidEvictedEvent**.
4. Choose **BidEvictedEvent** in the resulting list to view its details. Under **Event record**, you can find the instance ID.

For more information about using CloudTrail, see Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail.

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 RHEL and SUSE)	Charged for the seconds used	Charged for the seconds used
	RHEL and 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 RHEL and SUSE)	No charge	Charged for the seconds used
	RHEL and SUSE	No charge	Charged for the full hours used, but no charge for the inter rupted partial hour

EBS volume usage

While an interrupted Spot Instance is stopped, you are charged only for the EBS volumes, which are preserved.

With EC2 Fleet and Spot Fleet, if you have many stopped instances, you can exceed the limit on the number of EBS volumes for your account.

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 `m1.small` Spot Instances, this field is set to `SpotUsage`. For all other instance types, this field is set to `SpotUsage:` {*instance-type*}. For example, `SpotUsage:c1.medium`.
`Operation`	The product being charged for. For Linux Spot Instances, this field is set to `RunInstances`. For Windows Spot Instances, this field is set to `RunInstances:0002`. Spot usage is grouped according to Availability Zone.
`InstanceID`	The ID of the Spot Instance that generated this instance usage.
`MyBidID`	The ID for the Spot Instance request that generated this instance usage.
`MyMaxPrice`	The maximum price specified for this Spot request.
`MarketPrice`	The Spot price at the time specified in the `Timestamp` field.
`Charge`	The price charged for this instance usage.
`Version`	The data feed version. The possible version is 1.0.

Amazon S3 bucket requirements

When you subscribe to the data feed, you must specify an Amazon S3 bucket to store the data feed files.

Before you choose an Amazon S3 bucket for the data feed, consider the following:

- You must have `FULL_CONTROL` permission to the bucket. If you're the bucket owner, you have this permission by default. Otherwise, the bucket owner must grant your AWS account this permission.
- When you subscribe to a data feed, these permissions are used to update the bucket ACL to give the AWS data feed account `FULL_CONTROL` permission. The AWS data feed account writes data feed files to the bucket. If your account doesn't have the required permissions, the data feed files cannot be written to the bucket. For more information, see Logs sent to Amazon S3 in the *Amazon CloudWatch Logs User Guide*.

> **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	• Example 1: 4 x `r5.4xlarge` + 4 x `r5.2xlarge` • Example 2: 1 x `r5.12xlarge` + 1 x `r5.4xlarge` + 1 x `r5.2xlarge` + 5 x `r5.xlarge` + 2 x `r5.large`	
C5	• Example 1: 1 x `c5.9xlarge` + 2 x `c5.4xlarge` + 1 x `c5.xlarge` • Example 2: 4 x `c5.4xlarge` + 1 x `c5.xlarge` + 2 x `c5.large`	
M5	• Example 1: 4 x `m5.4xlarge` + 4 x `m5.2xlarge`	

Instance family	Example instance size combinations	
	• 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. For more information about Microsoft Licensing, see Amazon Web Services and Microsoft Licensing.
2. After you have verified that your machine image can be used within Amazon EC2, import it using VM Import/Export. For information about how to import your machine image, see the VM Import/Export User Guide.
3. After you import your machine image, you can launch instances from it onto active Dedicated Hosts in your account.

4. When you run these instances, depending on the operating system, you might be required to activate these instances against your own KMS server (for example, Windows Server or Windows SQL Server). You can't activate your imported Windows AMI against the Amazon Windows KMS server.

> **Note**
>
> To track how your images are used in AWS, enable host recording in AWS Config. You can use AWS Config to record configuration changes to a Dedicated Host and use the output as a data source for license reporting. For more information, see Track configuration changes.

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.

Insta type	vCPUs	Memory (GiB)	Baseline CPU utilization per vCPU	Network burst bandwidth (Gbps)	Amazon EBS burst bandwidth (Mbps)	Max number of instances per Dedicated Host
t3.na	2	0.5	5%	5	Up to 2,085	192
t3.m	2	1	10%	5	Up to 2,085	192
t3.sr	2	2	20%	5	Up to 2,085	192
t3.m	2	4	20%	5	Up to 2,085	192
t3.la	2	8	30%	5	2,780	96
t3.xl	4	16	40%	5	2,780	48

Insta type	vCPUs	Memory (GiB)	Baseline CPU utilization per vCPU	Network burst bandwidth (Gbps)	Amazon EBS burst bandwidth (Mbps)	Max number of instances per Dedicated Host
t3.2 e	8	32	40%	5	2,780	24

Monitor CPU utilization for T3 Dedicated Hosts

You can use the `DedicatedHostCPUUtilization` Amazon CloudWatch metric to monitor the vCPU utilization of a Dedicated Host. The metric is available in the `EC2` namespace and `Per-Host-Metrics` dimension. For more information, see Dedicated Host metrics.

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.

```
C:\>  aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

```
"AvailableInstanceCapacity": [
    { "AvailableCapacity": 2,
      "InstanceType": "c5.xlarge",
      "TotalCapacity": 18 },
    { "AvailableCapacity": 4,
      "InstanceType": "c5.large",
      "TotalCapacity": 36 }
  ],
"AvailableVCpus": 8
```

PowerShell

To view the instance capacity of a Dedicated Host

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.

State	Description
`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 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**. 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, G2, 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 your instance.

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: `M4` | `M5` | `M5a` | `M5n` | `M5zn` | `M6a` | `M6i` | `M6in` | `M7a` | `M7i` | `T3`
- Compute optimized: `C4` | `C5` | `C5a` | `C5n` | `C6a` | `C6i` | `C6in` | `C7i`
- Memory optimized: `R4` | `R5` | `R5a` | `R5b` | `R5n` | `R6a` | `R6i` | `R6in` | `R7a` | `R7iz` | `u-12tb1` | `u-18tb1` | `u-24tb1` | `u-3tb1` | `u-6tb1` | `u-9tb1` | `X2iezn`
- Accelerated computing: `G3` | `G5g` | `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 your instance.

> **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:

- Windows
- Windows with SQL Server
- Windows with SQL Server Web
- Windows with SQL Server Standard
- Windows with SQL Server Enterprise

When you purchase a Capacity Reservation, you must specify the *platform* that represents the operating system for your instance.

- For Windows with SQL Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web, you must choose the specific platform.
- For all other Windows versions, excluding BYOL which is not supported, choose the **Windows** platform.

For more information about the supported Linux platforms, see Supported platforms in the *Amazon EC2 User Guide for Linux Instances*.

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 can't be used with Bring Your Own License (BYOL).
- Capacity Reservations do not ensure that a hibernated instance can resume after you try to start it.

Capacity Reservation pricing and billing

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 Linux platforms, see Supported platforms in the *Amazon EC2 User Guide for Linux 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 Linux platforms, see Supported platforms in the *Amazon EC2 User Guide for Linux Instances*.

For example, the following command creates a Capacity Reservation that reserves capacity for three `m5.2xlarge` instances running Windows with SQL Server AMIs in the `us-east-1a` Availability Zone.

```
aws ec2 create-capacity-reservation --instance-type m5.2xlarge --instance-
platform Windows with SQL Server --availability-zone us-east-1a --instance-count 3
```

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`.

```
C:\> aws resource-groups create-group --name MyCRGroup --
configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-
types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

The following shows example output.

```
{
    "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`.

```
C:\> 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`.

```
C:\> 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.

```
C:\> aws ec2 get-groups-for-capacity-reservation --capacity-reservation-
id cr-1234567890abcdef1
```

The following shows example output.

```
{
    "CapacityReservationGroups": [
```

```
        {
            "OwnerId": "123456789012",
            "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/
MyCRGroup"
        }
    ]
}
```

> **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`.

```
C:\> 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`.

```
C:\> aws resource-groups delete-group --group MyCRGroup
```

The following shows example output.

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",
        "Name": "MyCRGroup"
    }
}
```

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.

```
C:\> 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-wl1-bos-wlz-1`.

Wavelength Zones are not available in every Region. For information about the Regions that support Wavelength Zones, see Available Wavelength Zones in the *AWS Wavelength Developer Guide*.

Considerations

You can't use Capacity Reservation groups in a Wavelength Zone.

To use a Capacity Reservation in a Wavelength Zone

1. Enable the Wavelength Zone for use in your AWS account. For more information, see 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-wl1-bos-wlz-1`. For more information, see Create a Capacity Reservation.
3. Create a subnet in the Wavelength Zone. For **Availability Zone**, choose the Wavelength Zone. For more information, see Create a subnet in your VPC in the *Amazon VPC User Guide*.
4. Launch an instance. For **Subnet**, choose the subnet in the Wavelength Zone (for example `subnet-123abc | us-east-1-wl1-bos-wlz-1`), and for **Capacity Reservation**, choose the specification (either open or target it by ID) that's required for the Capacity Reservation that you created in the Wavelength. For more information, see Launch instances into an existing Capacity Reservation.

Capacity Reservations on AWS Outposts

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region.

You can create Capacity Reservations on Outposts that you have created in your account. This allows you to reserve compute capacity on an Outpost at your site. You create and use Capacity Reservations on Outposts in the same way that you create and use Capacity Reservations in regular Availability Zones. The same features and instance matching behavior apply.

You can also share Capacity Reservations on Outposts with other AWS accounts within your organization using AWS Resource Access Manager. For more information about sharing Capacity Reservations, see 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 prequisites for sharing are met. If the Capacity Reservation is shared with external accounts, they receive an invitation to join the resource share and are granted access to the shared Capacity Reservation after accepting the invitation.

> **Important**
>
> Before launching instances into a Capacity Reservation that is shared with you, verify that you have access to the shared Capacity Reservation by viewing it in the console or by describing it using the describe-capacity-reservations AWS CLI command. If you can

view the shared Capacity Reservation in the console or describe it using the AWS CLI, it is available for your use and you can launch instances into it. If you attempt to launch instances into the Capacity Reservation and it is not accessible due to a sharing failure, the instances will launch into On-Demand capacity.

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 the run in the shared Capacity Reservation.

Instance limits

All Capacity Reservation usage counts toward the Capacity Reservation owner's On-Demand Instance limits. This includes:

- Unused reserved capacity
- Usage by instances owned by the Capacity Reservation owner
- Usage by instances owned by consumers

Instances launched into the shared capacity by consumers count towards the Capacity Reservation owner's On-Demand Instance limit. Consumers' instance limits are a sum of their own On-Demand Instance limits and the capacity available in the shared Capacity Reservations to which they have access.

Capacity Reservation Fleets

An *On-Demand Capacity Reservation Fleet* is a group of Capacity Reservations.

A Capacity Reservation Fleet request contains all of the configuration information that's needed to launch a Capacity Reservation Fleet. Using a single request, you can reserve large amounts of Amazon EC2 capacity for your workload across multiple instance types, up to a target capacity that you specify.

After you create a Capacity Reservation Fleet, you can manage the Capacity Reservations in the fleet collectively by modifying or canceling the Capacity Reservation Fleet.

Topics

- How Capacity Reservation Fleets work
- 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 total target capacity/64 instance type weight=6 instances), or 4 `m5.24xlarge` instances (384 / 96 = 4).

If you do not assign instance type weights, or if you assign an instance type weight of 1, the total target capacity is based purely on instance count. For example, if a Capacity Reservation Fleet with a total target capacity of 384 units uses the instance types in the preceding example, but omits the weights or specifies a weight of 1 for both instance types, the Fleet could reserve capacity for either 384 `m4.16xlarge` instances or 384 `m5.24xlarge` instances.

Instance type priority

The *instance type priority* is a value that you assign to the instance types in the Fleet. The priorities are used to determine which of the instance types specified for the Fleet should be prioritized for use.

Lower priority values indicate a higher priority for use.

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.

```
C:\> 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

```
C:\> 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.

```
C:\> 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

```
C:\> 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.

```
C:\> 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

```
C:\> aws ec2 modify-capacity-reservation-fleet \
--capacity-reservation-fleet-id crf-01234567890abcedf \
--total-target-capacity 160
```

Example: Modify end date

```
C:\> aws ec2 modify-capacity-reservation-fleet \
```

```
--capacity-reservation-fleet-id crf-01234567890abcedf \
--end-date 2021-07-04T23:59:59.000Z
```

Example: Remove end date

```
C:\> aws ec2 modify-capacity-reservation-fleet \
--capacity-reservation-fleet-id crf-01234567890abcedf \
--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.

```
C:\> 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

```
C:\> 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.

```
C:\> 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.

Related resources

After you create a Capacity Block, you can do the following with the Capacity Block:

- Launch instances into the Capacity Block. See Launch instances into Capacity Blocks.

- Create an Amazon EC2 Auto Scaling group. See Use Capacity Blocks for machine learning workloads in the *Amazon EC2 Auto Scaling User Guide*.
- Create an Amazon EKS self-managed node group. See Capacity Blocks for ML in the Amazon EKS User Guide.

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.

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. Linux and Ubuntu Pro operating system prices are billed at per-second granularity. Red Hat Enterprise Linux (RHEL), RHEL with HA, and SUSE Linux are billed at a flat, hourly rate (minimum one-hour billing).

For more information about pricing, 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 then the month you purchased your reservation, the upfront price and reservation usage show up under separate billing months. In your AWS Cost and Usage Report, the Capacity Block reservation ID is listed in the reservation/ReservationARN line item of your upfront fee and the lineitem/ResourceID in your anniversary bill so that you can associate the usage to the corresponding upfront price.

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 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": "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"
```

```
      }
 }
```

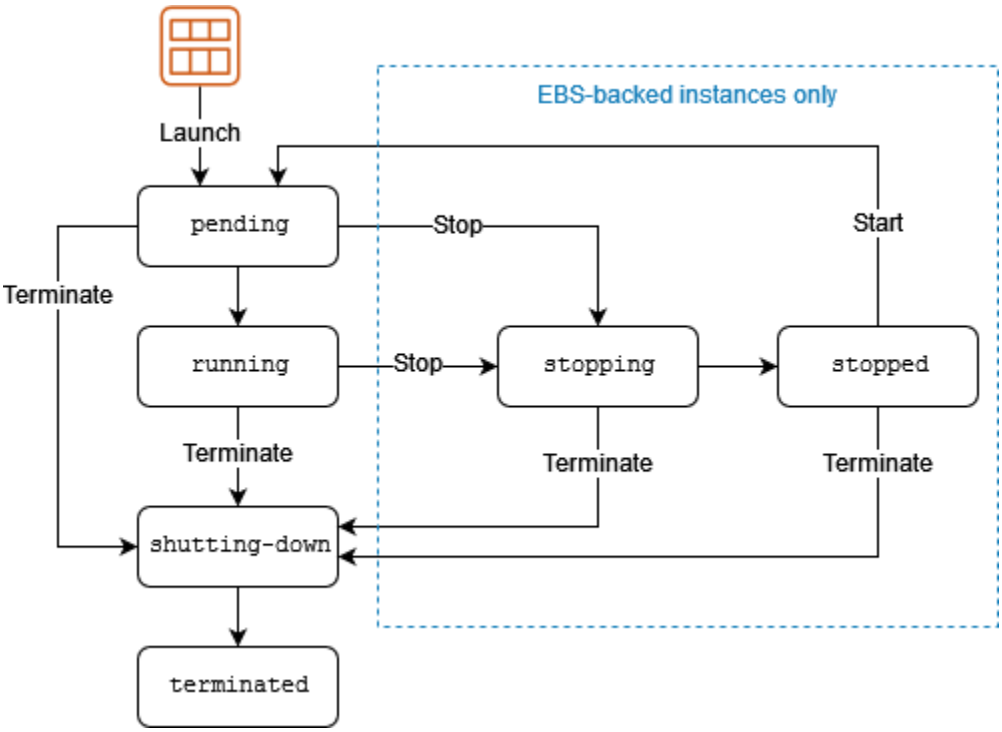
CapacityBlockExpired

```
{
  "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.



The following table provides a brief description of each instance state and indicates whether it is billed.

> **Note**
>
> The table indicates billing for instance usage only. Some AWS resources, such as Amazon EBS volumes and Elastic IP addresses, incur charges regardless of the instance's state. For more information, see Avoiding Unexpected Charges in the *AWS Billing 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	Not billed

Instance state	Description	Instance usage billing
	started after being in the `stopped` state.	
`running`	The instance is running and ready for use.	Billed
`stopping`	The instance is preparing to be stopped.	Not billed
`stopped`	The instance is shut down and cannot be used. The instance can be started at any time.	Not billed
`shutting down`	The instance is preparing to be terminated.	Not billed
`terminated`	The instance has been permanently deleted and cannot be started.	Not billed **Note** Reserved Instances that applied to terminated instances are billed until the end of their term according to their payment option. For more information, see Reserved Instances

Note

Rebooting an instance doesn't start a new instance billing period because the instance stays in the `running` state.

Topics

- Instance launch

- Instance stop and start (Amazon EBS-backed instances only)
- Instance hibernate (Amazon EBS-backed instances only)
- Instance reboot
- Instance retirement
- Instance termination
- Differences between reboot, stop, hibernate, and terminate
- Launch your instance
- Stop and start your instance
- Hibernate your Amazon EC2 instance
- Reboot your instance
- Instance retirement
- Terminate your instance
- 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 Windows 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 your instance.

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 retirement

An instance is scheduled to be retired when AWS detects the irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information, see Instance retirement.

Instance termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

If you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You can also describe a terminated instance using the CLI and API. Resources (such as tags) are gradually disassociated from the terminated instance, therefore may no longer be visible on the terminated instance after a short while. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate shutdown from within the instance itself. The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see Terminate your instance.

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 address, which doesn't change during a stop/start.	The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address, which doesn't change during a stop/start.	None

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
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 *AWS CloudFormation User Guide*
[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 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. For more information about bare metal instances, see Instances built on the Nitro System.

The instance receives a public DNS name that you can use to contact the instance from the internet. The instance also receives a private DNS name that other instances within the same VPC can use to contact the instance. For more information about connecting to your instance, see Connect to your Windows instance.

When you are finished with an instance, be sure to terminate it. For more information, see Terminate your instance.

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 Windows, 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 **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.
- 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 Windows 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 RDP (port 3389). 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 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.

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 Windows instance is joined after launch. If you select a domain, you must select an IAM role with the required permissions. For more information, see Seamlessly join a Windows EC2 instance.
- **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 GPU**: Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.
- **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 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 Windows 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 Windows, Apache, and your website.

When you launch an instance, you can either select an AMI from the list, or you can select a Systems Manager parameter that points to an AMI ID. For more information, see Using a Systems Manager parameter to find an AMI .

On the **Choose an Amazon Machine Image (AMI)** page, use one of two options to choose an AMI. Either search the list of AMIs, 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 **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM.
3. Check the **Boot mode** listed for each AMI. Notice which AMIs use the boot mode that you need, either `legacy-bios` or `uefi`. For more information, see Boot modes.
4. Choose an AMI that meets your needs, and then choose **Select**.

By Systems Manager parameter

1. Choose **Search by Systems Manager parameter** (at top right).
2. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.

3. Choose **Search**. The AMIs that match the AMI ID appear in the list.
4. Select the AMI from the list, and choose **Select**.

Step 2: Choose an Instance Type

On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see Instance types.

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 Windows instance is joined after launch. If you select a domain, you must select an IAM role with the required permissions. For more information, see Seamlessly join a Windows EC2 instance.
- **Placement group**: A placement group determines the placement strategy of your instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see Placement groups.
- **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.
- **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 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 Windows 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 RDP (port 3389).
- You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow internet traffic.

  To add a rule, choose **Add Rule**, select the protocol to open to network traffic, and then specify the source. Choose **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

  > **Warning**
  >
  > Rules that enable all IP addresses (`0.0.0.0/0`) to access your instance over SSH or RDP are acceptable for this short exercise, but are unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

Step 7: Review Instance Launch and Select Key Pair

On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.

When you are ready, choose **Launch**.

In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up. For more information, see Amazon EC2 key pairs and 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.

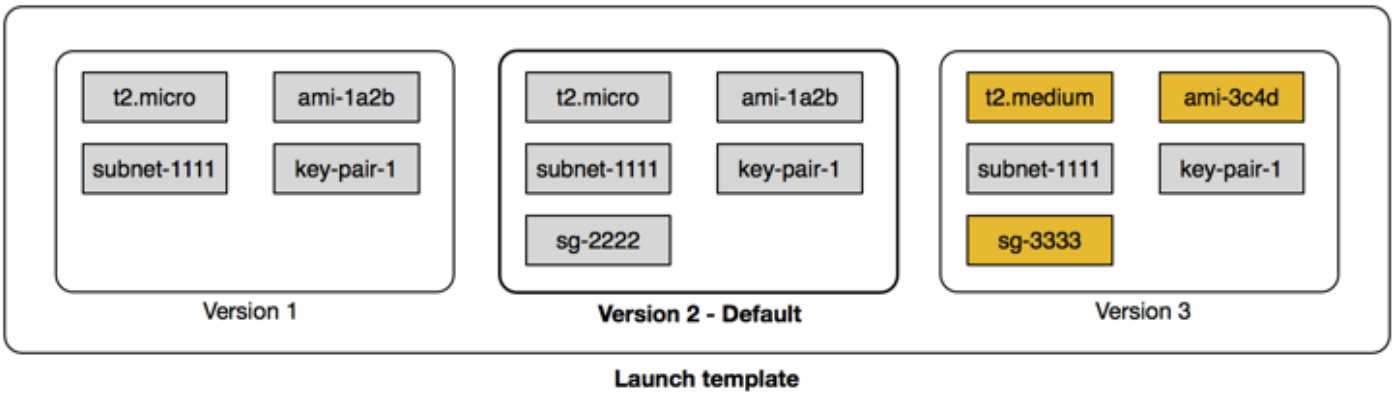
If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see Troubleshoot instance launch issues.

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.

> **Note**
>
> Anyone using launch templates will require permissions to use and create resources specified in the launch template. For example:
>
> - To launch an instance from a shared private Amazon Machine Image (AMI) that's specified in the launch template, the user must have launch permission for the AMI.
> - To create EBS volumes, tags, or other resources that are defined in the launch template, the user must have permissions to create those resources.

Permissions to create and manage launch templates work as follows.

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 new launch template using parameters that you define, or use an existing launch template or an instance as the basis for a new launch template.

Tasks

- Create a new launch template using parameters you define
- 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 new launch template using parameters you define

You can create a launch template using the console or the AWS CLI:

- Console
- AWS CLI

Console

To create a launch template, you must specify the launch template name and at least one instance configuration parameter.

The launch template parameters are grouped in the launch template. The following instructions take you through each parameter group.

Parameters for launch template configuration

- Start launch template creation
- Launch template name, description, and tags

- Application and OS Images (Amazon Machine Image)
- Instance type
- Key pair (login)
- Network settings
- Configure storage
- Resource tags
- Advanced details
- Summary

Start launch template creation

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**.

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 Windows, 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 **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.
- 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 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 Windows 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 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. 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.

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 paramater 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 GPU**: Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.
- **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 Windows instance at launch.

Summary

Use the **Summary** panel to review your launch template configuration and to create your launch template.

- Review the details of your launch template, 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 create your launch template, choose **Create launch template**.

AWS CLI

To create a launch template, you must specify the launch template name and at least one instance configuration parameter.

To create a launch template using the AWS CLI

- Use the create-launch-template command. The following example creates a launch template that specifies the following:
  - A name for the launch template (*TemplateForWebServer*)
  - A description for the launch template (*WebVersion1*)
  - A tag for the launch template (*purpose*=*production*)
  - The data for the instance configuration, specified in a JSON file:
    - The instance type (*r4.4xlarge*) and AMI (*ami-8c1be5f6*) to launch
    - The number of cores (*4*) and threads per core (*2*) for a total of 8 vCPUs (4 cores x 2 threads)
    - The subnet in which to launch the instance (*subnet-7b16de0c*)
    - A public IP address and an IPv6 address to be assigned to the instance
    - A tag for the instance (*Name*=*webserver*)

```
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 an example JSON file that contains the launch template data for the instance configuration.

```
{
    "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"
    }
}
```

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

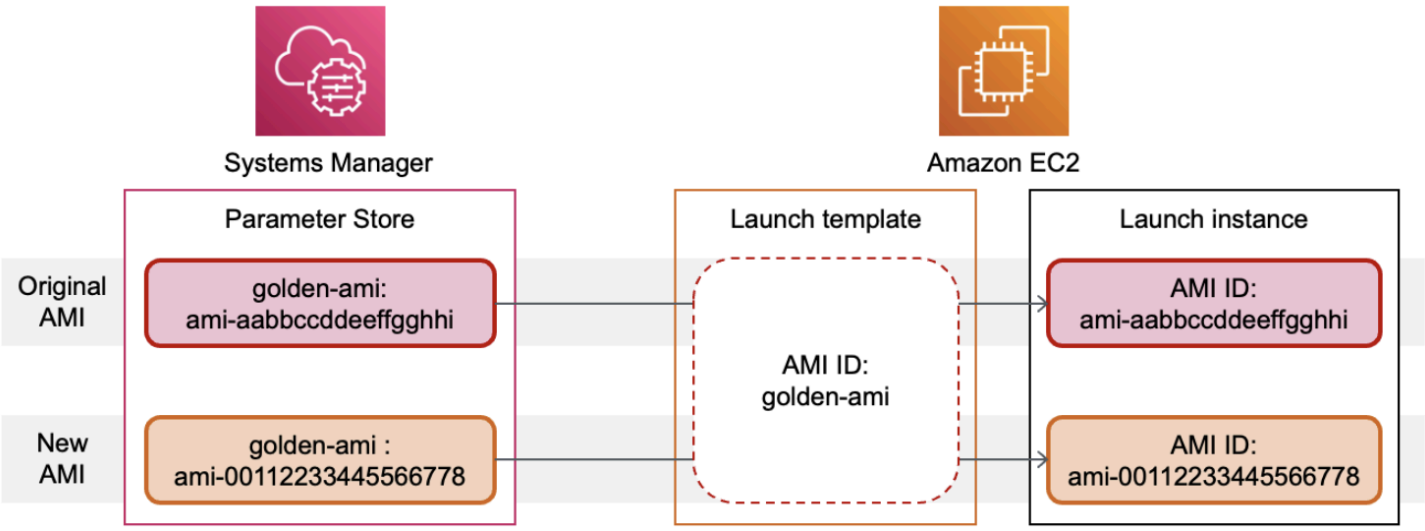
- Use the create-launch-template command to create a launch template using the output from the previous procedure. For more information about creating a launch template using the AWS CLI, see Create a new launch template using parameters you define.

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 new launch template using parameters you define.

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"
        }]
    }]
  }
}
```

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 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
- Association information. If the selected instance is associated with a configuration file, the same file is automatically associated with the new instance. If the configuration file includes a joined domain configuration, the new instance is joined to the same domain. For more information about joining a domain, see Seamlessly Join a Windows EC2 Instance in the *AWS Directory Service Administration Guide*.

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.

> **Note**
>
> You're not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type when you select an instance type. Additional taxes might also apply to the product.

6. For **Instance type**, select an instance type for your instance. The instance type defines the hardware configuration and size of the instance to launch.
7. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or create a new one.
8. Under **Network settings**, **Firewall (security groups)**, take note of the new security group that was created according to the vendor's specifications for the product. The security group might include rules that allow all IPv4 addresses (`0.0.0.0/0`) access on SSH (port 22) on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.
9. You can use the other fields on the screen to configure your instance, add storage, and add tags. For information about the different options that you can configure, see 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 Windows 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.
8. Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, choose **View Instances** to go to the Instances page.

   > **Note**
   >
   > You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

9. When your instance is in the `running` state, you can connect to it. To do this, select your instance in the list and choose **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see Connect to your Windows instance.

> **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.

10. 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
Query API	Use the RunInstances request.

Stop and start your instance

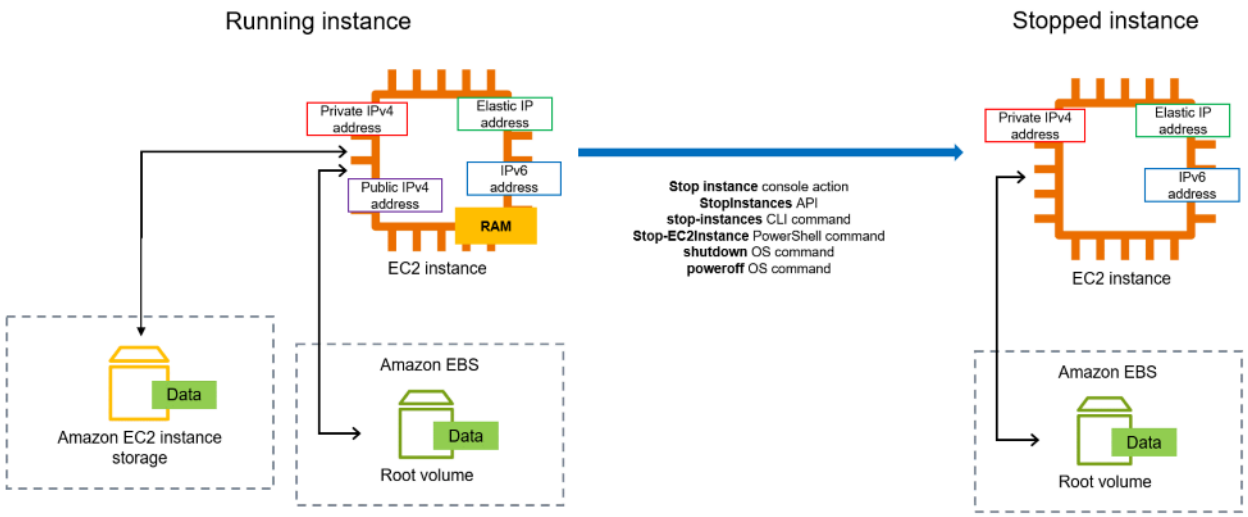
You can stop and start your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can be modified as described in the Modify a stopped instance section. When you stop an instance, the instance shuts down. When you start an instance, the instance is typically 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 your instance. 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.

AWS can schedule events for your instances, such as reboot, stop/start, or retirement. For information about the type of scheduled events that are managed by AWS, and how to view and receive notifications about scheduled events, see Scheduled events for your instances.

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.



Topics

- Costs related to starting and stopping an instance
- Find all running and stopped instances
- Prerequisites for stopping an instance
- Manually stop and start an instances
- Automatically stop and start your instances
- What happens when you stop an instance
- What happens when you start an instance
- Modify a stopped instance
- Enable stop protection

- Test application response to stop and start
- Troubleshoot stopping your instance

Costs related to starting and stopping an instance

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.

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.

Prerequisites for stopping an instance

You can stop an Amazon EBS-backed instance. Instances backed by an instance store volume do not support the **Stop** action. To better understand the differences between the two volume types, see Storage options for your Amazon EC2 instances.

To verify the root device type of an instance, you can use the Amazon EC2 console or the AWS CLI.

Amazon EC2 console

In the Amazon EC2 console, open the **Instances** pane and select an instance. The **Root device type** is listed on the **Storage** tab, under **Root device details**.

AWS CLI

You can verify the root device type of an instance by running the `describe-instances` AWS CLI command and checking the output for `RootDeviceType`: `ebs` or `instance-store`. For more information, see describe-instances in the *AWS CLI Command Reference*.

Manually stop and start an instances

You can stop and start your Amazon EBS-backed instance using the console or the command line.

> **Warning**
>
> When you stop an instance, the data on any attached instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3. Instances *backed* by an instance store volume do not support the **Stop** action.

Console

To stop and start an Amazon EBS-backed instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances** and select the instance.
3. Choose **Instance state**, **Stop instance**. If this option is disabled, either the instance is already stopped or its root device is an instance store volume.
4. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
5. (Optional) While your instance is stopped, you can modify certain instance attributes. For more information, see Modify a stopped instance.
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.

Command line

To stop and start an Amazon EBS-backed instance

Run 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.

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.

What happens when you stop an instance

When you stop an instance, changes are registered at the OS level of the instance, some system resources are lost, and some persist.

When you stop an instance, the following is *registered at the OS level*:

- The API request sends a button press event to the guest.
- Various system services are stopped as a result of the button press event. Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.

- 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*.
- When you stop and start a Windows instance, the launch agent performs tasks on the instance, such as changing the drive letters for any attached Amazon EBS volumes. For more information about these defaults and how you can change them, see Configure a Windows instance using EC2Launch v2.

When you stop an instance, the following is *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.

When you stop an instance, the following *persists*:

- 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

When you start an instance, changes are registered at the instance level.

When you start an EC2 instance, the following occurs:

- 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.

Modify a stopped instance

When an instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the `IncorrectInstanceState` error.

You can change the following attributes of a stopped instance using the Amazon EC2 console or the AWS CLI:

- Instance type
- User data
- EBS-optimization

Modification of the following attributes using the Amazon EC2 console is not supported:

- `DeleteOnTermination`
- Kernel
- RAM disk

Modify an instance attribute

You can modify an instance attribute using the Amazon EC2 console or the command line.

Console

To modify the following in the AWS Management Console	See the following resource
Instance type	Change the instance type
User data	Run commands on your Windows instance at launch
EBS optimization	Enable EBS optimization
The `DeleteOnTermination` attribute of the root volume	Update the block device mapping of a running instance. You are not required to stop the instance to change this attribute.

Command line

Modify an instance attribute using the command line

You can use the following commands to modify instance attributes. For more information about these command line interfaces see Access Amazon EC2.

- modify-instance-attribute (AWS CLI)
- Edit-EC2InstanceAttribute (AWS Tools for PowerShell)

Modify the root volume of an instance

You can modify the root volume of an instance by performing the following steps.

1. Detach the volume from the stopped instance.
2. Attach the volume to a running instance.
3. Modify the volume.
4. Detach the volume from the running instance.
5. Reattach the volume to the stopped instance.

Make sure that you reattach the root volume using the storage device name specified as the root device in the block device mapping for the instance. For more information about specifying block device mappings, see Block device mappings.

Enable stop protection

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.

Limitations

- 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 your instance.

- 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 API Reference*.

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
```

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.

Troubleshoot stopping your instance

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.

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.

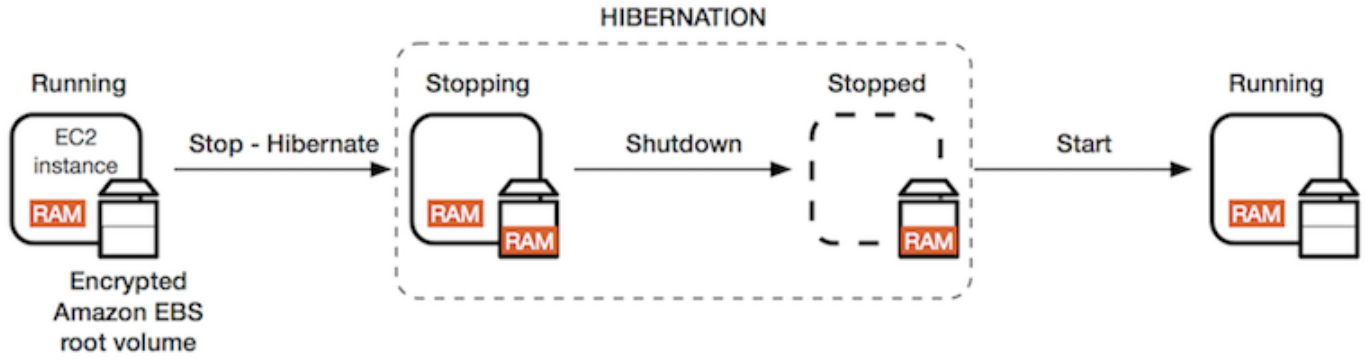
If you no longer need an instance, you can terminate it at any time, including when it is in a `stopped` (hibernated) state. For more information, see Terminate your instance.

Contents

- How Amazon EC2 instance hibernation works
- Prerequisites for Amazon EC2 instance hibernation
- Enable hibernation for an Amazon EC2 instance
- 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.

Supported AMIs

- Linux AMIs
- Windows AMIs

Linux AMIs

AMI	Xen - supported instance families only	Nitro - supported instance families only
AL2023 AMI released 2023.09.20 or later	Supported	Supported
Amazon Linux 2 AMI released 2019.08.29 or later	Supported	Supported
Amazon Linux AMI 2018.03 released 2018.11.16 or later	Supported	Supported
CentOS version 8 AMI [1]	Not supported	Supported
Fedora version 34 or later AMI [1]	Not supported	Supported
Red Hat Enterprise Linux (RHEL) 9 AMI [1]	Not supported	Supported
Red Hat Enterprise Linux (RHEL) 8 AMI [1]	Not supported	Supported
Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI released with serial number 20230303 or later [2]	Supported	Supported

AMI	Xen - supported instance families only	Nitro - supported instance families only
Ubuntu 20.04 LTS (Focal Fossa) AMI released with serial number 20210820 or later [2]	Supported	Supported
Ubuntu 18.04 LTS (Bionic Beaver) AMI released with serial number 20190722.1 or later [2] [4]	Supported	Supported
Ubuntu 16.04 LTS (Xenial Xerus) AMI [2] [3] [4]	Supported	Supported

[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).

[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.

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.

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.

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.

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.

If Windows is installing updates on your instance, we recommend that you do not reboot or shut down your instance using the Amazon EC2 console or the command line until all the updates are installed. When you use the Amazon EC2 console or the command line to reboot or shut down your instance, there is a risk that your instance will be hard rebooted. A hard reboot while updates are being installed could throw your instance into an unstable state.

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.

Instance retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware that hosts the instance. When an instance reaches its scheduled retirement date, it is stopped by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware.

For more information about the types of instance events, see Scheduled events for your instances.

Contents

- Identify instances scheduled for retirement
- Actions to take for 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.

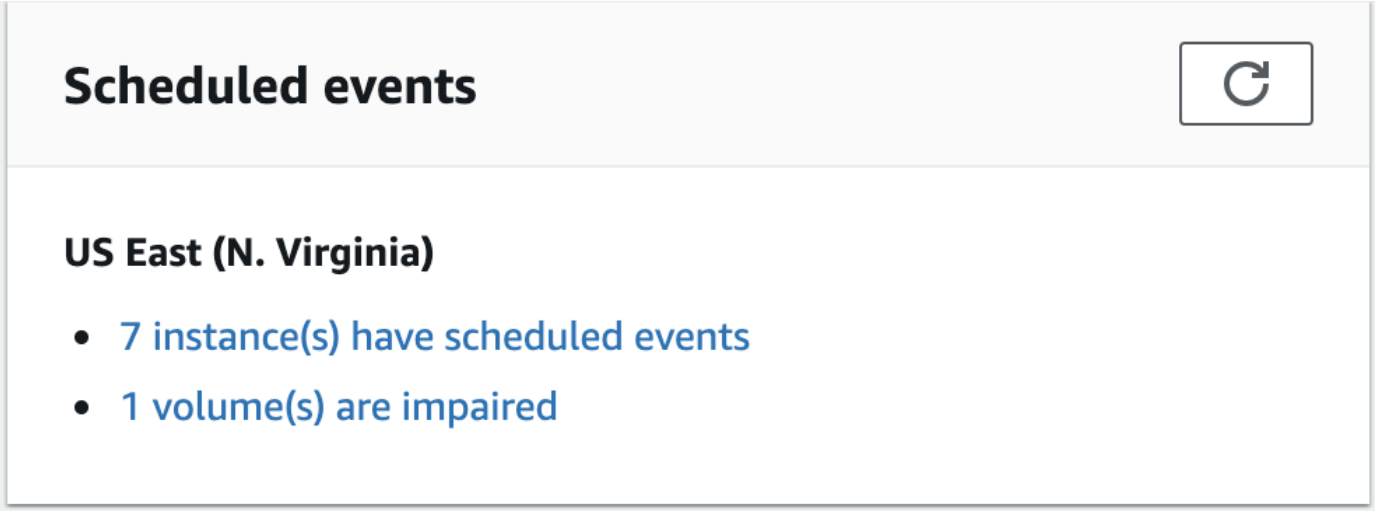
The email is sent to the primary account holder and the operations contact. For more information, see Adding, changing, or removing alternate contacts in the *AWS Billing User Guide*.

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 instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

Check if your instance is reachable

When you are notified that your instance is scheduled for retirement, we recommend that you take the following action as soon as possible:

- Check if your instance is reachable by either connecting to or pinging your instance.
- If your instance is reachable, you should plan to stop/start your instance at an appropriate time before the scheduled retirement date, when the impact is minimal. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private, and Elastic IP addresses that are associated with your instance, see Stop and start your instance. 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 a custom Windows 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.

Terminate your instance

You can delete your instance when you no longer need it. This is referred to as *terminating* your instance. As soon as the state of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

You can't connect to or start an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and start your instance, or hibernate it, see Stop and start your instance or Hibernate your Amazon EC2 instance. For more information, see Differences between reboot, stop, hibernate, and terminate.

For more information about what happens when you terminate an instance, see What happens when you terminate an instance.

Topics

- Termination considerations
- Terminate an instance

Termination considerations

Before you terminate an instance, you might want to consider the following.

- Enable termination protection – Prevent an instance from being accidentally terminated by someone using the AWS Management Console, AWS Command Line Interface (AWS CLI), or the API.
- Change the instance initiated shutdown behavior – Control whether an instance should stop or terminate when shutdown is initiated from the instance using an operating system command for system shutdown.
- Preserve data when an instance is terminated – Ensure data isn't lost when the instance is terminated.

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.

Limitations

- 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)

Terminating multiple instances with termination protection across Availability Zones

If you terminate multiple instances across multiple Availability Zones, and one or more of the specified instances are enabled for termination protection, the request fails with the following results:

- The specified instances that are in the same Availability Zone as the protected instance are not terminated.
- The specified instances that are in different Availability Zones, where no other specified instances are protected, are successfully terminated.

For example, say you have the following instances:

Instance	Availability Zone	Terminate protection
Instance A	`us-east-1a`	`Disabled`
Instance B		`Disabled`
Instance C	`us-east-1b`	`Enabled`
Instance D		`Disabled`

If you attempt to terminate all of these instances in the same request, the request reports failure with the following results:

- **Instance A** and **Instance B** are successfully terminated because none of the specified instances in `us-east-1a` are enabled for termination protection.
- **Instance C** and **Instance D** fail to terminate because at least one of the specified instances in `us-east-1b` (**Instance C**) is enabled for termination protection.

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, 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 Amazon EBS root volume to persist at launch using the console or command line

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 commmand 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 Amazon EBS root volume of a running instance to persist using the command line

You can use one of the following commands to change the root device volume of a running EBS-backed instance to persist. For more information about these command line interfaces, see Access Amazon EC2.

- 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
    }
  }
]
```

Terminate an instance

Before you terminate an instance, review the following.

- 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.

- 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.

Topics

- What happens when you terminate an instance
- Terminate an instance
- Related resources

What happens when you terminate an instance

When you terminate an instance, changes are registered at the OS level of the instance, some system resources are lost, and some persist.

When you terminate an instance, the following is *registered* at the OS level:

- The API request will send a button press event to the guest.
- Various system services will be stopped as a result of the button press event. In Windows, the **System** process 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.

When you terminate an instance, the following is *lost*:

- Data stored on an instance store volume.
- Data stored on Amazon EBS root device volumes if the `DeleteOnTermination` attribute is set to true.

When you terminate an instance, the following *persists*:

- Data stored on additional Amazon EBS volumes attached at launch or after launch of an instance.

Terminate an instance

You can terminate an instance using the AWS Management Console or the command line.

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.

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)

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.

Related resources

For more information about deleting Amazon EBS volumes, see Delete an Amazon EBS volume in the *Amazon EBS User Guide*.

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:** M3 | M4 | M5 | M5a | M5n | M5zn | M6a | M6i | M6in | M7a | M7i | M7i-flex | T1 | T2 | T3 | T3a
  - **Compute optimized:** C3 | C4 | C5 | C5a | C5n | C6a | C6i | C6in | C7a | C7i
  - **Memory optimized:** R3 | R4 | R5 | R5a | R5b | R5n | R6a | R6g | R6in | R7a | R7i | R7iz | u-3tb1 | u-6tb1 | u-9tb1 | u-12tb1 | u-18tb1 | u-24tb1 | X1 | X1e | X2iezn
  - **Accelerated computing:** G3 | G3s | P2 | P3
  - **High-performance computing** Hpc7a
- 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 your instance.

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 Windows Instances* provides information to help you connect to your Windows instance after you launch it. It also provides information to help you connect your Windows instance to another AWS resource.

For information about how to connect to a **Linux instance**, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*.

Topics

- Connect to your Windows instance
- Connect to your instances without requiring a public IPv4 address using EC2 Instance Connect Endpoint
- Connect your EC2 instance to an AWS resource

Connect to your Windows instance

You can connect to Amazon EC2 instances created from most Windows Amazon Machine Images (AMIs) using Remote Desktop. Remote Desktop uses the Remote Desktop Protocol (RDP) to connect to and use your instance in the same way you use a computer sitting in front of you (local computer). It is available on most editions of Windows and is also available for Mac OS.

The license for the Windows Server operating system allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your Windows instance. If you require more than two simultaneous remote connections, you must purchase a Remote Desktop Services (RDS) license. If you attempt a third connection, an error occurs.

If you need to connect to your instance in order to troubleshoot boot, network configuration, and other issues for instances built on the AWS Nitro System, you can use the EC2 Serial Console for Windows instances.

For information about connecting to a Linux instance, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*.

> **Tip**
>
> You can create an EC2 Instance Connect Endpoint to connect to your instance using SSH or RDP without a public IPv4 address.

Contents

- Prerequisites
- Connect to your Windows instance using RDP
- Connect to your Windows instance using Fleet Manager

- Connect to a Windows instance using its IPv6 address
- Connect to a Windows instance using Session Manager
- Configure your accounts
- Transfer files to Windows instances

Prerequisites

To connect using RDP

- **Install an RDP client**
  - [Windows] Windows includes an RDP client by default. To verify, type **mstsc** at a Command Prompt window. If your computer doesn't recognize this command, see the Windows home page and search for the download for the Microsoft Remote Desktop app.
  - [Mac OS X] Download the Microsoft Remote Desktop app from the Mac App Store.
  - [Linux] Use Remmina.
- **Locate the private key**

  Get the fully-qualified path to the location on your computer of the `.pem` file for the key pair that you specified when you launched the instance. For more information, see Identify the 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 Windows instance?
- **Enable inbound RDP traffic from your IP address to your instance**

  Ensure that the security group associated with your instance allows incoming RDP traffic (`port 3389`) from your IP address. The default security group does not allow incoming RDP traffic by default. For more information, see Authorize inbound traffic for your Windows instances.

  > **Note**
  > You do not need to specifically allow incoming RDP traffic from your IP address if you use Fleet Manager to connect. Fleet Manager handles that for you.

- **To connect using Fleet Manager**

  For prerequisites, see Connect using Remote Desktop in the *AWS Systems Manager User Guide*.

Connect to your Windows instance using RDP

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

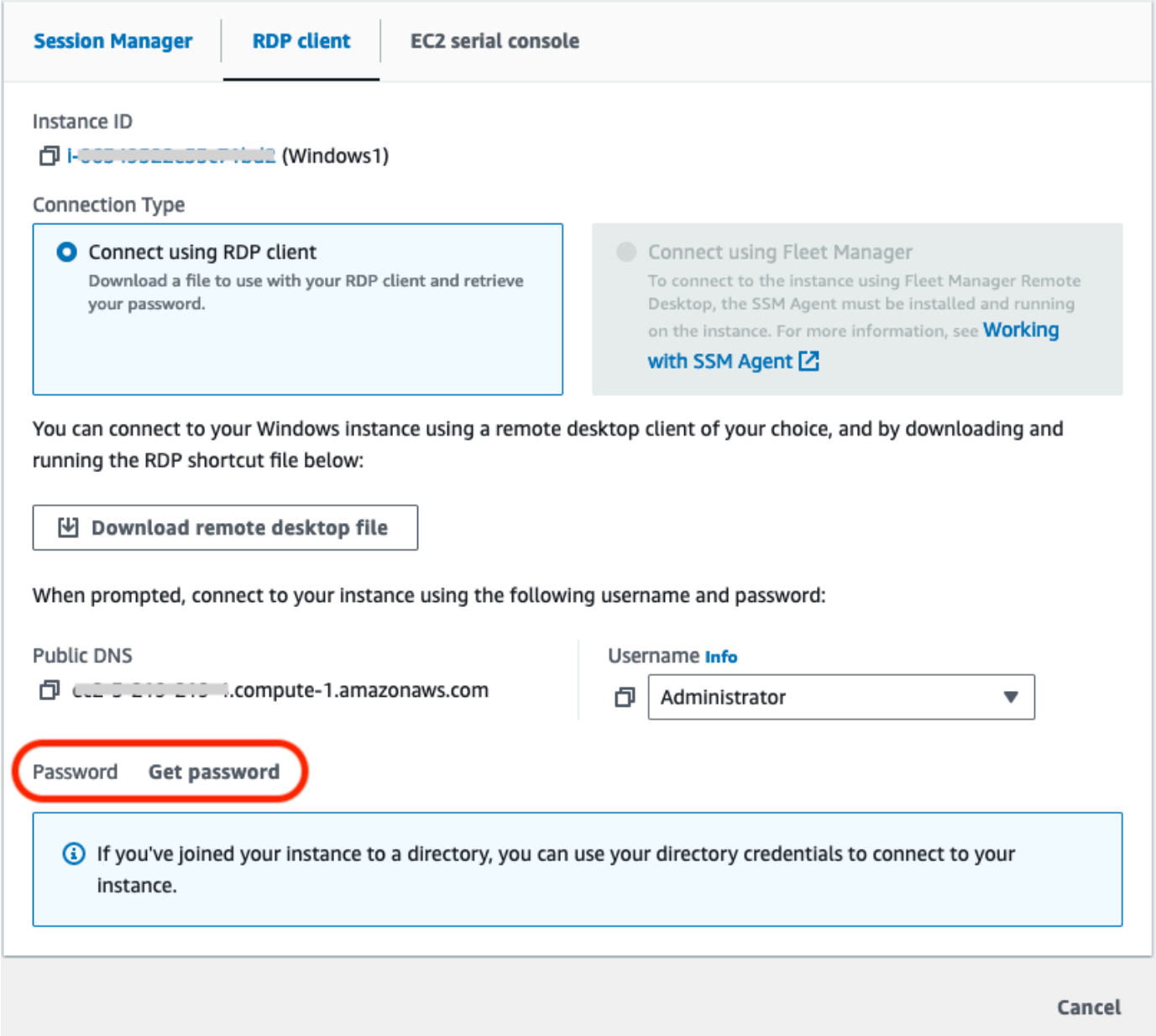
The default username for the Administrator account depends on the language of the operating system (OS) contained in the AMI. To ascertain the correct username, identify the language of your AMI's OS, and then choose the corresponding username. For example, for an English OS, the username is `Administrator`,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

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

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

To connect to your Windows instance using an RDP client

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and then choose **Connect**.
4. On the **Connect to instance** page, choose the **RDP client** tab.
5. For **Username**, choose the default username for the Administrator account. The username you choose must match the language of the operating system (OS) contained in the AMI that you used to launch your instance. If there is no username in the same language as your OS, choose **Administrator (Other)**.
6. Choose **Get password**.

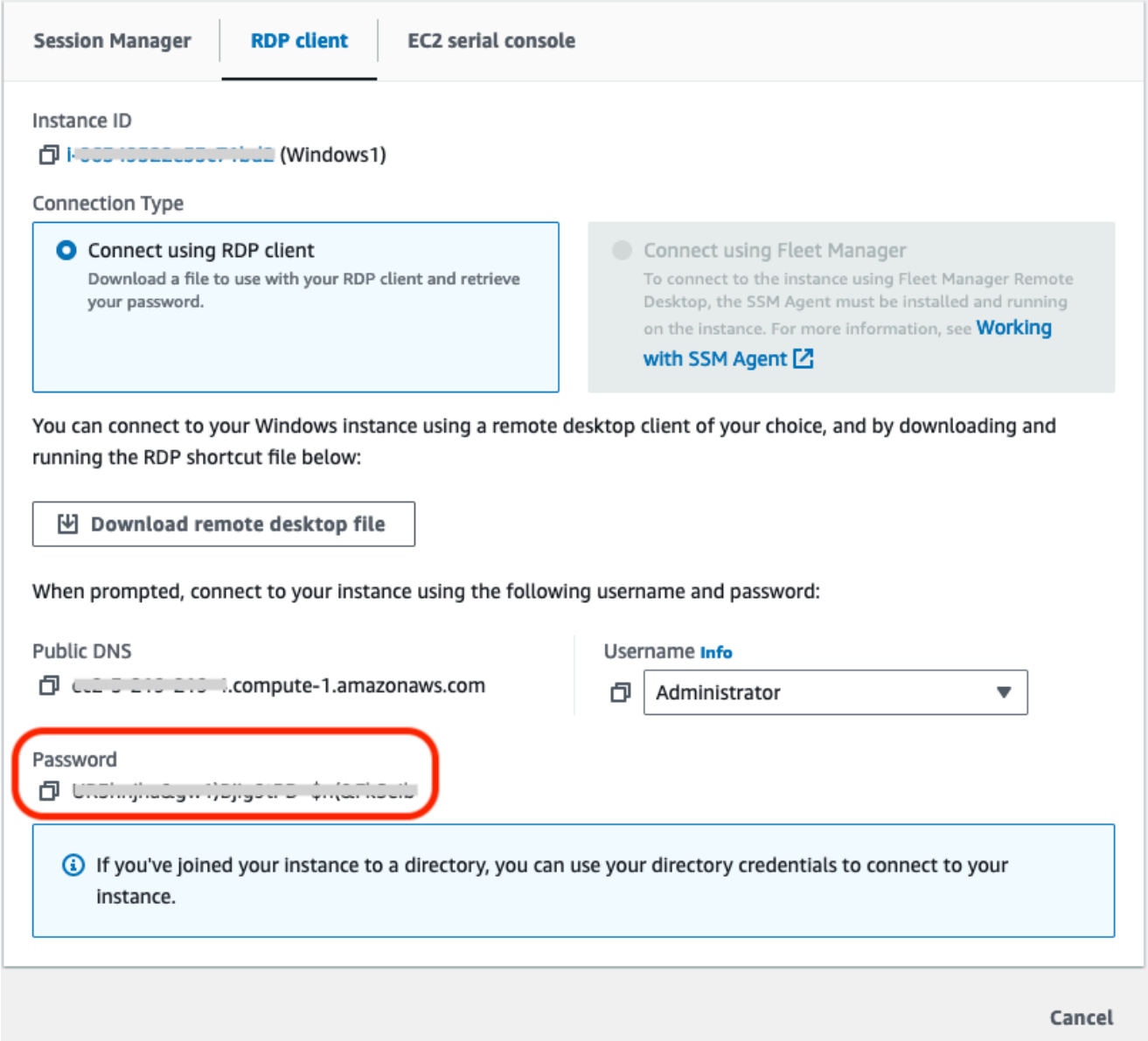


7. On the **Get Windows password** page, do the following:

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

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

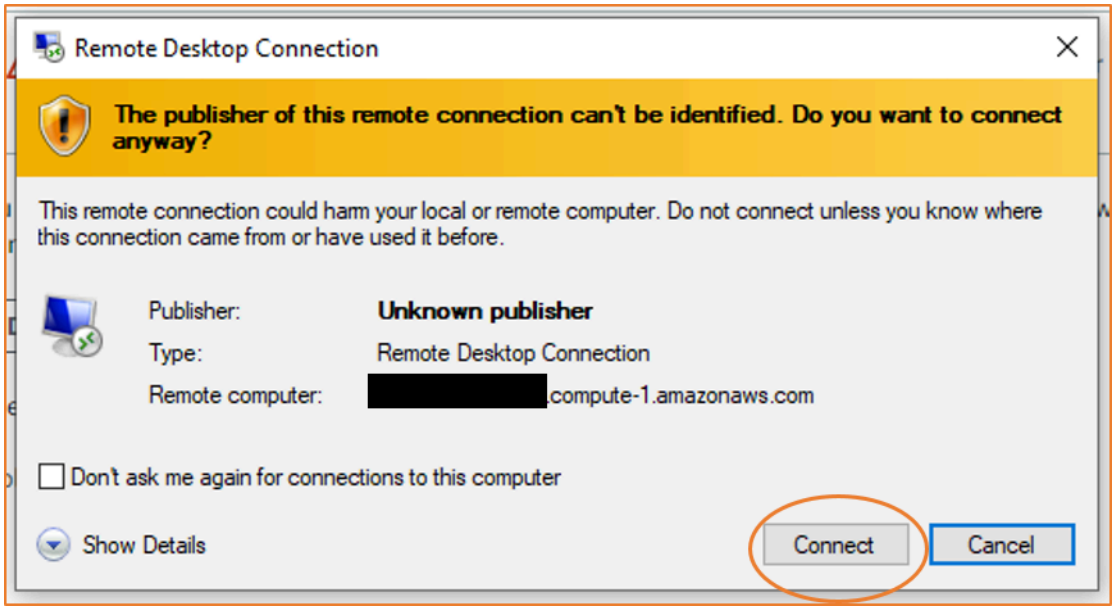
   c. Copy the password and save it in a safe place. This password is required to connect to the instance.



8. Choose **Download remote desktop file**. Your browser prompts you to either open or save the RDP shortcut file. When you have finished downloading the file, choose **Cancel** to return to the **Instances** page.

   - If you opened the RDP file, you'll see the **Remote Desktop Connection** dialog box.
   - If you saved the RDP file, navigate to your downloads directory, and open the RDP file to display the dialog box.

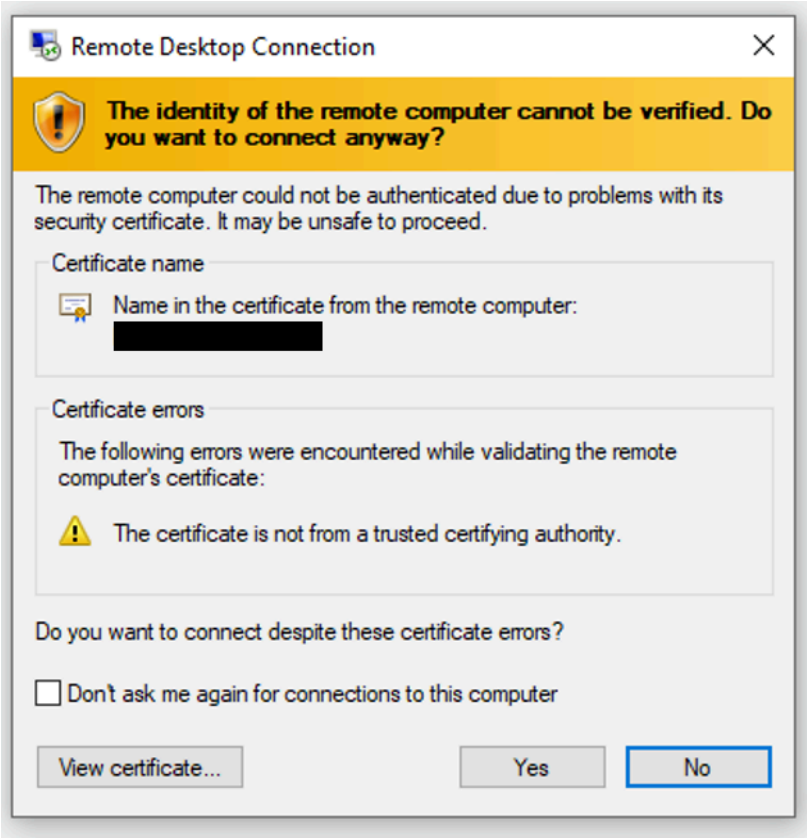
9. You might get a warning that the publisher of the remote connection is unknown. Choose **Connect** to continue to connect to your instance.



10. The administrator account is chosen by default. Paste the password that you copied previously, and then choose **Continue**.

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

11. Due to the nature of self-signed certificates, you might get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer. Alternatively, if you trust the certificate, choose **Yes** (Windows) or **Continue** (Mac OS X) to skip the following steps.



a. [Windows] Choose **View certificate**.

   [Mac OS X] Choose **Show Certificate**.

b. [Windows] Choose the **Details** tab, and scroll down to **Thumbprint**.

   [Mac OS X] Expand **Details**, and scroll down to **SHA1 Fingerprints**.

   This is the unique identifier for the remote computer's security certificate.

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

d. In the system log output, look for `RDPCERTIFICATE-THUMBPRINT`. If this value matches the thumbprint (Windows) or fingerprint (Mac OS X) of the certificate, you have verified the identity of the remote computer.

e. [Windows] Return to the **Certificate** dialog box and choose **OK**.

   [Mac OS X computer] Return to the **Verify Certificate** dialog box and choose **Continue**.

f. [Windows] Choose **Yes** in the **Remote Desktop Connection** window to connect to your instance.

[Mac OS X] The process automatically commences connecting to your instance. Note that you might need to switch spaces to see the Windows instance screen. For more information, see See open windows and spaces in Mission Control on Mac.

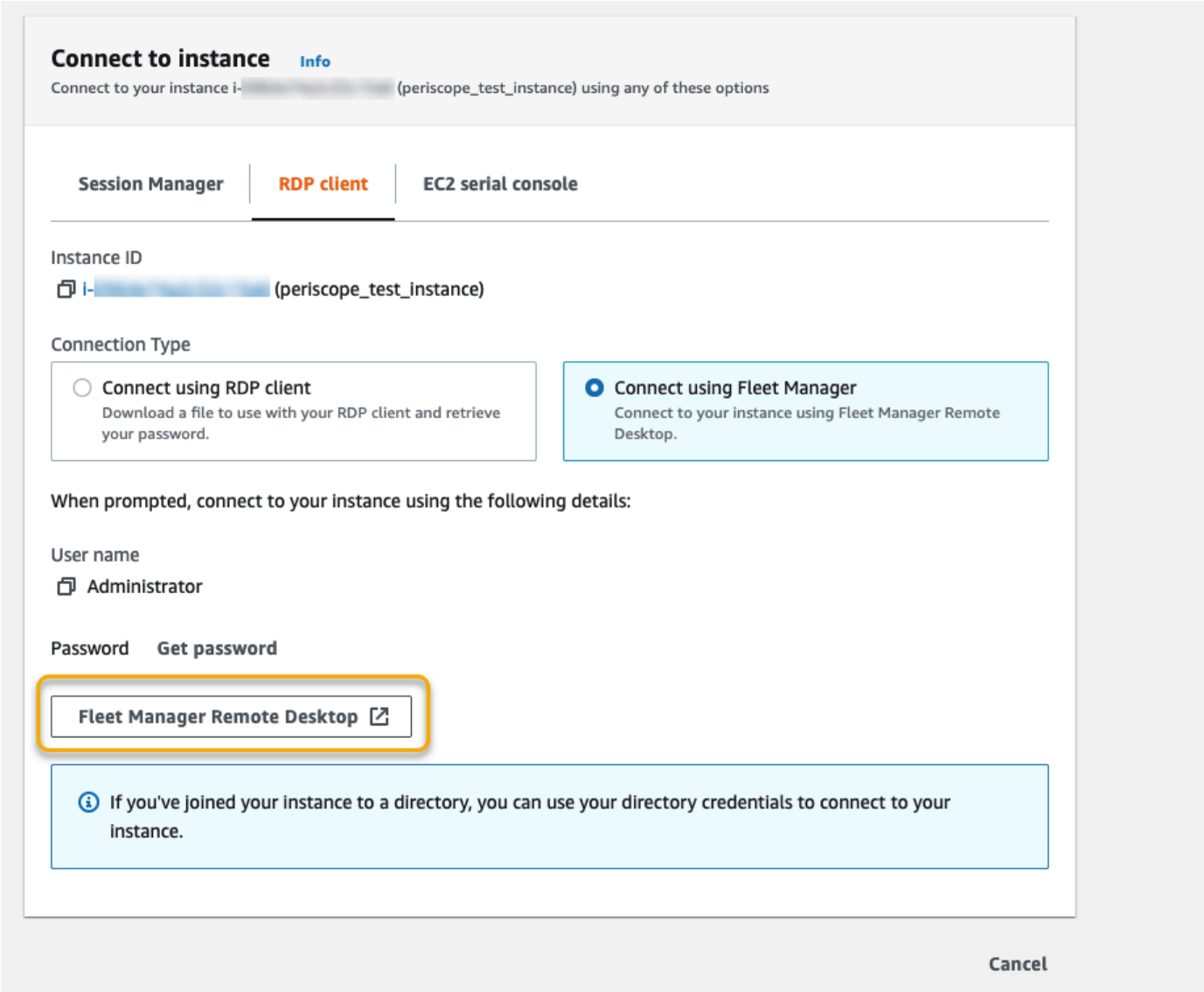
Connect to your Windows instance using Fleet Manager

You can use Fleet Manager, a capability of AWS Systems Manager, to connect to Windows instances using the Remote Desktop Protocol (RDP) and display up to four Windows instances on the same page in the AWS Management Console. You can connect to the first instance in the Fleet Manager Remote Desktop directly from the **Instances** page in the Amazon EC2 console. For more information about Fleet Manager, see Connect to a managed node using Remote Desktop in the *AWS Systems Manager User Guide*.

Before attempting to connect to an instance using Fleet Manager, ensure that the necessary setup steps have been completed. For more information, see Setting up Fleet Manager.

To connect to instances using RDP with Fleet Manager (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation pane, choose **Instances**.
3. Select the instance and then choose **Connect**.
4. On the **Connect to instance** page, choose the option to **Connect using Fleet Manager**, then choose **Fleet Manager Remote Desktop**. This opens the **Fleet Manager Remote Desktop** page in the AWS Systems Manager console.



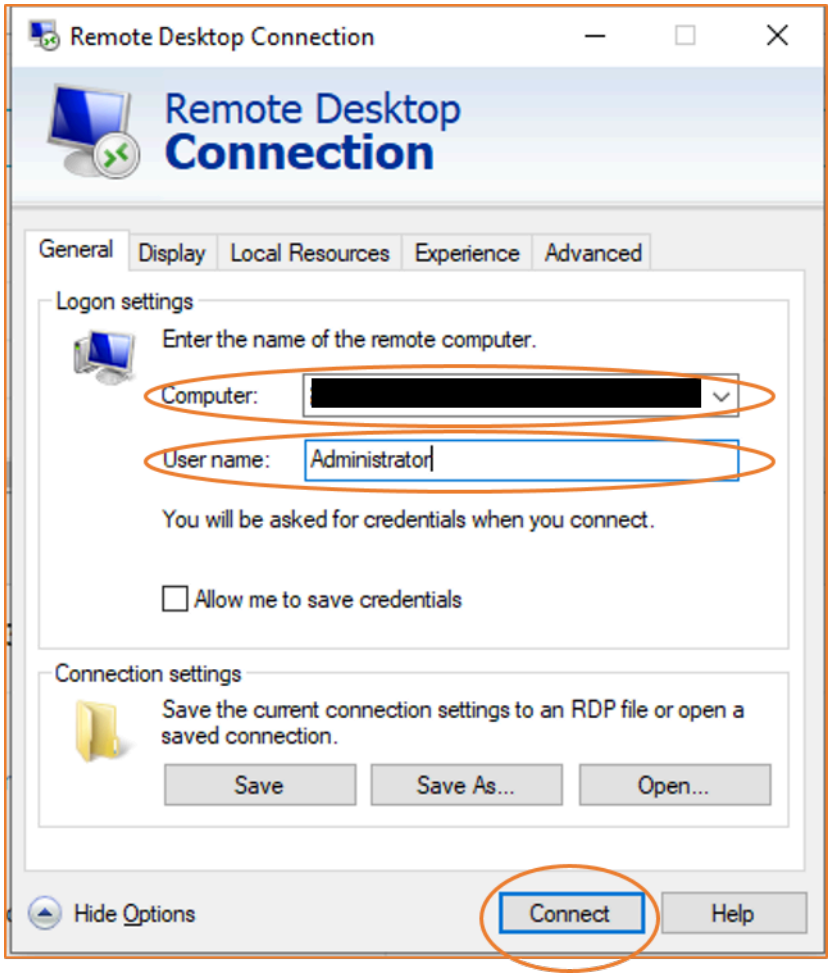
For more information about connecting to Windows instances from the **Fleet Manager Remote Desktop** page, see Connect using Remote Desktop in the *AWS Systems Manager User Guide*.

Connect to a Windows instance using its IPv6 address

If you've enabled your VPC for IPv6 and assigned an IPv6 address to your Windows instance, you can use an RDP client to connect to your instance using its IPv6 address (for example, `2001:db8:1234:1a00:9691:9503:25ad:1761`) instead of using its public IPv4 address or public DNS hostname.

To connect to your Windows instance using its IPv6 address

1. Get the initial administrator password for your instance, as described in Connect to your Windows instance using RDP. This password is required to connect to your instance.
2. [Windows] Open the RDP client on your Windows computer, choose **Show Options**, and do the following:



   - For **Computer**, enter the IPv6 address of your Windows instance.
   - For **User name**, enter **Administrator**.
   - Choose **Connect**.
   - When prompted, enter the password that you saved previously.

   [Mac OS X] Open the RDP client on your computer and do the following:

   - Choose **New**.

- For **PC Name**, enter the IPv6 address of your Windows instance.
- For **User name**, enter **Administrator**.
- Close the dialog box. Under **My Desktops**, select the connection, and choose **Start**.
- When prompted, enter the password that you saved previously.

3. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. If you trust the certificate, you can choose **Yes** or **Continue**. Otherwise, you can verify the identity of the remote computer, as described in Connect to your Windows instance using RDP.

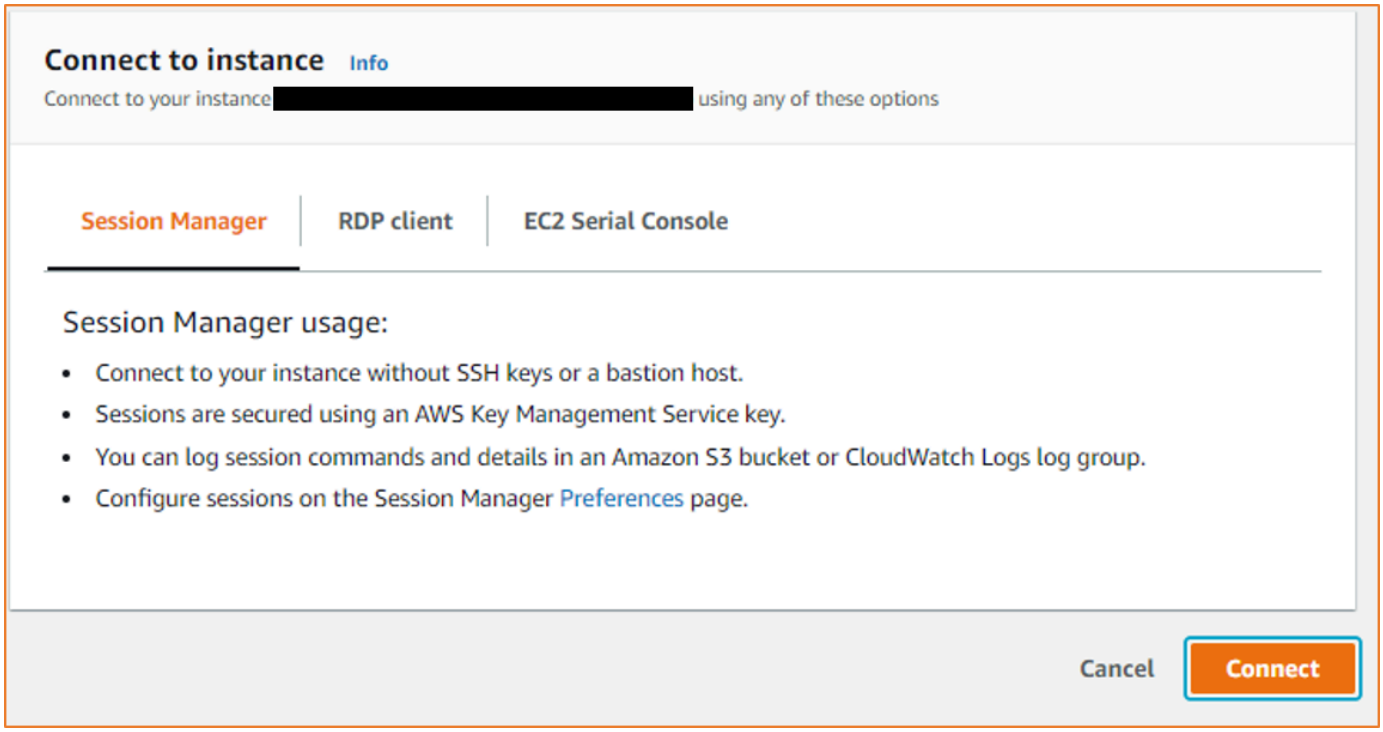
Connect to a Windows instance using Session Manager

Session Manager is a fully-managed AWS Systems Manager capability for managing your Amazon EC2 instances through an interactive, one-click, browser-based shell, or through the AWS CLI. You can use Session Manager to start a session with an instance in your account. After the session is started, you can run PowerShell commands as you would for any other connection type. For more information about Session Manager, see AWS Systems Manager Session Manager in the *AWS Systems Manager User Guide*.

Before attempting to connect to an instance using Session Manager, ensure that the necessary setup steps have been completed. For more information, see Setting up Session Manager.

To connect to a Windows instance using Session Manager on the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Connect**.
4. For **Connection method**, choose **Session Manager**.
5. Choose **Connect**.



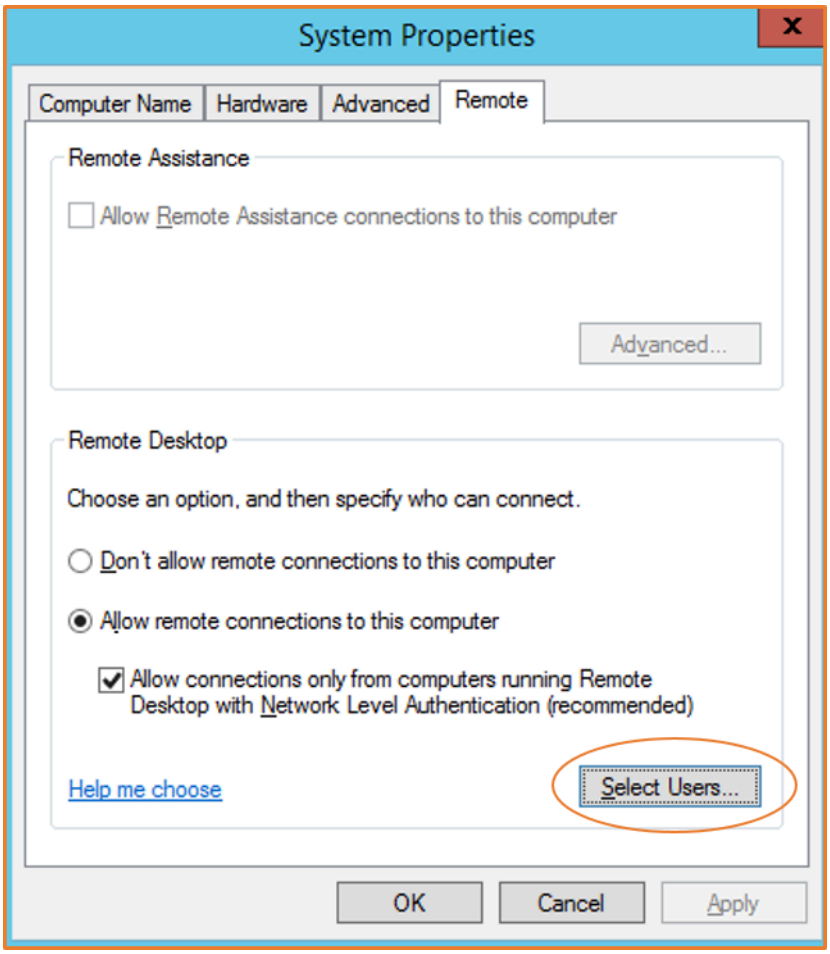
> **Tip**
>
> If you receive an error that you're not authorized to perform one or more Systems Manager actions (`ssm:`*`command-name`*), then you must update your policies to allow you to start sessions from the Amazon EC2 console. For more information and instructions, see Quickstart default IAM policies for Session Manager in the *AWS Systems Manager User Guide*.

Configure your accounts

After you connect, we recommend that you perform the following:

- Change the administrator password from the default value. You can change the password while you are logged on to the instance itself, just as you would on any computer running Windows Server.
- Create another user with administrator privileges on the instance. This is a safeguard in case you forget the administrator password or have a problem with the administrator account. The new user must have permission to access the instance remotely. Open **System Properties** by right-clicking on the **This PC** icon on your Windows desktop or File Explorer and selecting **Properties**.

Choose **Remote settings**, and choose **Select Users** to add the user to the **Remote Desktop Users** group.



Transfer files to Windows instances

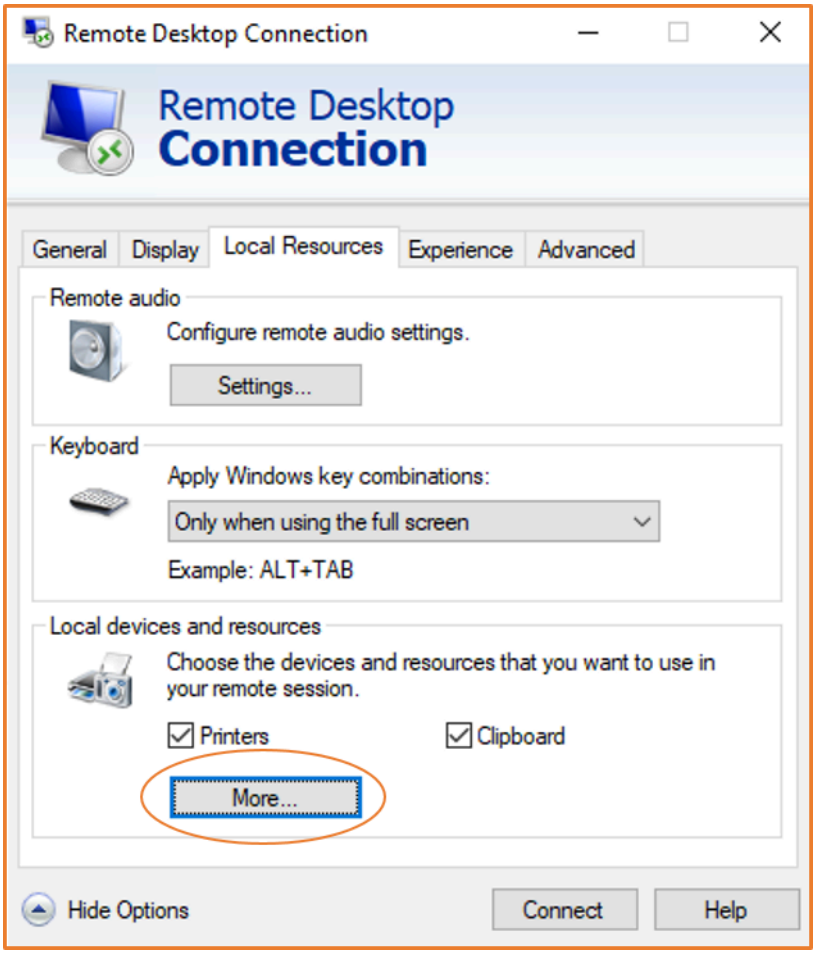
You can work with your Windows instance in the same way that you would work with any Windows server. For example, you can transfer files between a Windows instance and your local computer using the local file sharing feature of the Microsoft Remote Desktop Connection software. You can access local files on hard disk drives, DVD drives, portable media drives, and mapped network drives.

To access your local files from your Windows instances, you must enable the local file sharing feature by mapping the remote session drive to your local drive. The steps are slightly different depending on whether your local computer operating system is Windows or macOS X.

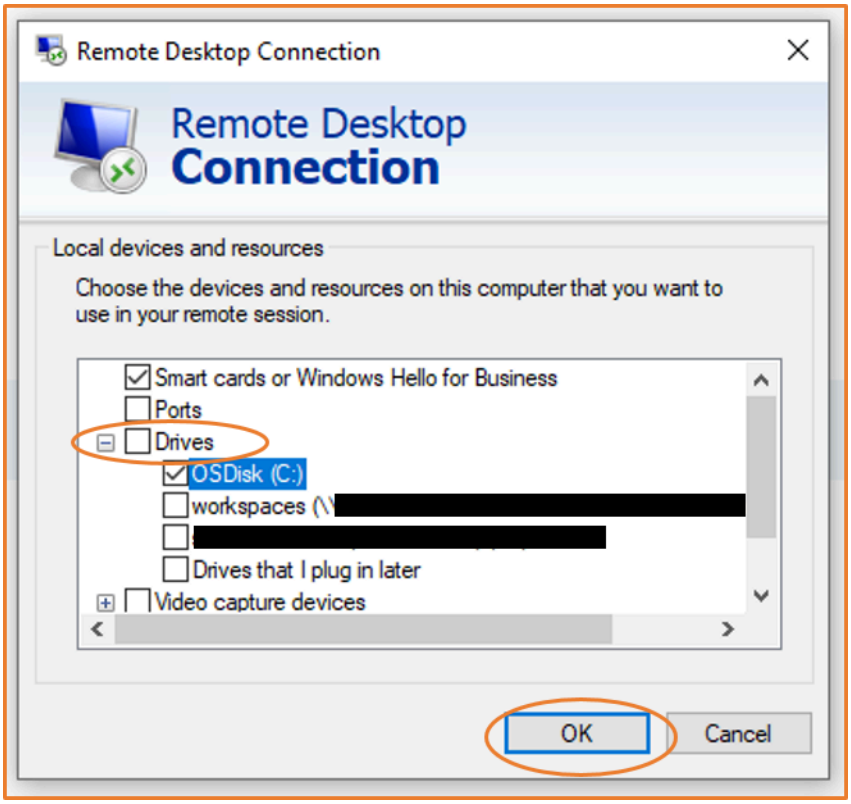
Windows

To map the remote session drive to your local drive on your local Windows computer

1. Open the Remote Desktop Connection client.
2. Choose **Show Options**.
3. Add the instance host name to the **Computer** field and user name to the **User name** field, as follows:
   a. Under **Connection settings**, choose **Open...**, and browse to the RDP shortcut file that you downloaded from the Amazon EC2 console. The file contains the Public IPv4 DNS host name, which identifies the instance, and the Administrator user name.
   b. Select the file and choose **Open**. The **Computer** and **User name** fields are populated with the values from the RDP shortcut file.
   c. Choose **Save**.
4. Choose the **Local Resources** tab.
5. Under **Local devices and resources**, choose **More...**



6. Open **Drives** and select the local drive to map to your Windows instance.
7. Choose **OK**.

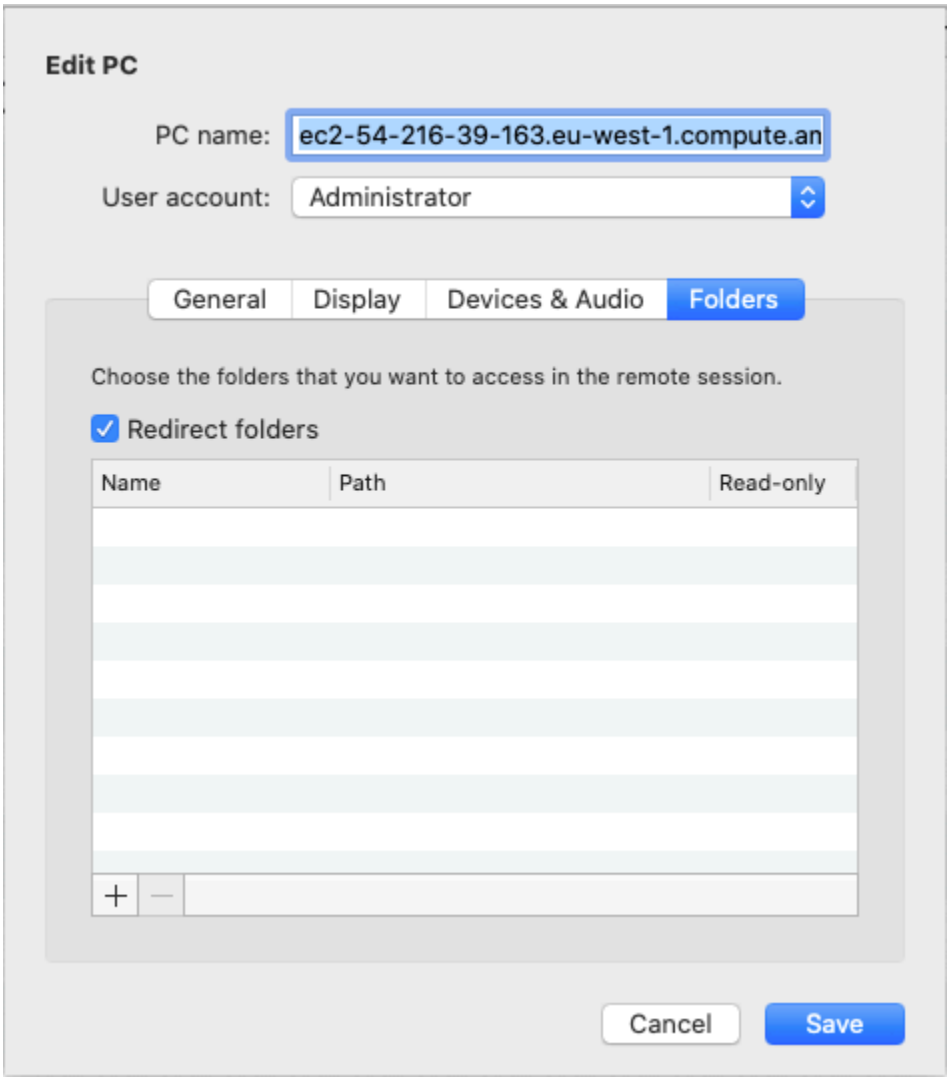


8. Choose **Connect** to connect to your Windows instance.

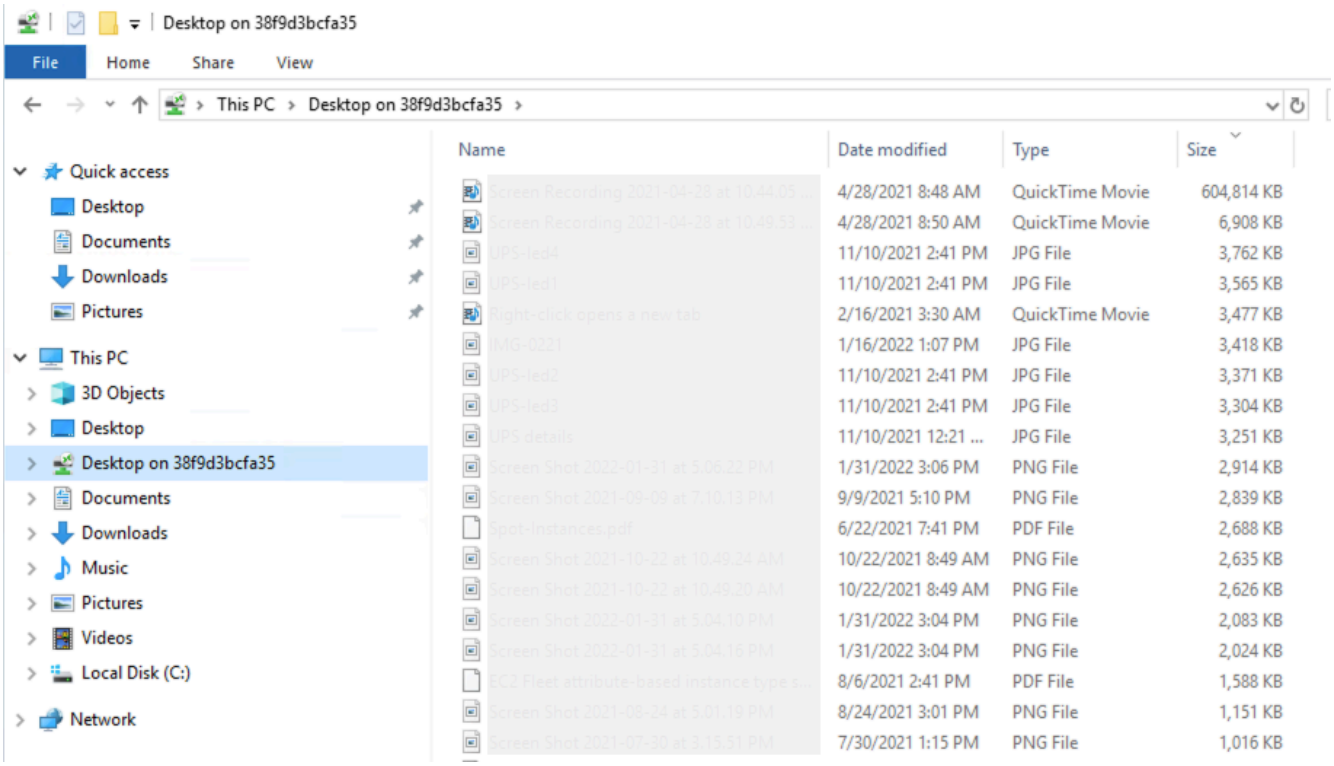
macOS X

To map the remote session drive to your local folder on your local macOS X computer

1. Open the Remote Desktop Connection client.
2. Browse to the RDP file that you downloaded from the Amazon EC2 console (when you initially connected to the instance), and drag it onto the Remote Desktop Connection client.
3. Right-click the RDP file, and choose **Edit**.
4. Choose the **Folders** tab, and select the **Redirect folders** check box.



5. Choose the **+** icon at bottom left, browse to the folder to map, and choose **Open**. Repeat this step for every folder to map.
6. Choose **Save**.
7. Choose **Connect** to connect to your Windows instance. You'll be prompted for the password.
8. On the instance, in File Explorer, expand **This PC**, and find the shared folder from which you can access your local files. In the following screenshot, the **Desktop** folder on the local computer was mapped to the remote session drive on the instance.



For more information on making local devices available to a remote session on a Mac computer, see Get started with the macOS client.

Connect to your instances without requiring a public IPv4 address using EC2 Instance Connect Endpoint

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.

How it works

First, you create an EC2 Instance Connect Endpoint in a subnet in your virtual private cloud (VPC). Then, when you want to connect to an instance, you specify the ID of the instance. You can optionally provide the EC2 Instance Connect Endpoint. The endpoint acts as a private tunnel to the instance.

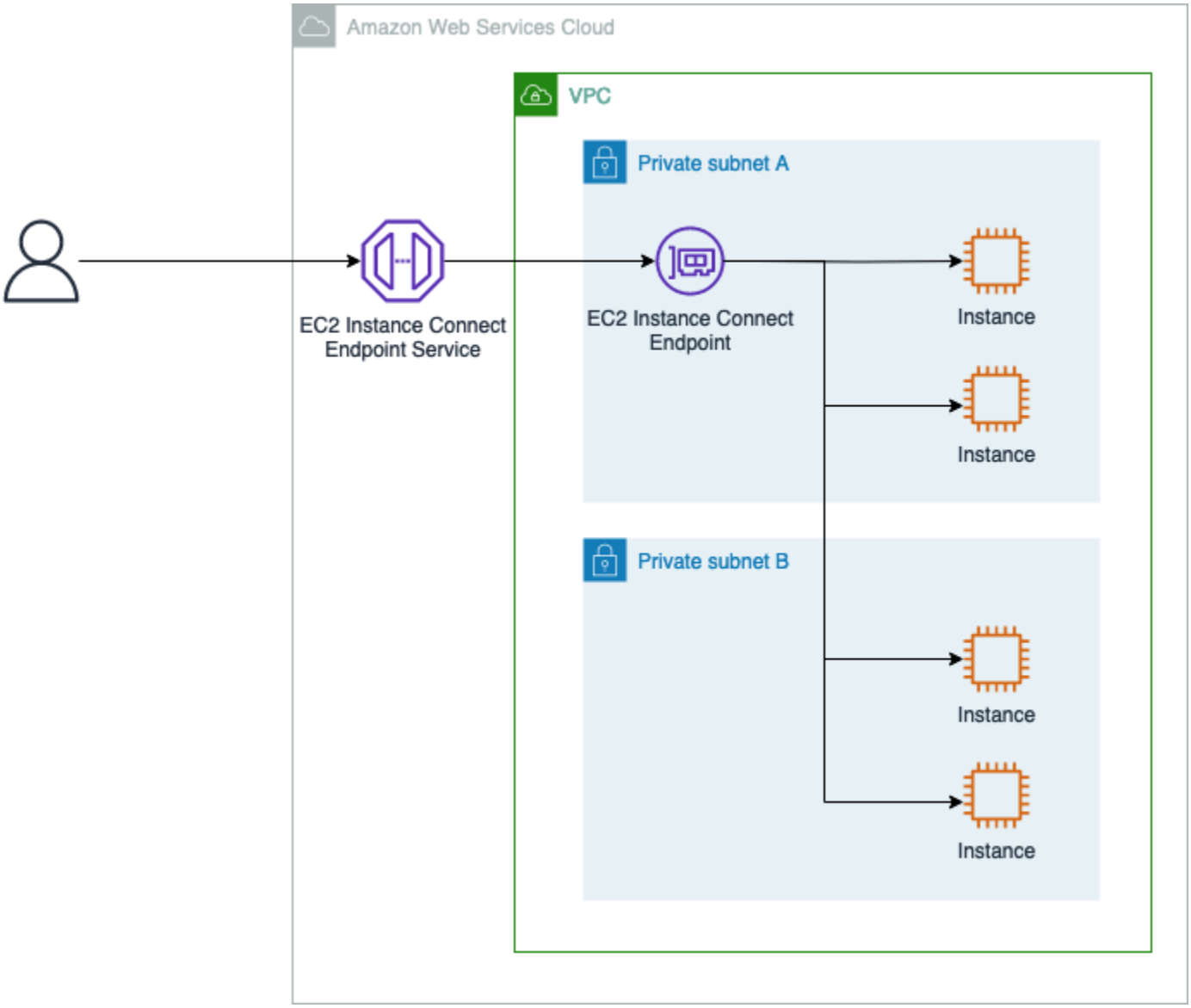
Once you create an EC2 Instance Connect Endpoint in a subnet, you can use the endpoint to connect to any instance in any subnet in your VPC provided your VPC is configured to allow subnets to communicate.

> **Note**
>
> If you use an EC2 Instance Connect Endpoint in one subnet to connect to an instance in another subnet that is in a different Availability Zone, there is an additional charge for data transfer across Availability Zones.

The following diagram shows a user from the internet connecting to their instances, which are located in private subnets in a VPC. The diagram illustrates the following key components:

- The **EC2 Instance Connect Endpoint Service** is an AWS service that allows the user to use the EC2 Instance Connect Endpoint to connect from the internet to their instances that are in private subnets.
- The **EC2 Instance Connect Endpoint** in Private subnet A acts as a private tunnel so that the user can connect to their instances that are in the private subnets.

  Access to create and connect to EC2 Instance Connect Endpoints is controlled by IAM permissions. You can configure additional security group rules on your instances to restrict inbound traffic. For example, you can use inbound rules on your instances to only allow traffic on management ports from the EC2 Instance Connect Endpoint.
- **Private subnet A** has an EC2 Instance Connect Endpoint, but **Private subnet B** does not. Based on your VPC configuration, if Private subnet A and Private subnet B are allowed to communicate, then you can use the EC2 Instance Connect Endpoint in Private subnet A to connect to an instance in Private subnet B.



Benefits

EC2 Instance Connect Endpoint provides the following 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 your VPC to have an internet gateway.
- You can control access to the creation and use of the EC2 Instance Connect Endpoints to connect to instances with IAM policies and permissions.

- All attempts to connect to instances, both successful and unsuccessful, are logged to CloudTrail.

Contents

- Prerequisites
- Grant IAM permissions to use EC2 Instance Connect Endpoint
- Security groups for EC2 Instance Connect Endpoint
- Create an EC2 Instance Connect Endpoint
- Connect using EC2 Instance Connect Endpoint to a Windows instance
- Log connections established over EC2 Instance Connect Endpoint
- Remove EC2 Instance Connect Endpoint
- Service-linked role for EC2 Instance Connect Endpoint
- Quotas

Prerequisites

The following are the prerequisites for using EC2 Instance Connect Endpoint to connect to an instance:

- AWS Regions
- AMIs
- IPv4 address
- Security groups
- Grant permissions

AWS Regions

Supported in all AWS Regions except Canada West (Calgary).

AMIs

The supported AMIs are dependent on how you connect to your instance with EC2 Instance Connect Endpoint.

You can connect to your instance using one of the following methods: EC2 console, SSH, or RDP.

IPv4 address

Your instance must have an IPv4 address (either private or public). EC2 Instance Connect Endpoint does not support connecting using an IPv6 address.

Security groups

The EC2 Instance Connect Endpoint and the instance that you want to connect to are each assigned security groups. For the recommended way to configure the security group rules, see Security groups for EC2 Instance Connect Endpoint.

Grant permissions

You must grant the required permissions to every IAM user who will use EC2 Instance Connect Endpoint to connect to an instance. For more information, see Grant IAM permissions to use EC2 Instance Connect Endpoint.

Grant IAM permissions to use EC2 Instance Connect Endpoint

To create or use an EC2 Instance Connect Endpoint, you must create an IAM policy that grants your users permissions for the following:

- Create, describe, and delete EC2 Instance Connect Endpoints
- Use the `ec2-instance-connect:OpenTunnel` action to use EC2 Instance Connect Endpoint to connect to instances

For information about creating IAM policies, see Creating IAM policies in the *IAM User Guide*.

Example IAM policies for EC2 Instance Connect Endpoint

- Allow users to create, describe, and delete EC2 Instance Connect Endpoints
- Allow users to use EC2 Instance Connect Endpoint to connect to instances
- Allow users to connect only from a specified source IP address range

Allow users 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": "*"
        }
    ]
}
```

Allow users 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` – Specifies 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` – Specifies 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` – Specifies 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 using EC2 Instance Connect Endpoint to a Windows instance.

A user can also be granted 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*.

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 `eice-123456789abcdef`. The connection is successfully established only if all the conditions are satisfied, for example, if the RDP connection is established on port 3389 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": "3389"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
```

```
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Allow users to connect only from a specified source 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 will be `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

The EC2 Instance Connect Endpoint and the instance that you want to connect to are each assigned security groups. We recommend that you configure the security group rules in the way described in this topic.

Topics

- EC2 Instance Connect Endpoint security group rule
- Instance security group rule
- Example

EC2 Instance Connect Endpoint security group rule

When you create the EC2 Instance Connect Endpoint, if you don't specify a security group, the default security group for the VPC is assigned. The default outbound rule allows all outbound traffic to all destinations. To limit connectivity to only the instances in the VPC, we recommend that the outbound rule only allows traffic to the specified destination.

Recommended outbound rule

- Allow outbound traffic to the specified destination (a security group or the VPC CIDR, depending on your security needs).

Instance security group rule

The instance needs at least one inbound rule to allow traffic from the EC2 Instance Connect Endpoint.

Recommended inbound rules

Specify one or more of the following rules, depending on your security needs and whether client IP preservation is enabled:

- Allow inbound traffic from the EC2 Instance Connect Endpoint security group.
- Allow inbound traffic from the client IP address.
- Allow inbound traffic from the VPC CIDR so that any instances in the VPC can send traffic to the destination instance.

The inbound rule that you specify is dependent on whether the EC2 Instance Connect Endpoint is configured to enable client IP preservation. Not all instance types support client IP preservation. For more information, see Limitations.

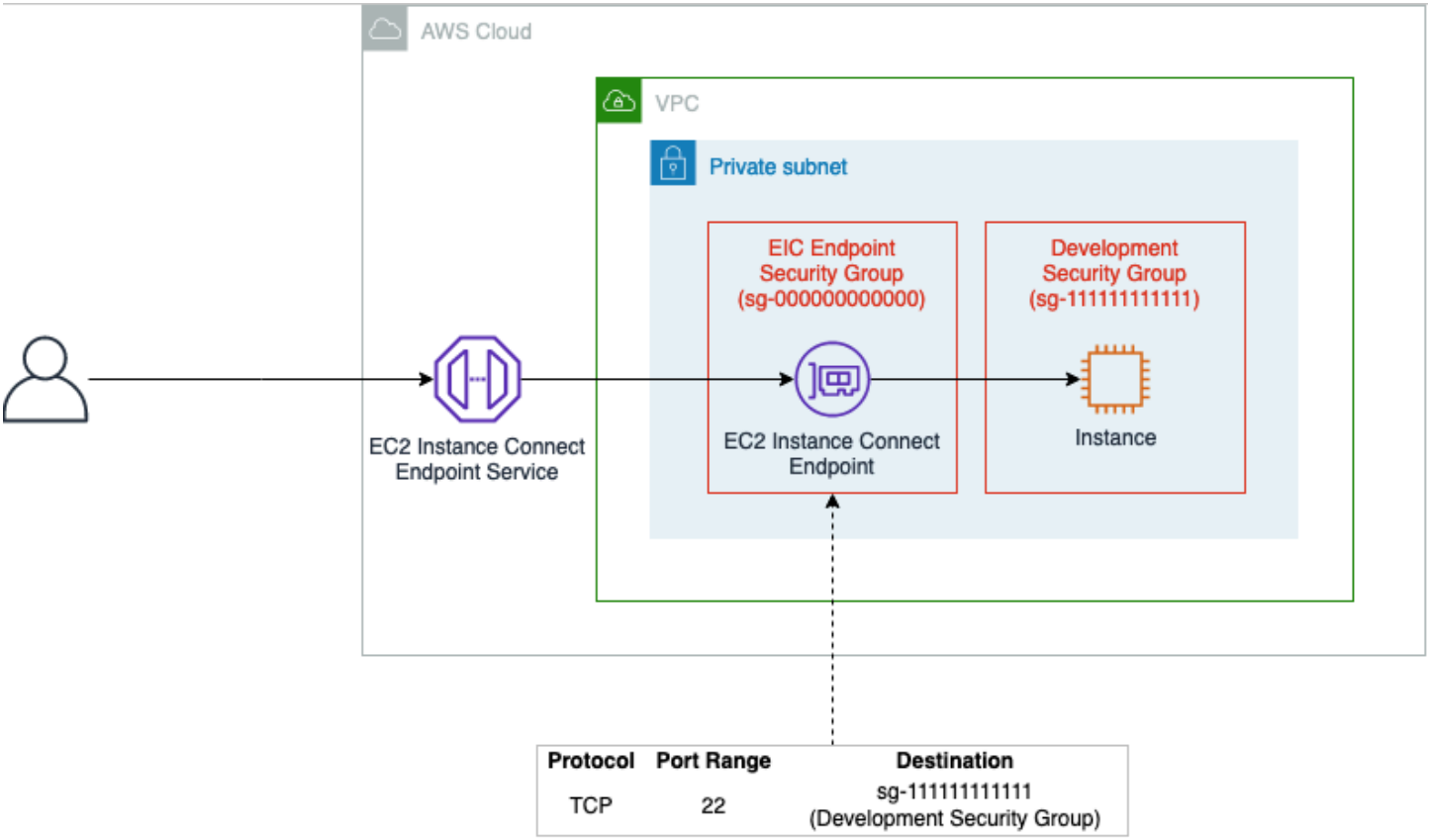
The following table lists the security group rules for the instance that can be configured depending on the value set for `preserveClientIp`.

Client IP preservation	Supported security group rules for the instance
`preserveClientIp=false`	• Allow inbound traffic from the EC2 Instance Connect Endpoint security group. • Allow inbound traffic from the VPC CIDR.
`preserveClientIp=true`	• Allow inbound traffic from the EC2 Instance Connect Endpoint security group. • Allow inbound traffic from the client IP address.

Example

In the following image, the EC2 Instance Connect Endpoint is assigned the security group **EIC Endpoint Security Group**. The **EIC Endpoint Security Group** has one outbound rule that allows

TCP traffic to the **Development Security Group**. This configuration means that the EC2 Instance Connect Endpoint can only send traffic to instances that are assigned the **Development Security Group**. In the image, the instance is assigned the **Development Security Group**, which means that, in this example, the EC2 Instance Connect Endpoint can send TCP traffic to the instance.



Create an EC2 Instance Connect Endpoint

You can create an EC2 Instance Connect Endpoint in a subnet in your VPC. You can then use the EC2 Instance Connect Endpoint to connect to instances in your VPC without requiring the instances to have a public IPv4 address.

EC2 Instance Connect Endpoint supports client IP preservation. You can configure the EC2 Instance Connect Endpoint to use your client's IP address as the source (`preserveClientIp` parameter is `true`) when connecting to an instance.

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.

> **Note**
>
> You can't modify an EC2 Instance Connect Endpoint after you've created it. If you want different endpoint settings, you'll need to delete the EC2 Instance Connect Endpoint and create a new one with the desired settings.

Prerequisites

You must have the required IAM permissions to create an EC2 Instance Connect Endpoint. For more information, see Allow users to create, describe, and delete EC2 Instance Connect Endpoints.

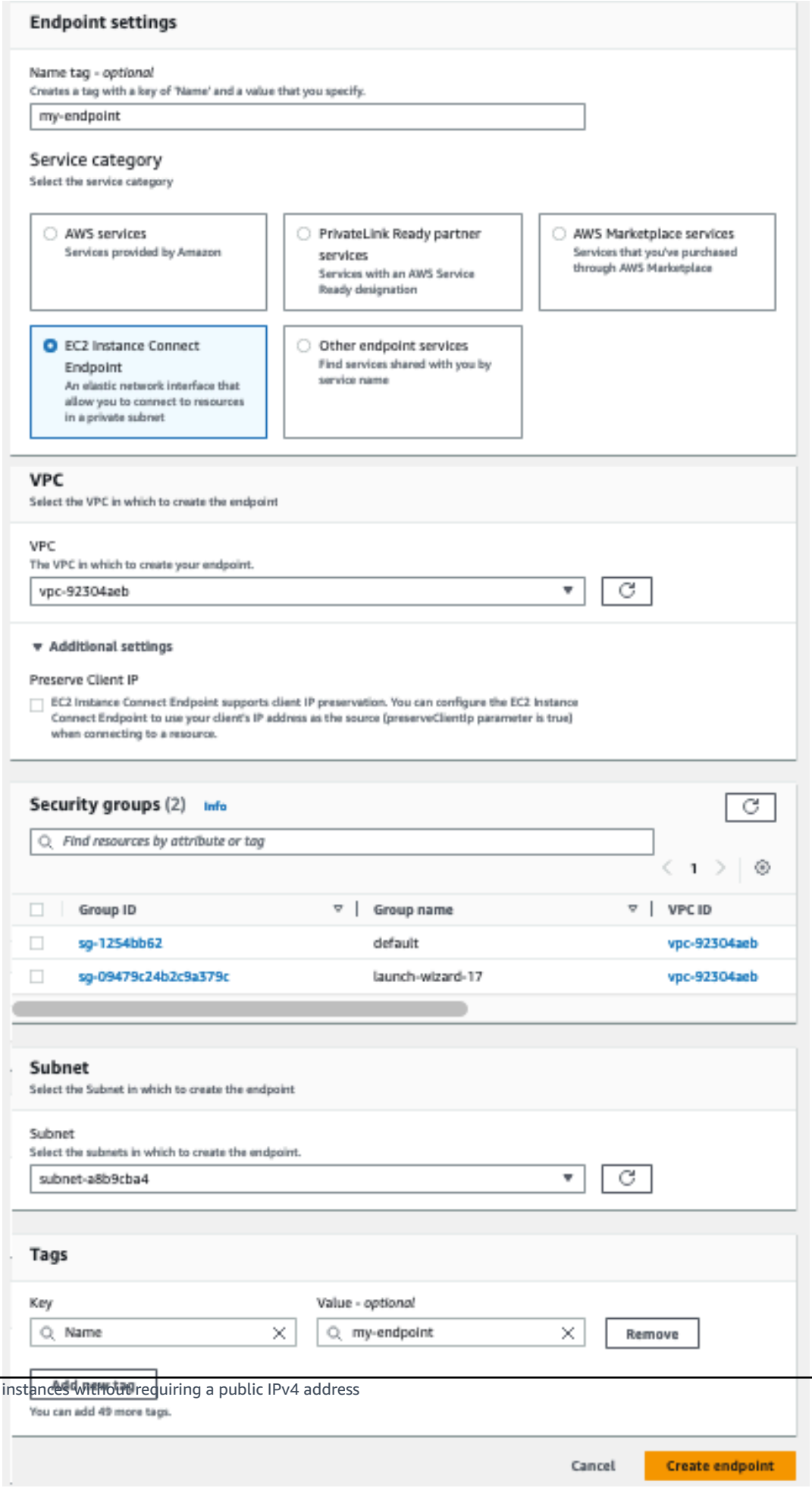
Create an EC2 Instance Connect Endpoint

Use one of the following methods to create an EC2 Instance Connect Endpoint.

Console

To create an EC2 Instance Connect Endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the left navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**, and then complete the settings in the dialog box, as follows:


Endpoint settings
Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
my-endpoint
Service category
Select the service category
AWS services
Services provided by Amazon
PrivateLink Ready partner services
Services with an AWS Service Ready designation
AWS Marketplace services
Services that you've purchased through AWS Marketplace
EC2 Instance Connect Endpoint
An elastic network interface that allow you to connect to resources in a private subnet
Other endpoint services
Find services shared with you by service name
VPC
Select the VPC in which to create the endpoint
VPC
The VPC in which to create your endpoint.
vpc-92304aeb
Additional settings
Preserve Client IP
EC2 Instance Connect Endpoint supports client IP preservation. You can configure the EC2 Instance Connect Endpoint to use your client's IP address as the source (preserveClientIp parameter is true) when connecting to a resource.
Security groups (2) Info
Find resources by attribute or tag
1
Group ID
Group name
VPC ID
sg-1254bb62
default
vpc-92304aeb
sg-09479c24b2c9a379c
launch-wizard-17
vpc-92304aeb
Subnet
Select the Subnet in which to create the endpoint
Subnet
Select the subnets in which to create the endpoint.
subnet-a8b9cba4
Tags
Key
Value - optional
Name
my-endpoint
Remove
Add new tag
You can add 49 more tags.
Cancel
Create endpoint

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 in which to create the endpoint.

d. Expand **Additional settings**, and for **Preserve Client IP** do one of the following:

- If you want your client's IP address to be used as the source when you connect to an instance, select the check box.

  Note: When **Preserve Client IP** is turned on, your instance's security group must allow traffic from your client IP address. For more information, see Instance security group rule.

- If you want the elastic network interface IP address to be used as the source when you connect to an instance, clear the check box. When **Preserve Client IP** is turned off, you can connect to any IP address that's routable from the VPC.

e. (Optional) For **Security groups**, select the security group to associate with the endpoint. If you don't select a security group, the default security group for your VPC will be associated with the endpoint. 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.

h. Choose **Create endpoint**.

   The initial status is **Pending**. Before you can connect to an instance using this endpoint, wait until the status is **Available**. This can take a few minutes. To monitor the endpoint status, see Describe an EC2 Instance Connect Endpoint.

AWS CLI

To create an EC2 Instance Connect Endpoint

Use the create-instance-connect-endpoint AWS CLI command and specify the subnet in which to create your EC2 Instance Connect Endpoint. Make sure you're using the latest version of the AWS CLI.

```
aws ec2 create-instance-connect-endpoint --region us-east-1 --subnet-
id subnet-0123456789example
```

Example output

```
 {
        "VpcId": "vpc-0123abcd",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "AvailabilityZone": "us-east-1a",
        "NetworkInterfaceIds": [
            "eni-0123abcd"
        ],
        "PreserveClientIp": true,
        "Tags": [],
        "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "StateMessage": "",
        "State": "create-complete",
        "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "SubnetId": "subnet-0123abcd",
        "OwnerId": "111111111111",
        "SecurityGroupIds": [
            "sg-0123abcd"
        ],
        "InstanceConnectEndpointId": "eice-0123456789example",
        "CreatedAt": "2023-04-07T15:43:53.000Z"
}
```

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`. This can take a few minutes. To monitor the endpoint state, see Describe an EC2 Instance Connect Endpoint.

Describe an EC2 Instance Connect Endpoint

Use one of the following methods to describe an EC2 Instance Connect Endpoint.

Console

To view 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. Find the endpoint in the table and select it to view its details. To use the endpoint to connect to an instance, the **Status** field must display **Available**.

AWS CLI

To describe an EC2 Instance Connect Endpoint

Use the describe-instance-connect-endpoints AWS CLI command and specify the EC2 Instance Connect Endpoint ID.

```
aws ec2 describe-instance-connect-endpoints --region us-east-1 --instance-connect-
endpoint-ids eice-0123456789example
```

Example output - To use the endpoint for connecting to an instance, the `State` field must show `create-complete`.

```
{
    "InstanceConnectEndpoints": [
        {
            "OwnerId": "111111111111",
            "InstanceConnectEndpointId": "eice-0123456789example",
            "InstanceConnectEndpointArn": "arn:aws:ec2:us-
east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
            "State": "create-complete",
            "StateMessage": "",
            "DnsName": "eice-0123456789example.b67b86ba.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
            "NetworkInterfaceIds": [
                "eni-0123456789example"
            ],
            "VpcId": "vpc-0123abcd",
            "AvailabilityZone": "us-east-1d",
            "CreatedAt": "2023-02-07T12:05:37+00:00",
            "SubnetId": "subnet-0123abcd",
            "Tags": []
        }
    ]
}
```

Connect using EC2 Instance Connect Endpoint to a Windows instance

EC2 Instance Connect Endpoint allows you to connect to an instance without requiring the instance to have a public IPv4 address. You can connect to any instances that support TCP.

To connect to an instance, specify the instance ID. You can optionally provide the EC2 Instance Connect Endpoint.

For information about how to connect to a **Linux instance**, see Connect using EC2 Instance Connect Endpoint to a Linux instance in the *Amazon EC2 User Guide for Linux Instances*.

Topics

- Limitations
- Prerequisites
- Connect to your Windows instance using RDP
- Troubleshoot

Limitations

- Only ports 22 and 3389 are supported.
- EC2 Instance Connect Endpoint doesn't support connections to an instance using IPv6 addresses.
- Each EC2 Instance Connect Endpoint can support up to 20 concurrent connections.
- EC2 Instance Connect Endpoint is intended specifically for management traffic use cases and not for high volume data transfers. High volume data transfers are throttled.
- Maximum duration for an established TCP connection: 1 hour (3,600 seconds). You can specify the maximum *allowed* duration in an IAM policy, which can be 3,600 seconds or less. For more information, see Allow users to use EC2 Instance Connect Endpoint to connect to instances.
- When client IP preservation is enabled, the instance to connect to must be in the same VPC as the EC2 Instance Connect Endpoint.
- Client IP preservation is not supported when traffic is routed through an AWS Transit Gateway.
- The following instance types do not support client IP preservation: C1, CG1, CG2, G1, G2, HI1, M1, M2, M3, and T1. If you are using these instance types, set the `preserveClientIp` parameter to `false`, otherwise attempting to connect to these instance types using EC2 Instance Connect Endpoint will fail. For more information about the `preserveClientIp` parameter, see step 3.d in the Create an EC2 Instance Connect Endpoint console procedure.

Prerequisites

- You must have the required IAM permission to connect to an EC2 Instance Connect Endpoint. For more information, see Allow users 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. To monitor the endpoint state, see Describe 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 AMIs.
- Ensure that the security group of the instance that you want to connect to is configured correctly for inbound traffic. For more information, see Instance security group rule.
- If you're using the AWS CLI, make sure 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*.

Connect to your Windows instance using RDP

You can use Remote Desktop Protocol (RDP) over EC2 Instance Connect Endpoint to connect to a Windows instance without a public IPv4 address or public DNS name.

To connect to your Windows instance using an RDP client

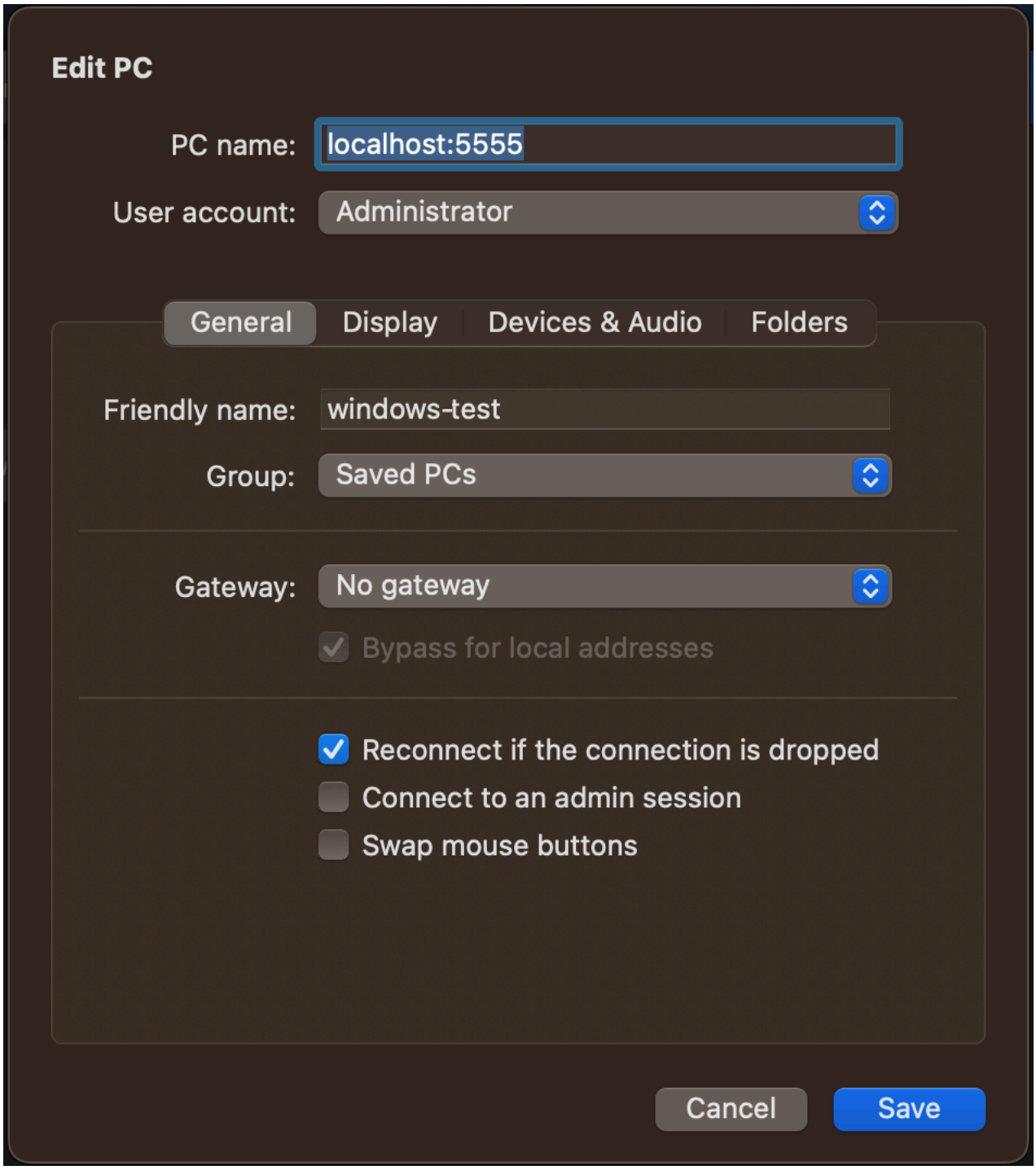
1. Complete Steps 1 – 8 in Connect to your Windows instance using RDP. After downloading the RDP desktop file at Step 8, you'll get an **Unable to connect** message, which is to be expected because your instance does not have a public IP address.
2. Run the following command to establish a private tunnel to the VPC in which the instance is located. `--remote-port` must be 3389 because RDP uses port 3389 by default.

```
aws ec2-instance-connect open-tunnel \
    --instance-id i-0123456789example \
    --remote-port 3389 \
    --local-port any-port
```

3. In your **Downloads** folder, find the RDP desktop file that you downloaded, and drag it onto the RDP client window.
4. Right-click the RDP desktop file and choose **Edit**.

5. In the **Edit PC** window, for **PC name** (the instance to connect to), enter `localhost:`*`local-port`*, where *`local-port`* uses the same value as you specified in Step 2, and then choose **Save**.

   Note that the following screenshot of the **Edit PC** window is from Microsoft Remote Desktop on a Mac. If you are using a Windows client, the window might be different.



6. In the RDP client, right-click the PC (that you just configured) and choose **Connect** to connect to your instance.
7. At the prompt, enter the decrypted password for the administrator account.

Troubleshoot

Use the following information to help diagnose and fix issues that you might encounter when using EC2 Instance Connect Endpoint to connect an instance.

Can't connect to your instance

The following are common reasons why you might not be able to connect to your instance.

- Security groups – Check the security groups assigned to the EC2 Instance Connect Endpoint and your instance. For more information about the required security group rules, see Security groups for EC2 Instance Connect Endpoint.
- Instance state – Verify that your instance is in the `running` state.
- Key pair – If the command you're using to connect requires a private key, verify that your instance has a public key and that you have the corresponding private key.
- IAM permissions – Verify that you have the required IAM permissions. For more information, see Grant IAM permissions to use EC2 Instance Connect Endpoint.

For more troubleshooting tips, see Troubleshoot connecting to your Windows 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": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
         "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
         "accountId": "123456789012",
         "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
         "userName": "IAM-friendly-name"
     },
     "eventTime": "2023-04-11T23:50:40Z",
     "eventSource": "ec2-instance-connect.amazonaws.com",
     "eventName": "OpenTunnel",
     "awsRegion": "us-east-1",
     "sourceIPAddress": "1.2.3.4",
```

```
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceConnectEndpointId": "eici-0123456789EXAMPLE",
        "maxTunnelDuration": "3600",
        "remotePort": "22",
        "privateIpAddress": "10.0.1.1"
    },
    "responseElements": null,
    "requestID": "98deb2c6-3b3a-437c-a680-03c4207b6650",
    "eventID": "bbba272c-8777-43ad-91f6-c4ab1c7f96fd",
    "readOnly": false,
    "resources": [{
        "accountId": "123456789012",
        "type": "AWS::EC2::InstanceConnectEndpoint",
        "ARN": "arn:aws:ec2:us-east-1:123456789012:instance-connect-endpoint/
eici-0123456789EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


Remove EC2 Instance Connect Endpoint

To remove EC2 Instance Connect Endpoint from your VPC, delete the endpoint that was created in a subnet.

When you delete an EC2 Instance Connect Endpoint, it first enters the **Deleting** (console) or `delete-in-progress` (AWS CLI) state, and then the `delete-complete` (AWS CLI) state. In the console, a deleted endpoint no longer appears. If the delete action fails, the state is `delete-failed`, and the **Status message** (console) or `StateMessage` (AWS CLI) provides the failure reason.

Use one of the following methods to delete an EC2 Instance Connect Endpoint.

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 eice-03f5e49b83924bbc7
```

Example output

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

Service-linked role for EC2 Instance Connect Endpoint

Amazon EC2 uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon EC2. Service-linked roles are predefined by Amazon EC2 and include all the permissions that Amazon EC2 requires to call other

AWS services on your behalf. For more information, see Using service-linked roles in the *IAM User Guide*.

When you create an EC2 Instance Connect Endpoint, the service-linked role named **AWSServiceRoleForEC2InstanceConnect** and the managed policy named **EC2InstanceConnectEndpoint** are automatically created in your AWS account, and the managed policy is automatically attached to the service-linked role.

Amazon EC2 uses **AWSServiceRoleForEC2InstanceConnect** to manage network interfaces in your account that are required when creating EC2 Instance Connect Endpoints.

Permissions granted by AWSServiceRoleForEC2InstanceConnect

Amazon EC2 uses **AWSServiceRoleForEC2InstanceConnect** to complete the following actions:

- `ec2:CreateNetworkInterface` – Create network interfaces
- `ec2:DeleteNetworkInterface` – Delete network interfaces
- `ec2:DescribeNetworkInterfaces` – Describe network interfaces
- `ec2:DescribeAvailabilityZones` – Describe Availability Zones
- `ec2:ModifyNetworkInterfaceAttribute` – Disable source and destination checks

Use service-linked roles

EC2 Instance Connect Endpoint uses the service-linked role named **AWSServiceRoleForEC2InstanceConnect** to provision network interfaces in your account that are required to use the service.

If you create an EC2 Instance Connect Endpoint, the **EC2InstanceConnectEndpoint** managed policy is automatically created in your AWS account and attached to the **AWSServiceRoleForEC2InstanceConnect** service-linked role.

Service Linked Roles for EC2 Instance Connect Endpoint

The **AWSServiceRoleForEC2InstanceConnect** service-linked role trusts the following services to assume the role:

- **ec2-instance-connect.amazonaws.com**

The role permissions policy, named **EC2InstanceConnectEndpoint**, allows EC2 Instance Connect Endpoint to complete the following actions on the specified resources:

- Action: `ec2:CreateNetworkInterface` – On all subnets and all network interfaces with the non-null tag key **InstanceConnectEndpointId** to create network interfaces for EC2 Instance Connect Endpoint
- Action: `ec2:CreateTags` – On all network interfaces created for an EC2 Instance Connect Endpoint at creation time with the tag key **InstanceConnectEndpointId**
- Action: `ec2:DeleteNetworkInterface` – On network interfaces created for an EC2 Instance Connect Endpoint with tag key **InstanceConnectEndpointId**
- Action: `ec2:DescribeNetworkInterfaces` – On network interfaces for an Instance Connect Endpoint
- Action: `ec2:DescribeAvailabilityZones` – For internal mapping of customer's Availability Zone
- Action: `ec2:ModifyNetworkInterfaceAttribute` – On all network interfaces to disable source and destination checks

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2-instance-connect.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
```

```
    ]
}
```

Permissions policies

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": "arn:aws:ec2:*:*:subnet/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "InstanceConnectEndpointId"
                    ]
                },
                "Null": {
                    "aws:RequestTag/InstanceConnectEndpointId": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/InstanceConnectEndpointId": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "InstanceConnectEndpointId"
                    ]
                },
                "Null": {
                    "aws:RequestTag/InstanceConnectEndpointId": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/InstanceConnectEndpointId": [
                        "eice-*"
                    ]
                }
            }
        }
    ]
```

```
}
```

Creating a service-linked role for EC2 Instance Connect Endpoint

When you create an EC2 Instance Connect Endpoint, the service-linked role **AWSServiceRoleForEC2InstanceConnect** is automatically created for you.

> **Important**
>
> Ensure that the AWS account used to create the EC2 Instance Connect Endpoint has an IAM policy attached to it that permits the `iam:CreateServiceLinkedRole` action.

Editing a service-linked role for EC2 Instance Connect Endpoint

EC2 Instance Connect Endpoint doesn't allow you to edit the **AWSServiceRoleForEC2InstanceConnect** service-linked role.

Deleting 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.

> **Note**
>
> You can delete the service-linked role only after you delete all EC2 Instance Connect Endpoint resources.

Use the AWS CLI to delete the service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Follow these steps to delete the service-linked role using the AWS CLI:

1. Delete all EC2 Instance Connect Endpoints using the `delete-instance-connect-endpoint` command, this will also remove the associated resources.
2. Delete the service linked role using the `delete-service-linked-role` command. Deleting the service-linked role will also delete the associated managed policy.

EC2 Instance Connect Endpoint supports using the **AWSServiceRoleForEC2InstanceConnect** service-linked role in every AWS Region where the service is available.

AWS managed policies for EC2 Instance Connect Endpoint

AWS managed policy: EC2InstanceConnectEndpoint

This policy is attached to a service-linked role that allows EC2 Instance Connect Endpoint to perform actions on your behalf. For more information, see EC2InstanceConnectEndpoint.

To view the permissions for this policy, see Ec2InstanceConnectEndpoint in the AWS Management Console.

EC2 Instance Connect Endpoint updates to AWS managed policies

View details about updates to AWS managed policies for EC2 Instance Connect Endpoint since this service began tracking these changes.

Change	Description	Date
EC2 Instance Connect Endpoint started tracking changes	EC2 Instance Connect Endpoint started tracking changes for its AWS managed policies.	June 13, 2023

Quotas

You can create a maximum number of EC2 Instance Connect Endpoints per AWS Region as follows:

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

Each EC2 Instance Connect Endpoint can support a maximum number of concurrent connections as follows:

Description	Quota
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**.

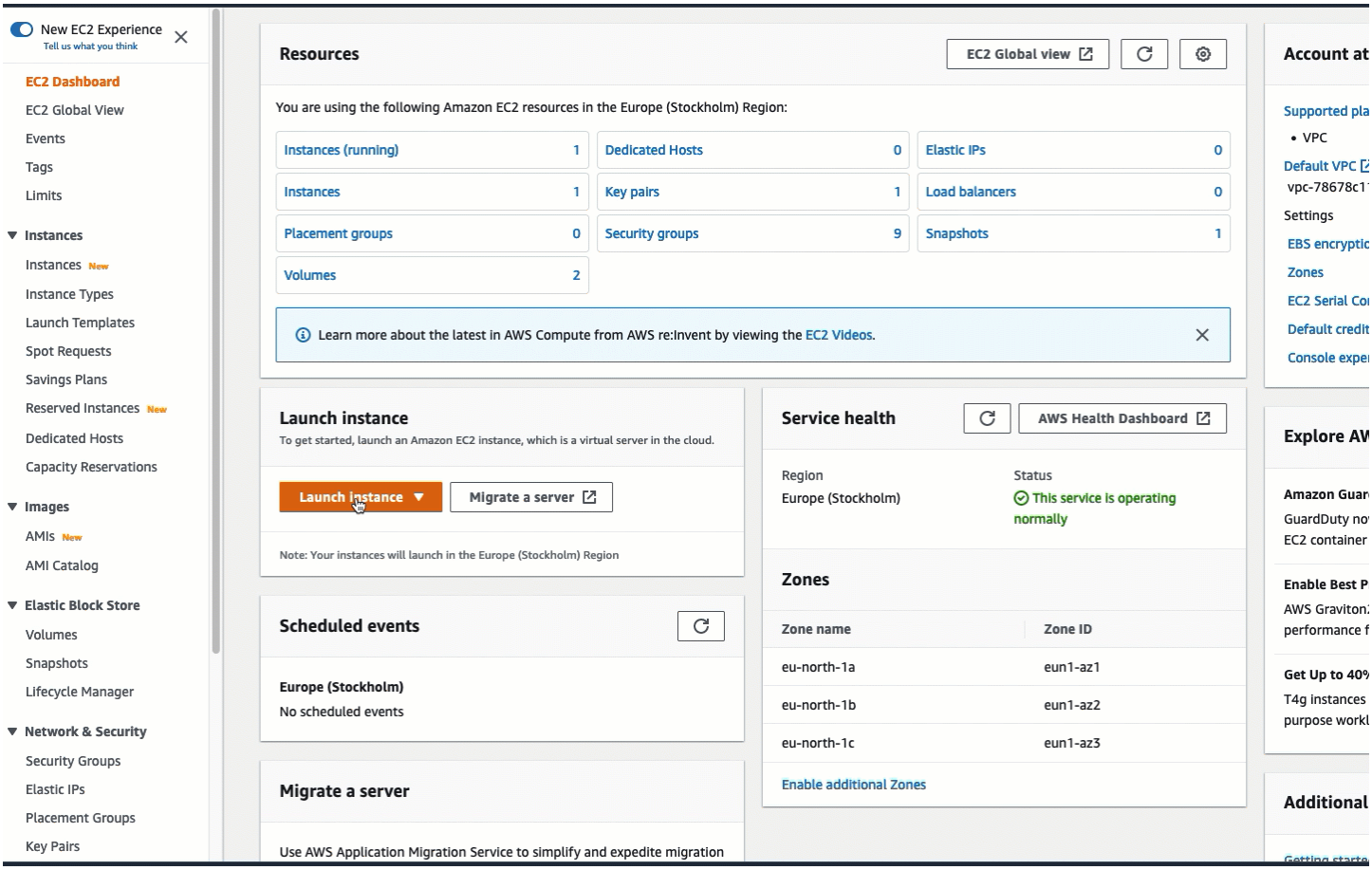
To automatically connect a newly-launched EC2 instance to an RDS database using the EC2 console

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



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:

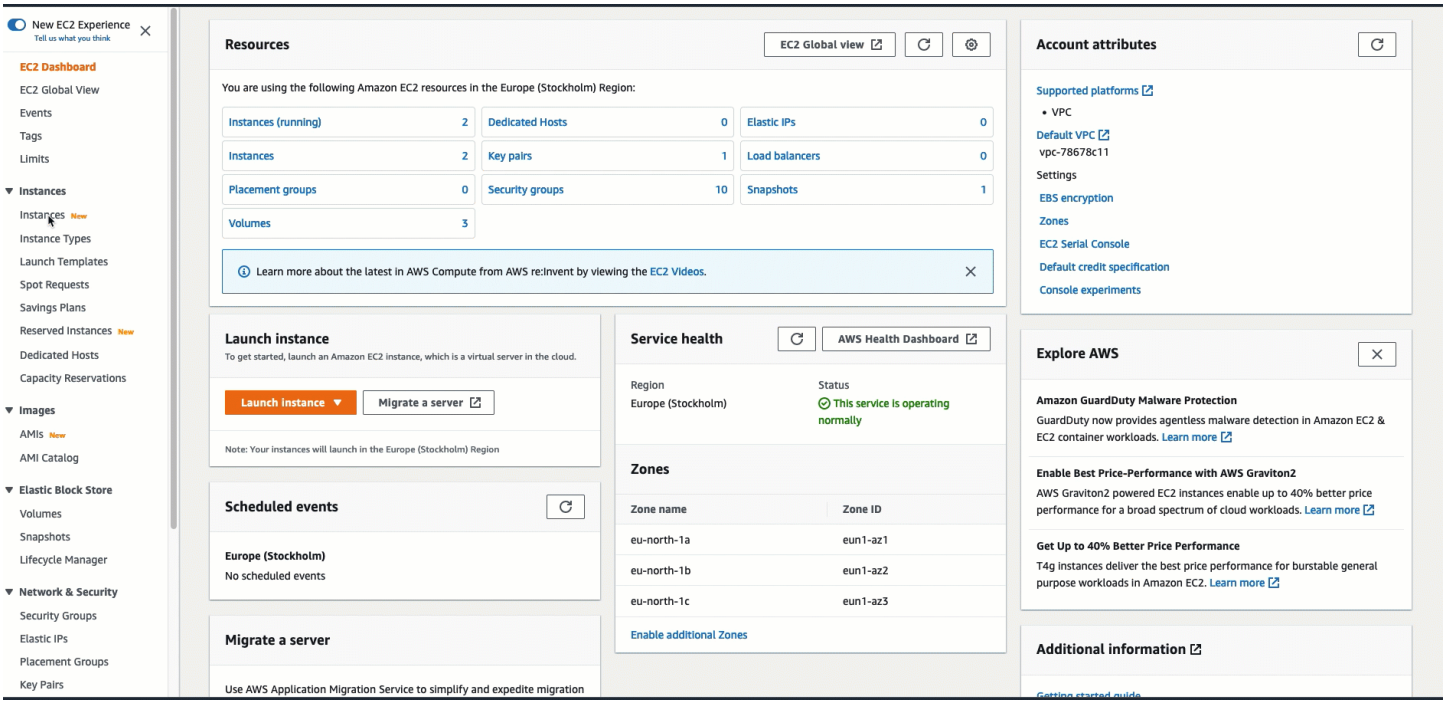
   a. For **Database role**, choose either **Cluster** or **Instance**.

   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



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

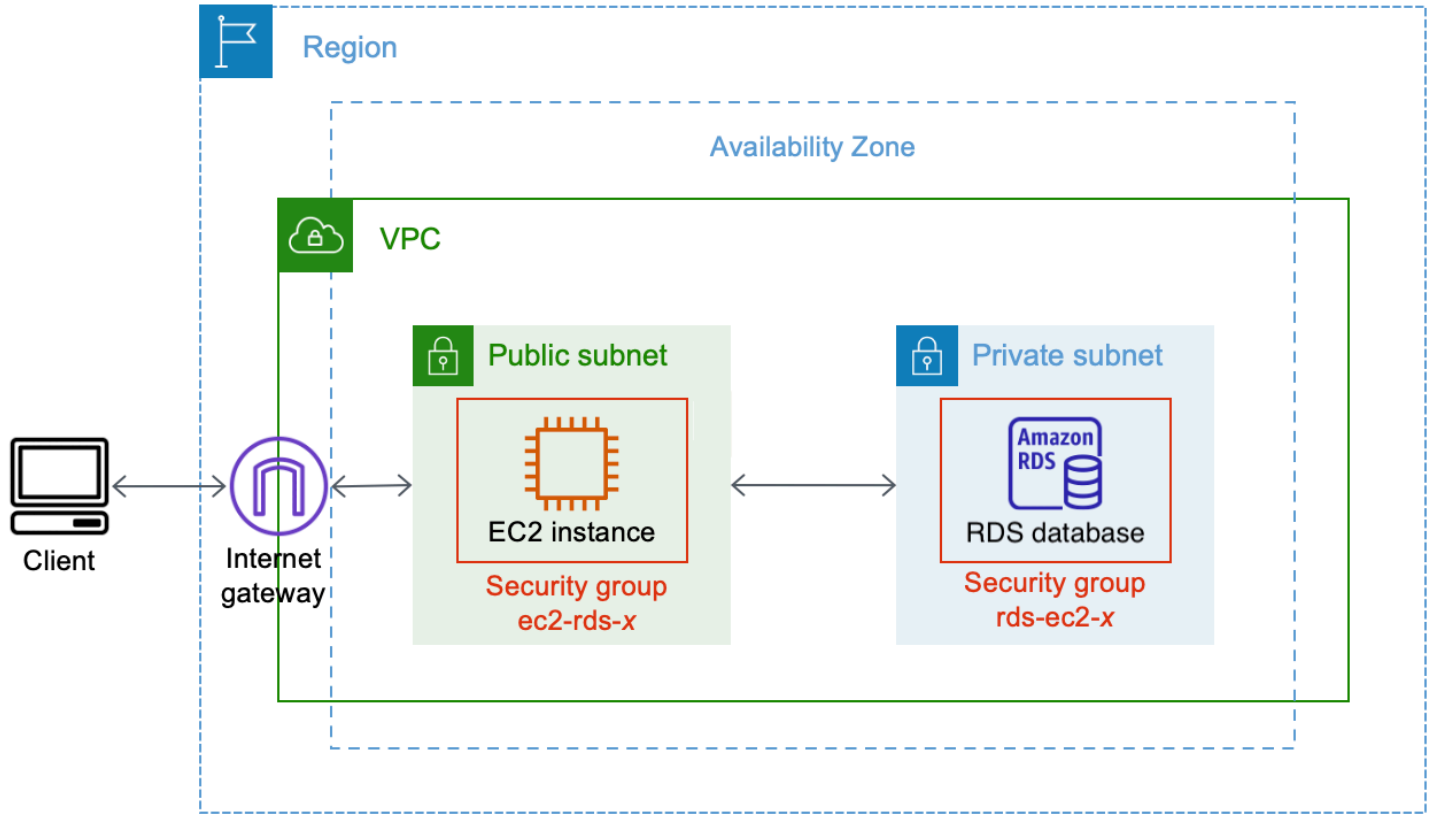
  Configure the connection between your EC2 instance to your RDS database by manually configuring and assigning the security groups to reproduce the configuration that is automatically created by the automatic connection feature in Option 1 and Option 2.

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`**.

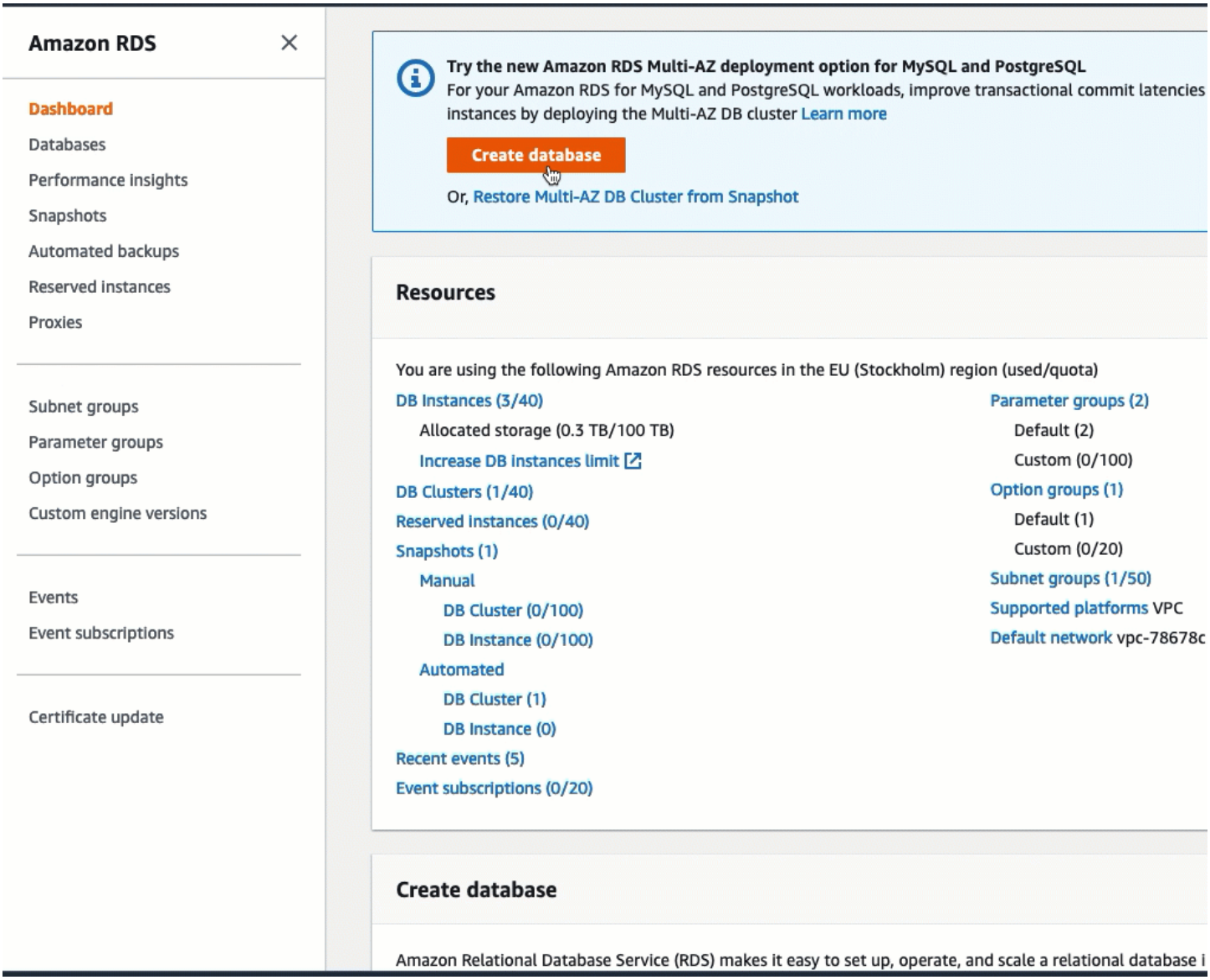
   c. For **Master password**, enter a password that you can remember for this tutorial, and then, for **Confirm password**, enter the password again.
8. Under **Instance configuration**, for **DB instance class**, leave the default, which is **db.t3.micro**. If your account is less than 12 months, you 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



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`.

   > **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:

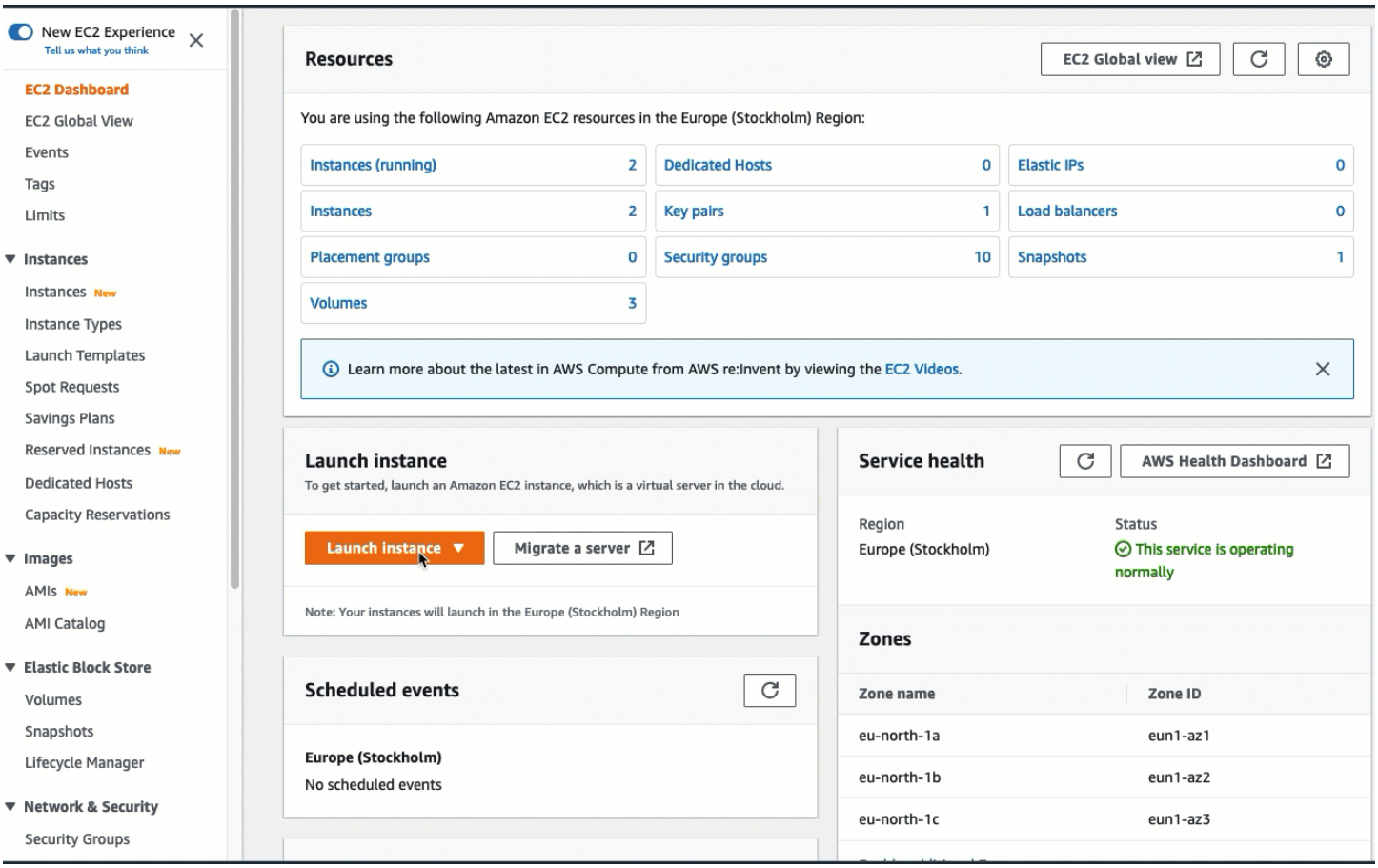
   - **Allow HTTPs traffic from the internet**
   - **Allow HTTP traffic from the internet**

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



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.

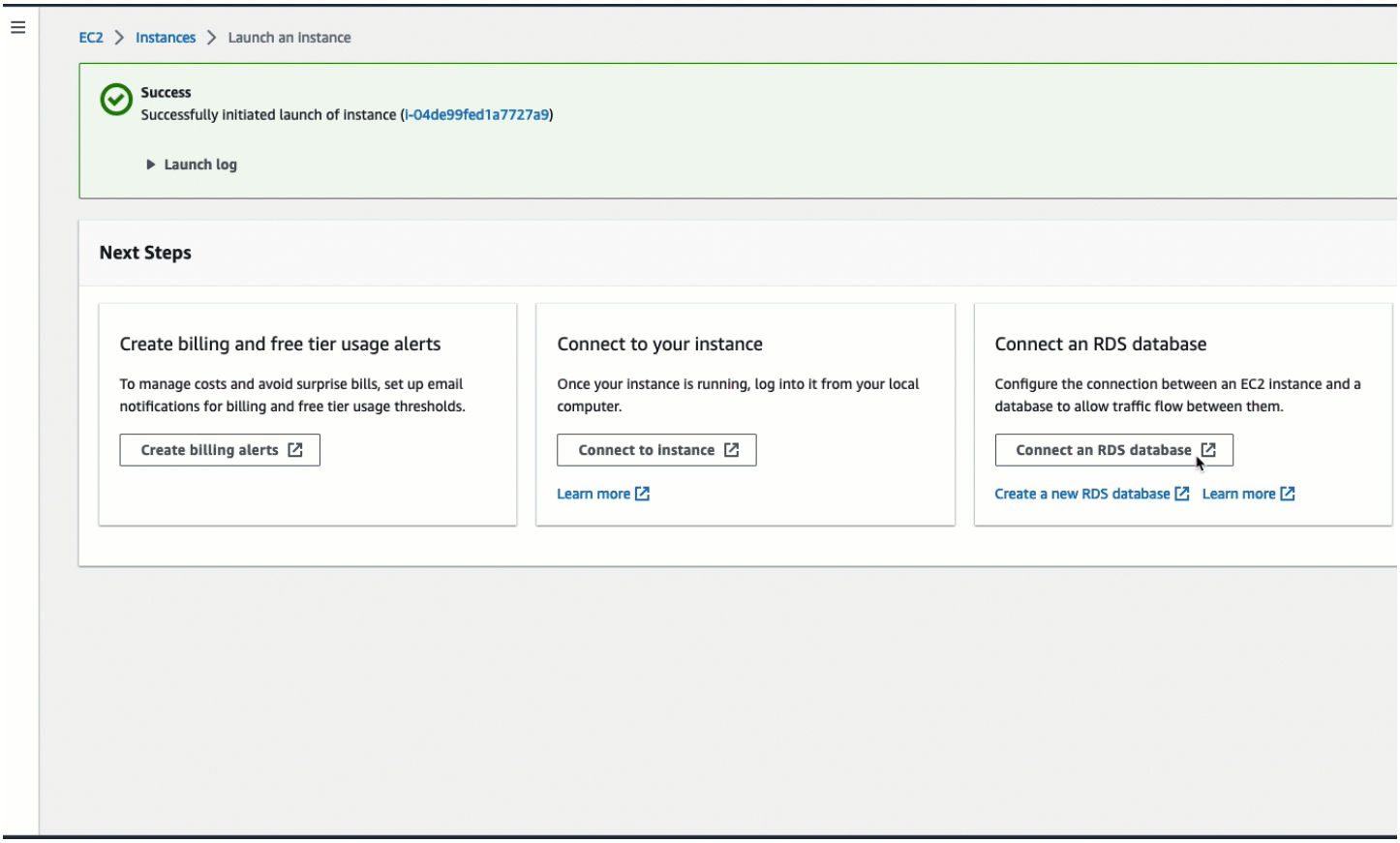
2. For **Database role**, choose **Instance**. *Instance* in this case refers to the database instance.

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



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

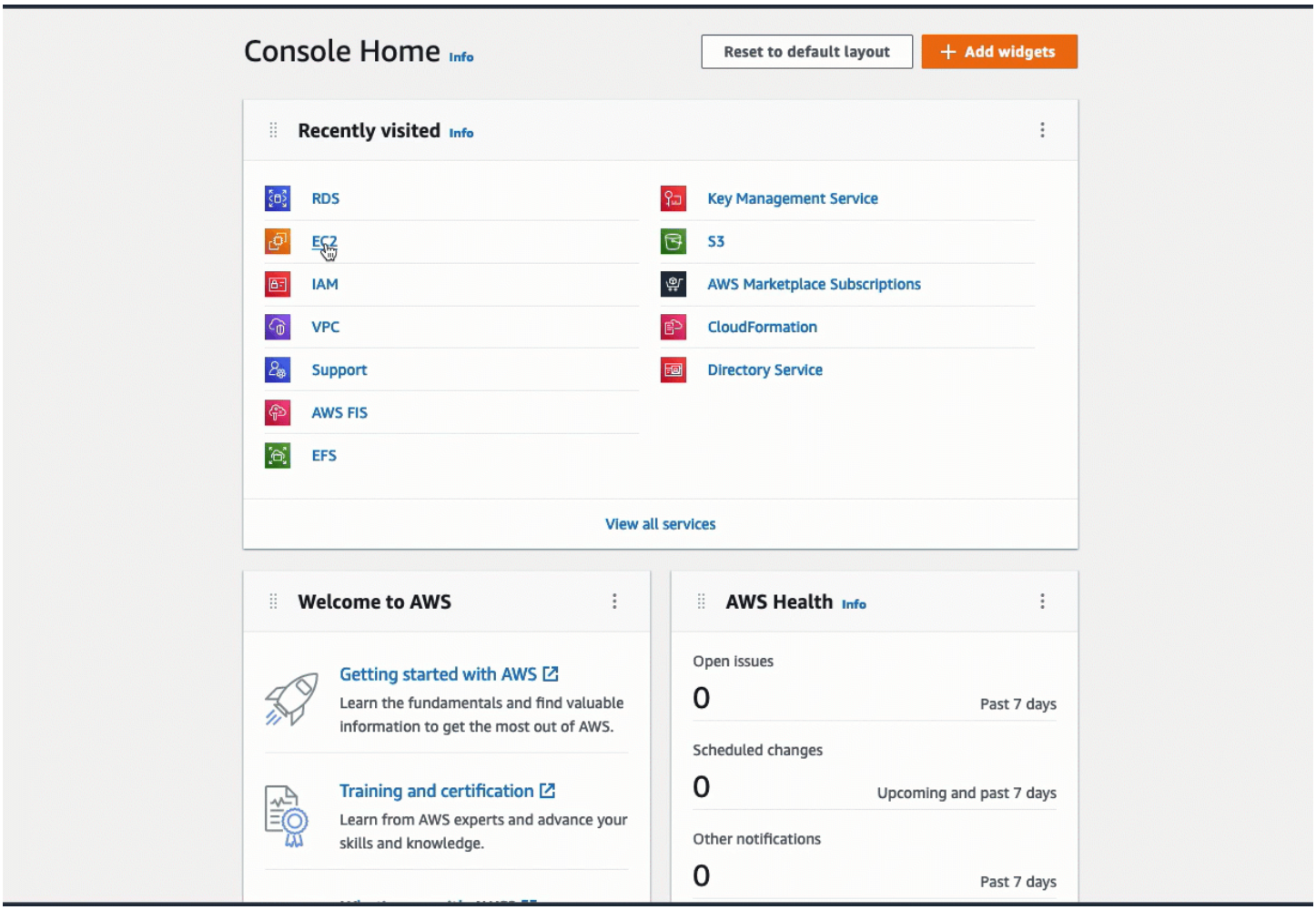
1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation page, choose **Databases**.
3. Choose the RDS database that you created for this tutorial.
4. On the **Connectivity & security** tab, under **Security**, **VPC security groups**, verify that a security group called **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



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`.

   > **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:

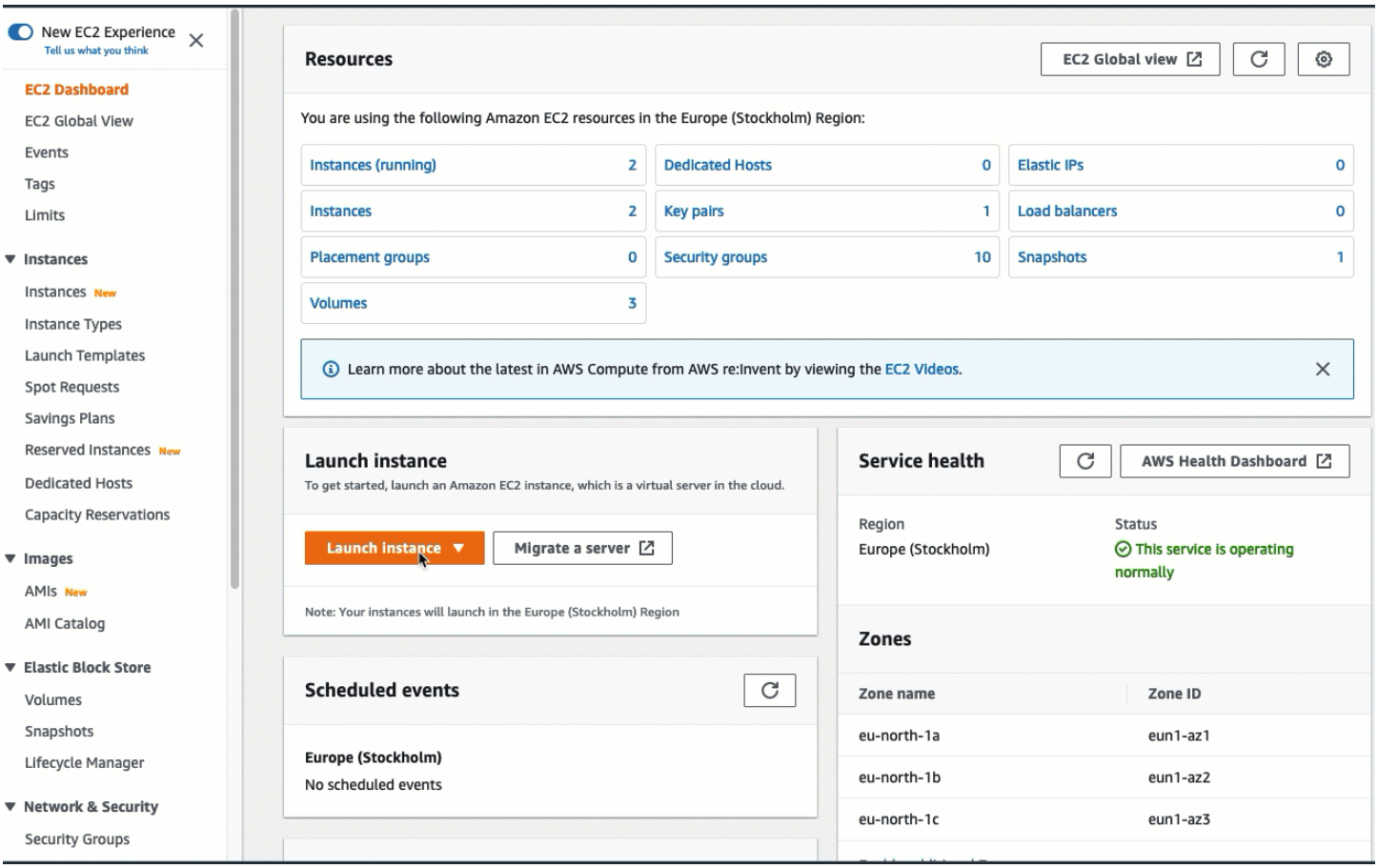
   - **Allow HTTPs traffic from the internet**
   - **Allow HTTP traffic from the internet**

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



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.

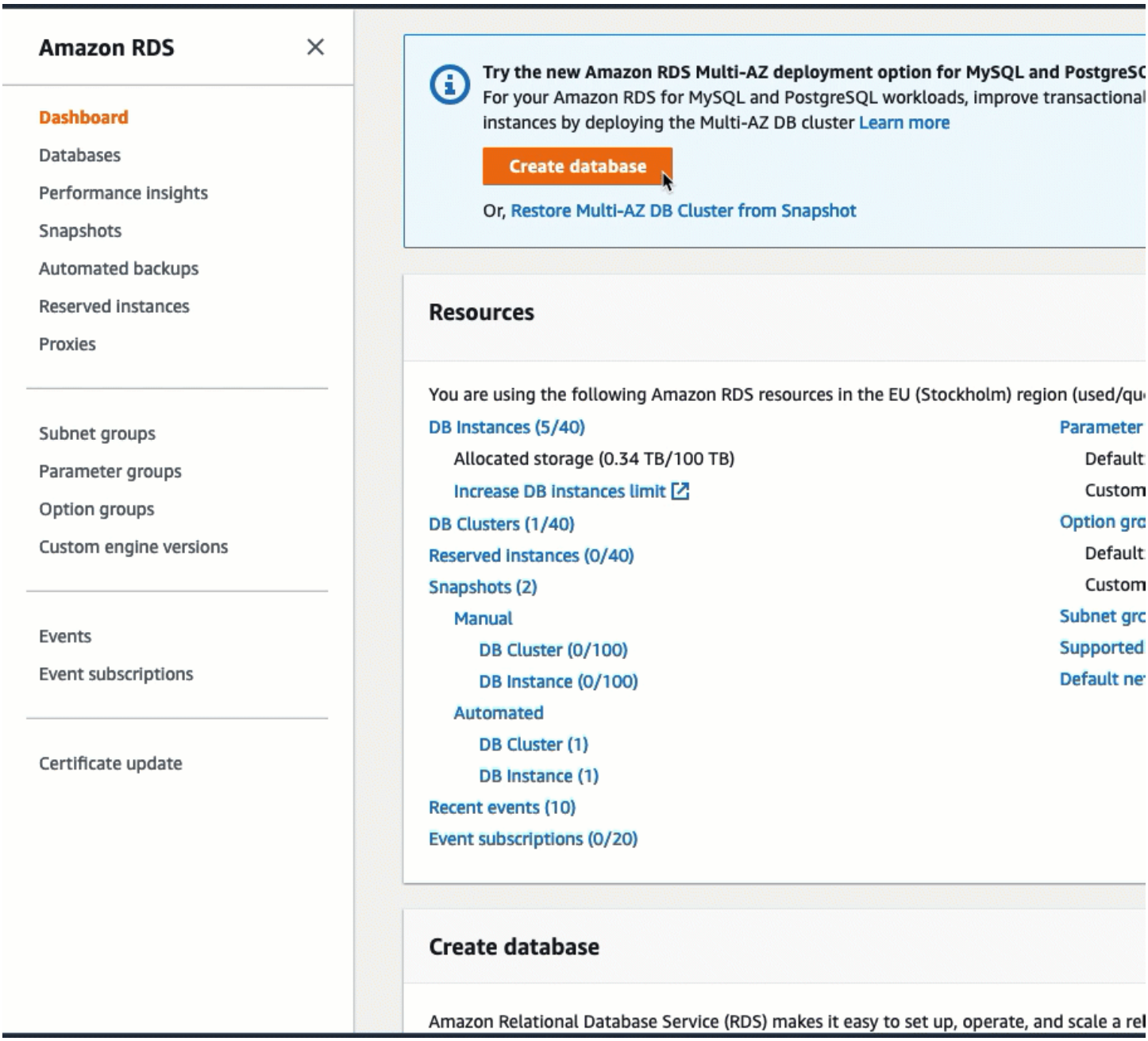
8. Under **Instance configuration**, for **DB instance class**, leave the default, which is **db.t3.micro**. If your account is less than 12 months, you can use this instance for free. Other restrictions apply. For more information, see AWS Free Tier.
9. Under **Connectivity**, for **Compute resource**, 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 configure to allow traffic between the EC2 instance and the RDS database. In the next task, you'll verify that the security groups were created and assigned to the EC2 instance and RDS database.
11. Choose **Create database**.

    On the **Databases** screen, the **Status** of the new database is **Creating** until the database is ready to use. When the status changes to **Available**, you can connect to the database. Depending on the database class and the amount of storage, it can take up to 20 minutes before the new database is available.

To learn more, see Configure automatic network connectivity with an EC2 instance in the *Amazon RDS User Guide*.

View an animation: Create an RDS database and automatically connect it to an EC2 instance



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

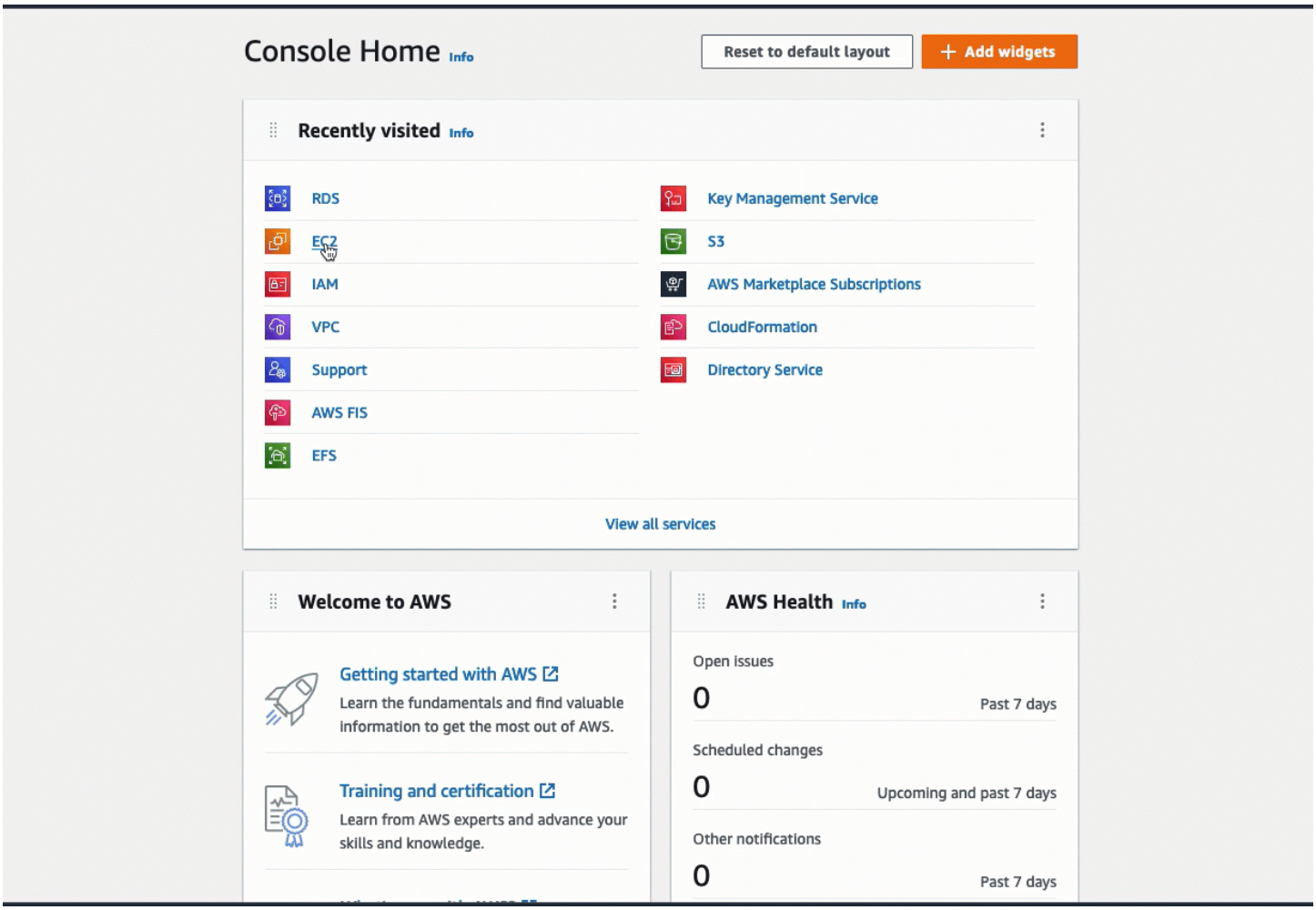
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Choose the EC2 instance that you selected to connect to the RDS database in the previous task, and choose the **Security** tab.
4. Under **Security details**, **Security groups**, 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:

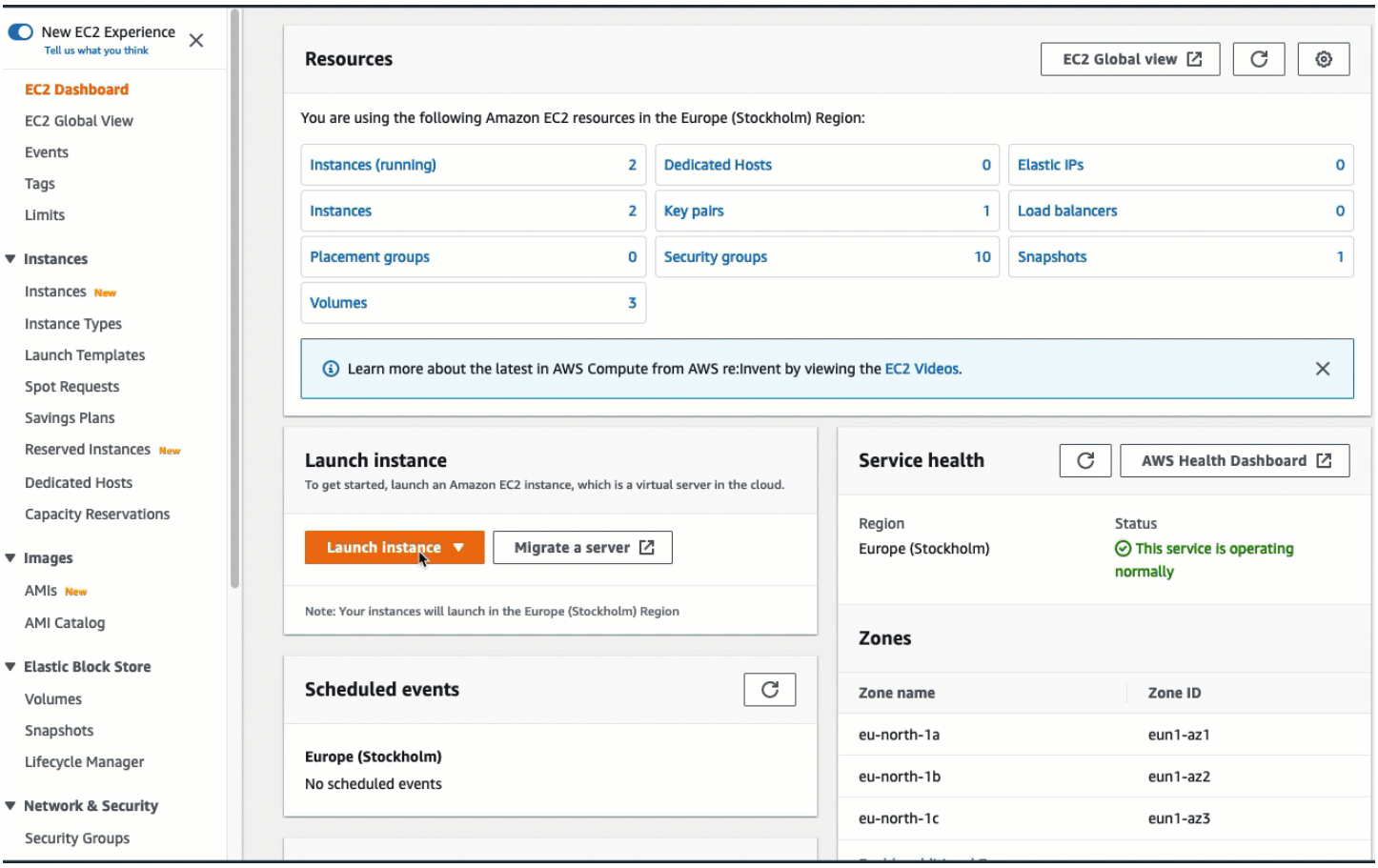
- **Allow HTTPs traffic from the internet**
- **Allow HTTP traffic from the internet**

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



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.

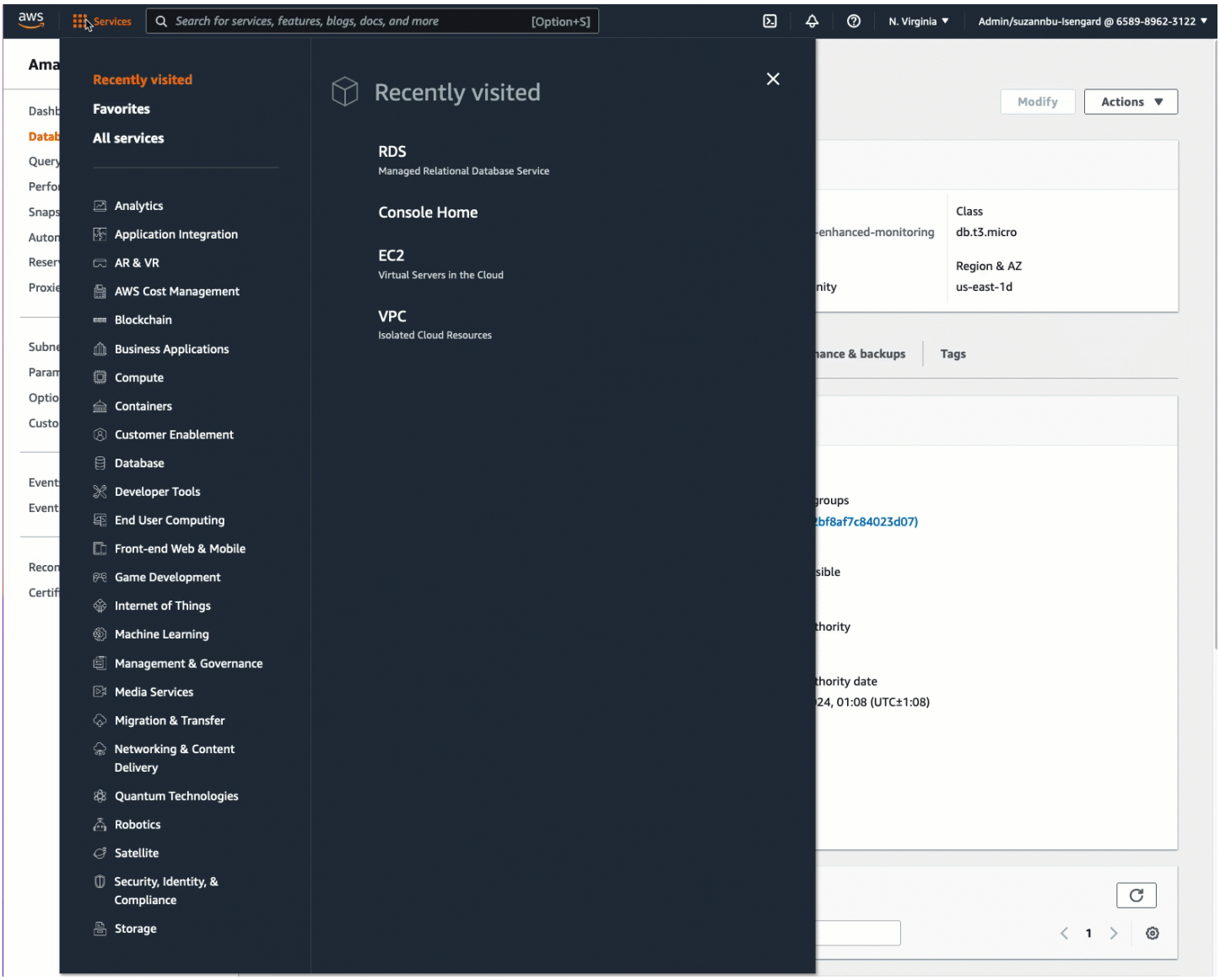
To create a MySQL DB instance

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 Windows instance

A Windows instance is a virtual server running Windows Server in the cloud.

After you have successfully launched and logged into your instance, you can make changes to it so that it's configured to meet the needs of a specific application. The following are some common tasks to help you get started.

Contents

- Configure launch settings for Amazon EC2 instances
- Paravirtual drivers for Windows instances
- AWS NVMe drivers for Windows instances
- Optimize CPU options
- Set the time for your Windows instance
- Set the password for a Windows instance
- Add Windows components using installation media
- Configure a secondary private IPv4 address for your Windows instance
- Run commands on your Windows instance at launch
- Instance metadata and user data
- Best practices and recommendations for SQL Server clustering in Amazon EC2
- Install WSL on your Windows instance

Configure launch settings for Amazon EC2 instances

Amazon EC2 launch agents perform tasks during instance startup and run if an instance is stopped and later started, or restarted. For information about a specific agent, see the detail pages in the following list.

- Configure a Windows instance using EC2Launch v2
- Configure a Windows instance using EC2Launch
- Configure a Windows instance using the EC2Config service (legacy)

Content

- Compare Amazon EC2 launch agents

- Configure DNS Suffix

Compare Amazon EC2 launch agents

The following table shows the major functional differences between EC2Config, EC2Launch v1, and EC2Launch v2.

Feature	EC2Config	EC2Launch v1	EC2Launch v2
Run as	Windows Service	PowerShell Scripts	Windows Service
Supports	Legacy OS only	Windows 2016 Windows 2019 (LTSC and SAC)	Windows 2016 Windows 2019 (LTSC and SAC) Windows 2022
Configuration file	XML	XML	YAML
Set Administrator username	No	No	Yes
User data size	16 KB	16 KB	60 KB (compressed)
Local user data baked on AMI	No	No	Yes, configurable
Task configuration in user data	No	No	Yes
Configurable wallpaper	No	No	Yes
Customize task run order	No	No	Yes
Configurable tasks	15	9	20 at launch

Feature	EC2Config	EC2Launch v1	EC2Launch v2
Supports Windows Event Viewer	Yes	No	Yes
Number of Event Viewer event types	2	0	30

> **Note**
>
> Microsoft no longer supports the Windows Server 2012 operating system. EC2Config documentation is provided for historical references only. We strongly recommend that you upgrade to the latest launch service for a supported operating system such as Windows Server 2022.

Configure DNS Suffix

With Amazon EC2 launch agents, you can configure a list of DNS suffixes that Windows instances use for domain name resolution. The launch agents override the standard Windows settings in the `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` registry key by adding the following values to the DNS suffix search list:

- The domain of the instance
- The suffixes that result from devolution of the instance domain
- NV domain
- The domains specified by each network interface cards

All launch agents support DNS suffix configuration. For more information, see your specific launch agent version:

- For information about the `setDnsSuffix` task and how to configure DNS suffixes in EC2Launch v2, see setDnsSuffix .
- For information about DNS suffix list setup and how to enable or disable devolution for EC2Launch v1, see Configure EC2Launch.

- For information about DNS suffix list setup and how to enable or disable devolution for EC2Config, see EC2Config settings files.

Domain name devolution

Domain name devolution is an Active Directory behavior that allows computers in a child domain to access resources in the parent domain without using a fully qualified domain name. By default, domain name devolution continues until there are only two nodes left in the domain name progression.

Launch agents perform devolution on the domain name if the instance is connected to a domain, and add the results to the DNS suffix search list that's maintained in the **`System\CurrentControlSet\Services\Tcpip\Parameters\SearchList`** registry key. The agents use the settings from the following registry keys, to determine the devolution behavior.

- **`System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution`**
  - When not set, disables devolution
  - When set to `1`, enables devolution (default)
  - When set to `0`, disables devolution
- **`System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`**
  - When not set, use level of 2 (default)
  - When set to 3 or greater, use value to set level

When you disable devolution or change your devolution settings to a higher level, the `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` registry key stil contains the suffixes that were added previously. They are not automatically removed. You can manually update the list, or you can clear the list and let your agent run through the process to set up the new list.

> **Note**
>
> To clear the DNS suffix list from the registry, you can run the following command.

```
PS C:\> Invoke-CimMethod -ClassName Win32_NetworkAdapterConfiguration -
MethodName "SetDNSSuffixSearchOrder" -Arguments @{ DNSDomainSuffixSearchOrder =
 $null } | Out-Null
```

Devolution examples

The following examples show domain name progression through the devolution process.

`corp.example.com`

- Progresses to `example.com`

`locale.region.corp.example.com`

1. Progresses to `region.corp.example.com`
2. Progresses to `corp.example.com`
3. Progresses to `example.com`

`locale.region.corp.example.com` with a setting of `DomainNameDevolutionLevel=3`

1. Progresses to `region.corp.example.com`
2. Progresses to `corp.example.com`. The progression stops here, due to the level setting.

Configure a Windows instance using EC2Launch v2

All supported instances of Amazon EC2 running Windows Server 2022 include the EC2Launch v2 launch agent (`EC2Launch.exe`) by default. We also provide Windows Server 2016 and 2019 AMIs with EC2Launch v2 installed as the default launch agent. These AMIs are provided in addition to the Windows Server 2016 and 2019 AMIs that include EC2Launch v1. You can search for Windows AMIs that include EC2Launch v2 by default by entering the following prefix in your search from the **AMIs** page in the Amazon EC2 console: `EC2LaunchV2-Windows_Server-*`.

EC2Launch v2 performs tasks during instance startup and runs if an instance is stopped and later started, or restarted. EC2Launch v2 can also perform tasks on demand. Some of these tasks are

automatically enabled, while others must be enabled manually. The EC2Launch v2 service supports all EC2Config and EC2Launch features.

This service uses a configuration file to control its operation. You can update the configuration file using either a graphical tool or by directly editing it as a single .yml file (`agent-config.yml`). The service binaries are located in the `%ProgramFiles%\Amazon\EC2Launch` directory.

EC2Launch v2 publishes Windows event logs to help you troubleshoot errors and set triggers. For more information, see Windows event logs.

Supported operating systems

- Windows Server 2022
- Windows Server 2019 (Long-Term Servicing Channel and Semi-Annual Channel)
- Windows Server 2016

EC2Launch v2 section contents

- EC2Launch v2 overview
- Install the latest version of EC2Launch v2
- Migrate to EC2Launch v2
- Stop, restart, delete, or uninstall EC2Launch v2
- Subscribe to EC2Launch v2 service notifications
- EC2Launch v2 settings
- Troubleshoot EC2Launch v2
- EC2Launch v2 version histories

EC2Launch v2 overview

EC2Launch v2 is a service that performs tasks during instance startup and runs if an instance is stopped and later started, or restarted.

Overview topics

- EC2Launch v2 concepts
- EC2Launch v2 tasks
- Telemetry

To compare launch agent version features, see Compare Amazon EC2 launch agents.

EC2Launch v2 concepts

The following concepts are useful to understand when considering EC2Launch v2.

Task

You can invoke a task to perform an action on an instance. You can configure tasks in the `agent-config.yml` file or through user data. For a list of available tasks for EC2Launch v2, see EC2Launch v2 tasks. For task configuration schema and details, see EC2Launch v2 task configuration.

Stage

A stage is a logical grouping of tasks that the EC2Launch v2 agent runs. Some tasks can run only in a specific stage. Others can run in multiple stages. When using `agent-config.yml`, you must specify a list of stages, and a list of tasks to run within each stage.

The service runs stages in the following order:

Stage 1: Boot

Stage 2: Network

Stage 3: PreReady

Windows is ready

After the PreReady stage completes, the service sends the `Windows is ready` message to the Amazon EC2 console.

Stage 4: PostReady

User data runs during the *PostReady* stage. Some script versions run before the `agent-config.yml` file *PostReady* stage, and some run after, as follows:

Before `agent-config.yml`

- YAML user data version 1.1
- XML user data

After `agent-config.yml`

- YAML user data version 1.0 (legacy version for backwards compatibility)

For example stages and tasks, see Example: `agent-config.yml`.

When you use user data, you must specify a list of tasks for the launch agent to run. The stage is implied. For example tasks, see Example: user data.

EC2Launch v2 runs the list of tasks in the order that you specify in `agent-config.yml` and in user data. Stages run sequentially. The next stage starts after the previous stage completes. Tasks also run sequentially.

Frequency

Task frequency determines when tasks should run, depending on the boot context. Most tasks have only one allowed frequency. You can specify a frequency for `executeScript` tasks.

You will see the following frequencies in the EC2Launch v2 task configuration.

- Once – The task runs once, when the AMI has booted for the first time (finished Sysprep).
- Always – The task runs every time that the launch agent runs. The launch agent runs when:
  - an instance starts or restarts
  - the EC2Launch service runs
  - `EC2Launch.exe run` is invoked

agent-config

`agent-config` is a file that is located in the configuration folder for EC2Launch v2. It includes configuration for the boot, network, PreReady, and PostReady stages. This file is used to specify the instance configuration for tasks that should run when the AMI is either booted for the first time or for subsequent times.

By default, the EC2Launch v2 installation installs an `agent-config` file that includes recommended configurations that are used in standard Amazon Windows AMIs. You can update the configuration file to alter the default boot experience for your AMI that EC2Launch v2 specifies.

User data

User data is data that is configurable when you launch an instance. You can update user data to dynamically change how custom AMIs or quickstart AMIs are configured. EC2Launch v2 supports 60 kB user data input length. User data includes only the UserData stage, and therefore runs after the `agent-config` file. You can enter user data when you launch an instance using the launch

instance wizard, or you can modify user data from the EC2 console. For more information about working with user data, see Run commands on your Windows instance at launch.

EC2Launch v2 tasks

EC2Launch v2 can perform the following tasks at each boot:

- Set up new and optionally customized wallpaper that renders information about the instance.
- Set the attributes for the administrator account that is created on the local machine.
- Add DNS suffixes to the list of search suffixes. Only suffixes that do not already exist are added to the list.
- Set drive letters for any additional volumes and extend them to use available space.
- Write files to the disk, either from the internet or from the configuration. If the content is in the configuration, it can be base64 decoded or encoded. If the content is from the internet, it can be unzipped.
- Run scripts either from the internet or from the configuration. If the script is from the configuration, it can be base64 decoded. If the script is from the internet, it can be unzipped.
- Run a program with given arguments.
- Set the computer name.
- Send instance information to the Amazon EC2 console.
- Send the RDP certificate thumbprint to the Amazon EC2 console.
- Dynamically extend the operating system partition to include any unpartitioned space.
- Run user data. For more information about specifying user data, see EC2Launch v2 task configuration.
- Set non-persistent static routes to reach the metadata service and AWS KMS servers.
- Set non-boot partitions to MBR or GPT.
- Start the Systems Manager service following Sysprep.
- Optimize ENA settings.
- Enable OpenSSH for later Windows versions.
- Enable Jumbo Frames.
- Set Sysprep to run with EC2Launch v2.
- Publish Windows event logs.

Telemetry

Telemetry is additional information that helps AWS to better understand your requirements, diagnose issues, and deliver features to improve your experience with AWS services.

EC2Launch v2 version `2.0.592` and later collect telemetry, such as usage metrics and errors. This data is collected from the Amazon EC2 instance on which EC2Launch v2 runs. This includes all Windows AMIs owned by AWS.

The following types of telemetry are collected by EC2Launch v2:

- **Usage information** — agent commands, install method, and scheduled run frequency.
- **Errors and diagnostic information** — agent installation error codes, run error codes, and error call stacks.

Examples of collected data:

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

Telemetry is enabled by default. You can disable telemetry collection at any time. If telemetry is enabled, EC2Launch v2 sends telemetry data without additional customer notifications.

Telemetry visibility

When telemetry is enabled, it appears in the Amazon EC2 console output as follows.

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

Disable telemetry on an instance

To disable telemetry for a single instance, you can either set a system environment variable, or use the MSI to modify the installation.

To disable telemetry by setting a system environment variable, run the following command as an administrator.

```
setx /M EC2LAUNCH_TELEMETRY 0
```

To disable telemetry using the MSI, run the following command after you download the MSI.

```
msiexec /i ".\AmazonEC2Launch.msi" Remove="Telemetry" /q
```

Install the latest version of EC2Launch v2

You can use one of the following methods to install the EC2Launch v2 agent on your EC2 instance:

- Download the agent from Amazon S3 and install with Windows PowerShell. For download URLs, see EC2Launch v2 downloads on Amazon S3.
- Install with SSM Distributor.
- Install from an EC2 Image Builder component.
- Launch your instance from an AMI that has EC2Launch v2 pre-installed.

> **Warning**
>
> AmazonEC2Launch.msi uninstalls previous versions of the EC2 launch services, such as EC2Launch (v1) and EC2Config.

For install steps, select the tab that matches your preferred method.

Windows PowerShell

To install the latest version of EC2Launch v2 agent with Windows PowerShell, follow these steps.

1. Create your local directory.

   ```
   New-Item -Path "$env:USERPROFILE\Desktop\EC2Launchv2" -ItemType Directory
   ```

2. Set the URL for your download location. Run the following command with the Amazon S3 URL you'll use. For download URLs, see EC2Launch v2 downloads on Amazon S3

   ```
   $Url = "Amazon S3 URL/AmazonEC2Launch.msi"
   ```

3. Use the following compound command to download the agent and run the install

```
$DownloadFile = "$env:USERPROFILE\Desktop\EC2Launchv2\" + $(Split-Path -Path
 $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadFile
msiexec /i "$DownloadFile"
```

4. To verify the install, check that the msi file exists in the EC2Launch v2 directory on your instance (`C:\ProgramData\Amazon\EC2Launch`).

AWS Systems Manager Distributor

You can install the `AWSEC2Launch-Agent` package from AWS Systems Manager Distributor. For instructions on how to install a package from Systems Manager Distributor, see Install or update packages in the *AWS Systems Manager User Guide*.

EC2 Image Builder component

You can install the `ec2launch-v2-windows` component when you build a custom image with EC2 Image Builder. For instructions on how to build a custom image with EC2 Image Builder, see Create an image pipeline using the EC2 Image Builder console wizard in the *EC2 Image Builder User Guide*.

AMI

EC2Launch v2 is preinstalled by default on the followingWindows Server 2022 and UEFI AMIs:

- Windows_Server-2022-English-Full-Base
- Windows_Server-2022-English-Core-Base
- Windows Server 2022 AMIs with all other languages
- Windows Server 2022 AMIs with SQL installed
- Windows_Server-2022-English-Core-EKS_Optimized

EC2Launch v2 is also preinstalled on the following Windows Server AMIs. You can find these AMIs from the Amazon EC2 console, or by using the following search prefix: `EC2LaunchV2-` in the AWS CLI.

- EC2LaunchV2-Windows_Server-2019-English-Core-Base
- EC2LaunchV2-Windows_Server-2019-English-Full-Base

- EC2LaunchV2-Windows_Server-2016-English-Core-Base
- EC2LaunchV2-Windows_Server-2016-English-Full-Base
- EC2LaunchV2-Windows_Server-2012_R2_RTM-English-Full-Base
- EC2LaunchV2-Windows_Server-2012_RTM-English-Full-Base

EC2Launch v2 downloads on Amazon S3

To install the latest version of EC2Launch v2, download the installer from one of the following locations:

> **Note**
>
> The 32-bit installation link will be deprecated. We recommend that you use the 64-bit installation link to install EC2Launch v2. If you require a 32-bit launch agent, use EC2Config.

- **64Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2/windows/amd64/latest/AmazonEC2Launch.msi
- **32Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2/windows/386/latest/AmazonEC2Launch.msi

Configure install options

When you install or upgrade EC2Launch v2, you can configure installation options with the EC2Launch v2 install dialog or with the **msiexec** command in a command line shell.

The first time the EC2Launch v2 installer runs on an instance, it initializes launch agent settings on your instance as follows:

- It creates the local path and writes the launch agent file to it. This is sometimes referred to as a *clean install*.
- It creates the `EC2LAUNCH_TELEMETRY` environment variable if it doesn't already exist, and sets it based on your configuration.

For configuration details, select the tab that matches the configuration method that you'll use.

Amazon EC2Launch Setup dialog

When you install or upgrade EC2Launch v2, you can configure the following installation options through the EC2Launch v2 install dialog.

Basic Install options

Send Telemetry

When you include this feature in the setup dialog, the installer sets the `EC2LAUNCH_TELEMETRY` environment variable to a value of `1`. If you disable **Send Telemetry**, the installer sets the environment variable to a value of `0`.

When the EC2Launch v2 agent runs, it reads the `EC2LAUNCH_TELEMETRY` environment variable to determine whether to upload telemetry data. If the value equals 1, it uploads the data. Otherwise, it doesn't upload.

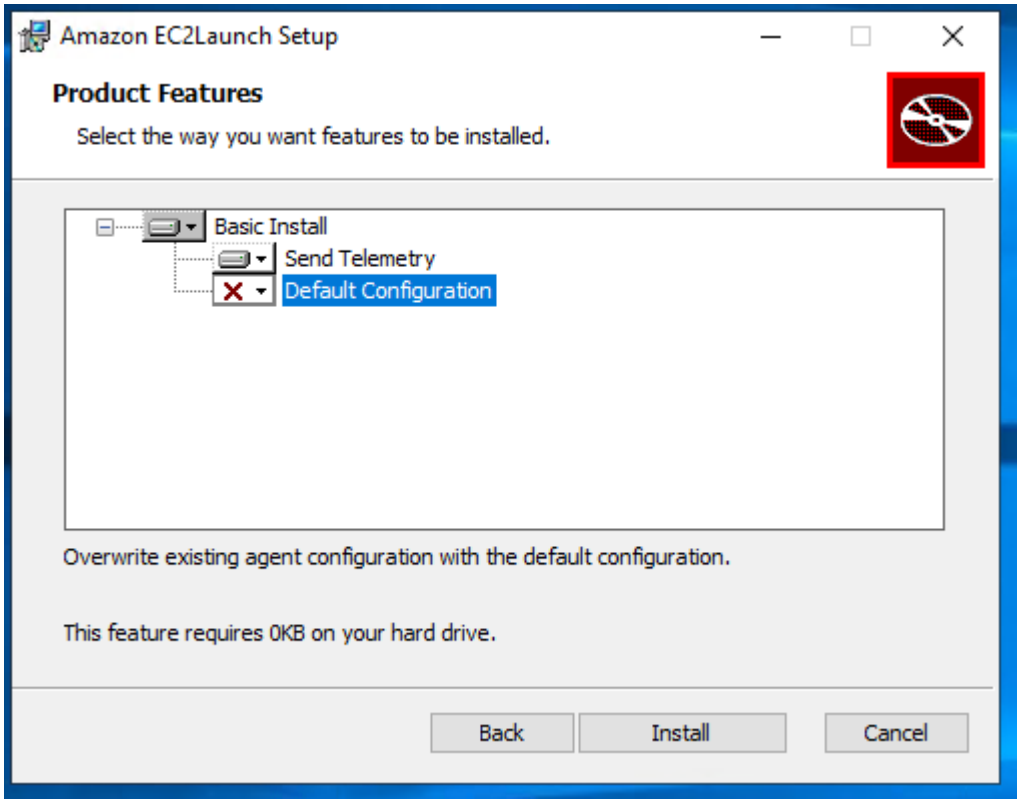
Default configuration

The default configuration for EC2Launch v2 is to overwrite the local launch agent if it exists already. The first time you run an install on an instance, the default configuration performs a clean install. If you disable the default configuration on the initial install, the installation fails.

If you run the install again on the instance, you can disable the default configuration to perform an upgrade that doesn't replace the `%ProgramData%/Amazon/EC2Launch/config/agent-config.yml` file.

Example: Upgrade EC2Launch v2 with telemetry

The following example shows the EC2Launch v2 setup dialog configured to upgrade the current installation and enable telemetry. This configuration performs an install without replacing the agent configuration file, and sets the `EC2LAUNCH_TELEMETRY` environment variable to a value of 1.



Command line

When you install or upgrade EC2Launch v2, you can configure the following installation options with the **msiexec** command in a command line shell.

`ADDLOCAL` parameter values

Basic (required)

Install the launch agent. If this value is not present in the `ADDLOCAL` parameter, the installation ends.

Clean

When you include the `Clean` value in the `ADDLOCAL` parameter, the installer writes the agent configuration file to the following location: `%ProgramData%/Amazon/EC2Launch/config/agent-config.yml`. If the agent configuration file already exists, it overwrites the file.

When you leave the `Clean` value out of the `ADDLOCAL` parameter, the installer performs an upgrade that doesn't replace the agent configuration file.

Telemetry

When you include the `Telemetry` value in the `ADDLOCAL` parameter, the installer sets the `EC2LAUNCH_TELEMETRY` environment variable to a value of `1`.

When you leave the `Telemetry` value out of the `ADDLOCAL` parameter, the installer sets the environment variable to a value of `0`.

When the EC2Launch v2 agent runs, it reads the `EC2LAUNCH_TELEMETRY` environment variable to determine whether to upload telemetry data. If the value equals `1`, it uploads the data. Otherwise, it doesn't upload.

Example: install EC2Launch v2 with telemetry

```
& msiexec /i "C:\Users\Administrator\Desktop\EC2Launchv2\AmazonEC2Launch.msi"
 ADDLOCAL="Basic,Clean,Telemetry" /q
```

Verify the EC2Launch v2 version

Use one of the following procedures to verify the version of EC2Launch v2 that is installed on your instances.

Windows PowerShell

Verify the installed version of EC2Launch v2 with Windows PowerShell, as follows.

1. Launch an instance from your AMI and connect to it.
2. Run the following command in PowerShell to verify the installed version of EC2Launch v2:

   ```
   & "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" version
   ```

Windows Control Panel

Verify the installed version of EC2Launch v2 in the Windows Control Panel, as follows.

1. Launch an instance from your AMI and connect to it.
2. Open the Windows Control Panel and choose **Programs and Features**.

3. Look for `Amazon EC2Launch` in the list of installed programs. Its version number appears in the **Version** column.

For information about the EC2Launch v2 versions included in the Windows AMIs, see AWS Windows AMIs.

For the latest version of EC2Launch v2, see EC2Launch v2 version history.

For the latest version of the EC2Launch v2 migration tool, see EC2Launch v2 migration tool version history.

You can receive notifications when new versions of the EC2Launch v2 service are released. For more information, see Subscribe to EC2Launch v2 service notifications.

Migrate to EC2Launch v2

The EC2Launch migration tool upgrades the installed launch agent (EC2Config and EC2Launch v1) by uninstalling it and installing EC2Launch v2. Applicable configurations from previous launch services are automatically migrated to the new service. The migration tool does not detect any scheduled tasks linked to EC2Launch v1 scripts; therefore, it does not automatically set up those tasks in EC2Launch v2. To configure these tasks, edit the `agent-config.yml` file, or use the EC2Launch v2 settings dialog box. For example, if an instance has a scheduled task that runs `InitializeDisks.ps1`, then after you run the migration tool, you must specify the volumes you want to initialize in the EC2Launch v2 settings dialog box. See Step 6 of the procedure to Change settings using the EC2Launch v2 settings dialog box.

You can download the migration tool or install with an SSM RunCommand document.

You can download the tool from the following locations:

> **Note**
>
> The 32-bit migration tool link will be deprecated. We recommend that you use the 64-bit link to migrate to EC2Launch v2. If you require a 32-bit launch agent, use EC2Config.

- **64Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2-utils/MigrationTool/windows/amd64/latest/EC2LaunchMigrationTool.zip
- **32Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2-utils/MigrationTool/windows/386/latest/EC2LaunchMigrationTool.zip

> **Note**
>
> You must run the EC2Launch v2 migration tool as an Administrator. EC2Launch v2 is installed as a service after you run the migration tool. It does not run immediately. By default, it runs during instance startup and runs if an instance is stopped and later started, or restarted.

Use the `AWSEC2Launch-RunMigration` SSM document to migrate to the latest EC2Launch v2 version with SSM Run Command. The document does not require any parameters. For more information about using SSM Run Command, see AWS Systems Manager Run Command.

The migration tool applies the following configurations from EC2Config to EC2Launch v2.

- If `Ec2DynamicBootVolumeSize` is set to `false`, removes EC2Launch v2 `boot` stage
- If `Ec2SetPassword` is set to `Enabled`, sets EC2Launch v2 password type to `random`
- If `Ec2SetPassword` is set to `Disabled`, sets EC2Launch v2 password type to `donothing`
- If `SetDnsSuffixList` is set to `false`, removes EC2Launch v2 `setDnsSuffix` task
- If `EC2SetComputerName` is set to true, adds EC2Launch v2 `setHostName` task to `yaml` configuration

The migration tool applies the following configurations from EC2Launch v1 to EC2Launch v2.

- If `ExtendBootVolumeSize` is set to `false`, removes EC2Launch v2 `boot` stage
- If `AdminPasswordType` is set to `Random`, sets EC2Launch v2 password type to `random`
- If `AdminPasswordType` is set to `Specify`, sets EC2Launch v2password type to `static` and password data to the password specified in `AdminPassword`
- If `SetWallpaper` is set to `false`, removes EC2Launch v2 `setWallpaper` task
- If `AddDnsSuffixList` is set to `false`, removes EC2Launch v2 `setDnsSuffix` task
- If `SetComputerName` is set to `true`, adds EC2Launch v2 `setHostName` task

Stop, restart, delete, or uninstall EC2Launch v2

You can manage the EC2Launch v2 service just as you would any other Windows service.

EC2Launch v2 runs once on boot and runs all of the configured tasks. After executing tasks, the service enters a stopped state. When you restart the service, the service will run all of the configured tasks again and return to a stopped state.

To apply updated settings to your instance, you can stop and restart the service. If you are manually installing EC2Launch v2, you must first stop the service first.

To stop the EC2Launch v2 service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, choose **Administrative Tools**, and then open **Services**.
3. In the list of services, right-click **Amazon EC2Launch**, and select **Stop**.

To restart the EC2Launch v2 service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, choose **Administrative Tools**, and then open **Services**.
3. In the list of services, right-click **Amazon EC2Launch**, and select **Restart**.

If you don't need to update the configuration settings, create your own AMI, or use AWS Systems Manager, you can delete and uninstall the service. Deleting a service removes its registry subkey. Uninstalling a service removes the files, the registry subkeys, and any shortcuts to the service.

To delete the EC2Launch v2 service

1. Start a command prompt window.
2. Run the following command:

```
sc delete EC2Launch
```

To uninstall EC2Launch v2

1. Launch and connect to your Windows instance.
2. On the **Start** menu, choose **Control Panel**.
3. Open **Programs** and then **Programs and Features**.

4. In the list of programs, choose **Amazon EC2Launch**. To confirm that you're choosing v2, check the **Version** column.
5. Choose **Uninstall**.

Subscribe to EC2Launch v2 service notifications

Amazon SNS can notify you when new versions of the EC2Launch v2 service are released. Use the following procedure to subscribe to these notifications.

Subscribe to EC2Launch v2 notifications

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select this Region because the SNS notifications that you are subscribing to were created in this Region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the Create subscription dialog box, do the following:
   a. For **Topic ARN**, use the following Amazon Resource Name (ARN): **arn:aws:sns:us-east-1:309726204594:amazon-ec2launch-v2**.
   b. For **Protocol**, choose **Email**.
   c. For **Endpoint**, enter an email address that you can use to receive the notifications.
   d. Choose **Create subscription**.
6. You'll receive an email asking you to confirm your subscription. Open the email and follow the directions to complete your subscription.

Whenever a new version of the EC2Launch v2 service is released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

1. Open the Amazon SNS console.
2. In the navigation pane, choose **Subscriptions**.

3. Select the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

EC2Launch v2 settings

This section contains information about how to configure settings for EC2Launch v2.

Topics include:

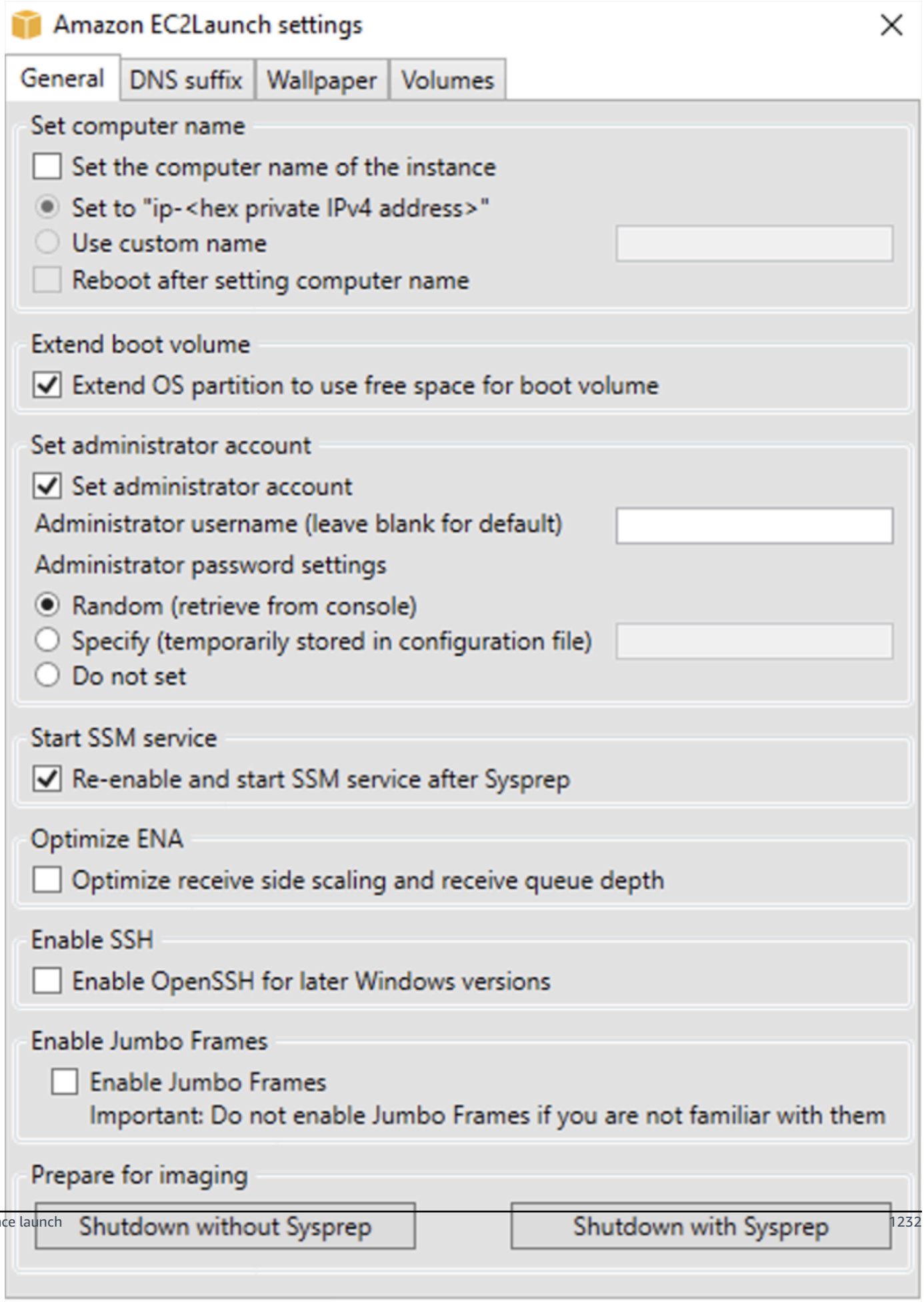
- Change settings using the EC2Launch v2 settings dialog box
- EC2Launch v2 directory structure
- Configure EC2Launch v2 using the CLI
- EC2Launch v2 task configuration
- EC2Launch v2 exit codes and reboots
- EC2Launch v2 and Sysprep

Change settings using the EC2Launch v2 settings dialog box

The following procedure describes how to use the EC2Launch v2 settings dialog box to enable or disable settings.

> **Note**
>
> If you improperly configure custom tasks in the agent-config.yml file, and you attempt to open the Amazon EC2Launch settings dialog box, you will receive an error. For example schema, see Example: `agent-config.yml`.

1. Launch and connect to your Windows instance.
2. From the Start menu, choose **All Programs**, and then navigate to **EC2Launch settings**.


Amazon EC2Launch settings
General
DNS suffix
Wallpaper
Volumes
Set computer name
Set the computer name of the instance
Set to "ip-<hex private IPv4 address>"
Use custom name
Reboot after setting computer name
Extend boot volume
Extend OS partition to use free space for boot volume
Set administrator account
Set administrator account
Administrator username (leave blank for default)
Administrator password settings
Random (retrieve from console)
Specify (temporarily stored in configuration file)
Do not set
Start SSM service
Re-enable and start SSM service after Sysprep
Optimize ENA
Optimize receive side scaling and receive queue depth
Enable SSH
Enable OpenSSH for later Windows versions
Enable Jumbo Frames
Enable Jumbo Frames
Important: Do not enable Jumbo Frames if you are not familiar with them
Prepare for imaging
Shutdown without Sysprep
Shutdown with Sysprep

3. On the **General** tab of the **EC2Launch settings** dialog box, you can enable or disable the following settings.

   a. **Set Computer Name**

      If this setting is enabled (it is disabled by default), the current host name is compared to the desired host name at each boot. If the host names do not match, the host name is reset, and the system then optionally reboots to pick up the new host name. If a custom host name is not specified, it is generated using the hexadecimal-formatted private IPv4 address, for example, `ip-AC1F4E6`. To prevent your existing host name from being modified, do not enable this setting.

   b. **Extend Boot Volume**

      This setting dynamically extends `Disk 0/Volume 0` to include any unpartitioned space. This can be useful when the instance is booted from a root device volume that has a custom size.

   c. **Set Administrator Account**

      When enabled, you can set the username and password attributes for the administrator account that is created on your local machine. If this feature is not enabled, an administrator account is not created on the system following Sysprep. Provide a password in `adminPassword` only if `adminPasswordtype` is `Specify`.

      The password types are defined as follows:

      i. `Random`

         EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.

      ii. `Specify`

         EC2Launch uses the password that you specify in `adminPassword`. If the password does not meet the system requirements, EC2Launch generates a random password instead. The password is stored in `agent-config.yml` as clear text and is deleted after Sysprep sets the administrator password. EC2Launch encrypts the password using the user's key.

      iii. `Do not set`

EC2Launch uses the password that you specify in the unattend.xml file. If you don't specify a password in unattend.xml, the administrator account is disabled.

d. **Start SSM Service**

When selected, the Systems Manager service is enabled to start following Sysprep. EC2Launch v2 performs all of the tasks described earlier, and the SSM Agent processes requests for Systems Manager capabilities, such as Run Command and State Manager.

You can use Run Command to upgrade your existing instances to use the latest version of the EC2Launch v2 service and SSM Agent. For more information, see Update SSM Agent by using Run Command in the *AWS Systems Manager User Guide*.

e. **Optimize ENA**

When selected, ENA settings are configured to ensure that ENA Receive Side Scaling and Receive Queue Depth settings are optimized for AWS. For more information, see Configure RSS CPU affinity.

f. **Enable SSH**

This setting enables OpenSSH for later Windows versions to allow for remote system administration.

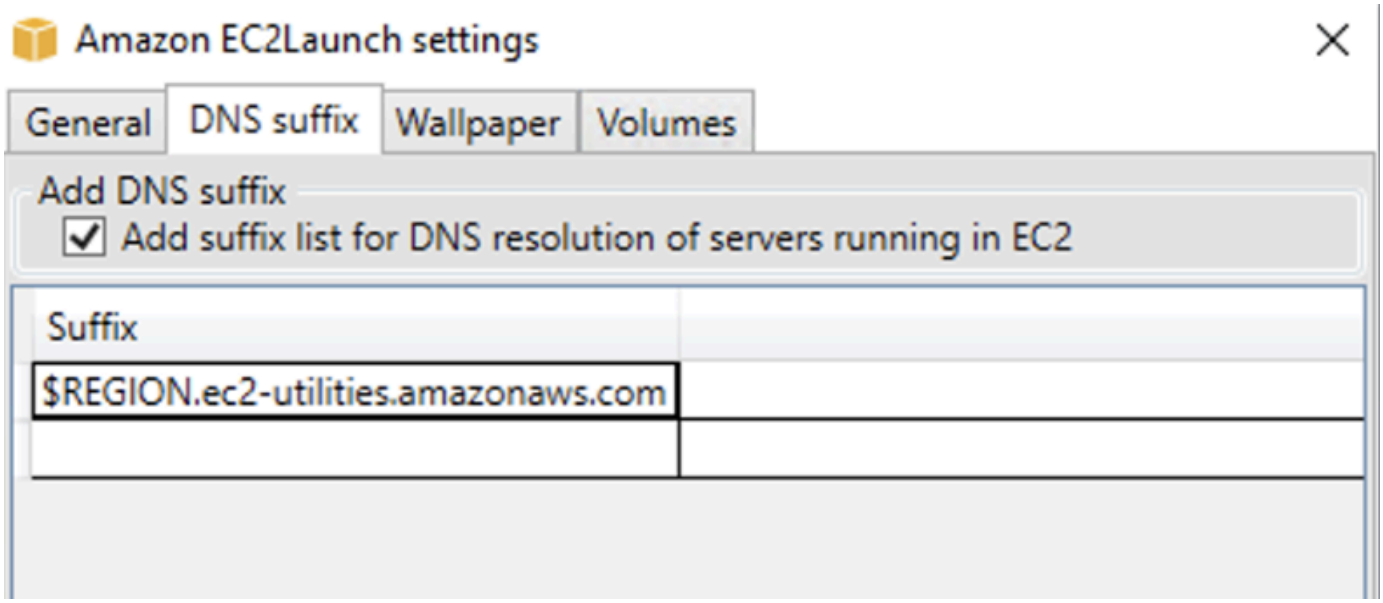
g. **Enable Jumbo Frames**

Select to enable Jumbo Frames. Jumbo Frames can have unintended effects on your network communications, so ensure you understand how Jumbo Frames will impact your system before enabling. For more information about Jumbo Frames, see Jumbo frames (9001 MTU).

h. **Prepare for Imaging**

Select whether you want your EC2 instance to shut down with or without Sysprep. When you want to run Sysprep with EC2Launch v2, choose **Shutdown with Sysprep**.

4. On the **DNS Suffix** tab, you can select whether you want to add a DNS suffix list for DNS resolution of servers running in EC2, without providing the fully qualified domain name. DNS suffixes can contain the variables `$REGION` and `$AZ`. Only suffixes that do not already exist will be added to the list.

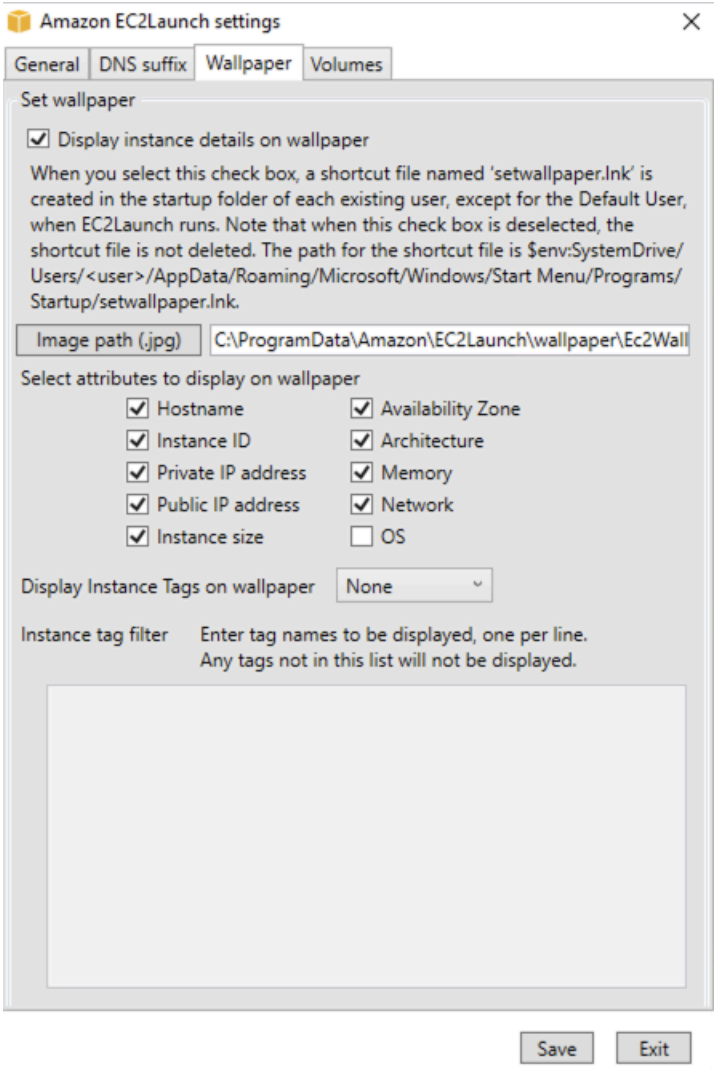


5. On the **Wallpaper** tab, you can configure your instance wallpaper with a background image, and specify instance details for the wallpaper to display. Amazon EC2 generates the details each time you log in.

   You can configure your wallpaper with the following controls.

   - **Display instance details on wallpaper** – This checkbox activates or deactivates instance detail display on the wallpaper.
   - **Image path (.jpg)** – Specify the path to the image to use as the wallpaper background.
   - **Select attributes to display on wallpaper** – Select the check boxes for the instance details that you want to appear on the wallpaper. Clear the check boxes for previously selected instance details that you want to remove from the wallpaper.
   - **Display Instance Tags on wallpaper** – Select one of the following settings to display instance tags on the wallpaper:
     - **None** – Don't display any instance tags on the wallpaper.
     - **Show all** – Display all instance tags on the wallpaper.
     - **Show filtered** – Display specified instance tags on the wallpaper. When you select this setting, you can add instance tags that you want to display on your wallpaper in the **Instance tag filter** box.

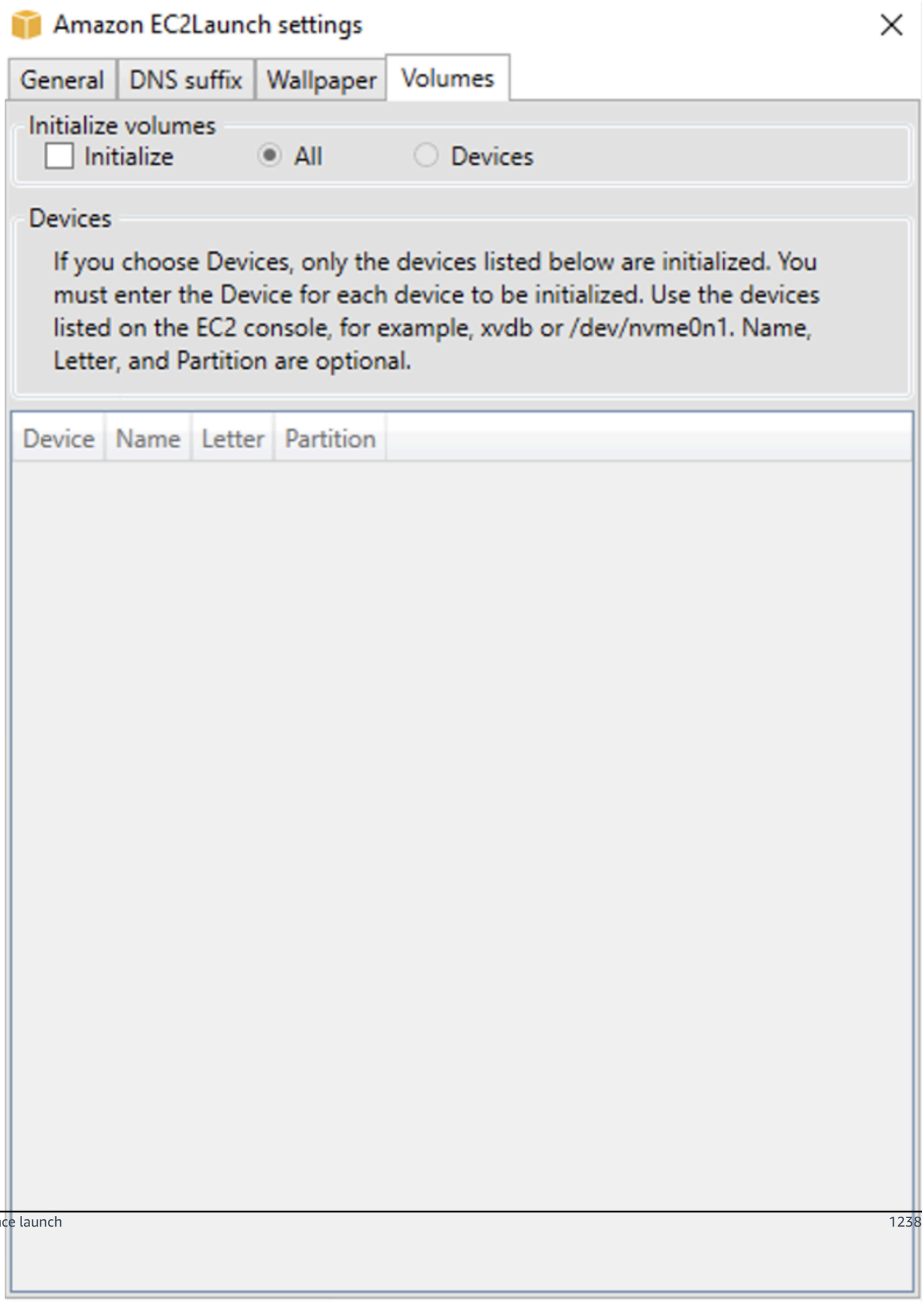
> **Note**
>
> You must enable tags in metadata to show tags on the wallpaper. For more information about instance tags and metadata, see Work with instance tags in instance metadata.



6. On the **Volumes** tab, select whether you want to initialize the volumes that are attached to the instance. Enabling sets drive letters for any additional volumes and extends them to use available space. If you select **All**, all of the storage volumes are initialized. If you select **Devices**, only devices that are specified in the list are initialized. You must enter the device for each device to be initialized. Use the devices listed on the EC2 console, for example, `xvdb` or `/dev/`

`nvme0n1`. The dropdown list displays the storage volumes that are attached to the instance. To enter a device that is not attached to the instance, enter it in the text field.

Name, **Letter**, and **Partition** are optional fields. If no value is specified for **Partition**, storage volumes larger than 2 TB are initialized with the GPT partition type, and those smaller than 2 TB are initialized with the MBR partition type. If devices are configured, and a non-NTFS device either contains a partition table, or the first 4 KB of the disk contain data, then the disk is skipped and the action logged.


Amazon EC2Launch settings
General
DNS suffix
Wallpaper
Volumes
Initialize volumes
Initialize
All
Devices
Devices
If you choose Devices, only the devices listed below are initialized. You must enter the Device for each device to be initialized. Use the devices listed on the EC2 console, for example, xvdb or /dev/nvme0n1. Name, Letter, and Partition are optional.
Device
Name
Letter
Partition

The following is an example configuration YAML file created from the settings entered in the EC2Launch dialog.

```
version: 1.0
config:
  - stage: boot
 tasks:
      - task: extendRootPartition
  - stage: preReady
    tasks:
      - task: activateWindows
        inputs:
          activation:
            type: amazon
      - task: setDnsSuffix
        inputs:
          suffixes:
            - $REGION.ec2-utilities.amazonaws.com
      - task: setAdminAccount
        inputs:
          password:
            type: random
      - task: setWallpaper
        inputs:
          path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
          attributes:
            - hostName
            - instanceId
            - privateIpAddress
            - publicIpAddress
            - instanceSize
            - availabilityZone
            - architecture
            - memory
            - network
  - stage: postReady
    tasks:
      - task: startSsm
```

EC2Launch v2 directory structure

EC2Launch v2 should be installed in the following directories:

- Service binaries: `%ProgramFiles%\Amazon\EC2Launch`
- Service data (settings, log files, and state files): `%ProgramData%\Amazon\EC2Launch`

> **Note**
>
> By default, Windows hides files and folders under `C:\ProgramData`. To view EC2Launch v2 directories and files, you must either enter the path in Windows Explorer or change the folder properties to show hidden files and folders.

The `%ProgramFiles%\Amazon\EC2Launch` directory contains binaries and supporting libraries. It includes the following subdirectories:

- `settings`
  - `EC2LaunchSettingsUI.exe` — user interface for modifying the `agent-config.yml` file
  - `YamlDotNet.dll` — DLL for supporting some operations in the user interface
- `tools`
  - `ebsnvme-id.exe` — tool for examining the metadata of the EBS volumes on the instance
  - `AWSAcpiSpcrReader.exe` — tool for determining the correct COM port to use
  - `EC2LaunchEventMessage.dll` — DLL for supporting the Windows event logging for EC2Launch.
- `service`
  - `EC2LaunchService.exe` — Windows service executable that is launched when the launch agent runs as a service.
- `EC2Launch.exe` — main EC2Launch executable
- `EC2LaunchAgentAttribution.txt` — attribution for code used within EC2 Launch

The `%ProgramData%\Amazon\EC2Launch` directory contains the following subdirectories. All of the data produced by the service, including logs, configuration, and state, is stored in this directory.

- `config` — Configuration

  The service configuration file is stored in this directory as `agent-config.yml`. This file can be updated to modify, add, or remove default tasks run by the service. Permission to create files in this directory is restricted to the administrator account to prevent privilege escalation.

- `log` — Instance logs

  Logs for the service (`agent.log`), console (`console.log`), performance (`bench.log`), and errors (`error.log`) are stored in this directory. Log files are appended to on subsequent executions of the service.

- `state` — Service state data

  The state that the service uses to determine which tasks should run is stored here. There is a `.run-once` file that indicates whether the service has already run after Sysprep (so tasks with a frequency of once will be skipped on the next run). This subdirectory includes a `state.json` and `previous-state.json` to track the status of each task.

- `sysprep` — Sysprep

  This directory contains files that are used to determine which operations to perform by Sysprep when it creates a customized Windows AMI that can be reused.

Configure EC2Launch v2 using the CLI

You can use the Command Line Interface (CLI) to configure your EC2Launch settings and manage the service. The following section contains descriptions and usage information for the CLI commands that you can use to manage EC2Launch v2.

Commands

- collect-logs
- get-agent-config
- list-volumes
- reset
- run
- status
- sysprep
- validate
- version
- wallpaper

collect-logs

Collects log files for EC2Launch, zips the files, and places them in a specified directory.

Example

```
ec2launch collect-logs -o C:\Mylogs.zip
```

Usage

`ec2launch collect-logs [flags]`

Flags

`-h, --help`

help for `collect-logs`

`-o, --output string`

path to zipped output log files

get-agent-config

Prints `agent-config.yml` in the format specified (JSON or YAML). If no format is specified, `agent-config.yml` is printed in the format previously specified.

Example

```
ec2launch get-agent-config -f json
```

Example 2

The following PowerShell commands show how to edit and save the `agent-config` file in JSON format.

```
$config = ec2launch get-agent-config --format json | ConvertFrom-Json
$jumboFrame =@"
{
   "task": "enableJumboFrames"
}
"@
```

```
$config.config | %{if($_.stage -eq 'postReady'){$_.tasks += (ConvertFrom-Json -
InputObject $jumboFrame)}}
$config | ConvertTo-Json -Depth 6 | Out-File -encoding UTF8 $env:ProgramData/Amazon/
EC2Launch/config/agent-config.yml
```

Usage

`ec2launch get-agent-config [flags]`

Flags

`-h`, `--help`

help for `get-agent-config`

`-f`, `--format string`

output format of `agent-config` file: `json`, `yaml`

list-volumes

Lists all of the storage volumes attached to the instance, including ephemeral and EBS volumes.

Example

```
ec2launch list-volumes
```

Usage

`ec2launch list-volumes`

Flags

`-h`, `--help`

help for `list-volumes`

reset

The main goal of this task is to reset the agent for the next time that it runs. To do that, the **reset** command deletes all of the agent state data for EC2Launch v2 from the local `EC2Launch` directory (see EC2Launch v2 directory structure). Reset optionally deletes the service and Sysprep logs.

Script behavior depends on what mode the agent runs the scripts in – inline, or detached.

Inline (default)

The EC2Launch v2 agent runs scripts one at a time (`detach: false`). This is the default setting.

> **Note**
>
> When your inline script issues a **reset** or **sysprep** command, it runs immediately and resets the agent. The current task finishes, then the agent shuts down without running any further tasks.
> For example, if the task that issues the command would have been followed by a `startSsm` task (included by default after user data runs), the task doesn't run and the Systems Manager service never starts.

Detached

The EC2Launch v2 agent runs scripts concurrently with other tasks (`detach: true`).

> **Note**
>
> When your detached script issues a **reset** or **sysprep** command, those commands wait for the agent to finish before they run. Tasks after the executeScript will still run.

Example

```
ec2launch reset -c
```

Usage

`ec2launch reset [flags]`

Flags

`-c, --clean`

cleans instance logs before `reset`

`-h, --help`

help for `reset`

run

Runs EC2Launch v2.

Example

```
ec2launch run
```

Usage

`ec2launch run [flags]`

Flags

`-h`, `--help`

help for `run`

status

Gets the status of the EC2Launch v2 agent. Optionally blocks the process until the agent is finished. The process exit code determines the agent state:

- `0` –the agent ran and was successful.
- 1 – the agent ran and failed.
- 2 – the agent is still running.
- 3 – the agent is in an unknown state. The agent state is not running or stopped.
- 4 – an error occurred when attempting to retrieve the agent state.
- 5 – the agent is not running and the status of the last known run is unknown. This could mean one of the following:
  - both the `state.json` and `previous-state.json` are deleted.
  - the `previous-state.json` is corrupted.

  This is the agent state after running the `reset` command.

Example:

```
ec2launch status -b
```

Usage

`ec2launch status [flags]`

Flags

`-b,--block`

blocks the process until the agent finishes running

`-h,--help`

help for `status`

sysprep

The main goal of this task is to reset the agent for the next time that it runs. To do that, the **sysprep** command resets the agent state, updates the `unattend.xml` file, disables RDP, and runs Sysprep.

Script behavior depends on what mode the agent runs the scripts in – inline, or detached.

Inline (default)

The EC2Launch v2 agent runs scripts one at a time (`detach: false`). This is the default setting.

> **Note**
>
> When your inline script issues a **reset** or **sysprep** command, it runs immediately and resets the agent. The current task finishes, then the agent shuts down without running any further tasks.
> For example, if the task that issues the command would have been followed by a `startSsm` task (included by default after user data runs), the task doesn't run and the Systems Manager service never starts.

Detached

The EC2Launch v2 agent runs scripts concurrently with other tasks (`detach: true`).

> **Note**
>
> When your detached script issues a **reset** or **sysprep** command, those commands wait for the agent to finish before they run. Tasks after the executeScript will still run.

Example:

```
ec2launch sysprep
```

Usage

`ec2launch sysprep [flags]`

Flags

`-c,--clean`

cleans instance logs before `sysprep`

`-h,--help`

help for Sysprep

`-s,--shutdown`

shuts down the instance after `sysprep`

validate

Validates the `agent-config` file `C:\ProgramData\Amazon\EC2Launch\config\agent-config.yml`.

Example

```
ec2launch validate
```

Usage

`ec2launch validate [flags]`

Flags

`-h ,--help`

help for `validate`

version

Gets the executable version.

Example

```
ec2launch version
```

Usage

`ec2launch version [flags]`

Flags

`-h, --help`

help for `version`

wallpaper

Sets new wallpaper to the wallpaper path that is provided (.jpg file), and displays the selected instance details.

Syntax

```
ec2launch wallpaper ^
--path="C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg" ^
--all-tags ^
--
attributes=hostName,instanceId,privateIpAddress,publicIpAddress,instanceSize,availabilityZone,a
```

Inputs

Parameters

--allowed-tags [*tag-name-1*, *tag-name-n*]

(Optional) Base64 encoded JSON array of instance tag names to display on the wallpaper. You can use this tag or the `--all-tags`, but not both.

--attributes *attribute-string-1*, *attribute-string-n*

(Optional) A comma-separated list of `wallpaper` attribute strings to apply settings to the wallpaper.

[--path | -p] *path-string*

(Required) Specifies the `wallpaper` background image file path.

Flags

--all-tags

(Optional) Displays all of the instance tags on the wallpaper. You can use this tag or the `--allowed-tags`, but not both.

[--help | -h]

Displays help for the **wallpaper** command.

EC2Launch v2 task configuration

This section includes the configuration schema, tasks, details, and examples for `agent-config.yml` and user data.

Tasks and examples

- Schema: agent-config.yml
- Schema: user data
- Task definitions

Schema: `agent-config.yml`

The structure of the `agent-config.yml` file is shown below. Note that a task cannot be repeated in the same stage. For task properties, see the task descriptions that follow.

Document structure: agent-config.yml

JSON

```
{
 "version": "1.0",
```

```
 "config": [
  {
   "stage": "string",
   "tasks": [
    {
     "task": "string",
     "inputs": {
      ...
     }
    },
    ...
   ]
  },
  ...
 ]
}
```

YAML

```
version: 1.0
config:
- stage: string
  tasks:
  - task: string
 inputs:
   ...
  ...
...
```

Example: `agent-config.yml`

The following example shows settings for the `agent-config.yml` configuration file.

```
version: 1.0
config:
- stage: boot
  tasks:
  - task: extendRootPartition
- stage: preReady
  tasks:
  - task: activateWindows
    inputs:
      activation:
```

```
        type: amazon
  - task: setDnsSuffix
    inputs:
      suffixes:
      - $REGION.ec2-utilities.amazonaws.com
  - task: setAdminAccount
    inputs:
      password:
        type: random
  - task: setWallpaper
    inputs:
      path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
      attributes:
      - hostName
      - instanceId
      - privateIpAddress
      - publicIpAddress
      - instanceSize
      - availabilityZone
      - architecture
      - memory
      - network
- stage: postReady
  tasks:
  - task: startSsm
```

Schema: user data

The following JSON and YAML examples show the document structure for user data. Amazon EC2 parses each task named in the `tasks` array that you specify in the document. Each task has its own set of properties and requirements. For details, see the Task definitions.

> **Note**
>
> A task must only appear once in user data tasks array.

Document structure: user data

JSON

```
{
```

```
 "version": "1.1",
 "tasks": [
  {
   "task": "string",
   "inputs": {
    ...
   },
  },
  ...
 ]
}
```

YAML

```
version: 1.1
tasks:
- task: string
  inputs:
    ...
...
```

Example: user data

For more information about user data, see Run commands on your Windows instance at launch.

The following YAML document example shows a PowerShell script that EC2Launch v2 runs as user data to create a file.

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
    content: |-
      New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

You can use an XML format for the user data that's compatible with previous versions of the launch agent. EC2Launch v2 runs the script as an `executeScript` task in the `UserData` stage. To conform with EC2Launch v1 and EC2Config behavior, the user data script runs as an attached/inline process by default.

You can add optional tags to customize how your script runs. For example, to run the user data script when the instance reboots in addition to one time when the instance launches, you can use the following tag:

```
<persist>true</persist>
```

Example:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

To run an XML user data script as a detached process, add the following tag to your user data.

```
<detach>true</detach>
```

Example:

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

> **Note**
>
> The detach tag is not supported on previous launch agents.

Change log: user data

The following table lists changes for user data, and cross-references them to the EC2Launch v2 agent version that applies.

User data version	Details	Introduced in
1.1	• User data tasks run before the `PostReady` stage in the agent config file. • Runs user data before starting the Systems Manager Agent (same behavior as EC2Launch v1 and EC2Config).*	EC2Launch v2 version 2.0.1245
1.0	• Will be deprecated. • User data tasks run after the `PostReady` stage in the agent config file. This is not backwards compatible with EC2Launch v1. • Impacted by a race condition between Systems Manager Agent start and user data tasks.	EC2Launch v2 version 2.0.0

* When used with the default `agent-config.yml` file.

Task definitions

Each task has its own set of properties and requirements. For details, see the individual tasks that you want to include in your document.

Tasks

- activateWindows
- enableJumboFrames
- enableOpenSsh
- executeProgram
- executeScript
- extendRootPartition
- initializeVolume
- optimizeEna

- setAdminAccount
- setDnsSuffix
- setHostName
- setWallpaper
- startSsm
- sysprep
- writeFile

activateWindows

Activates Windows against a set of AWS KMS servers. Activation is skipped if the instance is detected as Bring-Your-Own-License (BYOL).

Frequency — once

AllowedStages — `[PreReady]`

Inputs —

`activation`: (map)

`type`: (string) activation type to use, set to `amazon`

Example

```
task: activateWindows
inputs:
  activation:
    type: amazon
```

enableJumboFrames

Enables Jumbo Frames, which increase the maximum transmission unit (MTU) of the network adapter. For more information, see Jumbo frames (9001 MTU).

Frequency — always

AllowedStages — `[PostReady, UserData]`

Inputs — none

Example

```
task: enableJumboFrames
```

enableOpenSsh

Enables Windows OpenSSH and adds the public key for the instance to the authorized keys folder.

Frequency — once

AllowedStages — `[PreReady, UserData]`

Inputs — none

Example

The following example shows how to enable OpenSSH on an instance, and to add the public key for the instance to the authorized keys folder. This configuration works only on instances running Windows Server 2019 and later versions.

```
task: enableOpenSsh
```

executeProgram

Runs a program with optional arguments and a specified frequency.

Stages: You can run the `executeProgram` task during the `PreReady`, `PostReady`, and `UserData` stages.

Frequency: configurable, see *Inputs*.

Inputs

You can configure runtime parameters as follows:

frequency (string)

(Required) Specify exactly one of the following values:

- `once`
- `always`

path (string)

(Required) The file path for the executable to run.

arguments (list of strings)

(Optional) A comma separated list of arguments to provide to the program as input.

runAs (string)

(Required) Must be set to `localSystem`

Output

All of the tasks write logfile entries to the `agent.log` file. Additional output from the `executeProgram` task is stored separately in a dynamically named folder, as follows:

`%LocalAppData%\Temp\`*`EC2Launch########`*`\`*`outputfilename.tmp`*

The exact path to the output files is included in the `agent.log` file, for example:

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\ExecuteProgramInputs.tmp
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

Output files for the `executeProgram` task

`ExecuteProgramInputs.tmp`

Contains the path for the executable, and all of the input parameters that the `executeProgram` task passes to it when it runs.

`Output.tmp`

Contains runtime output from the program that the `executeProgram` task runs.

`Err.tmp`

Contains runtime error messages from the program that the `executeProgram` task runs.

Examples

The following examples show how to run an executable file from a local directory on an instance with the `executeProgram` task.

Example 1: Setup executable with one argument

This example shows an `executeProgram` task that runs a setup executable in quiet mode.

```
task: executeProgram
inputs:
- frequency: always
  path: C:\Users\Administrator\Desktop\setup.exe
  arguments: ['-quiet']
```

Example 2: VLC executable with two arguments

This example shows an `executeProgram` task that runs a VLC executable file with two arguments passed as input parameters.

```
task: executeProgram
inputs:
- frequency: always
  path: C:\vlc-3.0.11-win64.exe
  arguments: ['/L=1033','/S']
  runAs: localSystem
```

executeScript

Runs a script with optional arguments and a specified frequency. Script behavior depends on what mode the agent runs the scripts in – inline, or detached.

Inline (default)

The EC2Launch v2 agent runs scripts one at a time (`detach: false`). This is the default setting.

> **Note**
>
> When your inline script issues a **reset** or **sysprep** command, it runs immediately and resets the agent. The current task finishes, then the agent shuts down without running any further tasks.
> For example, if the task that issues the command would have been followed by a `startSsm` task (included by default after user data runs), the task doesn't run and the Systems Manager service never starts.

Detached

The EC2Launch v2 agent runs scripts concurrently with other tasks (`detach: true`).

> **Note**
>
> When your detached script issues a **reset** or **sysprep** command, those commands wait for the agent to finish before they run. Tasks after the executeScript will still run.

Stages: You can run the `executeScript` task during the `PreReady`, `PostReady`, and `UserData` stages.

Frequency: configurable, see *Inputs*.

Inputs

You can configure runtime parameters as follows:

frequency (string)

(Required) Specify exactly one of the following values:

- `once`
- `always`

type (string)

(Required) Specify exactly one of the following values:

- `batch`
- `powershell`

arguments (list of strings)

(Optional) A list of string arguments to pass to the shell. This parameter isn't supported for `type: batch`.

content (string)

(Required) Script content.

runAs (string)

(Required) Specify exactly one of the following values:

- `admin`
- `localSystem`

detach (Boolean)

(Optional) The EC2Launch v2 agent defaults to run scripts one at a time (`detach: false`). To run the script concurrently with other tasks, set the value to `true` (`detach: true`).

> **Note**
>
> Script exit codes (including `3010`) have no effect when `detach` is set to `true`.

Output

All of the tasks write logfile entries to the `agent.log` file. Additional output from script that the `executeScript` task runs is stored separately in a dynamically named folder, as follows:

`%LocalAppData%\Temp\`*`EC2Launch########`*`\`*`outputfilename.ext`*

The exact path to the output files is included in the `agent.log` file, for example:

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\UserScript.ps1
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

Output files for the `executeScript` task

`UserScript.`*`ext`*

Contains the script that the `executeScript` task ran. The file extension depends on the type of script you specified in the `type` parameter for the `executeScript` task, as follows:

- If the type is `batch`, then the file extension is `.bat`.
- If the type is `powershell`, then the file extension is `.ps1`.

`Output.tmp`

Contains runtime output from the script that the `executeScript` task runs.

`Err.tmp`

Contains runtime error messages from the script that the `executeScript` task runs.

Examples

The following examples show how to run an inline script with the `executeScript` task.

Example 1: Hello world output text file

This example shows an `executeScript` task that runs a PowerShell script to create a "Hello world" text file on the `C:` drive.

```
task: executeScript
inputs:
- frequency: always
  type: powershell
  runAs: admin
  content: |-
    New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
    Set-Content 'C:\PowerShellTest.txt' "Hello world"
```

Example 2: Run two scripts

This example shows that the `executeScript` task can run more than one script, and the script type doesn't necessarily need to match.

The first script (`type: powershell`) writes a summary of the processes that are currently running on the instance to a text file located on the `C:` drive.

The second script (`batch`) writes the system information to the `Output.tmp` file.

```
task: executeScript
inputs:
- frequency: always
  type: powershell
  content: |
    Get-Process | Out-File -FilePath C:\Process.txt
  runAs: localSystem
- frequency: always
  type: batch
  content: |
```

```
    systeminfo
```

Example 3: Idempotent system configuration with reboots

This example shows an `executeScript` task that runs an idempotent script to perform the following system configuration with a reboot between each step:

- Rename the computer.
- Join the computer to the domain.
- Enable Telnet.

The script ensures that each operation runs one time only. This prevents a reboot loop and makes the script idempotent.

```
task: executeScript
inputs:
- frequency: always
  type: powershell
  runAs: localSystem
  content: |-
    $name = $env:ComputerName
     if ($name -ne $desiredName) {
       Rename-Computer -NewName $desiredName
       exit 3010
     }
     $domain = Get-ADDomain
     if ($domain -ne $desiredDomain)
     {
       Add-Computer -DomainName $desiredDomain
       exit 3010
     }
    $telnet = Get-WindowsFeature -Name Telnet-Client
    if (-not $telnet.Installed)
    {
      Install-WindowsFeature -Name "Telnet-Client"
      exit 3010
    }
```

extendRootPartition

Extends the root volume to use all of the available space on the disk.

Frequency — once

AllowedStages — `[Boot]`

Inputs — none

Example

```
task: extendRootPartition
```

initializeVolume

Initializes empty volumes that are attached to the instance so that they're activated and partitioned. The launch agent skips initialization if it detects that the volume is not empty. A volume is considered empty if the first 4 KiB of the volume are empty, or if the volume doesn't have a Windows-recognizable drive layout.

The `letter` input parameter is always applied when this task runs, regardless of whether the drive is already initialized.

The `initializeVolume` task performs the following actions.

- Set disk attributes `offline` and `readonly` to false.
- Create a partition. If no partition type is specified in the `partition` input parameter, the following defaults apply:
  - If the disk size is smaller than 2 TB, set the partition type to `MBR`.
  - If the disk size is 2 TB or larger, set the partition type to `GPT`.
- Format the volume as NTFS.
- Set the volume label as follows:
  - Use the value of the `name` input parameter, if specified.
  - If the volume is ephemeral, and no name was specified, set the volume label to `Temporary Storage Z`.
- If the volume is ephemeral (SSD or HDD – not Amazon EBS), create an `Important.txt` file at the root of the volume with the following content:

  ```
  This is an 'Instance Store' disk and is provided at no additional charge.
  ```

```
*This disk offers increased performance since it is local to the host
*The number of Instance Store disks available to an instance vary by instance type
*DATA ON THIS DRIVE WILL BE LOST IN CASES OF IMPAIRMENT OR STOPPING THE INSTANCE.
 PLEASE ENSURE THAT ANY IMPORTANT DATA IS BACKED UP FREQUENTLY

For more information, please refer to: Amazon EC2 instance store.
```

- Set the drive letter to the value specified in the `letter` input parameter.

Stages: You can run the `initializeVolume` task during the `PostReady` and `UserData` stages.

Frequency: always.

Inputs

You can configure runtime parameters as follows:

devices (list of maps)

(Conditional) Configuration for each device that the launch agent initializes. This is required if the `initialize` input parameter is set to `devices`.

- **device (string, required)** – Identifies the device during instance creation. For example, `xvdb`, `xvdf`, or `\dev\nvme0n1`.
- **letter (string, optional)** – One character. The drive letter to assign.
- **name (string, optional)** – The volume name to assign.
- **partition (string, optional)** – Specify one of the following values for the type of partition to create, or let the launch agent default based on volume size:
  - MBR
  - GPT

initialize (string)

(Required) Specify exactly one of the following values:

- `all`
- `devices`

Examples

The following examples show sample input configurations for the `initializeVolume` task.

Example 1: Initialize two volumes on an instance

This example shows an `initializeVolume` task that initializes two secondary volumes on an instance. The device named `DataVolume2` in the example is ephemeral.

```
task: initializeVolume
inputs:
  initialize: devices
  devices:
  - device: xvdb
    name: DataVolume1
    letter: D
    partition: mbr
  - device: /dev/nvme0n1
    name: DataVolume2
    letter: E
    partition: gpt
```

Example 2: Initialize EBS volumes attached to an instance

This example shows an `initializeVolume` task that initializes all empty EBS volumes that are attached to the instance.

```
task: initializeVolume
inputs:
  initialize: all
```

optimizeEna

Optimizes ENA settings based on the current instance type; might reboot the instance.

Frequency — always

AllowedStages — `[PostReady, UserData]`

Inputs — none

Example

```
task: optimizeEna
```

setAdminAccount

Sets attributes for the default administrator account that is created on the local machine.

Frequency — once

AllowedStages — `[PreReady]`

Inputs —

`name`: (string) name of the administrator account

`password`: (map)

`type`: (string) strategy to set the password, either as `static`, `random`, or `doNothing`

`data`: (string) stores data if the `type` field is static

Example

```
task: setAdminAccount
inputs:
  name: Administrator
  password:
    type: random
```

setDnsSuffix

Adds DNS suffixes to the list of search suffixes. Only suffixes that do not already exist are added to the list. For more information about how launch agents set DNS suffixes, see Configure DNS Suffix.

Frequency — always

AllowedStages — `[PreReady]`

Inputs —

`suffixes`: (list of strings) list of one or more valid DNS suffixes; valid substitution variables are `$REGION` and `$AZ`

Example

```
task: setDnsSuffix
```

```
inputs:
  suffixes:
  - $REGION.ec2-utilities.amazonaws.com
```

setHostName

Sets the hostname of the computer to a custom string or, if `hostName` is not specified, the private IPv4 address.

Frequency — always

AllowedStages — `[PostReady, UserData]`

Inputs —

`hostName`: (string) optional host name, which must be formatted as follows.

- Must be 15 characters or less
- Must contain only alphanumeric (a-z, A-Z, 0-9) and hyphen (-) characters.
- Must not consist entirely of numerical characters.

`reboot`: (boolean) denotes whether a reboot is permitted when the hostname is changed

Example

```
task: setHostName
inputs:
  reboot: true
```

setWallpaper

Creates the `setwallpaper.lnk` shortcut file in the startup folder of each existing user except for `Default User`. This shortcut file runs when the user logs in for the first time after instance boot. It sets up the instance with a custom wallpaper that displays the instance attributes.

The shortcut file path is:

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/
Startup/setwallpaper.lnk
```

> **Note**
>
> When you remove the `setWallpaper` task, it does not delete this shortcut file. For more information, see `setWallpaper` task is not enabled but the wallpaper resets at reboot.

Stages: You can configure wallpaper during the `PreReady` and `UserData` stages.

Frequency: `always`

Wallpaper configuration

You can use the following settings to configure your wallpaper.

Inputs

Input parameters that you provide, and attributes that you can set to configure your wallpaper:

attributes (list of strings)

(Optional) You can add one or more of the following attributes to your wallpaper:

- `architecture`
- `availabilityZone`
- `hostName`
- `instanceId`
- `instanceSize`
- `memory`
- `network`
- `privateIpAddress`
- `publicIpAddress`

instanceTags

(Optional) You can use exactly one of the following options for this setting.

- **AllTags** (string) – Add all instance tags to your wallpaper.

```
instanceTags: AllTags
```

- **instanceTags** (list of strings) – Specify a list of instance tag names to add to your wallpaper. For example:

```
instanceTags:
  - Tag 1
  - Tag 2
```

path (string)

(Required) The filename path of the local .jpg format image file to use for your wallpaper image.

Example

The following example shows wallpaper configuration inputs that set the file path for the wallpaper background image, along with instance tags named `Tag 1` and `Tag 2`, and attributes that include the host name, instance ID, and private and public IP addresses for the instance.

```
task: setWallpaper
inputs:
  path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
  attributes:
  - hostName
  - instanceId
  - privateIpAddress
  - publicIpAddress
  instanceTags:
  - Tag 1
  - Tag 2
```

> **Note**
>
> You must enable tags in metadata to show tags on the wallpaper. For more information about instance tags and metadata, see Work with instance tags in instance metadata.

startSsm

Starts the Systems Manager (SSM) service following Sysprep.

Frequency — always

AllowedStages — `[PostReady, UserData]`

Inputs — none

Example

```
task: startSsm
```

sysprep

Resets the service state, updates `unattend.xml`, disables RDP, and runs Sysprep. This task runs only after all other tasks are completed.

Frequency — once

AllowedStages — `[UserData]`

Inputs —

`clean`: (boolean) cleans instance logs before running Sysprep

`shutdown`: (boolean) shuts down the instance after running Sysprep

Example

```
task: sysprep
inputs:
  clean: true
  shutdown: true
```

writeFile

Writes a file to a destination.

Frequency — see *Inputs*

AllowedStages — `[PostReady, UserData]`

Inputs —

`frequency`: (string) one of `once` or `always`

`destination`: (string) path to which to write the content

`content`: (string) text to write to the destination

Example

```
task: writeFile
inputs:
- frequency: once
  destination: C:\Users\Administrator\Desktop\booted.txt
  content: Windows Has Booted
```

EC2Launch v2 exit codes and reboots

You can use EC2Launch v2 to define how exit codes are handled by your scripts. By default, the exit code of the last command that is run in a script is reported as the exit code for the entire script. For example, if a script includes three commands and the first command fails but the following ones succeed, the run status is reported as `success` because the final command succeeded.

If you want a script to reboot an instance, then you must specify `exit 3010` in your script, even when the reboot is the last step in your script. `exit 3010` instructs EC2Launch v2 to reboot the instance and call the script again until it returns an exit code that is not `3010`, or until the maximum reboot count has been reached. EC2Launch v2 permits a maximum of 5 reboots per task. If you attempt to reboot an instance from a script by using a different mechanism, such as `Restart-Computer`, then the script run status will be inconsistent. For example, it may get stuck in a restart loop or not perform the restart.

If you are using an XML user data format that is compatible with older agents, the user data may run more times than you intend it to. For more information, see Service runs user data more than once in the Troubleshooting section.

EC2Launch v2 and Sysprep

The EC2Launch v2 service runs Sysprep, a Microsoft tool that enables you to create a customized Windows AMI that can be reused. When EC2Launch v2 calls Sysprep, it uses the files in `%ProgramData%\Amazon\EC2Launch` to determine which operations to perform. You can edit these files indirectly using the **EC2Launch settings** dialog box, or directly using a YAML editor or a text editor. However, there are some advanced settings that aren't available in the **EC2Launch settings** dialog box, so you must edit those entries directly.

If you create an AMI from an instance after updating its settings, the new settings are applied to any instance that's launched from the new AMI. For information about creating an AMI, see Create a custom Windows AMI.

Troubleshoot EC2Launch v2

This section shows common troubleshooting scenarios for EC2Launch v2, information about viewing Windows event logs, and console log output and messages.

Troubleshooting topics

- Common troubleshooting scenarios
- Windows event logs
- EC2Launch v2 console log output

Common troubleshooting scenarios

This section shows common troubleshooting scenarios and steps for resolution.

Scenarios

- Service fails to set the wallpaper
- Service fails to run user data
- Service runs a task only one time
- Service fails to run a task
- Service runs user data more than once
- Scheduled tasks from EC2Launch v1 fail to run after migration to EC2Launch v2
- Service initializes an EBS volume that is not empty
- setWallpaper task is not enabled but the wallpaper resets at reboot
- Service stuck in running status
- Invalid agent-config.yml prevents opening EC2Launch v2 settings dialog box
- task:executeScript should be unique and only invoked once

Service fails to set the wallpaper

Resolution

1. Check that `%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\setwallpaper.lnk` exists.
2. Check `%ProgramData%\Amazon\EC2Launch\log\agent.log` to see if any errors occurred.

Service fails to run user data

Possible cause: Service may have failed before running user data.

Resolution

1. Check `%ProgramData%\Amazon\EC2Launch\state\previous-state.json`.
2. See if `boot`, `network`, `preReady`, and `postReadyLocalData` have all been marked as success.
3. If one of the stages failed, check `%ProgramData%\Amazon\EC2Launch\log\agent.log` for specific errors.

Service runs a task only one time

Resolution

1. Check the frequency of the task.
2. If the service already ran after Sysprep, and the task frequency is set to `once`, the task will not run again.
3. Set the frequency of the task to `always` if you want it to run the task every time EC2Launch v2 runs.

Service fails to run a task

Resolution

1. Check the latest entries in `%ProgramData%\Amazon\EC2Launch\log\agent.log`.
2. If no errors occurred, try running the service manually from `"%ProgramFiles%\Amazon\EC2Launch\EC2Launch.exe" run` to see if the tasks succeed.

Service runs user data more than once

Resolution

User data is handled differently between EC2Launch v1 and EC2Launch v2. EC2Launch v1 runs user data as a scheduled task on the instance when `persist` is set to `true`. If `persist` is set to `false`, the task is not scheduled even when it exits with a reboot or is interrupted while running.

EC2Launch v2 runs user data as an agent task and tracks its run state. If user data issues a computer restart or if user data was interrupted while running, the run state persists as `pending` and the user data will run again at the next instance boot. If you want to prevent the user data script from running more than once, make the script idempotent.

The following example idempotent script sets the computer name and joins a domain.

```
<powershell>
  $name = $env:computername
  if ($name -ne $desiredName) {
 Rename-Computer -NewName $desiredName
  }
  $domain = Get-ADDomain
  if ($domain -ne $desiredDomain)
  {
 Add-Computer -DomainName $desiredDomain
  }
  $telnet = Get-WindowsFeature -Name Telnet-Client
  if (-not $telnet.Installed)
  {
 Install-WindowsFeature -Name "Telnet-Client"
  }
</powershell>
<persist>false</persist>
```

Scheduled tasks from EC2Launch v1 fail to run after migration to EC2Launch v2

Resolution

The migration tool does not detect any scheduled tasks linked to EC2Launch v1 scripts; therefore, it does not automatically set up those tasks in EC2Launch v2. To configure these tasks, edit the `agent-config.yml` file, or use the EC2Launch v2 settings dialog box. For example, if an instance has a scheduled task that runs `InitializeDisks.ps1`, then after you run the migration tool, you must specify the volumes you want to initialize in the EC2Launch v2 settings dialog box. See Step 6 of the procedure to Change settings using the EC2Launch v2 settings dialog box.

Service initializes an EBS volume that is not empty

Resolution

Before it initializes a volume, EC2Launch v2 attempts to detect whether it is empty. If a volume is not empty, it skips the initialization. Any volumes that are detected as not empty are not initialized.

A volume is considered empty if the first 4 KiB of a volume are empty, or if a volume does not have a Windows-recognizable drive layout. A volume that was initialized and formatted on a Linux system does not have a Windows-recognizable drive layout, for example MBR or GPT. Therefore, it will be considered as empty and initialized. If you want to preserve this data, do not rely on EC2Launch v2 empty drive detection. Instead, specify volumes that you would like to initialize in the EC2Launch v2 settings dialog box (see step 6) or in the `agent-config.yml`.

`setWallpaper` task is not enabled but the wallpaper resets at reboot

The `setWallpaper` task creates the `setwallpaper.lnk` shortcut file in the startup folder of each existing user except for `Default User`. This shortcut file runs when the user logs in for the first time after instance boot. It sets up the instance with a custom wallpaper that displays the instance attributes. Removing the `setWallpaper` task does not delete this shortcut file. You must manually delete this file or delete it using a script.

The shortcut path is:

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/
Programs/Startup/setwallpaper.lnk
```

Resolution

Manually delete this file, or delete it using a script.

Example PowerShell script to delete shortcut file

```
foreach ($userDir in (Get-ChildItem "C:\Users" -Force -Directory).FullName)
{
 $startupPath = Join-Path $userDir -ChildPath "AppData\Roaming\Microsoft\Windows\Start
 Menu\Programs\Startup"
 if (Test-Path $startupPath)
 {
  $wallpaperSetupPath = Join-Path $startupPath -ChildPath "setwallpaper.lnk"
  if (Test-Path $wallpaperSetupPath)
  {
   Remove-Item $wallpaperSetupPath -Force -Confirm:$false
  }
 }
}
```

Service stuck in running status

Description

EC2Launch v2 is blocked, with log messages (`agent.log`) similar to the following:

```
2022-02-24 08:08:58 Info:
 ********************************************************************************
2022-02-24 08:08:58 Info: EC2Launch Service starting
2022-02-24 08:08:58 Info: Windows event custom log exists: Amazon EC2Launch
2022-02-24 08:08:58 Info: ACPI SPCR table not supported. Bailing Out
2022-02-24 08:08:58 Info: Serial port is in use. Waiting for Serial Port...
2022-02-24 08:09:00 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:02 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:04 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:06 Info: ACPI SPCR table not supported. Use default console port.
```

Possible cause

SAC is enabled and using the serial port. For more information, see Use SAC to troubleshoot your Windows instance.

Resolution

Try the following steps to resolve this issue:

- Disable the service that is using the serial port.
- If you want the service to continue to use the serial port, write custom scripts to perform launch agent tasks and invoke them as scheduled tasks.

Invalid `agent-config.yml` prevents opening EC2Launch v2 settings dialog box

Description

EC2Launch v2 settings attempts to parse the `agent-config.yml` file before it opens the dialog box. If the YAML configuration file does not follow the supported schema, the dialog box will show the following error:

```
Unable to parse configuration file agent-config.yml. Review configuration
file. Exiting application.
```

Resolution

1. Verify that the configuration file follows the supported schema.
2. If you want to start from scratch, copy the default configuration file into `agent-config.yml`. You can use the example `agent-config.yml` provided in the Task Configuration section.
3. You can also start over by deleting `agent-config.yml`. EC2Launch v2 settings generates an empty configuration file.

`task:executeScript should be unique and only invoked once`

Description

A task cannot be repeated in the same stage.

Resolution

Some tasks must be input as an array, such as **executeScript** and **executeProgram**. For an example of how to write the script as an array, see **executeScript**.

Windows event logs

EC2Launch v2 publishes Windows event logs for important events, such as service starting, Windows is ready, and task success and failure. Event identifiers uniquely identify a particular event. Each event contains stage, task, and level information, and a description. You can set triggers for specific events using the event identifier.

Event IDs provide information about an event and uniquely identify some events. The least significant digit of an event ID indicates the severity of an event.

Event	Least significant digit
Success	. . .0
Informational	. . .1
Warning	. . .2
Error	. . .3

Service-related events that are generated when the service starts or stops include a single digit event identifier.

Event	Single digit identifier
`Success`	`0`
`Informational`	`1`
`Warning`	`2`
`Error`	`3`

The event messages for `EC2LaunchService.exe` events begin with `Service:`. The event messages for `EC2Launch.exe` events do not begin with `Service:`.

Four digit event IDs include information about the stage, task, and severity of an event.

Topics

- Event ID format
- Event ID examples
- Windows event log schema

Event ID format

The following table shows the format of an EC2Launch v2 event identifier.

3	2 1	0
S	T	L

The letters and numbers in the table represent the following event type and definitions.

Event type	Definition

Event type	Definition
S (Stage)	0 - Service-level message 1 - Boot 2 - Network 3 - PreReady 5 - Windows is Ready 6 - PostReady 7 - User Data
T (Task)	The tasks represented by the corresponding two values are different for each stage. To view the complete list of events, see Windows Event log schema.
L (Level of the event)	0 - Success 1 - Informational 2 - Warning 3 - Error

Event ID examples

The following are example event IDs.

- `5000` - Windows is ready to use
- `3010` - Activate windows task in PreReady stage was successful
- `6013` - Set wallpaper task in PostReady Local Data stage encountered an error

Windows event log schema

MessageId/Event Id	Event message
. . .0	Success
. . .1	Informational
. . .2	Warning
. . .3	Error
x	EC2Launch service-level logs
0	EC2Launch service exited successfully
1	EC2Launch service informational logs
2	EC2Launch service warning logs
3	EC2Launch service error logs
10	Replace state.json with previous-state.json
100	Serial Port
200	Sysprep
300	PrimaryNic
400	Metadata
x000	Stage (1 digit), Task (2 digits), Status (1 digit)
1000	Boot
1010	Boot - extend_root_partition

MessageId/Event Id	Event message
2000	Network
2010	Network - add_routes
3000	PreReady
3010	PreReady - activate_windows
3020	PreReady - install_egpu_manager
3030	PreReady - set_monitor_on
3040	PreReady - set_hibernation
3050	PreReady - set_admin_account
3060	PreReady - set_dns_suffix
3070	PreReady - set_wallpaper
3080	PreReady - set_update_schedule
3090	PreReady - output_log
3100	PreReady - enable_open_ssh
5000	Windows is Ready to use
6000	PostReadyLocalData
7000	PostReadyUserData
6010/7010	PostReadyLocal/UserData - set_wallpaper
6020/7020	PostReadyLocal/UserData - set_update_schedule

MessageId/Event Id	Event message
6030/7030	PostReadyLocal/UserData - set_hostname
6040/7040	PostReadyLocal/UserData - execute_program
6050/7050	PostReadyLocal/UserData - execute_script
6060/7060	PostReadyLocal/UserData - manage_package
6070/7070	PostReadyLocal/UserData - initialize_volume
6080/7080	PostReadyLocal/UserData - write_file
6090/7090	PostReadyLocal/UserData - start_ssm
7100	PostReadyUserData - enable_open_ssh
6110/7110	PostReadyLocal/UserData - enable_jumbo_frames

EC2Launch v2 console log output

This section contains sample console log output for EC2Launch v2 and lists all of the EC2Launch v2 console log error messages to help you to troubleshoot issues. For more information about instance console output and how to access it, see Instance console output.

Outputs

- EC2Launch v2 console log output
- EC2Launch v2 console log messages

EC2Launch v2 console log output

The following is sample console log output for EC2Launch v2.

```
2023/11/30 20:18:53Z: Windows sysprep configuration complete.
2023/11/30 20:18:57Z: Message: Waiting for access to metadata...
2023/11/30 20:18:57Z: Message: Meta-data is now available.
2023/11/30 20:18:57Z: AMI Origin Version: 2023.11.15
2023/11/30 20:18:57Z: AMI Origin Name: Windows_Server-2022-English-Full-Base
2023/11/30 20:18:58Z: OS: Microsoft Windows NT 10.0.20348
2023/11/30 20:18:58Z: OsVersion: 10.0
2023/11/30 20:18:58Z: OsProductName: Windows Server 2022 Datacenter
2023/11/30 20:18:58Z: OsBuildLabEx: 20348.1.amd64fre.fe_release.210507-1500
2023/11/30 20:18:58Z: OsCurrentBuild: 20348
2023/11/30 20:18:58Z: OsReleaseId: 2009
2023/11/30 20:18:58Z: Language: en-US
2023/11/30 20:18:58Z: TimeZone: UTC
2023/11/30 20:18:58Z: Offset: UTC +0000
2023/11/30 20:18:58Z: Launch: EC2 Launch v2.0.1643
2023/11/30 20:18:58Z: AMI-ID: ami-1234567890abcdef1
2023/11/30 20:18:58Z: Instance-ID: i-1234567890abcdef0
2023/11/30 20:18:58Z: Instance Type: c5.large
2023/11/30 20:19:00Z: Driver: AWS NVMe Driver v1.5.0.33
2023/11/30 20:19:00Z: SubComponent: AWS NVMe Driver v1.5.0.33;
 EnableSCSIPersistentReservations: 0
2023/11/30 20:19:00Z: Driver: AWS PV Driver Package v8.4.3
2023/11/30 20:19:01Z: Driver: Amazon Elastic Network Adapter v2.6.0.0
2023/11/30 20:19:01Z: RDPCERTIFICATE-SUBJECTNAME: EC2AMAZ-SO1T009
2023/11/30 20:19:01Z: RDPCERTIFICATE-THUMBPRINT:
 1234567890ABCDEF1234567890ABCDEF1234567890
2023/11/30 20:19:09Z: SSM: Amazon SSM Agent v3.2.1705.0
2023/11/30 20:19:13Z: Username: Administrator
2023/11/30 20:19:13Z: Password: <Password>
1234567890abcdef1EXAMPLEPASSWORD
</Password>
2023/11/30 20:19:14Z: User data format: no_user_data
2023/11/30 20:19:14Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2023/11/30 20:19:14Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2023/11/30 20:19:14Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2023/11/30 20:19:14Z: EC2LaunchTelemetry: AgentCommandErrorCode=0
2023/11/30 20:19:14Z: Message: Windows is Ready to use
```

EC2Launch v2 console log messages

The following is a list of all of the EC2Launch v2 console log messages.

```
Message: Error EC2Launch service is stopping. {error message}
 Error setting up EC2Launch agent folders
 See instance logs for detail
 Error stopping service
 Error initializing service
Message: Windows sysprep configuration complete
Message: Invalid administrator username: {invalid username}
Message: Invalid administrator password
Username: {username}
Password: <Password>{encrypted password}</Password>
AMI Origin Version: {amiVersion}
AMI Origin Name: {amiName}
Microsoft Windows NT {currentVersion}.{currentBuildNumber}
OsVersion: {currentVersion}
OsProductName: {productName}
OsBuildLabEx: {buildLabEx}
OsCurrentBuild: {currentBuild}
OsReleaseId: {releaseId}
Language: {language}
TimeZone: {timeZone}
Offset: UTC {offset}
Launch agent: EC2Launch {BuildVersion}
AMI-ID: {amiId}
Instance-ID: {instanceId}
Instance Type: {instanceType}
RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
SqlServerBilling: {sql billing}
SqlServerInstall: {sql patch leve, edition type}
Driver: AWS NVMe Driver {version}
Driver: Inbox NVMe Driver {version}
Driver: AWS PV Driver Package {version}
Microsoft-Hyper-V is installed.
Unable to get service status for vmms
Microsoft-Hyper-V is {status}
SSM: Amazon SSM Agent {version}
AWS VSS Version: {version}
Message: Windows sysprep configuration complete
Message: Windows is being configured. SysprepState is {state}
Windows is still being configured. SysprepState is {state}
```

```
Message: Windows is Ready to use
Message: Waiting for meta-data accessibility...
Message: Meta-data is now available.
Message: Still waiting for meta-data accessibility...
Message: Failed to find primary network interface...retrying...
User data format: {format}
```

EC2Launch v2 version histories

Version histories

- EC2Launch v2 version history
- EC2Launch v2 migration tool version history

EC2Launch v2 version history

The following table describes the released versions of EC2Launch v2.

Version	Details	Release date
2.0.1815	• Adjusted error handling to fail on critical setup issues before sysprep. • Fixed an issue where wallpaper and hostname tasks could use an incorrect IP address on instances with multiple IP addresses assigned to the primary network interface. • Wallpaper and hostname tasks changed to get private IP from IMDS first, then fail back to WMI if IMDS is disabled. • Fixed an issue with the `initializeVolume` task where `sc1` volumes failed to initialize due to a transient error.	March 6, 2024
2.0.1739	• Fixed an issue that prevented exit codes from being captured by `executeScript` tasks that were run as the Windows Administrator user.	January 17, 2024

Version	Details	Release date
2.0.1702	• Restricted `Telemetry.log` permissions to `read-execute` only for standard users. • Configured the EC2Launch Windows service to restart on start-up failure. • Made `add-routes` failures actionable by logging `route.exe stderr` output. • Fixed an issue that occurs when route metrics are outside of the range [1, 9999]. • Added wallpaper support to several new instance types. • Fixed an issue caused by user data scripts that run as the Windows Administrator user and send output to `stderr`.	January 4, 2024

Version	Details	Release date
2.0.1643	• Updated the `ebsnvme-id.exe` tool to version 1.1.0.7. • Fixed an issue with receive side scaling (RSS) and receive queue depth settings on metal instance types that begin with 'metal-*', such as metal-48x1. • Removed telemetry event that reports on XML userdata commands that block the agent. • Updated `setDnsSuffix` task to limit domain name devolution based on registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`. • Added a public task and CLI that adds network routes. • *Note* – This is the last version to officially support Windows Server 2012. • *Note* – This is the last version to officially support 32-bit operating systems.	October 4, 2023
2.0.1580	• Changed the way that the launch agent handles errors when you modify log file permissions. • Added a timeout for connecting to the serial port. The timeout allows the launch agent to continue running if the serial port is in use.	September 5, 2023

Version	Details	Release date

2.0.1521	• Deprecated the `–block` flag of the `EC2Launch.exe` **reset** and **sysprep** commands.
	• Updated `EC2Launch.exe` to detect and handle the **reset** and **sysprep** commands that are used in inline `executeScript` tasks. Those commands cause the agent to stop running after the `executeScript` task runs them.
	• Updated XML userdata scripts to run inline by default.
	• Enable XML userdata scripts to run detached with the new `detach` tag. For more details, see User data scripts.
	• Made the following changes to the agent log.
	  • Updated agent log messages.
	  • Removed `executeScript` content and output from the agent log.
	  • Removed `executeProgram` arguments and output from the agent log.
	• Made the following changes to the console log.
	  • Added `EnableSCSIPersistentReservations` value to the console log.
		July 3, 2023

Version	Details	Release date
2.0.1303	• Added additional error handling and log lines when adding network routes. • Allowed `executeScript` and `executeProgram` tasks in the PreReady stage. • Updated `executeProgram` task to generate output files similar to the output from the executeScript task. For more information, see executeProgram. • Added telemetry to monitor usage of blocking agent commands in XML user data.	May 3, 2023
2.0.1245	• Improved visibility into crashes by logging crash call stacks in clear text. • Added the EventLog service as a startup dependency to fix a crash when the Amazon EC2Launch service starts up faster than the EventLog service. • Made XML user data run before PostReady stage from the agent config file (like EC2Launch v1 and EC2Config). • Added YAML user data version 1.1 to make user data run before PostReady stage from the agent config file (YAML user data version 1.0 runs after PostReady stage from the agent config file).	March 8, 2023

Version	Details	Release date
2.0.1173	• Adds an optional feature to display instance tags on wallpaper. For more information, see setWallpaper . • Adds error handling when the security group for Elastic Graphics is not properly set up. • Fixes a timeout when the Instance Metadata Service is not enabled.	February 6, 2023
2.0.1121	• Fixes an issue where a 404 error is printed to the wallpaper when no public IPv4 address is assigned. • Fixes an issue where the volume's file system is formatted as `RAW` instead of `NTFS` when its device's drive letter is set to `D`. • Fixes an issue where NVMe SSD volumes are incorrectly identified as EBS volumes. • Fixes an error when activating Windows when IMDS is disabled.	January 4, 2023

Version	Details	Release date
2.0.1082	• Fixes an issue where the `setWallpaper :privateIp Address` field is blank when IMDS is disabled. • Fixes an issue with setting the hostname to the private IPv4 address when IMDS is disabled. • Fixes an issue with initializing volumes on Windows Server 2012. • Fixes an issue with setting jumbo frames. • Fixes an error when no SSH key is specified at instance launch. • Fixes an error on Windows Server 2012 when Windows does not have a 'ReleaseId' registry key.	December 7, 2022
2.0.1011	• Fixes logic for finding network adapter when PnPDeviceID is empty.	November 11, 2022
2.0.1009	• Uses PCI segment information to select the console port.	November 8, 2022
2.0.982	• Adds retry logic to get RDP information. • Fixes errors during volume initialization on `d2.8xlarge` instances. • Fixes issue where an incorrect network adapter can be selec ted after a reboot. • Removes false alarm error message when ACPI SPCR is unavailable.	October 31, 2022

Version	Details	Release date
2.0.863	• Updates IMDS wait logic to make only IMDSv2 requests. • Adds logic to assign drive letter to volumes that are already initialized but not mounted. • Prints a more specific error message when key pair type is not supported. • Fixes 3010 reboot code bug. • Adds check for invalid base64-encoded user data.	July 6, 2022
2.0.698	• Fixes typo in log output when executing scripts.	January 30, 2022
2.0.674	• Telemetry uploads the enabled/disabled privacy control. • Fixes `index out of bounds` bug. • Removes wallpaper shortcuts during `sysprep`.	November 15, 2021
2.0.651	• Adds logic to uninstall legacy agents during EC2Launch v2 installation. • Fixes `list-volume` CLI issue when root volume is not listed as volume 0.	October 7, 2021
2.0.592	• Fixes bug to correctly report stage status. • Removes false alarm error messages when log files are closed. • Adds telemetry.	August 31, 2021

Version	Details	Release date
2.0.548	• Adds leading zeros for hex IP hostname. • Fixes file permissions for `enableOpenSsh` task. • Fixes sysprep command crash.	August 4, 2021
2.0.470	• Fixes bug in network stage to wait for DHCP to assign an IP to the instance. • Fixes bug with `setDnsSuffix` when `SearchList` registry key does not exist. • Fixes bug in DNS devolution logic in `setDnsSuffix` . • Adds network routes after intermediate reboots. • Allows `initializeVolume` to re-letter existing volumes. • Removes extra information from version subcommand.	July 20, 2021
2.0.285	• Adds option to run user scripts in a detached process. • Legacy userdata (XML userdata) now runs in a detached process, which is similar behavior to the prior launch agent. • Adds CLI flag to the `sysprep` and `reset` commands, which allows them to block until the service stops. • Restricts the config folder permissions.	March 8, 2021

Version	Details	Release date
2.0.207	• Adds optional `hostName` field to `setHostName` task. • Fixes reboot bug. Reboot tasks `executeScript` and `executeProgram` will be marked as running. • Adds more return codes to the status command. • Adds bootstrap service to fix startup issue when running on `t2.nano` instance type. • Fixes clean installation mode to remove files not tracked by installer.	February 2, 2021
2.0.160	• Fixes `validate` command to detect invalid stage name. • Adds `w32tm resync` command in `addroutes` task. • Fixes issue with changing DNS suffix search order. • Adds check conditions to better report invalid user data.	December 4, 2020
2.0.153	Adds Sysprep functionality in UserData.	November 3, 2020

Version	Details	Release date
2.0.146	• Fixes issue with RootExtend on non-English AMIs. • Grants users group write permission to log files. • Creates MS Reserved partition for GPT volumes. • Adds list-volumes command and volume dropdown in Amazon EC2Launch settings. • Adds get-agent-config command for printing agent-co nfig.yml file in yaml or json format. • Erases static password if no public key detected.	October 6, 2020
2.0.124	• Adds option to display OS version on wallpaper. • Initializes encrypted EBS volumes. • Adds routes for VPCs with no local DNS name.	September 10, 2020
2.0.104	• Creates DNS suffix search list if it does not exist. • Skips Hibernation if not requested.	August 12, 2020
2.0.0	Initial release.	June 30, 2020

EC2Launch v2 migration tool version history

The following table describes the released versions of the EC2Launch v2 migration tool.

Version	Details	Release date
1.0.358	• Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1815.	March 8, 2024
1.0.345	• Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1739.	January 18, 2024
1.0.342	• Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1702.	January 5, 2024
1.0.331	• Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1643 • Fix an error that occurs when running `.Install.ps1 -DryRun`. • Fix an issue where password configuration is incorrectly set to `random` during migration from EC2Config. • Fix an error that occurs if `setWallpaper` is set to `False` during migration from EC2Launch.	November 3, 2023
1.0.303	Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1580.	September 14, 2023
1.0.286	Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1521.	July 14, 2023
1.0.272	Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1303.	May 3, 2023
1.0.262	Update migration tool with the latest version of the EC2Launch v2 agent: 2.0.1245.	March 9, 2023
1.0.241	Increments the version number of the EC2Launch v2 agent to 2.0.1011.	December 7, 2022

Version	Details	Release date
1.0.218	• Validates Region value retrieved from instance metadata. • Fixes migration failure bug in language packs. • Increments the version number of the EC2Launch v2 agent to 2.0.863.	September 3, 2022
1.0.162	• Moves logic to remove legacy agents to the EC2Launch v2 MSI. • Increments the version number of the EC2Launch v2 agent to 2.0.698.	March 18, 2022
1.0.136	Increments the version number of the EC2Launch v2 agent to 2.0.651.	October 13, 2021
1.0.130	Increments the version number of the EC2Launch v2 agent to 2.0.548.	August 5, 2021
1.0.113	Uses IMDSv2 in place of IMDSv1.	June 4, 2021
1.0.101	Increments the version number of the EC2Launch v2 agent to 2.0.285.	March 12, 2021
1.0.86	Increments the version number of the EC2Launch v2 agent to 2.0.207.	February 3, 2021
1.0.76	Increments the version number of the EC2Launch v2 agent to 2.0.160.	December 4, 2020
1.0.69	Increments the version number of the EC2Launch v2 agent to 2.0.153.	November 5, 2020
1.0.65	Increments the version number of the EC2Launch v2 agent to 2.0.146.	October 9, 2020
1.0.60	Increments the version number of the EC2Launch v2 agent to 2.0.124.	September 10, 2020

Version	Details	Release date
1.0.54	• Installs EC2Launch v2 if no agents are installed. • Increments the version number of the EC2Launch v2 agent to 2.0.104. • Decouples the SSM agent.	August 12, 2020
1.0.50	Removes NuGet dependency.	August 10, 2020
1.0.0	Initial release.	June 30, 2020

Configure a Windows instance using EC2Launch

EC2Launch is a set of Windows PowerShell scripts that replaced the EC2Config service on Windows Server 2016 and 2019 AMIs. Many of these AMIs are still available. EC2Launch v2 is the latest launch agent for all supported Windows versions, which replaces both EC2Config and EC2Launch. For more information, see Configure a Windows instance using EC2Launch v2.

> **Note**
>
> To use EC2Launch with IMDSv2, the version must be 1.3.2002730 or later.

Contents

- EC2Launch tasks
- Telemetry
- Install the latest version of EC2Launch
- Verify the EC2Launch version
- EC2Launch directory structure
- Configure EC2Launch
- EC2Launch version history

EC2Launch tasks

EC2Launch performs the following tasks by default during the initial instance boot:

- Sets up new wallpaper that renders information about the instance.
- Sets the computer name to the private IPv4 address of the instance.
- Sends instance information to the Amazon EC2 console.
- Sends the RDP certificate thumbprint to the EC2 console.
- Sets a random password for the administrator account.
- Adds DNS suffixes.
- Dynamically extends the operating system partition to include any unpartitioned space.
- Executes user data (if specified). For more information about specifying user data, see Work with instance user data.
- Sets persistent static routes to reach the metadata service and AWS KMS servers.

> **Important**
>
> If a custom AMI is created from this instance, these routes are captured as part of the OS configuration and any new instances launched from the AMI will retain the same routes, regardless of subnet placement. In order to update the routes, see Update metadata/KMS routes for Server 2016 and later when launching a custom AMI.

The following tasks help to maintain backward compatibility with the EC2Config service. You can also configure EC2Launch to perform these tasks during startup:

- Initialize secondary EBS volumes.
- Send Windows Event logs to the EC2 console logs.
- Send the *Windows is ready to use* message to the EC2 console.

For more information about Windows Server 2019, see Compare Features in Windows Server Versions on Microsoft.com.

Telemetry

Telemetry is additional information that helps AWS to better understand your requirements, diagnose issues, and deliver features to improve your experience with AWS services.

EC2Launch version `1.3.2003498` and later collect telemetry, such as usage metrics and errors. This data is collected from the Amazon EC2 instance on which EC2Launch runs. This includes all Windows AMIs owned by AWS.

The following types of telemetry are collected by EC2Launch:

- **Usage information** — agent commands, install method, and scheduled run frequency.
- **Errors and diagnostic information** — agent installation and run error codes.

Examples of collected data:

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

Telemetry is enabled by default. You can disable telemetry collection at any time. If telemetry is enabled, EC2Launch sends telemetry data without additional customer notifications.

Your choice to enable or disable telemetry is collected.

You can opt in or out of telemetry collection. Your selection to opt in or out of telemetry is collected to ensure that we adhere to your telemetry option.

Telemetry visibility

When telemetry is enabled, it appears in the Amazon EC2 console output as follows:

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

Disable telemetry on an instance

To disable telemetry by setting a system environment variable, run the following command as an administrator:

```
setx /M EC2LAUNCH_TELEMETRY 0
```

To disable telemetry during installation, run `install.ps1` as follows:

```
. .\install.ps1 -EnableTelemetry:$false
```

Install the latest version of EC2Launch

Use the following procedure to download and install the latest version of EC2Launch on your instances.

To download and install the latest version of EC2Launch

1. If you have already installed and configured EC2Launch on an instance, make a backup of the EC2Launch configuration file. The installation process does not preserve changes in this file. By default, the file is located in the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.
2. Download EC2-Windows-Launch.zip to a directory on the instance.
3. Download install.ps1 to the same directory where you downloaded `EC2-Windows-Launch.zip`.
4. Run `install.ps1`
5. If you made a backup of the EC2Launch configuration file, copy it to the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.

To download and install the latest version of EC2Launch using PowerShell

If you have already installed and configured EC2Launch on an instance, make a backup of the EC2Launch configuration file. The installation process does not preserve changes in this file. By default, the file is located in the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.

To install the latest version of EC2Launch using PowerShell, run the following commands from a PowerShell window

```
mkdir $env:USERPROFILE\Desktop\EC2Launch
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/EC2-Windows-
Launch.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -
Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/install.ps1"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -
Leaf)
```

```
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
& $env:USERPROFILE\Desktop\EC2Launch\install.ps1
```

Verify the installation by checking `C:\ProgramData\Amazon\EC2-Windows\Launch`.

Verify the EC2Launch version

Use the following Windows PowerShell command to verify the installed version of EC2Launch.

```
PS C:\> Test-ModuleManifest -Path "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1" | Select Version
```

EC2Launch directory structure

EC2Launch is installed by default on Windows Server 2016 and later AMIs in the root directory `C:\ProgramData\Amazon\EC2-Windows\Launch`.

> **Note**
>
> By default, Windows hides files and folders under `C:\ProgramData`. To view EC2Launch directories and files, you must either type the path in Windows Explorer or change the folder properties to show hidden files and folders.

The `Launch` directory contains the following subdirectories.

- `Scripts` — Contains the PowerShell scripts that make up EC2Launch.
- `Module` — Contains the module for building scripts related to Amazon EC2.
- `Config` — Contains script configuration files that you can customize.
- `Sysprep` — Contains Sysprep resources.
- `Settings` — Contains an application for the Sysprep graphical user interface.
- `Library` — Contains shared libraries for EC2 launch agents.
- `Logs` — Contains log files generated by scripts.

EC2Launch version `1.3.2004592` and later

Users of the `Administrators` group have `Full control` permissions to all EC2Launch directories. User that aren't in the Administrators group have `Read & execute` permissions to

all EC2Launch directories except `C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Config`. The `Config` directory is restricted to users that are members of the `Administrators` group.

EC2Launch version `1.3.2004491` and earlier

All EC2Launch directories inherit their permissions from `C:\ProgramData`, except `C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Scripts`. This folder inherits all initial permissions from `C:\ProgramData` when it is created, but removes access for normal users to `CreateFiles` in the directory.

Configure EC2Launch

After your instance has been initialized the first time, you can configure EC2Launch to run again and perform different start-up tasks.

Tasks

- Configure initialization tasks
- Schedule EC2Launch to run on every boot
- Initialize drives and map drive letters
- Send Windows event logs to the EC2 console
- Send Windows is ready message after a successful boot

Configure initialization tasks

Specify settings in the `LaunchConfig.json` file to enable or disable the following initialization tasks:

- Set the computer name to the instance private IPv4 address.
- Set the monitor to always stay on.
- Set up new wallpaper.
- Add DNS suffix list.

> **Note**
>
> This adds a DNS suffix lookup for the following domain and configures other standard suffixes. For more information about how launch agents set DNS suffixes, see Configure DNS Suffix.

```
region.ec2-utilities.amazonaws.com
```

- Extend the boot volume size.
- Set the administrator password.

To configure initialization settings

1. On the instance to configure, open the following file in a text editor: `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json`.
2. Update the following settings as needed and save your changes. Provide a password in `adminPassword` only if `adminPasswordtype` is `Specify`.

```
{
 "setComputerName": false,
 "setMonitorAlwaysOn": true,
 "setWallpaper": true,
 "addDnsSuffixList": true,
 "extendBootVolumeSize": true,
 "handleUserData": true,
 "adminPasswordType": "Random | Specify | DoNothing",
 "adminPassword": "password that adheres to your security policy (optional)"
}
```

The password types are defined as follows:

`Random`

EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.

`Specify`

EC2Launch uses the password you specify in `adminPassword`. If the password does not meet the system requirements, EC2Launch generates a random password instead. The password is stored in `LaunchConfig.json` as clear text and is deleted after Sysprep sets the administrator password. EC2Launch encrypts the password using the user's key.

`DoNothing`

EC2Launch uses the password you specify in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.

3. In Windows PowerShell, run the following command to schedule the script to run as a Windows Scheduled Task. The script runs one time during the next boot and then disables these tasks from running again.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

Schedule EC2Launch to run on every boot

You can schedule EC2Launch to run on every boot instead of only the initial boot.

To enable EC2Launch to run on every boot:

1. Open Windows PowerShell and run the following command:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

2. Or, run the executable with the following command:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe
```

Then select `Run EC2Launch on every boot`. You can specify that your EC2 instance `Shutdown without Sysprep` or `Shutdown with Sysprep`.

> **Note**
>
> When you enable EC2Launch to run on every boot, the following happens the next time EC2Launch runs:
>
> - If `AdminPasswordType` is still set to `Random`, EC2Launch will generate a new password at the next boot. After that boot, `AdminPasswordType` is automatically set to `DoNothing` to prevent EC2Launch from generating new passwords on subsequent boots.

To prevent EC2Launch from generating a new password on the first boot, manually set `AdminPasswordType` to `DoNothing` before you reboot.

- `HandleUserData` will be set back to `false` unless the user data has `persist` set to `true`. For more information about user data scripts, see User Data Scripts in the Amazon EC2 User Guide.

Initialize drives and map drive letters

Specify settings in the `DriveLetterMappingConfig.json` file to map drive letters to volumes on your EC2 instance. The script initializes drives that are not already initialized and partitioned. For more information about getting volume details in Windows, see Get-Volume in the Microsoft documentation.

To map drive letters to volumes

1. Open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` file in a text editor.
2. Specify the following volume settings and save your changes:

```
{
 "driveLetterMapping": [
  {
   "volumeName": "sample volume",
   "driveLetter": "H"
  }
 ]
}
```

3. Open Windows PowerShell and use the following command to run the EC2Launch script that initializes the disks:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

To initialize the disks each time the instance boots, add the `-Schedule` flag as follows:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -
Schedule
```

Send Windows event logs to the EC2 console

Specify settings in the `EventLogConfig.json` file to send Windows Event logs to EC2 console logs.

To configure settings to send Windows Event logs

1. On the instance, open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogConfig.json` file in a text editor.
2. Configure the following log settings and save your changes:

```
{
 "events": [
  {
   "logName": "System",
   "source": "An event source (optional)",
   "level": "Error | Warning | Information",
   "numEntries": 3
  }
 ]
}
```

3. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task each time the instance boots.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendEventLogs.ps1 -
Schedule
```

   The logs can take three minutes or more to appear in the EC2 console logs.

Send Windows is ready message after a successful boot

The EC2Config service sent the "Windows is ready" message to the EC2 console after every boot. EC2Launch sends this message only after the initial boot. For backwards compatibility with the EC2Config service, you can schedule EC2Launch to send this message after every boot. On the instance, open Windows PowerShell and run the following command. The system schedules the script to run as a Windows Scheduled Task.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -
Schedule
```

EC2Launch version history

Windows AMIs starting with Windows Server 2016 include a set of Windows Powershell scripts called EC2Launch. EC2Launch performs tasks during the initial instance boot. For information about the EC2Launch versions included in the Windows AMIs, see see AWS Windows AMIs.

To download and install the latest version of EC2Launch, see Install the latest version of EC2Launch.

The following table describes the released versions of EC2Launch. Note that the version format changed after version 1.3.610.

Version	Details	Release date
1.3.2004617	• Fixed an error when setting the wallpaper.	15 January 2024
1.3.2004592	• Updated access permissions set by install.ps1 for `%ProgramData%\Amazon\EC2-Windows\Launch` . • Restricted EC2Launch folder/file access to read-execute only for standard user accounts. • Changed the agent to stop waiting for the Instance Metadata Service (IMDS) to initialize if IMDS is not enabled for the instance. • Added a five minute timeout when waiting for the IMDS to initialize. • Changed the agent to write telemetry to the instance console log before the `Windows is Ready` message instead of after. • Added wallpaper support to several new instance types.	2 January 2024

Version	Details	Release date
	For more information about access permissions and user account permissions of EC2Launch directories, see EC2Launch directory structure.	
1.3.2004491	• Added telemetry to monitor usage of the **Specify admin password** option.	9 November 2023
1.3.2004462	• Added flush after every write to the serial console.	18 October 2023
1.3.2004438	• Limit domain name devolution based on registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`. • Limited `UserdataExecution.log` permissions to `Administrators` only. • Added error messages in the Windows Event Log when log initialization fails.	4 October 2023
1.3.2004256	• Added `EnableSCSIPersistentReservations` value to console log. • Added retry capability for **Get-ConsolePort**.	7 July 2023
1.3.2004052	• Fixed an error that occurred when no SSH key is specified at instance launch. • Updated to retry starting the AmazonSSMAgent Windows service on failure. • Updated to fail SysprepInstance.ps1 if BeforeSysprep.cmd fails with a non-zero exit code.	8 March 2023

Version	Details	Release date
1.3.2003975	• Fixed issue impacting Packer AMI builds where `SysprepInstance.ps1` returns a `$LastErrorCode` of 1.	24 December 2022
1.3.2003961	• Fixed issue where explicitly specified administrator passwords are overwritten with a random password on fast-launched instances. • Fixed issue where SSM Agent fails to start on smaller instance types. • Fixed an issue where the instance console log contains `RDPCERTIFICATE-THUMBPRINT: 0000000000000000000000000000` instead of a valid RDP certificate thumbprint value.	6 December 2022
1.3.2003923	• Fixes logic for finding network adapter when PnPDeviceID is empty.	9 November 2022
1.3.2003919	• Updated Get-ConsolePort to use PCI segment information. • Fixed issue where an incorrect network adapter can be selected after reboot. • Fixed start-SSM-Agent timeout logic. • Fixed backwards compatibility for Send-AdminCredentials function alias.	8 November 2022
1.3.2003857	• Prioritizes adapters with a default gateway when the primary network adapter is selected. • Extended in-memory password encryption.	3 October 2022

Version	Details	Release date
1.3.2003824	• Fixed error during `setComputerName` . • Added logic to skip Windows activation when a BYOL billing code is detected. • Added in-memory password encryption. • Fixed error during volume initialization on `m6id.4xlarge` .	30 August 2022
1.3.2003691	• Updated IMDS wait logic to make only IMDSv2 requests. • Fixed bug impacting eGPU installation.	21 June 2022
1.3.2003639	• Added network-adapter wait logic to prevent use before initialization. • Fixed minor issues.	10 May 2022
1.3.2003498	• Added telemetry. • Added shortcut to Settings UI. • Formatted PowerShell scripts. • Fixed issue with shutdown occurring before BeforeSys prep.cmd completes.	31 January 2022
1.3.2003411	Changed password generation logic to exclude passwords with low complexity.	04 August 2021
1.3.2003364	Updated Install-EgpuManager with IMDSv2 support.	07 June 2021
1.3.2003312	• Added log lines before and after `setMonitorAlwaysOn` setting. • Added AWS Nitro Enclaves package version to console log.	04 May 2021
1.3.2003284	Improved permission model by updating location for storing user data to `LocalAppData` .	23 March 2021

Version	Details	Release date
1.3.2003236	• Updated method for setting user password in `Set-AdminAccount` and `Randomize-LocalAdminPassword` . • `FixedInitializeDisks` to check whether disk is set to read only before setting it to writable.	11 February 2021
1.3.2003210	Localization fix for `install.ps1` .	7 January 2021
1.3.2003205	Security fix for `install.ps1` to update permissions on `%ProgramData%AmazonEC2-WindowsLaunchModuleScripts` directory.	28 December 2020
1.3.2003189	Added `w32tm resync` after adding routes.	4 December 2020
1.3.2003155	Updated instance type information.	25 August 2020
1.3.2003150	Added `OsCurrentBuild` and `OsReleaseId` to console output .	22 April 2020
1.3.2003040	Fixed IMDS version 1 fallback logic.	7 April 2020
1.3.2002730	Added support for IMDS V2.	3 March 2020
1.3.2002240	Fixed minor issues.	31 October 2019
1.3.2001660	Fixed automatic login issue for users without password after first time executing Sysprep.	2 July 2019
1.3.2001360	Fixed minor issues.	27 March 2019
1.3.2001220	All PowerShell scripts signed.	28 February 2019

Version	Details	Release date
1.3.2001200	Fixed issue with InitializeDisks.ps1 where running the script on a node in a Windows Server Failover Cluster would format drives on remote nodes whose drive letter matched the local drive letter.	27 February 2019
1.3.2001160	Fixed missing wallpaper in Windows 2019.	22 February 2019
1.3.2001040	• Added plugin for setting the monitor to never turn off to fix ACPI issues. • SQL Server edition and version written to console.	21 January 2019
1.3.2000930	Fix for adding routes to metadata on ipv6-enabled ENIs.	2 January 2019
1.3.2000760	• Added default configuration for RSS and Receive Queue settings for ENA devices. • Disabled hibernation during Sysprep.	5 December 2018
1.3.2000630	• Added route 169.254.169.253/32 for DNS server. • Added filter of setting Admin user. • Improvements made to instance hibernation. • Added option to schedule EC2Launch to run on every boot.	9 November 2018
1.3.20004 30.0	• Added route 169.254.169.123/32 to AMZN time service. • Added route 169.254.169.249/32 to GRID license service. • Added timeout of 25 seconds when attempting to start Systems Manager.	19 September 2018
1.3.200039.0	• Fixed improper drive lettering for EBS NVME volumes. • Added additional logging for NVME driver versions.	15 August 2018
1.3.2000080	Fixed minor issues.	

Version	Details	Release date
1.3.610	Fixed issue with redirecting output and errors to files from user data.	
1.3.590	• Added missing instances types in the wallpaper. • Fixed an issue with drive letter mapping and disk installation.	
1.3.580	• Fixed `Get-Metadata` to use the default system proxy settings for web requests. • Added a special case for NVMe in disk initialization. • Fixed minor issues.	
1.3.550	Added a `-NoShutdown` option to enable Sysprep with no shutdown.	
1.3.540	Fixed minor issues.	
1.3.530	Fixed minor issues.	
1.3.521	Fixed minor issues.	
1.3.0	• Fixed a hexadecimal length issue for computer name change. • Fixed a possible reboot loop for computer name change. • Fixed an issue in wallpaper setup.	
1.2.0	• Update to display information about installed operating system (OS) in EC2 system log. • Update to display EC2Launch and SSM Agent version in EC2 system log. • Fixed minor issues.	

Version	Details	Release date
1.1.2	• Update to display ENA driver information in EC2 system log. • Update to exclude Hyper-V from primary NIC filter logic. • Added AWS KMS server and port into registry key for KMS activation. • Improved wallpaper setup for multiple users. • Update to clear routes from persistent store. • Update to remove the z from availability zone in DNS suffix list. • Update to address an issue with the <runAsLocalSystem> tag in user data.	
1.1.1	Initial release.	

Configure a Windows instance using the EC2Config service (legacy)

> **Note**
>
> Microsoft no longer supports the Windows Server 2012 operating system. EC2Config documentation is provided for historical references only. We strongly recommend that you upgrade to the latest launch service for a supported operating system such as Windows Server 2022.
> The latest launch service for Windows Server 2022 is EC2Launch v2, which replaces both EC2Config and EC2Launch.

Windows AMIs for Windows Server 2012 R2 and earlier include an optional service, the EC2Config service (`EC2Config.exe`). EC2Config starts when the instance boots and performs tasks during startup and each time you stop or start the instance. EC2Config can also perform tasks on demand. Some of these tasks are automatically enabled, while others must be enabled manually. Although

optional, this service provides access to advanced features that aren't otherwise available. This service runs in the LocalSystem account.

> **Note**
>
> EC2Launch replaced EC2Config on Windows AMIs for Windows Server 2016 and 2019. For more information, see Configure a Windows instance using EC2Launch. The latest launch service for all supported Windows Server versions is EC2Launch v2, which replaces both EC2Config and EC2Launch.

EC2Config uses settings files to control its operation. You can update these settings files using either a graphical tool or by directly editing XML files. The service binaries and additional files are contained in the `%ProgramFiles%\Amazon\EC2ConfigService` directory.

Contents

- EC2Config tasks
- Install the latest version of EC2Config
- Stop, restart, delete, or uninstall EC2Config
- EC2Config and AWS Systems Manager
- EC2Config and Sysprep
- EC2 service properties
- EC2Config settings files
- Configure proxy settings for the EC2Config service
- EC2Config version history
- Troubleshoot issues with the EC2Config service

EC2Config tasks

EC2Config runs initial startup tasks when the instance is first started and then disables them. To run these tasks again, you must explicitly enable them prior to shutting down the instance, or by running Sysprep manually. These tasks are as follows:

- Set a random, encrypted password for the administrator account.
- Generate and install the host certificate used for Remote Desktop Connection.

- Dynamically extend the operating system partition to include any unpartitioned space.
- Execute the specified user data (and Cloud-Init, if it's installed). For more information about specifying user data, see Work with instance user data.

EC2Config performs the following tasks every time the instance starts:

- Change the host name to match the private IP address in Hex notation (this task is disabled by default and must be enabled in order to run at instance start).
- Configure the key management server (AWS KMS), check for Windows activation status, and activate Windows as necessary.
- Mount all Amazon EBS volumes and instance store volumes, and map volume names to drive letters.
- Write event log entries to the console to help with troubleshooting (this task is disabled by default and must be enabled in order to run at instance start).
- Write to the console that Windows is ready.
- Add a custom route to the primary network adapter to enable the following IP addresses when a single NIC or multiple NICs are attached: `169.254.169.250`, `169.254.169.251`, and `169.254.169.254`. These addresses are used by Windows Activation and when you access instance metadata.

> **Note**
>
> If the Windows OS is configured to use IPv4, these IPv4 link-local addresses can be used. If the Windows OS has the IPv4 network protocol stack disabled and uses IPv6 instead, add `[fd00:ec2::240]` in place of `169.254.169.250` and `169.254.169.251`. Then add `[fd00:ec2::254]` in place of `169.254.169.254`.

EC2Config performs the following task every time a user logs in:

- Display wallpaper information to the desktop background.

While the instance is running, you can request that EC2Config perform the following task on demand:

- Run Sysprep and shut down the instance so that you can create an AMI from it. For more information, see Create a standardized Amazon Machine Image (AMI) using Sysprep.

Install the latest version of EC2Config

By default, the EC2Config service is included in AMIs prior to Windows Server 2016. When the EC2Config service is updated, new Windows AMIs from AWS include the latest version of the service. However, you need to update your own Windows AMIs and instances with the latest version of EC2Config.

> **Note**
>
> EC2Launch replaces EC2Config on Windows Server 2016 and 2019. For more information, see Configure a Windows instance using EC2Launch. The latest launch service for all supported Windows Server versions is EC2Launch v2, which replaces both EC2Config and EC2Launch.

For information about how to receive notifications for EC2Config updates, see Subscribe to EC2Config service notifications. For information about the changes in each version, see the EC2Config version history.

Before you begin

- Verify that you have .NET framework 3.5 SP1 or greater.
- By default, Setup replaces your settings files with default settings files during installation and restarts the EC2Config service when the installation is completed. If you changed EC2Config service settings, copy the `config.xml` file from the `%Program Files%\Amazon\Ec2ConfigService\Settings` directory. After you update the EC2Config service, you can restore this file to retain your configuration changes.
- If your version of EC2Config is earlier than version 2.1.19 and you are installing version 2.2.12 or earlier, you must first install version 2.1.19. To install version 2.1.19, download EC2Install_2.1.19.zip, unzip the file, and then run `EC2Install.exe`.

> **Note**
>
> If your version of EC2Config is earlier than version 2.1.19 and you are installing version 2.3.313 or later, you can install it directly without installing version 2.1.19 first.

Verify the EC2Config version

Use the following procedure to verify the version of EC2Config that is installed on your instances.

To verify the installed version of EC2Config

1. Launch an instance from your AMI and connect to it.
2. In Control Panel, select **Programs and Features**.
3. In the list of installed programs, look for `Ec2ConfigService`. Its version number appears in the **Version** column.

Update EC2Config

Use the following procedure to download and install the latest version of EC2Config on your instances.

To download and install the latest version of EC2Config

1. Download and unzip the EC2Config installer.
2. Run `EC2Install.exe`. For a complete list of options, run `EC2Install` with the `/?` option. By default, setup displays prompts. To run the command with no prompts, use the `/quiet` option.

   > **Important**
   >
   > To keep the custom settings from the `config.xml` file that you saved, run `EC2Install` with the `/norestart` option, restore your settings, and then restart the EC2Config service manually.

3. If you are running EC2Config version 4.0 or later, you must restart SSM Agent on the instance from the Microsoft Services snap-in.

> **Note**
>
> The updated EC2Config version information will not appear in the instance System Log or Trusted Advisor check until you reboot or stop and start your instance.

To download and install the latest version of EC2Config using PowerShell

To download, unzip, and install the latest version of EC2Config using PowerShell, run the following commands from a PowerShell window:

```
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Config/EC2Install.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\" + $(Split-Path -Path $Url -Leaf)
$ExtractPath = "$env:USERPROFILE\Desktop\"
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$ExtractShell = New-Object -ComObject Shell.Application
$ExtractFiles = $ExtractShell.Namespace($DownloadZipFile).Items()
$ExtractShell.NameSpace($ExtractPath).CopyHere($ExtractFiles)
Start-Process $ExtractPath
Start-Process `
    -FilePath $env:USERPROFILE\Desktop\EC2Install.exe `
    -ArgumentList "/S"
```

Verify the installation by checking `C:\Program Files\Amazon\` for the `Ec2ConfigService` directory.

Stop, restart, delete, or uninstall EC2Config

You can manage the EC2Config service just as you would any other service.

To apply updated settings to your instance, you can stop and restart the service. If you're manually installing EC2Config, you must stop the service first.

To stop the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Stop**.

To restart the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Restart**.

If you don't need to update the configuration settings, create your own AMI, or use AWS Systems Manager, you can delete and uninstall the service. Deleting a service removes its registry subkey. Uninstalling a service removes the files, the registry subkey, and any shortcuts to the service.

To delete the EC2Config service

1. Start a command prompt window.
2. Run the following command:

```
sc delete ec2config
```

To uninstall EC2Config

1. Launch and connect to your Windows instance.
2. On the **Start** menu, click **Control Panel**.
3. Double-click **Programs and Features**.
4. On the list of programs, select **EC2ConfigService**, and click **Uninstall** .

EC2Config and AWS Systems Manager

The EC2Config service processes Systems Manager requests on instances created from AMIs for versions of Windows Server prior to Windows Server 2016 that were published before November 2016.

Instances created from AMIs for versions of Windows Server prior to Windows Server 2016 that were published after November 2016 include the EC2Config service *and* SSM Agent. EC2Config performs all of the tasks described earlier, and SSM Agent processes requests for Systems Manager capabilities like Run Command and State Manager.

You can use Run Command to upgrade your existing instances to use to the latest version of the EC2Config service and SSM Agent. For more information, see Update SSM Agent by using Run Command in the *AWS Systems Manager User Guide*.

EC2Config and Sysprep

The EC2Config service runs Sysprep, a Microsoft tool that enables you to create a customized Windows AMI that can be reused. When EC2Config calls Sysprep, it uses the files in `%ProgramFiles%\Amazon\EC2ConfigService\Settings` to determine which operations to perform. You can edit these files indirectly using the **EC2 Service Properties** dialog box, or directly using an XML editor or a text editor. However, there are some advanced settings that aren't available in the **Ec2 Service Properties** dialog box, so you must edit those entries directly.

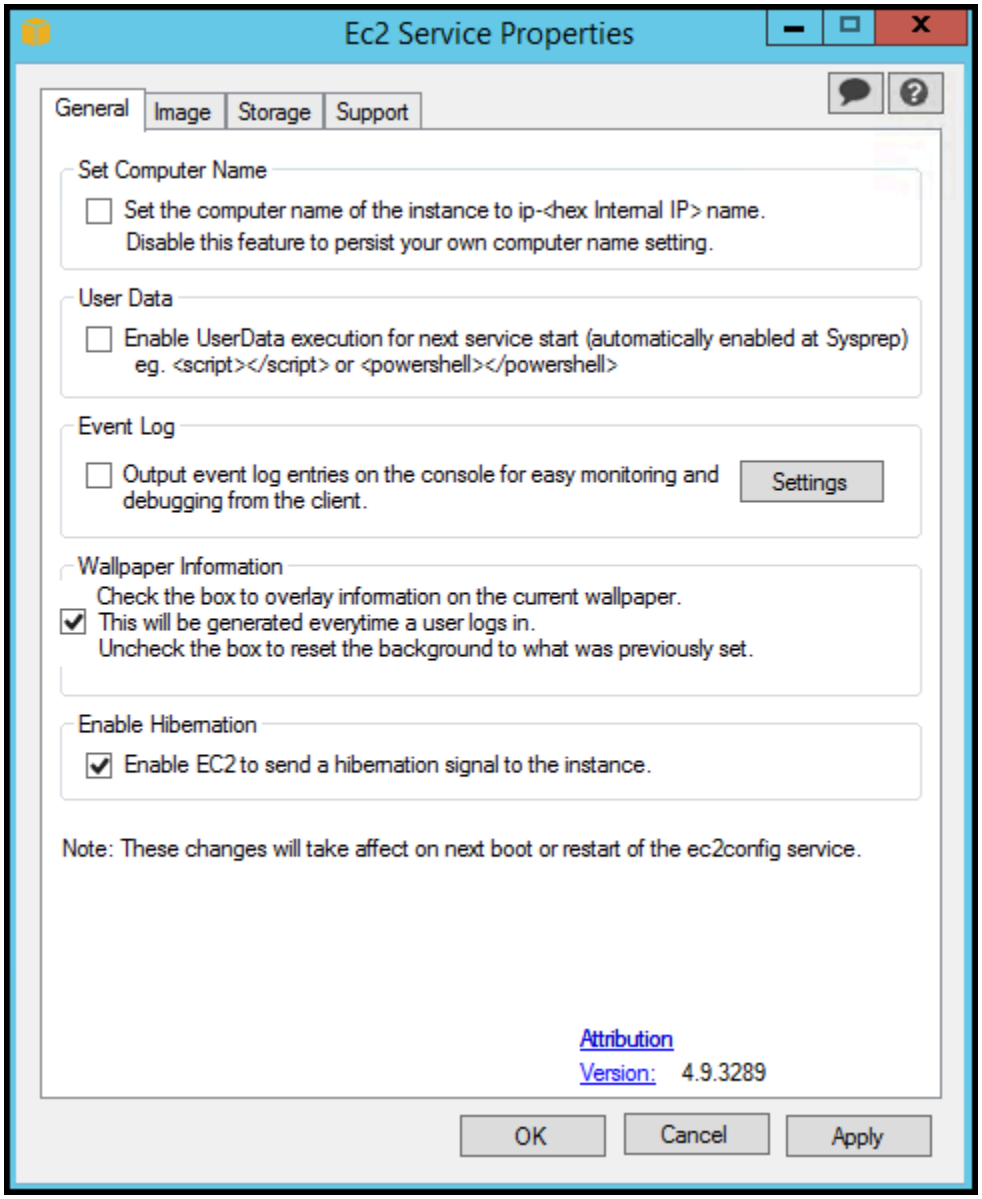
If you create an AMI from an instance after updating its settings, the new settings are applied to any instance that's launched from the new AMI. For information about creating an AMI, see Create a custom Windows AMI.

EC2 service properties

The following procedure describes how to use the **Ec2 Service Properties** dialog box to enable or disable settings.

To change settings using the Ec2 Service Properties dialog box

1. Launch and connect to your Windows instance.
2. From the **Start** menu, click **All Programs**, and then click **EC2ConfigService Settings**.



3. On the **General** tab of the **EC2 Service Properties** dialog box, you can enable or disable the following settings.

   Set Computer Name

   If this setting is enabled (it is disabled by default), the host name is compared to the current internal IP address at each boot; if the host name and internal IP address do not match, the host name is reset to contain the internal IP address and then the system reboots to pick up the new host name. To set your own host name, or to prevent your existing host name from being modified, do not enable this setting.

User Data

User data execution enables you to specify scripts in the instance metadata. By default, these scripts are run during the initial launch. You can also configure them to run the next time you reboot or start the instance, or every time you reboot or start the instance.

If you have a large script, we recommend that you use user data to download the script, and then run it.

For more information, see User data execution.

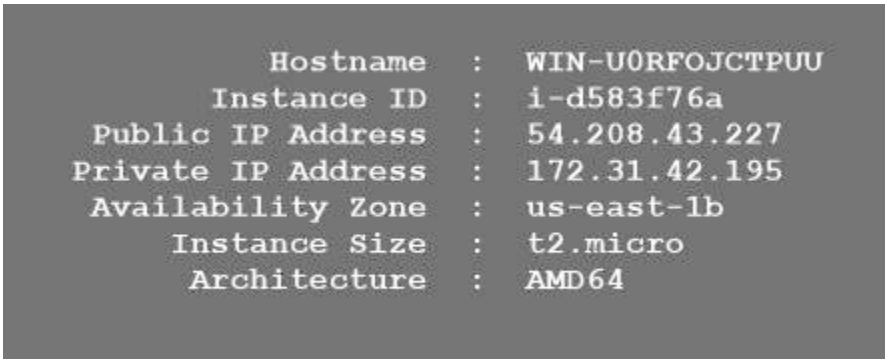
Event Log

Use this setting to display event log entries on the console during boot for easy monitoring and debugging.

Click **Settings** to specify filters for the log entries sent to the console. The default filter sends the three most recent error entries from the system event log to the console.

Wallpaper Information

Use this setting to display system information on the desktop background. The following is an example of the information displayed on the desktop background.

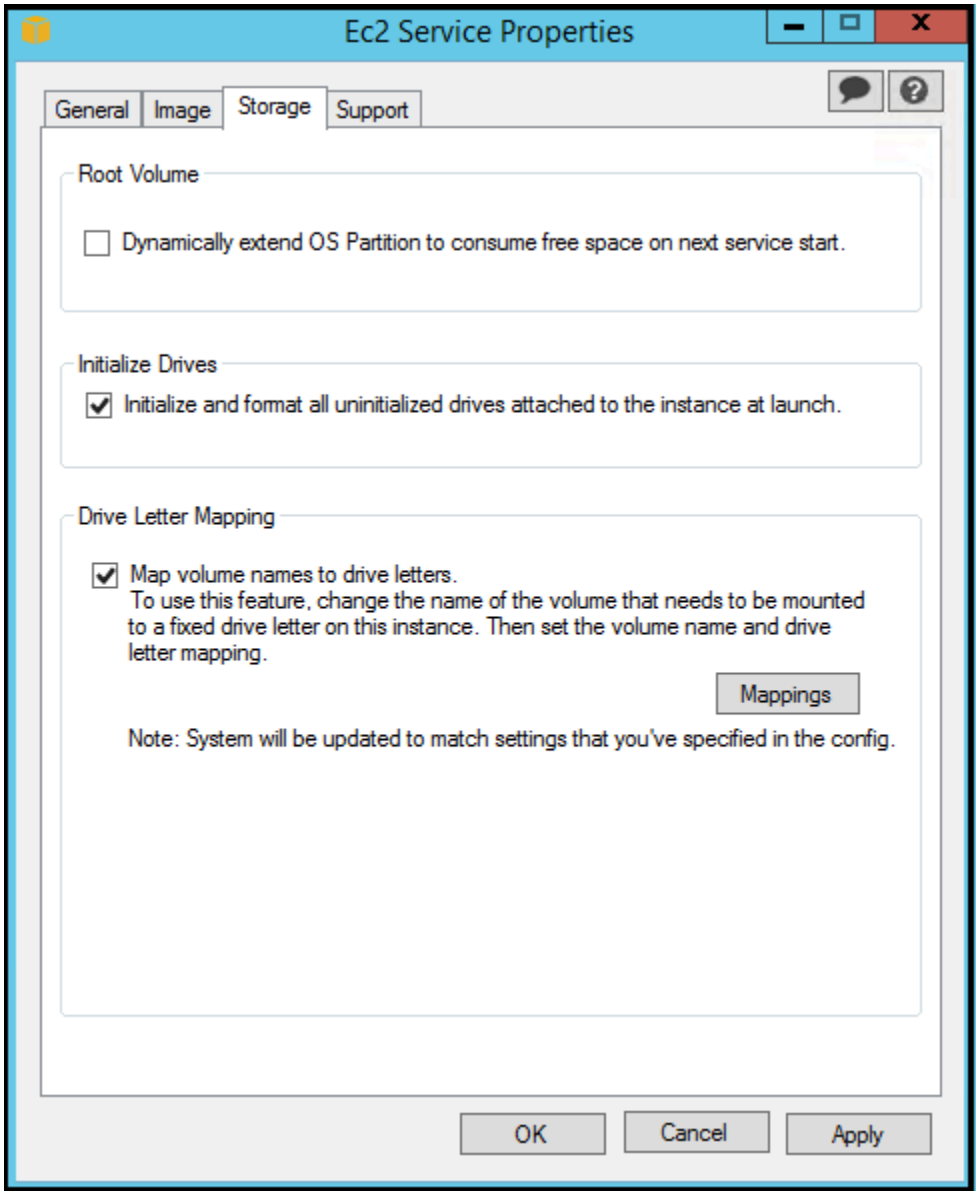


The information displayed on the desktop background is controlled by the settings file `EC2ConfigService\Settings\WallpaperSettings.xml`.

Enable Hibernation

Use this setting to allow EC2 to signal the operating system to perform hibernation.

4. Click the **Storage** tab. You can enable or disable the following settings.



Root Volume

This setting dynamically extends Disk 0/Volume 0 to include any unpartitioned space. This can be useful when the instance is booted from a root device volume that has a custom size.

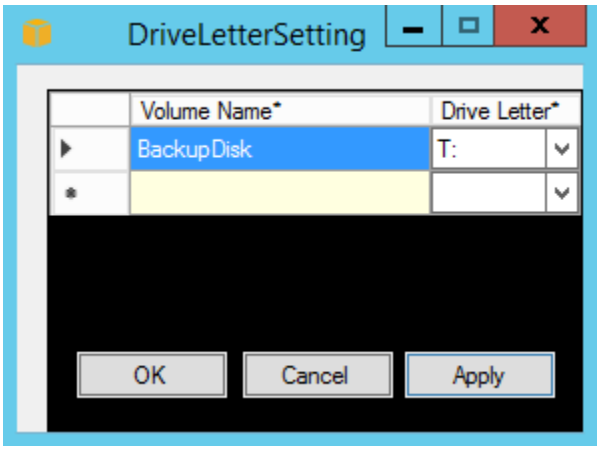
Initialize Drives

This setting formats and mounts all volumes attached to the instance during start.

Drive Letter Mapping

The system maps the volumes attached to an instance to drive letters. For Amazon EBS volumes, the default is to assign drive letters going from D: to Z:. For instance store volumes, the default depends on the driver. AWS PV drivers and Citrix PV drivers assign instance store volumes drive letters going from Z: to A:. Red Hat drivers assign instance store volumes drive letters going from D: to Z:.

To choose the drive letters for your volumes, click **Mappings**. In the **DriveLetterSetting** dialog box, specify the **Volume Name** and **Drive Letter** values for each volume, click **Apply**, and then click **OK**. We recommend that you select drive letters that avoid conflicts with drive letters that are likely to be in use, such as drive letters in the middle of the alphabet.



After you specify a drive letter mapping and attach a volume with same label as one of the volume names that you specified, EC2Config automatically assigns your specified drive letter to that volume. However, the drive letter mapping fails if the drive letter is already in use. Note that EC2Config doesn't change the drive letters of volumes that were already mounted when you specified the drive letter mapping.

5. To save your settings and continue working on them later, click **OK** to close the **EC2 Service Properties** dialog box. If you have finished customizing your instance and want to create an AMI from that instance, see Create a standardized Amazon Machine Image (AMI) using Sysprep.

EC2Config settings files

The settings files control the operation of the EC2Config service. These files are located in the `C:\Program Files\Amazon\Ec2ConfigService\Settings` directory:

- `ActivationSettings.xml`—Controls product activation using a key management server (AWS KMS).
- `AWS.EC2.Windows.CloudWatch.json`—Controls which performance counters to send to CloudWatch and which logs to send to CloudWatch Logs.
- `BundleConfig.xml`—Controls how EC2Config prepares an instance store-backed instance for AMI creation.
- `Config.xml`—Controls the primary settings.
- `DriveLetterConfig.xml`—Controls drive letter mappings.
- `EventLogConfig.xml`—Controls the event log information that's displayed on the console while the instance is booting.
- `WallpaperSettings.xml`—Controls the information that's displayed on the desktop background.

ActivationSettings.xml

This file contains settings that control product activation. When Windows boots, the EC2Config service checks whether Windows is already activated. If Windows is not already activated, it attempts to activate Windows by searching for the specified AWS KMS server.

- `SetAutodiscover`—Indicates whether to detect a AWS KMS automatically.
- `TargetKMSServer`—Stores the private IP address of a AWS KMS. The AWS KMS must be in the same Region as your instance.
- `DiscoverFromZone`—Discovers the AWS KMS server from the specified DNS zone.
- `ReadFromUserData`—Gets the AWS KMS server from UserData.
- `LegacySearchZones`—Discovers the AWS KMS server from the specified DNS zone.
- `DoActivate`—Attempts activation using the specified settings in the section. This value can be `true` or `false`.
- `LogResultToConsole`—Displays the result to the console.

BundleConfig.xml

This file contains settings that control how EC2Config prepares an instance for AMI creation.

- `AutoSysprep`—Indicates whether to use Sysprep automatically. Change the value to `Yes` to use Sysprep.
- `SetRDPCertificate`—Sets a self-signed certificate to the Remote Desktop server. This enables you to securely RDP into the instances. Change the value to `Yes` if the new instances should have the certificate.

  This setting is not used with Windows Server 2008 or Windows Server 2012 instances because they can generate their own certificates.
- `SetPasswordAfterSysprep`—Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value of this setting to `No` if the new instances should not be set to a random encrypted password.

Config.xml

Plug-ins

- `Ec2SetPassword`—Generates a random encrypted password each time you launch an instance. This feature is disabled by default after the first launch so that reboots of this instance don't change a password set by the user. Change this setting to `Enabled` to continue to generate passwords each time you launch an instance.

  This setting is important if you are planning to create an AMI from your instance.
- `Ec2SetComputerName`—Sets the host name of the instance to a unique name based on the IP address of the instance and reboots the instance. To set your own host name, or prevent your existing host name from being modified, you must disable this setting.
- `Ec2InitializeDrives`—Initializes and formats all volumes during startup. This feature is enabled by default.
- `Ec2EventLog`—Displays event log entries in the console. By default, the three most recent error entries from the system event log are displayed. To specify the event log entries to display, edit the `EventLogConfig.xml` file located in the `EC2ConfigService\Settings` directory. For information about the settings in this file, see Eventlog Key in the MSDN Library.
- `Ec2ConfigureRDP`—Sets up a self-signed certificate on the instance, so users can securely access the instance using Remote Desktop. This feature is disabled on Windows Server 2008 and Windows Server 2012 instances because they can generate their own certificates.

- `Ec2OutputRDPCert`—Displays the Remote Desktop certificate information to the console so that the user can verify it against the thumbprint.
- `Ec2SetDriveLetter`—Sets the drive letters of the mounted volumes based on user-defined settings. By default, when an Amazon EBS volume is attached to an instance, it can be mounted using the drive letter on the instance. To specify your drive letter mappings, edit the `DriveLetterConfig.xml` file located in the `EC2ConfigService\Settings` directory.
- `Ec2WindowsActivate`—The plug-in handles Windows activation. It checks to see if Windows is activated. If not, it updates the AWS KMS client settings, and then activates Windows.

  To modify the AWS KMS settings, edit the `ActivationSettings.xml` file located in the `EC2ConfigService\Settings` directory.
- `Ec2DynamicBootVolumeSize`—Extends Disk 0/Volume 0 to include any unpartitioned space.
- `Ec2HandleUserData`—Creates and runs scripts created by the user on the first launch of an instance after Sysprep is run. Commands wrapped in script tags are saved to a batch file, and commands wrapped in PowerShell tags are saved to a .ps1 file (corresponds to the User Data check box on the Ec2 Service Properties dialog box).
- `Ec2ElasticGpuSetup`—Installs the Elastic GPU software package if the instance is associated with an elastic GPU.
- `Ec2FeatureLogging`—Sends Windows feature installation and corresponding service status to the console. Supported only for the Microsoft Hyper-V feature and corresponding vmms service.

Global Settings

- `ManageShutdown`—Ensures that instances launched from instance store-backed AMIs do not terminate while running Sysprep.
- `SetDnsSuffixList`—Sets the DNS suffix of the network adapter for Amazon EC2. This allows DNS resolution of servers running in Amazon EC2 without providing the fully qualified domain name.

> **Note**
>
> This adds a DNS suffix lookup for the following domain and configures other standard suffixes. For more information about how launch agents set DNS suffixes, see Configure DNS Suffix.

```
region.ec2-utilities.amazonaws.com
```

- `WaitForMetaDataAvailable`—Ensures that the EC2Config service will wait for metadata to be accessible and the network available before continuing with the boot. This check ensures that EC2Config can obtain information from metadata for activation and other plug-ins.
- `ShouldAddRoutes`—Adds a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.
- `RemoveCredentialsfromSyspreponStartup`—Removes the administrator password from `Sysprep.xml` the next time the service starts. To ensure that this password persists, edit this setting.

DriveLetterConfig.xml

This file contains settings that control drive letter mappings. By default, a volume can be mapped to any available drive letter. You can mount a volume to a particular drive letter as follows.

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- `VolumeName`—The volume label. For example, *`My Volume`*. To specify a mapping for an instance storage volume, use the label `Temporary Storage X`, where `X` is a number from 0 to 25.
- `DriveLetter`—The drive letter. For example, *`M:`*. The mapping fails if the drive letter is already in use.

EventLogConfig.xml

This file contains settings that control the event log information that's displayed on the console while the instance is booting. By default, we display the three most recent error entries from the System event log.

- `Category`—The event log key to monitor.
- `ErrorType`—The event type (for example, `Error`, `Warning`, `Information`.)
- `NumEntries`—The number of events stored for this category.
- `LastMessageTime`—To prevent the same message from being pushed repeatedly, the service updates this value every time it pushes a message.
- `AppName`—The event source or application that logged the event.

WallpaperSettings.xml

This file contains settings that control the information that's displayed on the desktop background. The following information is displayed by default.

- `Hostname`—Displays the computer name.
- `Instance ID`—Displays the ID of the instance.
- `Public IP Address`—Displays the public IP address of the instance.
- `Private IP Address`—Displays the private IP address of the instance.
- `Availability Zone`—Displays the Availability Zone in which the instance is running.
- `Instance Size`—Displays the type of instance.
- `Architecture`—Displays the setting of the `PROCESSOR_ARCHITECTURE` environment variable.

You can remove any of the information that's displayed by default by deleting its entry. You can add additional instance metadata to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

You can add additional System environment variables to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

This file contains settings that control how EC2Config initializes drives.

By default, EC2Config initialize drives that were not brought online with the operating system. You can customize the plugin as follows.

```
<InitializeDrivesSettings>
    <SettingsGroup>setting</SettingsGroup>
</InitializeDrivesSettings>
```

Use a settings group to specify how you want to initialize drives:

FormatWithTRIM

Enables the TRIM command when formatting drives. After a drive has been formatted and initialized, the system restores TRIM configuration.

Starting with EC2Config version 3.18, the TRIM command is disabled during the disk format operation by default. This improves formatting times. Use this setting to enable TRIM during the disk format operation for EC2Config version 3.18 and later.

FormatWithoutTRIM

Disables the TRIM command when formatting drives and improves formatting times in Windows. After a drive has been formatted and initialized, the system restores TRIM configuration.

DisableInitializeDrives

Disables formatting for new drives. Use this setting to initialize drives manually.

Configure proxy settings for the EC2Config service

You can configure the EC2Config service to communicate through a proxy using one of the following methods: the AWS SDK for .NET, the `system.net` element, or Microsoft Group Policy

and Internet Explorer. Using the AWS SDK for .NET is the preferred method because you can specify sign-in credentials.

Methods

- Configure proxy settings using the AWS SDK for .NET (Preferred)
- Configure proxy settings using the system.net element
- Configure proxy settings using Microsoft Group Policy and Microsoft Internet Explorer

Configure proxy settings using the AWS SDK for .NET (Preferred)

You can configure proxy settings for the EC2Config service by specifying the `proxy` element in the `Ec2Config.exe.config` file. For more information, see Configuration Files Reference for AWS SDK for .NET.

To specify the proxy element in Ec2Config.exe.config

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory: `%ProgramFiles%\Amazon\Ec2ConfigService`.
2. Add the following `aws` element to the `configSections`. Do not add this to any existing `sectionGroups`.

   For EC2Config versions 3.17 or earlier

   ```
   <configSections>
      <section name="aws" type="Amazon.AWSSection, AWSSDK"/>
   </configSections>
   ```

   For EC2Config versions 3.18 or later

   ```
   <configSections>
        <section name="aws" type="Amazon.AWSSection, AWSSDK.Core"/>
   </configSections>
   ```

3. Add the following `aws` element to the `Ec2Config.exe.config` file.

   ```
   <aws>
      <proxy
        host="string value"
   ```

```
     port="string value"
     username="string value"
     password="string value" />
</aws>
```

4. Save your changes.

Configure proxy settings using the system.net element

You can specify proxy settings in a `system.net` element in the `Ec2Config.exe.config` file. For more information, see defaultProxy Element (Network Settings) on MSDN.

To specify the system.net element in Ec2Config.exe.config

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory: `%ProgramFiles%\Amazon\Ec2ConfigService`.
2. Add a `defaultProxy` entry to `system.net`. For more information, see defaultProxy Element (Network Settings) on MSDN.

   For example, the following configuration routes all traffic to use the proxy that is currently configured for Internet Explorer, with the exception of the metadata and licensing traffic, which will bypass the proxy.

```
<defaultProxy>
    <proxy usesystemdefault="true" />
    <bypasslist>
        <add address="169.254.169.250" />
        <add address="169.254.169.251" />
        <add address="169.254.169.254" />
        <add address="[fd00:ec2::250]" />
        <add address="[fd00:ec2::254]" />
    </bypasslist>
</defaultProxy>
```

3. Save your changes.

Configure proxy settings using Microsoft Group Policy and Microsoft Internet Explorer

The EC2Config service runs under the Local System user account. You can specify instance-wide proxy settings for this account in Internet Explorer after you change Group Policy settings on the instance.

To configure proxy settings using Group Policy and Internet Explorer

1. On an instance where you want the EC2Config service to communicate through a proxy, open a Command prompt as an Administrator, type **`gpedit.msc`**, and press Enter.
2. In the Local Group Policy Editor, under **Local Computer Policy**, choose **Computer Configuration**, **Administrative Templates**, **Windows Components**, **Internet Explorer**.
3. In the right-pane, choose **Make proxy settings per-machine (rather than per-user)** and then choose **Edit policy setting**.
4. Choose **Enabled**, and then choose **Apply**.
5. Open Internet Explorer, and then choose the **Tools** button.
6. Choose **Internet Option**, and then choose the **Connections** tab.
7. Choose **LAN settings**.
8. Under **Proxy server**, choose the **Use a proxy server for your LAN** option.
9. Specify address and port information and then choose **OK**.

EC2Config version history

Windows AMIs prior to Windows Server 2016 include an optional service called the EC2Config service (`EC2Config.exe`). EC2Config starts when the instance boots and performs tasks during startup and each time you stop or start the instance. For information about the EC2Config versions included in the Windows AMIs, see AWS Windows AMIs.

You can receive notifications when new versions of the EC2Config service are released. For more information, see Subscribe to EC2Config service notifications.

The following table describes the released versions of EC2Config. For information about the updates for SSM Agent, see Systems Manager SSM Agent Release Notes.

Version	Details	Release date
4.9.5554	• Limit domain name devolution based on registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`. • New version of SSM Agent `3.2.1630.0`.	4 October 2023
4.9.5467	• Added retry capability for discovering console port. • New version of SSM Agent `3.1.2282.0`.	1 August 2023
4.9.5288	• Updated AWS Core SDK to version `3.7.103.23`. • Fixed issue where the `AWS-UpdateEC2Config` SSM document fails to update `EC2Config` on instances enabled with only IMDSv2. • New version of SSM Agent `3.1.2144.0`.	8 March 2023
4.9.5231	• New version of SSM Agent 3.1.1927.0.	14 February 2023
4.9.5103	• Fixed issue where ephemeral volumes are incorrectly identified on r5d and i4i instance families. • New version of SSM Agent 3.1.1856.0.	5 December 2022
4.9.5064	• Updated to use PCI segment information to select the console port. • Signed PowerShell scripts and added copyright headers. • Fixed primary network adapter selection logic. •	16 November 2022

Version	Details	Release date
	New version of SSM Agent 3.1.1732.0.	
4.9.4588	• Updated IMDS wait logic to make only IMDSv2 requests. • Added libec2launch.dll launch-agent shared library. • New version of SSM Agent 3.1.1188.0.	31 May 2022
4.9.4556	• Added wait logic to ensure full initialization of NIC before use. • New version of Log4Net 2.0.14.0 picks up security patch. • New version of SSM Agent 3.1.1045.0 picks up security patch.	1 March 2022
4.9.4536	• Fixed issue where userdata crashes when the Temp folder is missing. • New version of SSM Agent 3.1.804.0.	31 January 2022
4.9.4508	• Fixed issue to correctly compute diskpart script path. • New version of SSM Agent 3.1.338.0.	6 October 2021
4.9.4500	• Updated `Install-EgpuManagerConfig` with IMDS v2 support. • Updated web links to use https. • New version of SSM Agent 3.1.282.0	7 September 2021

Version	Details	Release date
4.9.4419	• Fixed IMDS version 1 fallback logic • Updated all usage of Windows temp directory to EC2Config temp directory • New version of SSM Agent 3.0.1124.0	2 June 2021
4.9.4381	• Added support for SSM document schema version 2.2 in EC2ConfigUpdater • Added AWS Nitro Enclaves package version to console log • New version of SSM Agent 3.0.529.0	4 May 2021
4.9.4326	• Removed all links in the settings UI • This is the last EC2Config version that supports Windows Server 2008.	3 March 2021
4.9.4279	• Fixed security issue related to `Ec2ConfigMonitor` scheduled task • Fixed drive letter mapping issue and incorrect ephemeral disk count • Added `OsCurrentBuild` and `OsReleaseId` to console output • New version of SSM Agent 2.3.871.0	11 December 2020
4.9.4222	• Fixed IMDS version 1 fallback logic • New version of SSM Agent 2.3.842.0	7 April 2020
4.9.4122	• Added support for IMDS v2 • New version of SSM Agent 2.3.814.0	4 March 2020

Version	Details	Release date
4.9.3865	• Fixed issue detecting COM port for Windows Server 2008 R2 on metal instances • New version of SSM Agent 2.3.722.0	31 October 2019
4.9.3519	• New version of SSM Agent 2.3.634.0	18 June 2019
4.9.3429	• New version of SSM Agent 2.3.542.0	25 April 2019
4.9.3289	• New version of SSM Agent 2.3.444.0	11 February 2019
4.9.3270	• Added plugin for setting the monitor to never turn off to fix ACPI issues • SQL Server edition and version written to console • New version of SSM Agent 2.3.415.0	22 January 2019
4.9.3230	• Drive Letter Mapping description updated to better align to functionality • New version of SSM Agent 2.3.372.0	10 January 2019
4.9.3160	• Increased wait time for primary NIC • Added default configuration for RSS and Receive Queue settings for ENA devices • Disabled hibernation during Sysprep • New version of SSM Agent 2.3.344.0 • Upgraded AWS SDK to 3.3.29.13	15 December 2018
4.9.3067	• Improvements made to instance hibernation • New version of SSM Agent 2.3.235.0	8 November 2018
4.9.3034	• Added route 169.254.169.253/32 for DNS server • New version of SSM Agent 2.3.193.0	24 October 2018
4.9.2986	• Added signing for all EC2Config related binaries • New version of SSM Agent 2.3.136.0	11 October 2018

Version	Details	Release date
4.9.2953	New version of SSM Agent (2.3.117.0)	2 October 2018
4.9.2926	New version of SSM Agent (2.3.68.0)	18 September 2018
4.9.2905	• New version of SSM Agent (2.3.50.0) • Added route 169.254.169.123/32 to AMZN time service • Added route 169.254.169.249/32 to GRID license service • Fixed an issue causing EBS NVMe volumes to be marked as ephemeral	17 September 2018
4.9.2854	New version of SSM Agent (2.3.13.0)	17 August 2018
4.9.2831	New version of SSM Agent (2.2.916.0)	7 August 2018
4.9.2818	New version of SSM Agent (2.2.902.0)	31 July 2018
4.9.2756	New version of SSM Agent (2.2.800.0)	27 June 2018
4.9.2688	New version of SSM Agent (2.2.607.0)	25 May 2018
4.9.2660	New version of SSM Agent (2.2.546.0)	11 May 2018
4.9.2644	New version of SSM Agent (2.2.493.0)	26 April 2018
4.9.2586	New version of SSM Agent (2.2.392.0)	28 March 2018

Version	Details	Release date
4.9.2565	• New version of SSM Agent (2.2.355.0) • Fixed an issue on M5 and C5 instances (unable to find PV drivers) • Add console logging for instance type, newest PV drivers, and NVMe drivers	13 March 2018
4.9.2549	New version of SSM Agent (2.2.325.0)	8 March 2018
4.9.2461	New version of SSM Agent (2.2.257.0)	15 February 2018
4.9.2439	New version of SSM Agent (2.2.191.0)	6 February 2018
4.9.2400	New version of SSM Agent (2.2.160.0)	16 January 2018
4.9.2327	• New version of SSM Agent (2.2.120.0) • Added COM port discovery on Amazon EC2 bare metal instances • Added Hyper-V status logging on Amazon EC2 bare metal instances	2 January 2018
4.9.2294	New version of SSM Agent (2.2.103.0)	4 December 2017
4.9.2262	New version of SSM Agent (2.2.93.0)	15 November 2017
4.9.2246	New version of SSM Agent (2.2.82.0)	11 November 2017

Version	Details	Release date
4.9.2218	New version of SSM Agent (2.2.64.0)	29 October 2017
4.9.2212	New version of SSM Agent (2.2.58.0)	23 October 2017
4.9.2203	New version of SSM Agent (2.2.45.0)	19 October 2017
4.9.2188	New version of SSM Agent (2.2.30.0)	10 October 2017
4.9.2180	• New version of SSM Agent (2.2.24.0) • Added the Elastic GPU plugin for GPU instances	5 October 2017
4.9.2143	New version of SSM Agent (2.2.16.0)	1 October 2017
4.9.2140	New version of SSM Agent (2.1.10.0)	
4.9.2130	New version of SSM Agent (2.1.4.0)	
4.9.2106	New version of SSM Agent (2.0.952.0)	
4.9.2061	New version of SSM Agent (2.0.922.0)	
4.9.2047	New version of SSM Agent (2.0.913.0)	
4.9.2031	New version of SSM Agent (2.0.902.0)	
4.9.2016	• New version of SSM Agent (2.0.879.0) • Fixed the CloudWatch Logs directory path for Windows Server 2003	

Version	Details	Release date
4.9.1981	• New version of SSM Agent (2.0.847.0) • Fixed the issue with `important.txt` being generated in EBS volumes.	
4.9.1964	New version of SSM Agent (2.0.842.0)	
4.9.1951	• New version of SSM Agent (2.0.834.0) • Fixed the issue with drive letter not being mapped from Z: for ephemeral drives.	
4.9.1925	• New version of SSM Agent (2.0.822.0) • [Bug] This version is not a valid update target from SSM Agent v4.9.1775.	
4.9.1900	New version of SSM Agent (2.0.805.0)	
4.9.1876	• New version of SSM Agent (2.0.796.0) • Fixed an issue with output/error redirection for admin us erdata execution.	
4.9.1863	• New version of SSM Agent (2.0.790.0) • Fixed problems with attaching multiple EBS volumes to an Amazon EC2 instance. • Improved CloudWatch to take a configuration path, keeping the backwards compatibility.	
4.9.1791	New version of SSM Agent (2.0.767.0)	

Version	Details	Release date
4.9.1775	New version of SSM Agent (2.0.761.0)	
4.9.1752	New version of SSM Agent (2.0.755.0)	
4.9.1711	New version of SSM Agent (2.0.730.0)	
4.8.1676	New version of SSM Agent (2.0.716.0)	
4.7.1631	New version of SSM Agent (2.0.682.0)	
4.6.1579	• New version of SSM Agent (2.0.672.0) • Fixed agent update issue with v4.3, v4.4, and v4.5	
4.5.1534	New version of SSM Agent (2.0.645.1)	
4.4.1503	New version of SSM Agent (2.0.633.0)	
4.3.1472	New version of SSM Agent (2.0.617.1)	
4.2.1442	New version of SSM Agent (2.0.599.0)	
4.1.1378	New version of SSM Agent (2.0.558.0)	

Version	Details	Release date
4.0.1343	• Run Command, State Manager, the CloudWatch agent, and domain join support have been moved into another agent called SSM Agent. SSM Agent will be installed as part of the EC2Config upgrade. For more information, see EC2Config and AWS Systems Manager. • If you have a proxy set up in EC2Config, you will need to update your proxy settings for SSM Agent before upgrading. If you do not update the proxy settings, you will not be able to use Run Command to manage your instances. To avoid this, see the following information before updating to the newer version: Installing and Configuring SSM Agent on Windows Instances in the *AWS Systems Manager User Guide*. • If you previously enabled CloudWatch integration on your instances by using a local configuration file (`AWS.EC2.Windows.CloudWatch.json`), you will need to configure the file to work with SSM Agent.	
3.19.1153	• Re-enabled activation plugin for instances with old AWS KMS configuration. Skip activation for BYOL users. • Change default TRIM behavior to be disabled during disk format operation and added FormatWithTRIM for overriding InitializeDisks plugin with userdata.	
3.18.1118	• Fix to reliably add routes to the primary network adapter. • Updates to improve support for AWS services.	

Version	Details	Release date
3.17.1032	• Fixes duplicate system logs appearing when filters set to same category. • Fixes to prevent from hanging during disk initialization.	
3.16.930	Added support to log "Window is Ready to use" event to Windows Event Log on start.	
3.15.880	Fix to allow uploading Systems Manager Run Command output to S3 bucket names with '.' character.	
3.14.786	Added support to override InitializeDisks plugin settings. For example: To speed up SSD disk initialize, you can temporarily disable TRIM by specifying this in userdata: <InitializeDrivesSettings><SettingsGroup>FormatWithout TRIM</SettingsGroup></InitializeDrivesSettings	
3.13.727	Systems Manager Run Command - Fixes to process commands reliably after windows reboot.	
3.12.649	• Fix to gracefully handle reboot when running commands/scripts. • Fix to reliably cancel running commands. • Add support for (optionally) uploading MSI logs to S3 when installing applications via Systems Manager Run Command.	

Version	Details	Release date
3.11.521	• Fixes to enable RDP thumbprint generation for Windows Server 2003. • Fixes to include timezone and UTC offset in the EC2Config log lines. • Systems Manager support to run Run Command commands in parallel. • Roll back previous change to bring partitioned disks online.	
3.10.442	• Fix Systems Manager configuration failures when installing MSI applications. • Fix to reliably bring storage disks online. • Updates to improve support for AWS services.	
3.9.359	• Fix in post Sysprep script to leave the configuration of windows update in a default state. • Fix the password generation plugin to improve the reliability in getting GPO password policy settings. • Restrict EC2Config/SSM log folder permissions to the local Administrators group. • Updates to improve support for AWS services.	

Version	Details	Release date
3.8.294	• Fixed an issue with CloudWatch that prevented logs from getting uploaded when not on primary drive. • Improved the disk initialization process by adding retry lo gic. • Added improved error handling when the SetPassword plugin occasionally failed during AMI creation. • Updates to improve support for AWS services.	
3.7.308	• Improvements to the ec2config-cli utility for config testing and troubleshooting within instance. • Avoid adding static routes for AWS KMS and meta-data service on an OpenVPN adapter. • Fixed an issue where user-data execution was not honoring the "persist" tag. • Improved error handling when logging to the EC2 console is not available. • Updates to improve support for AWS services.	
3.6.269	• Windows activation reliability fix to first use link local ad dress 169.254.0.250/251 for activating windows via AWS KMS • Improved proxy handling for Systems Manager, Windows Activation and Domain Join scenarios • Fixed an issue where duplicate lines of user accounts were added to the Sysprep answer file	

Version	Details	Release date
3.5.228	• Addressed a scenario where the CloudWatch plugin may consume excessive CPU and memory reading Windows Event Logs • Added a link to the CloudWatch configuration documentation in the EC2Config Settings UI	
3.4.212	• Fixes to EC2Config when used in combination with VM-Import. • Fixed service naming issue in the WiX installer.	
3.3.174	• Improved exception handling for Systems Manager and domain join failures. • Change to support Systems Manager SSM schema versioning. • Fixed formatting ephemeral disks on Win2K3. • Change to support configuring disk size greater than 2TB. • Reduced virtual memory usage by setting GC mode to default. • Support for downloading artifacts from UNC path in `aws:psModule` and `aws:application` plugin. • Improved logging for Windows activation plugin.	

Version	Details	Release date
3.2.97	• Performance improvements by delay loading Systems Manager SSM assemblies. • Improved exception handling for malformed sysprep 2008.xml. • Command line support for Systems Manager "Apply" configuration. • Change to support domain join when there is a pending computer rename. • Support for optional parameters in the `aws:applications` plugin. • Support for command array in `aws:psModule` plugin.	
3.0.54	• Enable support for Systems Manager. • Automatically domain join EC2 Windows instances to an AWS directory via Systems Manager. • Configure and upload CloudWatch logs/metrics via Systems Manager. • Install PowerShell modules via Systems Manager. • Install MSI applications via Systems Manager.	

Version	Details	Release date
2.4.233	• Added scheduled task to recover EC2Config from service startup failures. • Improvements to the Console log error messages. • Updates to improve support for AWS services.	
2.3.313	• Fixed an issue with large memory consumption in some cases when the CloudWatch Logs feature is enabled. • Fixed an upgrade bug so that ec2config versions lower than 2.1.19 can now upgrade to latest. • Updated COM port opening exception to be more friendly and useful in logs. • Ec2configServiceSettings UI disabled resizing and fixed the attribution and version display placement in UI.	
2.2.12	• Handled NullPointerException while querying a registry key for determining Windows Sysprep state which returned null occasionally. • Freed up unmanaged resources in finally block.	
2.2.11	Fixed a issue in CloudWatch plugin for handling empty log lines.	
2.2.10	• Removed configuring CloudWatch Logs settings through UI. • Enable users to define CloudWatch Logs settings in `%ProgramFiles%\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` file to allow future enhancements.	

Version	Details	Release date
2.2.9	Fixed unhandled exception and added logging.	
2.2.8	• Fixes Windows OS version check in EC2Config Installer to support Windows Server 2003 SP1 and later. • Fixes null value handling when reading registry keys related to updating Sysprep config files.	
2.2.7	• Added support for EC2Config to run during Sysprep execution for Windows 2008 and greater. • Improved exception handling and logging for better d iagnostics	
2.2.6	• Reduced the load on the instance and on CloudWatch Logs when uploading log events. • Addressed an upgrade issue where the CloudWatch Logs plug-in did not always stay enabled	
2.2.5	• Added support to upload logs to CloudWatch Log Service. • Fixed a race condition issue in Ec2OutputRDPCert plug-in • Changed EC2Config Service recovery option to Restart from TakeNoAction • Added more exception information when EC2Config Crashes	
2.2.4	• Fixed a typo in PostSysprep.cmd • Fixed the bug which EC2Config does not pin itself onto start menu for OS2012+	

Version	Details	Release date
2.2.3	• Added option to install EC2Config without service starting immediately upon install. To use, run 'Ec2Install.exe s tart=false' from the command prompt • Added parameter in wallpaper plugin to control adding/re moving wallpaper. To use, run 'Ec2WallpaperInfo.exe set' or 'Ec2WallpaperInfo.exe revert' from the command prompt • Added checking for RealTimeIsUniversal key, output i ncorrect settings of the RealTimeIsUniveral registry key to the Console • Removed EC2Config dependency on Windows temp folder • Removed UserData execution dependency on .Net 3.5	
2.2.2	• Added check to service stop behavior to check that res ources are being released • Fixed issue with long execution times when joined to domai n	
2.2.1	• Updated Installer to allow upgrades from older versions • Fixed Ec2WallpaperInfo bug in .Net4.5 only environment • Fixed intermittent driver detection bug • Added silent install option. Execute Ec2Install.exe with the '-q' option. eg: 'Ec2Install.exe -q'	
2.2.0	• Added support for .Net4 and .Net4.5 only environments • Updated Installer	

Version	Details	Release date
2.1.19	• Added ephemeral disk labeling support when using Intel network driver (eg. C3 instance Type). For more information, see Enhanced networking on Windows. • Added AMI Origin Version and AMI Origin Name support to the console output • Made changes to the Console Output for consistent formatting/parsing • Updated Help File	
2.1.18	• Added EC2Config WMI Object for Completion notification (-Namespace root\Amazon -Class EC2_ConfigService) • Improved Performance of Startup WMI query with large Event Logs; could cause prolonged high CPU during initial execution	
2.1.17	• Fixed UserData execution issue with Standard Output and Standard Error buffer filling • Fixed incorrect RDP thumbprint sometimes appearing in Console Output for >= w2k8 OS • Console Output now contains 'RDPCERTIFICATE-SubjectName:' for Windows 2008+, which contains the machine name value • Added D:\ to Drive Letter Mapping dropdown • Moved Help button to top right and changed look/feel • Added Feedback survey link to top right	

Version	Details	Release date
2.1.16	• General Tab includes link to EC2Config download page for new Versions • Desktop Wallpaper overlay now stored in Users Local Appdata folder instead of My Documents to support MyDocs redirection • MSSQLServer name sync'd with system in Post-Sysprep script (2008+) • Reordered Application Folder (moved files to Plugin directory and removed duplicate files) • Changed System Log Output (Console): • *Moved to a date, name, value format for easier parsing (Please start migrating dependencies to new format) • *Added 'Ec2SetPassword' plugin status • *Added Sysprep Start and End times • Fixed issue of Ephemeral Disks not being labeled as 'Temporary Storage' for non-english Operating Systems • Fixed EC2Config Uninstall failure after running Sysprep	

Version	Details	Release date
2.1.15	• Optimized requests to the Metadata service • Metadata now bypass Proxy Settings • Ephemeral Disks labeled as 'Temporary Storage' and Important.txt placed on volume when found (Citrix PV drivers only). For more information, see Upgrade PV drivers on Windows instances. • Ephemeral Disks assigned drive letters from Z to A (Citrix PV drivers only) - assignment can be overwritten using Drive Letter Mapping plugin with Volume labels 'Temporary Storage X' where x is a number 0-25) • UserData now runs immediately following 'Windows is Ready'	
2.1.14	Desktop wallpaper fixes	
2.1.13	• Desktop wallpaper will display hostname by default • Removed dependency on Windows Time service • Route added in cases where multiple IPs are assigned to a single interface	
2.1.11	• Changes made to Ec2Activation Plugin • -Verifies Activation status every 30 days • -If Grace Period has 90 days remaining (out of 180), re attempts activation	

Version	Details	Release date
2.1.10	• Desktop wallpaper overlay no longer persists with Sysprep or Shutdown without Sysprep • Userdata option to run on every service start with <persist>true</persist> • Changed location and name of /DisableWinUpdate.cmd to /Scripts/PostSysprep.cmd • Administrator password set to not expire by default in /Scripts/PostSysprep.cmd • Uninstall will remove EC2Config PostSysprep script from c:\windows\setup\script\CommandComplete.cmd • Add Route supports custom interface metrics	
2.1.9	UserData Execution no longer limited to 3851 Characters	

Version	Details	Release date
2.1.7	• OS Version and language identifier written to console • EC2Config version written to console • PV driver version written to console • Detection of Bug Check and output to the console on next boot when found • Option added to config.xml to persist Sysprep credentials • Add Route Retry logic in cases of ENI being unavailable at start • User Data execution PID written to console • Minimum generated password length retrieved from GPO • Set service start to retry 3 attempts • Added S3_DownloadFile.ps1 and S3_Upload file.ps1 examples to /Scripts folder	

Version	Details	Release date
2.1.6	• Version information added to General tab • Renamed the Bundle tab to Image • Simplified the process of specifying passwords and moved the password-related UI from the General tab to the Image tab • Renamed the Disk Settings tab to Storage • Added a Support tab with common tools for troubleshooting • Windows Server 2003 `sysprep.ini` set to extend OS partition by default • Added the private IP address to the wallpaper • Private IP address displayed on wallpaper • Added retry logic for Console output • Fixed Com port exception for metadata accessibility -- caused EC2Config to terminate before console output is displayed • Checks for activation status on every boot -- activates as necessary • Fixed issue of relative paths -- caused when manually executing wallpaper shortcut from startup folder; pointing to Administrator/logs • Fixed default background color for Windows Server 2003 user (other than Administrator)	

Version	Details	Release date
2.1.2	• Console timestamps in UTC (Zulu) • Removed appearance of hyperlink on Sysprep tab • Addition of feature to dynamically expand Root Volume on first boot for Windows 2008+ • When Set-Password is enabled, now automatically enables EC2Config to set the password • EC2Config checks activation status prior to running Sysprep (presents warning if not activated) • Windows Server 2003 `Sysprep.xml` now defaults to UTC timezone instead of Pacific • Randomized Activation Servers • Renamed Drive Mapping tab to Disk Settings • Moved Initialize Drives UI items from General to the Disk Set tings tab • Help button now points to HTML help file • Updated HTML help file with changes • Updated 'Note' text for Drive Letter Mappings • Added InstallUpdates.ps1 to /Scripts folder for automating Patches and cleanup prior to Sysprep	

Version	Details	Release date
2.1.0	• Desktop wallpaper displays instance information by default upon first logon (not disconnect/reconnect) • PowerShell can be run from the userdata by surrounding the code with <powershell></powershell>	

Subscribe to EC2Config service notifications

Amazon SNS can notify you when new versions of the EC2Config service are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2Config notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select this Region because the SNS notifications that you are subscribing to were created in this Region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
   a. For **Topic ARN**, use the following Amazon Resource Name (ARN):

      ```
      arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config
      ```

   b. For **Protocol**, choose `Email`.
   c. For **Endpoint**, type an email address that you can use to receive the notifications.
   d. Choose **Create subscription**.
6. You'll receive a email asking you to confirm your subscription. Open the email and follow the directions to complete your subscription.

Whenever a new version of the EC2Config service is released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from EC2Config notifications

1. Open the Amazon SNS console.
2. In the navigation pane, choose **Subscriptions**.
3. Select the subscription and then choose **Actions**, **Delete subscriptions** When prompted for confirmation, choose **Delete**.

Troubleshoot issues with the EC2Config service

The following information can help you troubleshoot issues with the EC2Config service.

Update EC2Config on an unreachable instance

Use the following procedure to update the EC2Config service on a Windows Server instance that is inaccessible using Remote Desktop.

To update EC2Config on an Amazon EBS-backed Windows instance that you can't connect to

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Select the instance and choose **Instance state**, and then choose **Stop instance**.

   > **Warning**
   >
   > When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. Choose **Launch instances** and create a temporary `t2.micro` instance in the same Availability Zone as the affected instance. Use a different AMI than the one that you used to launch the affected instance.

   > **Important**
   >
   > If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the EC2 console, choose **Volumes**.

6. Locate the root volume of the affected instance. Detach the volume and then attach the volume to the temporary instance that you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to make the volume available for use.
8. Download the latest version of the EC2Config service. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.
9. On the temporary instance, open the Run dialog box, type **`regedit`**, and press Enter.
10. Choose `HKEY_LOCAL_MACHINE`. From the **File** menu, choose **Load Hive**. Choose the drive and then navigate to and open the following file: `Windows\System32\config\SOFTWARE`. When prompted, specify a key name.
11. Select the key you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Choose the `RunOnce` key. If this key doesn't exist, choose `CurrentVersion` from the context (right-click) menu, choose **New** and then choose **Key**. Name the key `RunOnce`.
12. From the context (right-click) menu choose the `RunOnce` key, choose **New** and then choose **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe /quiet` as the data.
13. Choose the `HKEY_LOCAL_MACHINE\`*`specified key name`*`\Microsoft\Windows NT \CurrentVersion\Winlogon` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter **`AutoAdminLogon`** as the name and **1** as the value data.
14. Choose the `HKEY_LOCAL_MACHINE\`*`specified key name`*`\Microsoft\Windows NT \CurrentVersion\Winlogon>` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter **`DefaultUserName`** as the name and **`Administrator`** as the value data.
15. Choose the `HKEY_LOCAL_MACHINE\`*`specified key name`*`\Microsoft\Windows NT \CurrentVersion\Winlogon` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Type **`DefaultPassword`** as the name and enter a password in the value data.
16. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened Registry Editor.
17. From the **File** menu, choose **Unload Hive**.
18. In Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.

19. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
20. Stop and start your instance the instance.
21. After the instance starts, check the system log and verify that you see the message Windows is ready to use.
22. Open Registry Editor and choose `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`. Delete the String Value keys you created earlier: **AutoAdminLogon**, **DefaultUserName**, and **DefaultPassword**.
23. Delete or stop the temporary instance you created in this procedure.

Paravirtual drivers for Windows instances

Windows AMIs contain a set of drivers to permit access to virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. The following table shows key differences between the different drivers.

	RedHat PV	Citrix PV	AWS PV
Instance type	Not supported for all instance types. If you specify an unsupported instance type, the instance is impaired.	Supported for Xen instance types.	Supported for Xen instance types.
Attached volumes	Supports up to 16 attached volumes.	Supports more than 16 attached volumes.	Supports more than 16 attached volumes.
Network	The driver has known issues where the network connectio n resets under high loads; for example, fast FTP file transfers.		The driver automatic ally configure s jumbo frames on the network

	RedHat PV	Citrix PV	AWS PV
			adapter when on a compatibl e instance type. When the instance is in a cluster placement group, this offers better network performan ce between instances in the cluster placement group.

The following table shows which PV drivers you should run on each version of Windows Server on Amazon EC2.

Windows Server version	PV driver version
Windows Server 2022	AWS PV latest version
Windows Server 2019	AWS PV latest version
Windows Server 2016	AWS PV latest version
Windows Server 2012 R2	AWS PV latest version
Windows Server 2012	AWS PV latest version
Windows Server 2008 R2	AWS PV version 8.3.5

Windows Server version	PV driver version
Windows Server 2008	Citrix PV 5.9
Windows Server 2003	Citrix PV 5.9

Contents

- AWS PV drivers
- Citrix PV drivers
- RedHat PV drivers
- Subscribe to notifications
- Upgrade PV drivers on Windows instances
- Troubleshoot PV drivers

AWS PV drivers

The AWS PV drivers are stored in the `%ProgramFiles%\Amazon\Xentools` directory. This directory also contains public symbols and a command line tool, `xenstore_client.exe`, that enables you to access entries in XenStore. For example, the following PowerShell command returns the current time from the Hypervisor:

```
PS C:\> [DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl
 AWSXenStoreBase).XenTime).ToString("hh:mm:ss")
11:17:00
```

The AWS PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`. These driver components are as follows: xenbus, xeniface, xennet, xenvbd, and xenvif.

AWS PV drivers also have a Windows service named LiteAgent, which runs in user-mode. It handles tasks such as shutdown and restart events from AWS APIs on Xen generation instances. You can access and manage services by running `Services.msc` from the command line. When running on Nitro generation instances, the AWS PV drivers are not used and the LiteAgent service will self-stop starting with driver version 8.2.4. Updating to the latest AWS PV driver also updates the LiteAgent and improves reliability on all instance generations.

Install the latest AWS PV drivers

Amazon Windows AMIs contain a set of drivers to permit access to virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. We recommend that you install the latest drivers to improve stability and performance of your EC2 Windows instances.

Installation options

- You can use AWS Systems Manager to automatically update the PV drivers. For more information, see Walkthrough: Automatically Update PV Drivers on EC2 Windows Instances (Console) in the *AWS Systems Manager User Guide*.
- You can download the driver package and run the install program manually. Be sure to check the `readme.txt` file for system requirements. For information about downloading and installing the AWS PV drivers, or upgrading a domain controller, see Upgrade Windows Server instances (AWS PV upgrade) manually.

AWS PV driver package history

The following table shows the changes to AWS PV drivers for each driver release.

Package version	Details	Release date
8.4.3	Fixed bugs in the package installer to improve the upgrade experience.	24 January 2023
8.4.2	Stability fixes to address race condition.	13 April 2022
8.4.1	Improved package installer.	7 January 2022
8.4.0	• Stability fixes to address rare cases of stuck disk IO. • Stability fixes to address rare cases of crashes during EBS volume detachment. • Added feature to distribute load across multiple cores for workloads that leverage more than 20,000 IOPS and experience degradation due to bottlenecks. To enable this	2 March 2021

Package version	Details	Release date
	feature, see Workloads that leverage more than 20,000 disk IOPS experience degradation due to CPU bottlenecks. • AWS PV 8.4 installation on Windows Server 2008 R2 will fail. AWS PV version 8.3.5 and earlier are supported on Windows Server 2008 R2.	
8.3.5	Improved package installer.	7 January 2022
8.3.4	Improved reliability of network device attachment.	4 August 2020
8.3.3	• Update to XenStore-facing component to prevent bug check during error-handling paths. • Update to storage component to avoid crashes when an invalid SRB is submitted. To update this driver on Windows Server 2008 R2 instances, you must first verify that the appropriate patches are installed to address the following Microsoft Security Advisory: Microsoft Security Advisory 3033929.	4 February 2020
8.3.2	Enhanced reliability of networking components.	30 July 2019
8.3.1	Improved performance and robustness of storage component.	12 June 2019
8.2.7	Improved efficiency to support migrating to latest generation instance types.	20 May 2019
8.2.6	Improved efficiency of crash dump path.	15 January 2019
8.2.5	Additional security enhancements. PowerShell installer now available in package.	12 December 2018

Package version	Details	Release date
8.2.4	Reliability improvements.	2 October 2018
8.2.3	Bug fixes and performance improvements. Report EBS volume ID as disk serial number for EBS volumes. This enables cluster scenarios such as S2D.	29 May 2018
8.2.1	Network and storage performance improvements plus multiple robustness fixes. To verify that this version has been installed, refer to the following Windows registry value: `HKLM\Software\Amazon\PVDriver\Version 8.2.1` .	8 March 2018
7.4.6	Stability fixes to make AWS PV drivers more resilient.	26 April 2017
7.4.3	Added support for Windows Server 2016. Stability fixes for all supported Windows OS versions. *AWS PV driver version 7.4.3's signature expires on March 29, 2019. We recommend updating to the latest AWS PV driver.	18 Nov 2016
7.4.2	Stability fixes for support of X1 instance type.	2 Aug 2016
7.4.1	• Performance improvement in AWS PV Storage driver. • Stability fixes in AWS PV Storage driver: Fixed an issue where the instances were hitting a system crash with bug check code 0x0000DEAD. • Stability fixes in AWS PV Network driver. • Added support for Windows Server 2008R2.	12 July 2016

Package version	Details	Release date
7.3.2	• Improved logging and diagnostics. • Stability fix in AWS PV Storage driver. In some cases disks may not surface in Windows after reattaching the disk to the instance. • Added support for Windows Server 2012.	24 June 2015
7.3.1	TRIM update: Fix related to TRIM requests. This fix stabilizes instances and improves instance performance when managing large numbers of TRIM requests.	
7.3.0	TRIM support: The AWS PV driver now sends TRIM requests to the hypervisor. Ephemeral disks will properly process TRIM requests given the underlying storage supports TRIM (SSD). Note that EBS-based storage does not support TRIM as of March 2015.	
7.2.5	• Stability fix in AWS PV Storage drivers: In some cases the AWS PV driver could dereference invalid memory and cause a system failure. • Stability fix while generating a crash dump: In some cases the AWS PV driver could get stuck in a race condition when writing a crash dump. Before this release, the issue could only be resolved by forcing the driver to stop and restart which lost the memory dump.	
7.2.4	Device ID persistence: This driver fix masks the platform PCI device ID and forces the system to always surface the same device ID, even if the instance is moved. More generally, the fix affects how the hypervisor surfaces virtual devices. The fix also includes modifications to the co-installer for the AWS PV drivers so the system persists mapped virtual devices.	

Package version	Details	Release date
7.2.2	• Load the AWS PV drivers in Directory Services Restore Mode (DSRM) mode: Directory Services Restore Mode is a safe mode boot option for Windows Server domain controllers. • Persist device ID when virtual network adapter device is reattached: This fix forces the system to check the MAC address mapping and persist the device ID. This fix ensures that adapters retain their static settings if the adapters are reattached.	
7.2.1	• Run in safe mode: Fixed an issue where the driver would not load in safe mode. Previously the AWS PV Drivers would only instantiate in normal running systems. • Add disks to Microsoft Windows Storage Pools: Previously we synthesized page 83 queries. The fix disabled page 83 support. Note this does not affect storage pools that are used in a cluster environment because PV disks are not valid cluster disks.	
7.2.0	Base: The AWS PV base version.	

Citrix PV drivers

The Citrix PV drivers are stored in the `%ProgramFiles%\Citrix\XenTools` (32-bit instances) or `%ProgramFiles(x86)%\Citrix\XenTools` (64-bit instances) directory.

The Citrix PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services`. These driver components are as follows: xenevtchn, xeniface, xennet, Xennet6, xensvc, xenvbd, and xenvif.

Citrix also has a driver component named XenGuestAgent, which runs as a Windows service. It handles tasks such as shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

If you are encountering networking errors while performing certain workloads, you may need to disable the TCP offloading feature for the Citrix PV driver. For more information, see TCP offloading.

RedHat PV drivers

RedHat drivers are supported for legacy instances, but are not recommended on newer instances with more than 12GB of RAM due to driver limitations. Instances with more than 12GB of RAM running RedHat drivers can fail to boot and become inaccessible. We recommend upgrading RedHat drivers to Citrix PV drivers, and then upgrade Citrix PV drivers to AWS PV drivers.

The source files for the RedHat drivers are in the `%ProgramFiles%\RedHat` (32-bit instances) or `%ProgramFiles(x86)%\RedHat` (64-bit instances) directory. The two drivers are `rhelnet`, the RedHat Paravirtualized network driver, and `rhelscsi`, the RedHat SCSI miniport driver.

Subscribe to notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use one of the following methods to subscribe to these notifications.

> **Note**
>
> You must specify the Region for the SNS Topic you subscribe to.

Subscribe to EC2 notifications from the console

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select this Region because the SNS notifications that you are subscribing to are in this Region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
   a. For **TopicARN**, copy the following Amazon Resource Name (ARN):

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers

   b. For **Protocol**, choose `Email`.
   c. For **Endpoint**, type an email address that you can use to receive the notifications.

d. Choose **Create subscription**.

6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Subscribe to EC2 notifications using the AWS CLI

To subscribe to EC2 notifications with the AWS CLI, use the following command.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-
windows-drivers --region us-east-1 --protocol email --notification-
endpoint YourUserName@YourDomainName.ext
```

Subscribe to EC2 notifications using the AWS Tools for PowerShell

To subscribe to EC2 notifications with Tools for Windows PowerShell, use the following command.

```
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-
drivers' -Region us-east-1 -Protocol email -Endpoint 'YourUserName@YourDomainName.ext'
```

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

Unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Subscriptions**.
3. Select the check box for the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Upgrade PV drivers on Windows instances

We recommend that you install the latest PV drivers to improve the stability and performance of your EC2 Windows instances. The directions on this page help you download the driver package and run the install program.

To verify which driver your Windows instance uses

Open **Network Connections** in Control Panel and view **Local Area Connection**. Check whether the driver is one of the following:

- AWS PV Network Device
- Citrix PV Ethernet Adapter
- RedHat PV NIC Driver

Alternatively, you can check the output from the `pnputil -e` command.

System requirements

Be sure to check the `readme.txt` file in the download for system requirements.

Contents

- Upgrade Windows Server instances (AWS PV upgrade) with Distributor
- Upgrade Windows Server instances (AWS PV upgrade) manually
- Upgrade a domain controller (AWS PV upgrade)
- Upgrade Windows Server 2008 and 2008 R2 instances (Redhat to Citrix PV upgrade)
- Upgrade your Citrix Xen guest agent service

Upgrade Windows Server instances (AWS PV upgrade) with Distributor

You can use Distributor, a capability of AWS Systems Manager, to install or upgrade the AWS PV driver package. The installation or upgrade can be performed one time, or you can install or update it on a schedule. The `In-place update` option for **Installation Type** isn't supported for this Distributor package.

> **Important**
>
> If your instance is a domain controller, see Upgrade a domain controller (AWS PV upgrade). The upgrade process for domain controller instances is different than standard editions of Windows.

1. We recommend that you create a backup in case you need to roll back your changes.

   > **Tip**
   >
   > Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For

more information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

a. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

b. In the navigation pane, choose **Instances**.

c. Select the instance that requires the driver upgrade, and choose **Instance state**, **Stop instance**.

d. After the instance is stopped, select the instance, choose **Actions**, then **Image and templates**, and then choose **Create image**.

e. Choose **Instance state**, **Start instance**.

2. Connect to the instance using Remote Desktop. For more information, see Connect to your Windows instance using RDP.
3. We recommend that you take all non-system disks offline and note any drive letter mappings to the secondary disks in Disk Management before you perform this upgrade. This step is not required if you are performing an in-place update of AWS PV drivers. We also recommend setting non-essential services to **Manual** start-up in the Services console.
4. For the instructions for how to install or upgrade the AWS PV driver package using Distributor, see the procedures in Install or update packages in the *AWS Systems Manager User Guide*.
5. For **Name**, choose **AWSPVDriver**.
6. For **Installation type**, select **Uninstall and reinstall**.
7. Configure the other parameters for the package as necessary and run installation or upgrade using the referenced procedure in Step 4.

   After running the Distributor package, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes.
8. After the upgrade is complete, and the instance passes both health checks in the Amazon EC2 console, verify that the new driver was installed by connecting to the instance using Remote Desktop.
9. Once you have connected, run the following PowerShell command:

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

10. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters noted in Step 3.

If you previously disabled TCP offloading using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

If you previously applied a static IP address or DNS configuration to the network interface, you might need to reapply the static IP address or DNS configuration after upgrading AWS PV drivers.

Upgrade Windows Server instances (AWS PV upgrade) manually

Use the following procedure to perform an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers on Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019, or Windows Server 2022. This upgrade is not available for RedHat drivers, or for other versions of Windows Server.

Some older versions of Windows Server can't use the latest drivers. To verify which driver version to use for your operating system, see the driver version table in the Paravirtual drivers for Windows instances page.

> **Important**
>
> If your instance is a domain controller, see Upgrade a domain controller (AWS PV upgrade). The upgrade process for domain controller instances is different than standard editions of Windows.

To upgrade AWS PV drivers manually

1. We recommend that you create a backup in case you need to roll back your changes.

> **Tip**
>
> Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For

more information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

a. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

b. In the navigation pane, choose **Instances**.

c. Select the instance that requires the driver upgrade, and choose **Instance state**, **Stop instance**.

d. After the instance is stopped, select the instance, choose **Actions**, then **Image and templates**, and then choose **Create image**.

e. Choose **Instance state**, **Start instance**.

2. Connect to the instance using Remote Desktop.

3. We recommend that you take all non-system disks offline and note any drive letter mappings to the secondary disks in Disk Management before you perform this upgrade. This step is not required if you are performing an in-place update of AWS PV drivers. We also recommend setting non-essential services to **Manual** start-up in the Services console.

4. Download the latest driver package to the instance.

   Or, run the following PowerShell command:

```
Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/AWSPV/
Latest/AWSPVDriver.zip -outfile $env:USERPROFILE\pv_driver.zip
Expand-Archive $env:userprofile\pv_driver.zip -DestinationPath
 $env:userprofile\pv_drivers
```

5. Extract the contents of the folder and then run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, you can verify that the new driver was installed by connecting to the instance using Remote Desktop and then running the following PowerShell command:

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters noted in Step 3.

If you previously disabled TCP offloading using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

If you previously applied a static IP address or DNS configuration to the network interface, you might need to reapply the static IP address or DNS configuration after upgrading AWS PV drivers.

Upgrade a domain controller (AWS PV upgrade)

Use the following procedure on a domain controller to perform either an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers.

To upgrade a domain controller

1. We recommend that you create a backup of your domain controller in case you need to roll back your changes. Using an AMI as a backup is not supported. For more information, see Backup and Restore Considerations for Virtualized Domain Controllers in the Microsoft documentation.
2. Run the following command to configure Windows to boot into Directory Services Restore Mode (DSRM).

   > **Warning**
   >
   > Before running this command, confirm that you know the DSRM password. You'll need this information so that you can log in to your instance after the upgrade is complete and the instance automatically reboots.

   ```
   bcdedit /set {default} safeboot dsrepair
   ```

   PowerShell:

```
PS C:\> bcdedit /set "{default}" safeboot dsrepair
```

The system must boot into DSRM because the upgrade utility removes Citrix PV storage drivers so it can install AWS PV drivers. Therefore we recommend noting any drive letter and folder mappings to the secondary disks in Disk Management. When Citrix PV storage drivers are not present, secondary drives are not detected. Domain controllers that use an NTDS folder on secondary drives will not boot because the secondary disk is not detected.

> **Warning**
>
> After you run this command do not manually reboot the system. The system will be unreachable because Citrix PV drivers do not support DSRM.

3. Run the following command to add **`DisableDCCheck`** to the registry:

```
reg add HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck /t
 REG_SZ /d true
```

4. Download the latest driver package to the instance.
5. Extract the contents of the folder and then run `AWSPVDriverSetup.msi`.

   After running the MSI, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes.
6. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop. Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters and folder mappings noted earlier.

   You must connect to the instance by specifying the user name in the following format *hostname*\administrator. For example, Win2k12TestBox\administrator.
7. Run the following command to remove the DSRM boot configuration:

```
bcdedit /deletevalue safeboot
```

8. Reboot the instance.
9. To complete the upgrade process, verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver

version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history.

10. Run the following command to delete **`DisableDCCheck`** from the registry:

```
reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck
```

> **Note**
>
> If you previously disabled TCP offloading using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV Drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

Upgrade Windows Server 2008 and 2008 R2 instances (Redhat to Citrix PV upgrade)

Before you start upgrading your RedHat drivers to Citrix PV drivers, make sure you do the following:

- Install the latest version of the EC2Config service. For more information, see Install the latest version of EC2Config.
- Verify that you have Windows PowerShell 3.0 installed. To verify the version that you have installed, run the following command in a PowerShell window:

  ```
  PS C:\> $PSVersionTable.PSVersion
  ```

  Windows PowerShell 3.0 is bundled in the Windows Management Framework (WMF) version 3.0 install package. If you need to install Windows PowerShell 3.0, see Windows Management Framework 3.0 in the Microsoft Download Center.
- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see Create a custom Windows AMI.

  > **Tip**
  >
  > Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For more

information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

If you create an AMI, make sure that you do the following:

- Write down your password.
- Do not run the Sysprep tool manually or using the EC2Config service.
- Set your Ethernet adapter to obtain an IP address automatically using DHCP. For more information, see Configure TCP/IP Settings in the Microsoft TechNet Library.

To upgrade RedHat drivers

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see Connect to your Windows instance.
2. In your instance, download the Citrix PV upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, choose **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and choose **Yes** if you are ready to start the upgrade.
6. In the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box, choose **Yes** to remove the RedHat software. Your instance will be rebooted.

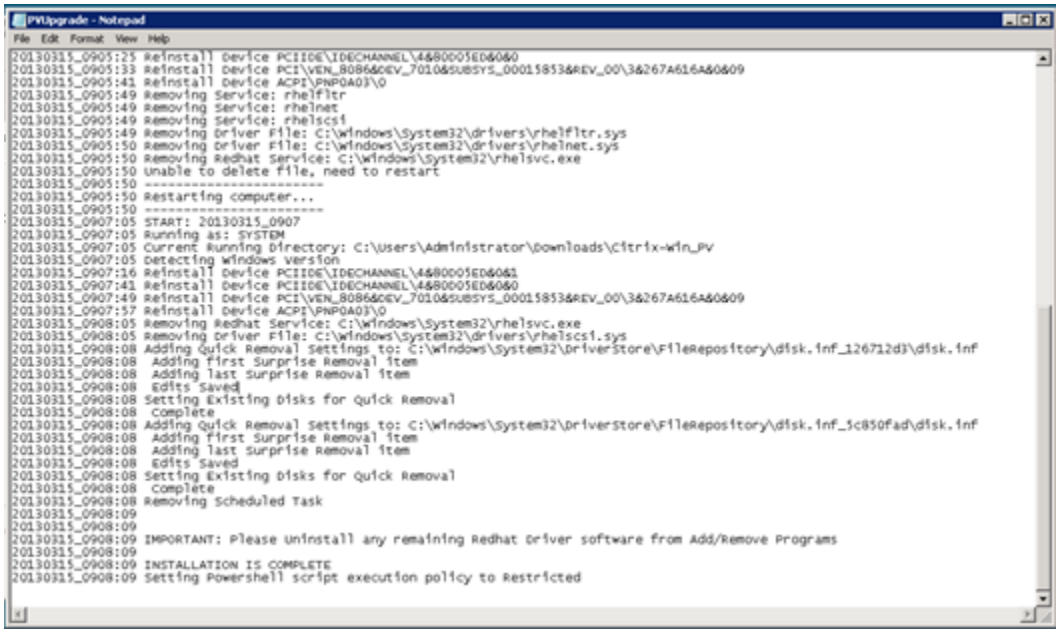
> **Note**
>
> If you do not see the uninstaller dialog box, choose **Red Hat Paravirtualize** in the Windows taskbar.

7. Check that the instance has rebooted and is ready to be used.
   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   b. On the **Instances** page, select **Actions**, then **Monitor and troubleshoot**, and then choose **Get system log**.

c. The upgrade operations should have restarted the server 3 or 4 times. You can see this in the log file by the number of times `Windows is Ready to use` is displayed.

```
    Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
 Ec2Config service v2.1.9.0
  RedHat PV NIC Driver v1.3.10.0
2013/03/15 17:11:01Z: Waiting for meta-data accessibility...
2013/03/15 17:11:02Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
<Username>Administrator</Username>
<Password>
L79ThJPF8LyIL38IZht0FBrjet3vnT2csTiU/XGVMRCH7kQtBnznAnXrKd1sirXlxl9BwVMsd9b38jFJqv0lIUpgNNJRZoCDc7IbUwM
</Password>
2013/03/15 17:11:30Z: Product activation was successful.
2013/03/15 17:11:32Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
 Ec2Config service v2.1.9.0
2013/03/15 21:04:24Z: There was an exception writing driver information to console: System.Exception: U
   at Ec2Config.Service1.Go()
2013/03/15 21:04:35Z: Waiting for meta-data accessibility...
2013/03/15 21:04:40Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:05:08Z: Product activation was successful.
2013/03/15 21:05:09Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
 Ec2Config service v2.1.9.0
  Citrix PV Ethernet Adapter v5.9.960.49119
2013/03/15 21:07:20Z: Waiting for meta-data accessibility...
2013/03/15 21:07:21Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BFD64F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:07:27Z: Message: Windows is Ready to use
```

8. Connect to your instance and log in as the local administrator.
9. Close the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box.
10. Confirm that the installation is complete. Navigate to the `Citrix-WIN_PV` folder that you extracted earlier, open the `PVUpgrade.log` file, and then check for the text `INSTALLATION IS COMPLETE`.



Upgrade your Citrix Xen guest agent service

If you are using Citrix PV drivers on Windows Server, you can upgrade the Citrix Xen guest agent service. This Windows service handles tasks such as shutdown and restart events from the API. You can run this upgrade package on any version of Windows Server, as long as the instance is running Citrix PV drivers.

> **Important**
>
> For Windows Server 2008 R2 and later, we recommend you upgrade to AWS PV drivers that include the Guest Agent update.

Before you start upgrading your drivers, make sure you back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see Create a custom Windows AMI.

> **Tip**
>
> Instead of creating the AMI from the Amazon EC2 console, you can use Systems Manager Automation to create the AMI using the `AWS-CreateImage` runbook. For more

information, see AWS-CreateImage in the *AWS Systems Manager Automation runbook reference User Guide*.

If you create an AMI, make sure you do the following:

- Do not enable the Sysprep tool in the EC2Config service.
- Write down your password.
- Set your Ethernet adapter to DHCP.

To upgrade your Citrix Xen guest agent service

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see Connect to your Windows instance.
2. On your instance, download the Citrix upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, choose **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and choose **Yes** if you are ready to start the upgrade.
6. When the upgrade is complete, the `PVUpgrade.log` file will open and contain the text `UPGRADE IS COMPLETE`.
7. Reboot your instance.

Troubleshoot PV drivers

The following are solutions to issues that you might encounter with older Amazon EC2 images and PV drivers.

Contents

- Windows Server 2012 R2 loses network and storage connectivity after an instance reboot
- TCP offloading
- Time synchronization
- Workloads that leverage more than 20,000 disk IOPS experience degradation due to CPU bottlenecks

Windows Server 2012 R2 loses network and storage connectivity after an instance reboot

> **Important**
>
> This issue occurs only with AMIs made available before September 2014.

Windows Server 2012 R2 Amazon Machine Images (AMIs) made available before September 10, 2014 can lose network and storage connectivity after an instance reboot. The error in the AWS Management Console system log states: “Difficulty detecting PV driver details for Console Output.” The connectivity loss is caused by the Plug and Play Cleanup feature. This features scans for and disables inactive system devices every 30 days. The feature incorrectly identifies the EC2 network device as inactive and removes it from the system. When this happens, the instance loses network connectivity after a reboot.

For systems that you suspect could be affected by this issue, you can download and run an in-place driver upgrade. If you are unable to perform the in-place driver upgrade, you can run a helper script. The script determines if your instance is affected. If it is affected, and the Amazon EC2 network device has not been removed, the script disables the Plug and Play Cleanup scan. If the network device was removed, the script repairs the device, disables the Plug and Play Cleanup scan, and enables your instance to reboot with network connectivity enabled.

Contents

- Choose how to fix problems
- Method 1 - Enhanced networking
- Method 2 - Registry configuration
- Run the remediation script

Choose how to fix problems

There are two methods for restoring network and storage connectivity to an instance affected by this issue. Choose one of the following methods:

Method	Prerequisites	Procedure Overview
Method 1 - Enhanced networking	Enhanced networking is only available in a virtual private	You change the server instance type to a C3

Method	Prerequisites	Procedure Overview
	cloud (VPC) which requires a C3 instance type. If the server does not currently use the C3 instance type, then you must temporarily change it.	instance. Enhanced networkin g then enables you to connect to the affected instance and fix the problem. After you fix the problem, you change the instance back to the original instance type. This method is typically faster than Method 2 and less likely to result in user error. You will incur additiona l charges as long as the C3 instance is running.
Method 2 - Registry configura tion	Ability to create or access a second server. Ability to change Registry settings.	You detach the root volume from the affected instance, attach it to a different instance, connect, and make changes in the Registry. You will incur additiona l charges as long as the additional server is running. This method is slower than Method 1, but this method has worked in situations where Method 1 failed to resolve the problem.

Method 1 - Enhanced networking

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Select the instance and choose **Instance state**, and then choose **Stop instance**.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. After the instance is stopped, create a backup. Select the instance and choose **Actions**, then **Image and templates**, and then choose **Create image**.
5. Change the instance type to any C3 instance type.
6. Start the instance.
7. Connect to the instance using Remote Desktop and then download the AWS PV Drivers Upgrade package to the instance.
8. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

   After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.
9. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history.
10. Stop the instance and change the instance back to its original instance type.
11. Start the instance and resume normal use.

Method 2 - Registry configuration

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Select the instance, choose **Instance state**, and then choose **Stop instance**.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. Choose **Launch instances** and create a temporary Windows Server 2008 or Windows Server 2012 instance in the same Availability Zone as the affected instance. Do not create a Windows Server 2012 R2 instance.

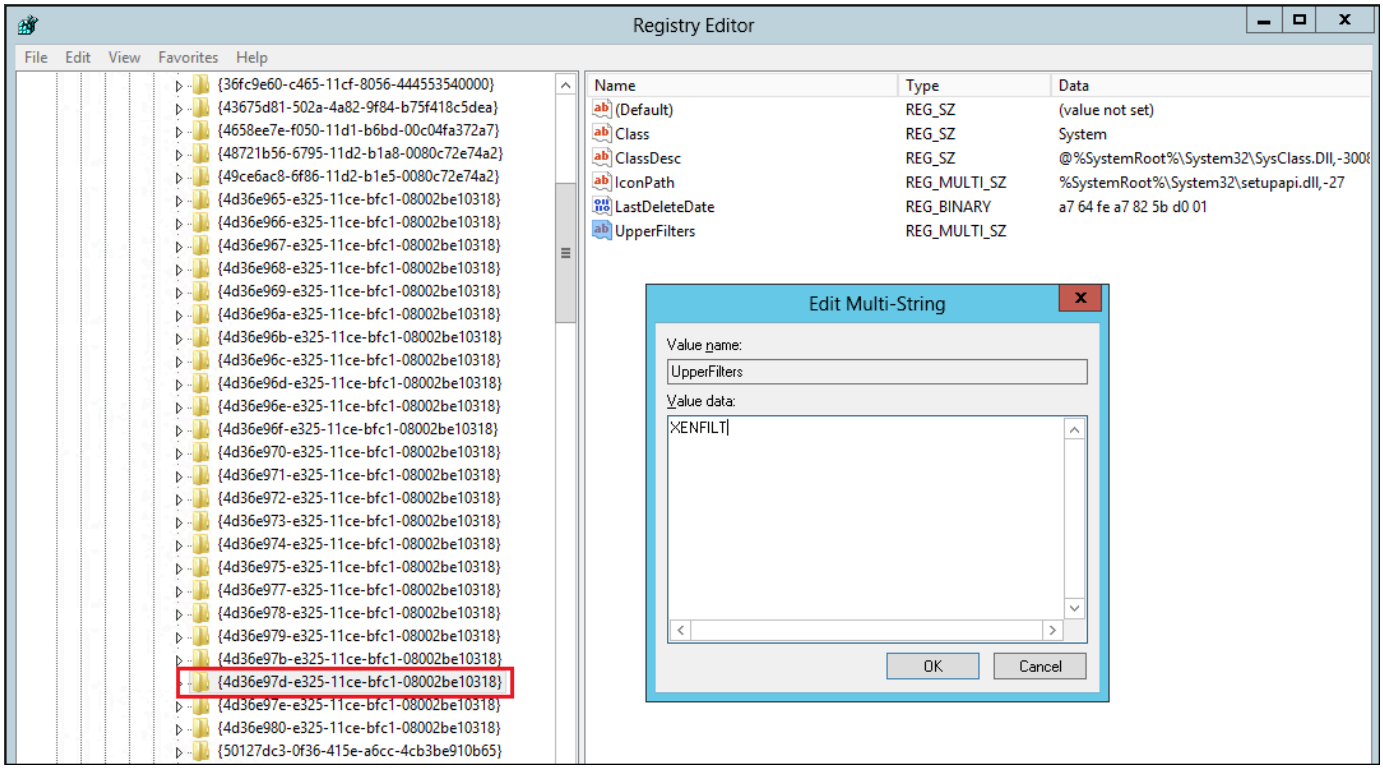
   > **Important**
   >
   > If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. Detach the volume and then attach the volume to the temporary instance that you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to make the volume available for use.
8. On the temporary instance, open the **Run** dialog box, type **`regedit`**, and press Enter.
9. In the Registry Editor navigation pane, choose **HKEY_Local_Machine**, and then from the **File** menu choose **Load Hive**.
10. In the **Load Hive** dialog box, navigate to *Affected Volume*\Windows\System32\config\System and type a temporary name in the **Key Name** dialog box. For example, enter OldSys.
11. In the navigation pane of the Registry Editor, locate the following keys:

    HKEY_LOCAL_MACHINE*`your_temporary_key_name`*\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

    HKEY_LOCAL_MACHINE*`your_temporary_key_name`*\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. For each key, double-click **UpperFilters**, enter a value of XENFILT, and then choose **OK**.



13. Locate the following key:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS\Parameters

14. Create a new string (REG_SZ) with the name ActiveDevice and the following value:

    PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. Locate the following key:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS

16. Change the **Count** from 0 to 1.

17. Locate and delete the following keys:

    HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\xenvbd\StartOverride

    HKEY_LOCAL_MACHINE *your_temporary_key_name*\ControlSet001\Services\xenfilt\StartOverride

18. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened the Registry Editor.

19. From the **File** menu, choose **Unload Hive**.
20. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
21. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your Windows Server 2012 R2 instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
22. Start the instance.
23. Connect to the instance using Remote Desktop and then download the AWS PV Drivers Upgrade package to the instance.
24. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

    After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.
25. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history.
26. Delete or stop the temporary instance you created in this procedure.

Run the remediation script

If you are unable to perform an in-place driver upgrade or migrate to a newer instance you can run the remediation script to fix the problems caused by the Plug and Play Cleanup task.

To run the remediation script

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance for which you want to run the remediation script. Choose **Instance state**, and then choose **Stop instance**.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. After the instance is stopped, create a backup. Select the instance, choose **Actions**, then **Image and templates**, and then choose **Create image**.
5. Choose **Instance state**, and then choose **Start instance**.
6. Connect to the instance by using Remote Desktop and then download the RemediateDriverIssue.zip folder to the instance.
7. Extract the contents of the folder.
8. Run the remediation script according to the instructions in the Readme.txt file. The file is located in the folder where you extracted RemediateDriverIssue.zip.

TCP offloading

> **Important**
>
> This issue does not apply to instances running AWS PV or Intel network drivers.

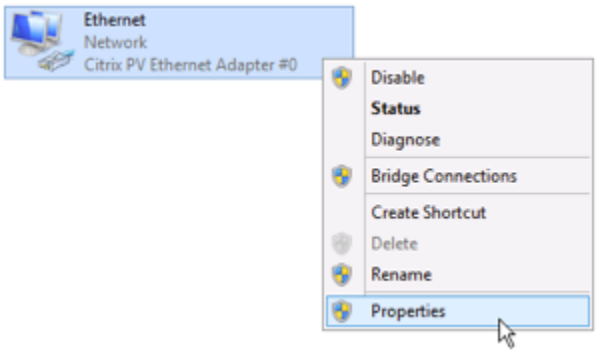
By default, TCP offloading is enabled for the Citrix PV drivers in Windows AMIs. If you encounter transport-level errors or packet transmission errors (as visible on the Windows Performance Monitor)—for example, when you're running certain SQL workloads—you may need to disable this feature.

> **Warning**
>
> Disabling TCP offloading may reduce the network performance of your instance.

To disable TCP offloading for Windows Server 2012 and 2008

1. Connect to your instance and log in as the local administrator.

2. If you're using Windows Server 2012, press **Ctrl+Esc** to access the **Start** screen, and then choose **Control Panel**. If you're using Windows Server 2008, choose **Start** and select **Control Panel**.
3. Choose **Network and Internet**, then **Network and Sharing Center**.
4. Choose **Change adapter settings**.
5. Right-click **Citrix PV Ethernet Adapter #0** and select **Properties**.



6. In the **Local Area Connection Properties** dialog box, choose **Configure** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
7. On the **Advanced** tab, disable each of the properties, except for **Correct TCP/UDP Checksum Value**. To disable a property, select it from **Property** and choose **Disabled** from **Value**.
8. Choose **OK**.
9. Run the following commands from a Command Prompt window.

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

10. Reboot the instance.

Time synchronization

Prior to the release of the 2013.02.13 Windows AMI, the Citrix Xen guest agent could set the system time incorrectly. This can cause your DHCP lease to expire. If you have issues connecting to your instance, you might need to update the agent.

To determine whether you have the updated Citrix Xen guest agent, check whether the `C:\Program Files\Citrix\XenGuestAgent.exe` file is from March 2013. If the date on this file

is earlier than that, update the Citrix Xen guest agent service. For more information, see Upgrade your Citrix Xen guest agent service.

Workloads that leverage more than 20,000 disk IOPS experience degradation due to CPU bottlenecks

You can be affected by this issue if you are using Windows instances running AWS PV drivers that leverage more than 20,000 IOPS, and you experience bug check code `0x9E: USER_MODE_HEALTH_MONITOR`.

Disk reads and writes (IOs) in the AWS PV drivers occur in two phases: **IO preparation** and **IO completion**. By default, the preparation phase runs on a single arbitrary core. The completion phase runs on core `0`. The amount of computation required to process an IO varies based on it size and other properties. Some IOs use more computation in the preparation phase, and others in the completion phase. When an instance drives more than 20,000 IOPS, the preparation or completion phase may result in a bottleneck, where the CPU upon which it runs is at 100% capacity. Whether or not the preparation or completion phase becomes a bottleneck depends on the properties of the IOs used by the application.

Starting with AWS PV drivers 8.4.0, the load of the preparation phase and the completion phase can be distributed across multiple cores, eliminating bottlenecks. Each application uses different IO properties. Therefore, applying one of the following configurations may raise, lower, or not impact the performance of your application. After you apply any of these configurations, monitor the application to verify that it is meeting your desired performance.

1. **Prerequisites**

   Before you begin this troubleshooting procedure, verify the following prerequisites:

   - Your instance uses AWS PV drivers version 8.4.0 or later. To upgrade, see Upgrade PV drivers on Windows instances.
   - You have RDP access to the instance. For steps to connect to your Windows instance using RDP, see Connect to your Windows instance using RDP.
   - You have administrator access on the instance.

2. **Observe CPU load on your instance**

   You can use Windows Task Manager to view the load on each CPU to determine potential bottlenecks to disk IO.

1. Verify that your application is running and handling traffic similar to your production workload.
2. Connect to your instance using RDP.
3. Choose the **Start** menu on your instance.
4. Enter `Task Manager` in the **Start** menu to open Task Manager.
5. If Task Manager displays the Summary View, choose **More details** to expand the detailed view.
6. Choose the **Performance** tab.
7. Select **CPU** in the left pane.
8. Right-click on the graph in the main pane and select **Change graph to**>**Logical processors** to display each individual core.
9. Depending on how many cores are on your instance, you may see lines displaying CPU load over time, or you may just see a number.
   - If you see graphs displaying load over time, look for CPUs where the box is almost entirely shaded.
   - If you see a number on each core, look for cores that consistently show 95% or greater.

10Note whether core `0` or a different core is experiencing a heavy load.

3. **Choose which configuration to apply**

Configuration name	When to apply this configuration	Notes
Default configuration	Workload is driving less than 20,000 IOPS, or other configurations did not improve performance or stability.	For this configuration, IO occurs on a few cores, which may benefit smaller workloads by increasing cache locality and reducing context switching.
Allow driver to choose whether to distribute completion	Workload is driving more than 20,000 IOPS and moderate or high load is observed on core `0`.	This configuration is recommended for all Xen instances using PV 8.4.0 or later and leveraging more than 20,000 IOPS,

Configuration name	When to apply this configuration	Notes
		whether or not problems are encountered.
Distribute both preparation and completion	Workload is driving more than 20,000 IOPS, and either allowing the driver to choose the distribution did not improve performan ce, or a core other than `0` is experiencing a high load.	This configuration enables distribution of both IO preparation and IO completion.

> **Note**
>
> We recommend that you do not distribute IO preparation without also distributing IO completion (setting `DpcRedirection` without setting `NotifierDistributed`) because the completion phase is sensitive to overload by the preparation phase when the preparation phase is running in parallel.

Registry key values

- *NotifierDistributed*

  Value `0` or not present — The completion phase will run on core `0`.

  Value 1 — The driver chooses to run the completion phase or core `0` or one additional core per attached disk.

  Value 2 — The driver runs the completion phase on one additional core per attached disk.
- *DpcRedirection*

  Value `0` or not present — The preparation phase will run on a single, arbitrary core.

  Value 1 — The preparation phase is distributed across multiple cores.

Default configuration

Apply the default configuration with AWS PV driver versions prior to 8.4.0, or if performance or stability degradation is observed after applying one of the other configurations in this section.

1. Connect to your instance using RDP.
2. Open a new PowerShell command prompt as an administrator.
3. Run the following commands to remove the `NotifierDistributed` and `DpcRedirection` registry keys.

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name NotifierDistributed
```

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name DpcRedirection
```

4. Reboot your instance.

Allow driver to choose whether to distribute completion

Set `NotiferDistributed` registry key to allow the PV storage driver to choose whether or not to distribute IO completion.

1. Connect to your instance using RDP.
2. Open a new PowerShell command prompt as an administrator.
3. Run the following command to set the `NotiferDistributed` registry key.

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name NotifierDistributed
```

4. Reboot your instance.

Distribute both preparation and completion

Set `NotifierDistributed` and `DpcRedirection` registry keys to always distribute both the preparation and completion phases.

1. Connect to your instance using RDP.
2. Open a new PowerShell command prompt as an administrator.
3. Run the following commands to set the `NotifierDistributed` and `DpcRedirection` registry keys.

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000002 -Name NotifierDistributed
```

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name DpcRedirection
```

4. Reboot your instance.

AWS NVMe drivers for Windows instances

Amazon EBS volumes and instance store volumes are exposed as NVMe block devices on Nitro-based instances. Windows Server 2012 R2 and later include an NVMe driver, StorNVMe, that is provided by Microsoft. However, to achieve the full performance and features provided by Amazon EBS you must have the AWS NVMe driver installed when using an NVMe block device. The latest AWS Windows AMIs for Windows Server 2008 R2 and later contain the required AWS NVMe driver.

For more information about EBS and NVMe, see Amazon EBS and NVMe in the *Amazon EBS User Guide*. For more information about SSD instance store and NVMe, see SSD instance store volumes.

Install or upgrade AWS NVMe drivers using PowerShell

If you are not using the latest AWS Windows AMIs provided by Amazon, use the following procedure to install the current AWS NVMe driver. You should perform this update at a time when it is convenient to reboot your instance. Either the install script will reboot your instance or you must reboot it as the final step.

Prerequisites

PowerShell 3.0 or later

To download and install the latest AWS NVMe driver

1. We recommend that you create an AMI as a backup as follows, in case you need to roll back your changes.

   a. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

   b. In the navigation pane, choose **Instances**.

   c. Select the instance that requires the driver upgrade, and choose **Instance state**, **Stop instance**.

   d. After the instance is stopped, select the instance, choose **Actions**, then **Image and templates**, and then choose **Create image**.

   e. Choose **Instance state**, **Start instance**.

2. Connect to your instance and log in as the local administrator.

3. Download and extract the drivers to your instance using one of the following options:

   - Using a browser:

     a. Download the latest driver package to the instance.

     b. Extract the zip archive.

   - Using PowerShell:

     ```
     Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/NVMe/
     Latest/AWSNVMe.zip -outfile $env:USERPROFILE\nvme_driver.zip
     Expand-Archive $env:userprofile\nvme_driver.zip -DestinationPath
      $env:userprofile\nvme_driver
     ```

4. Install the driver to your instance by running the `install.ps1` PowerShell script from the `nvme_driver` directory (`.\install.ps1`). If you get an error, make sure you are using PowerShell 3.0 or later.

   a. (Optional) Starting with AWS NVMe version `1.5.0`, Small Computer System Interface (SCSI) persistent reservations are supported for Windows Server 2016 and later. This feature adds support for Windows Server Failover Clustering with shared Amazon EBS storage. By default, this feature isn't enabled during installation.

You can enable the feature when running the `install.ps1` script to install the driver by specifying the `EnableSCSIPersistentReservations` parameter with a value of `$true`.

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $true
```

You can disable the feature when running the `install.ps1` script to install the driver by specifying the `EnableSCSIPersistentReservations` parameter with a value of `$false`.

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $false
```

b. Starting with AWS NVMe `1.5.0`, the `install.ps1` script always installs the `ebsnvme-id` tool with the driver.

   (Optional) For versions `1.4.0`, `1.4.1`, and `1.4.2`, the `install.ps1` script allows you to specify whether the `ebsnvme-id` tool should be installed with the driver.

   i. To install the `ebsnvme-id` tool, specify `InstallEBSNVMeIdTool 'Yes'`.

   ii. If you don't want to install the tool, specify `InstallEBSNVMeIdTool 'No'`.

   If you don't specify `InstallEBSNVMeIdTool`, and the tool is already present at `C:\ProgramData\Amazon\Tools`, the package will upgrade the tool by default. If the tool is not present, `install.ps1` will not upgrade the tool by default.

   If you don't want to install the tool as part of the package, and want to install it later, you can find the latest version or the tool in the driver package. Alternatively, you can download version `1.0.0` from Amazon S3:

   Download the `ebsnvme-id` tool.

5. If the installer does not reboot your instance, reboot the instance.

Install or upgrade AWS NVMe drivers with Distributor

You can use Distributor, a capability of AWS Systems Manager, to install the NVMe driver package one time or with scheduled updates.

1. For the instructions for how to install the NVMe driver package using Distributor, see the procedures in Install or update packages in the *Amazon EC2 Systems Manager User Guide*.
2. For **Name**, choose **AWSNVMe**.
3. For installation type, select **Uninstall and reinstall**.
4. (Optional) Customize the installation by specifying values for `AdditionalArguments`.

   a. Starting with AWS NVMe `1.5.0`, the driver supports SCSI persistent reservations for Windows Server 2016 and later. By default, this feature isn't enabled during installation. To enable this feature, specify `{"SSM_EnableSCSIPersistentReservations": $true}` for `AdditionalArguments`. If you don't want to enable this feature, specify `{"SSM_EnableSCSIPersistentReservations": $false}` for `AdditionalArguments`.

   b. Starting with AWS NVMe `1.5.0`, the `install.ps1` script will always install the `ebsnvme-id` tool.

      (Optional) For versions `1.4.0`, `1.4.1`, and `1.4.2`, the `install.ps1` script allows you to specify whether the ebsnvme-id tool should be installed with the driver.

      i. To install the ebsnvme-id tool, specify `{"SSM_InstallEBSNVMeIdTool": "Yes"}` for `AdditionalArguments`.

      ii. If you don't want to install the tool, specify `{"SSM_InstallEBSNVMeIdTool": "No"}` for `AdditionalArguments`.

      If `SSM_InstallEBSNVMeIdTool` is not specified for `AdditionalArguments`, and the tool is already present at `C:\ProgramData\Amazon\Tools`, the package will upgrade the tool by default. If the tool is not present, the package will not upgrade the tool by default. Additional arguments must be formatted using valid JSON syntax. For examples of how to pass additional arguments for the `aws configure` package, see the Amazon EC2 Systems Manager documentation.

      If you don't want to install the tool as part of the package, and want to install it later, you can find the latest version of the tool in the driver package. Alternatively, you can download version `1.0.0` from Amazon S3:

      Download the `ebsnvme-id` tool.

5. If the installer does not reboot your instance, reboot the instance.

Configure SCSI persistent reservations

After AWS NVMe driver version `1.5.0` or later has been installed, you can enable or disable SCSI persistent reservations using the Windows registry for Windows Server 2016 and later. You must reboot the instance for these registry changes to take effect.

You can enable SCSI persistent reservations with the following command which sets the `EnableSCSIPersistentReservations` to a value of 1.

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 1
```

You can disable SCSI persistent reservations with the following command which sets the `EnableSCSIPersistentReservations` to a value of `0`.

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 0
```

AWS NVMe driver version history

The following table shows the corresponding NVMe driver version to download for each Windows Server version on Amazon EC2.

Windows Server version	AWSNVMe driver version
Windows Server 2022	latest
Windows Server 2019	latest
Windows Server 2016	latest
Windows Server 2012 R2	latest
Windows Server 2012	latest
Windows Server 2008 R2	1.3.2 and earlier

The following table describes the released versions of the AWS NVMe driver.

Package version	Driver version	Details	Release date
1.5.1	1.5.0	Fixed the install script to create a folder for the ebsnvme-id tool if it is not present.	17 November 2023
1.5.0	1.5.0	Added support for Small Computer System Interface (SCSI) persistent reservations for instances running Windows Server 2016 and later. The ebsnvme-id tool (`ebsnvme-id.exe`) is now installed by default.	31 August 2023
1.4.2	1.4.2	Fixed a bug where the AWS NVMe driver did not support instance store volumes on D3 instances.	16 March 2023
1.4.1	1.4.1	Reports Namespace Preferred Write Granularity (NPGW) for EBS volumes that support this optional NVMe feature. For more information, see section 8.25, "Improving Performance through I/O Size and Alignment Adherence," in the NVMe Base Specification, version 1.4.	20 May 2022
1.4.0	1.4.0	• Added support for IOCTLs that allow applications to interact with NVMe devices. This support allows applications to get `IdentifyController` , `IdentifyNamespace` , and `NameSpace` list from the NVMe device. For more information, see Protocol-specific queries in the Microsoft documentation. • AWSNVMe 1.4.0 installation on Windows Server 2008 R2 will fail. AWSNVMe version 1.3.2 and earlier are supported on Windows Server 2008 R2. • The 1.4.0 driver version and the latest `ebsnvme-id` tool (`ebsnvme-id.exe`) are combined in a single package. This combination allows you to install both driver and tool from a single package.	23 November 2021

Package version	Driver version	Details	Release date
		For more details, see Install or upgrade AWS NVMe drivers using PowerShell. • Bug fixes and reliability improvements.	
1.3.2	1.3.2	Fixed issue with modifying EBS volumes actively processing IO, which may result in data corruption. Customers who do not modify online EBS volumes (for example, resizing or changing type) are not impacted.	10 September 2019
1.3.1	1.3.1	Reliability Improvements.	21 May 2019
1.3.0	1.3.0	Device optimization improvements.	31 August 2018
1.2.0	1.2.0	Performance and reliability improvements for AWS NVMe devices on all supported instances, including bare metal instances.	13 June 2018
1.0.0	1.0.0	AWS NVMe driver for supported instance types running Windows Server.	12 February 2018

Subscribe to notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2 notifications from the console

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select this Region because the SNS notifications that you are subscribing to are in this Region.
3. In the navigation pane, choose **Subscriptions**.

4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
   a. For **TopicARN**, copy the following Amazon Resource Name (ARN):

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. For **Protocol**, choose `Email`.
   c. For **Endpoint**, type an email address that you can use to receive the notifications.
   d. Choose **Create subscription**.
6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Subscriptions**.
3. Select the check box for the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

To subscribe to EC2 notifications using the AWS CLI

To subscribe to EC2 notifications with the AWS CLI, use the following command.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers --
protocol email --notification-endpoint YourUserName@YourDomainName.ext
```

To subscribe to EC2 notifications using AWS Tools for Windows PowerShell

To subscribe to EC2 notifications with AWS Tools for Windows PowerShell, use the following command.

```
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-
drivers' -Protocol email -Region us-east-1 -Endpoint 'YourUserName@YourDomainName.ext'
```

Optimize CPU options

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
m6i.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
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
m6in.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
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
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

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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c4.xlarge	4	2	2	1, 2	1, 2
c4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c4.8xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5.large	2	1	2	1	1, 2
c5.xlarge	4	2	2	2	1, 2
c5.2xlarge	8	4	2	2, 4	1, 2
c5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5.12xlarge	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
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
c5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c5d.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c5d.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
c5n.large	2	1	2	1	1, 2
c5n.xlarge	4	2	2	2	1, 2
c5n.2xlarge	8	4	2	2, 4	1, 2
c5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5n.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5n.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
c6a.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
c6a.xlarge	4	2	2	1, 2	1, 2
c6a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c6a.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c6a.8xlarge	32	16	2	4, 6, 8, 10, 12, 14, 16	1, 2
c6a.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 16, 24	1, 2
c6a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 32	1, 2
c6a.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 32, 48	1, 2
c6a.32xlarge	128	64	2	8, 12, 16, 20, 24, 28, 32, 64	1, 2
c6a.48xlarge	192	96	2	8, 12, 16, 20, 24, 28, 32, 64, 96	1, 2
c6i.large	2	1	2	1	1, 2
c6i.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
c6i.2xlarge	8	4	2	2, 4	1, 2
c6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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
c7i.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
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
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
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
r6id.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
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
r7a.xlarge	4	4	1	1, 2, 3, 4	1
r7a.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
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
r7a.32xlarge	128	128	1	4, 6, 8, 10, 12, 14, 16, 32, 48, 64, 80, 96, 112, 128	1

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
r7a.48xlarge	192	192	1	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 48, 72, 96, 120, 144, 168, 192	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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`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
`u-6tb1.56xlarge`	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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
`u-6tb1.112xlarge`	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-9tb1.112xlarge`	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-12tb1.112xlarge`	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-18tb1.112xlarge`	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-24tb1.112xlarge	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
x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
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
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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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

Accelerated computing instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
g2.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g2.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
g3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
g3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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
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
p2.xlarge	4	2	2	1, 2	1, 2
p2.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid CPU cores	Valid threads per core
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

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 new launch template using parameters you define 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 Task Manager to view the CPU information for your instance.

You can use AWS Config to record, assess, audit, and evaluate configuration changes for instances, including terminated instances. For more information, see Getting Started with AWS Config in the *AWS Config Developer Guide*.

Set the time for your Windows instance

A consistent and accurate time reference on your Windows 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 Linux instances, see Set the time for your Linux 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
- 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 Nitro System.)

NTP 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 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.

Starting with the August 2018 release, Windows AMIs use the Amazon Time Sync Service by default. No further configuration is required for instances launched from these AMIs and you can skip the following procedures.

Contents

- Connect to the IPv4 endpoint of the Amazon Time Sync Service
- Default network time protocol (NTP) settings for Amazon Windows AMIs
- Connect to the PTP hardware clock

Connect to the IPv4 endpoint of the Amazon Time Sync Service

First verify your current NTP configuration. If your instance is already using the IPv4 endpoint of the Amazon Time Sync Service, no further configuration is required. If your instance is not using

the Amazon Time Sync Service, then complete the procedure to change the NTP server to use the Amazon Time Sync Service.

To verify the NTP configuration

1. From your instance, open a Command Prompt window.
2. Get the current NTP configuration by typing the following command:

   ```
   w32tm /query /configuration
   ```

   This command returns the current configuration settings for the Windows instance and will show if you're connected to the Amazon Time Sync Service.
3. (Optional) Get the status of the current configuration by typing the following command:

   ```
   w32tm /query /status
   ```

   This command returns information such as the last time the instance synced with the NTP server and the poll interval.

To change the NTP server to use the Amazon Time Sync Service

1. From the Command Prompt window, run the following command:

   ```
   w32tm /config /manualpeerlist:169.254.169.123 /syncfromflags:manual /update
   ```
2. Verify your new settings by using the following command:

   ```
   w32tm /query /configuration
   ```

   In the output that's returned, verify that `NtpServer` displays the `169.254.169.123` IPv4 endpoint.

Default network time protocol (NTP) settings for Amazon Windows AMIs

Amazon Machine Images (AMIs) generally adhere to the out-of-the-box defaults except in cases where changes are required to function on EC2 infrastructure. The following settings have been

determined to work well in a virtual environment, as well as to keep any clock drift to within one second of accuracy:

- **Update Interval** – Governs how frequently the time service will adjust system time towards accuracy. AWS configures the update interval to occur once every two minutes.
- **NTP Server** – Starting with the August 2018 release, AMIs use the Amazon Time Sync Service by default. This time service is accessible from any AWS Region at the 169.254.169.123 IPv4 endpoint. Additionally, the 0x9 flag indicates that the time service is acting as a client, and to use `SpecialPollInterval` to determine how frequently to check in with the configured time server.
- **Type** – "NTP" means that the service acts as a standalone NTP client instead of acting as part of a domain.
- **Enabled and InputProvider** – The time service is enabled and provides time to the operating system.
- **Special Poll Interval** – Checks against the configured NTP Server every 900 seconds, or 15 minutes.

Registry path	Key name	Data
HKLM:\System\CurrentControlSet\services\w32time\Config	UpdateInterval	120
HKLM:\System\CurrentControlSet\services\w32time\Parameters	NtpServer	169.254.169.123,0x9
HKLM:\System\CurrentControlSet\services\w32time\Parameters	Type	NTP
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	Enabled	1

Registry path	Key name	Data
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	InputProvider	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	SpecialPollInterval	900

Connect to the PTP hardware clock

Windows instances support only an NTP connection to the PTP hardware clock.

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 connect to an NTP endpoint and is running on an instance with a PTP hardware clock, it will automatically be connected to the PTP hardware clock over NTP.

Requirements

The PTP hardware clock is available on an instance when the following requirements are met:

- Supported AWS Region: Asia Pacific (Tokyo)
- Supported instance family: R7g

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.

Change the time zone on your instance

Windows 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.

To change the time zone on a Windows instance

1. From your instance, open a Command Prompt window.
2. Identify the time zone to use on the instance. To get a list of time zones, use the following command:

```
tzutil /l
```

This command returns a list of all available time zones in the following format:

```
display name
time zone ID
```

3. Locate the time zone ID to assign to the instance.
4. Assign to another time zone by using the following command:

```
tzutil /s "Pacific Standard Time"
```

The new time zone should take effect immediately.

> **Note**
>
> You can assign the UTC time zone by using the following command:
>
> ```
> tzutil /s "UTC"
> ```

To prevent your time zone from changing once you set it for Windows Server 2008 and later

When you change the time zone on a Windows instance, you must ensure that the time zone persists through system restarts. Otherwise, when the instance restarts, it reverts back to using UTC time. For Windows Server 2008 and later, you can persist your time zone setting by adding a **RealTimeIsUniversal** registry key. This key is set by default on all current generation instances. To verify whether the **RealTimeIsUniversal** registry key is set, see step 4 in the following procedure. If the key is not set, follow these steps from the beginning.

To set the RealTimeIsUniversal registry key

1. From the instance, open a Command Prompt window.
2. Use the following command to add the registry key:

```
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /
v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

3. If you are using a Windows Server 2008 AMI (*not* Windows Server 2008 R2) that was created before February 22, 2013, we recommend updating to the latest AWS Windows AMI. If you are using an AMI running Windows Server 2008 R2 (*not* Windows Server 2008), you must verify that the Microsoft hotfix KB2922223 is installed. If this hotfix is not installed, we recommend updating to the latest AWS Windows AMI.
4. (Optional) Verify that the instance saved the key successfully using the following command:

```
reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
\TimeZoneInformation" /s
```

This command returns the subkeys for the **TimeZoneInformation** registry key. You should see the **RealTimeIsUniversal** key at the bottom of the list, similar to the following:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation
    Bias                            REG_DWORD     0x1e0
    DaylightBias                    REG_DWORD     0xffffffc4
    DaylightName                    REG_SZ        @tzres.dll,-211
    DaylightStart                   REG_BINARY    00000300020002000000000000000000
    StandardBias                    REG_DWORD     0x0
    StandardName                    REG_SZ        @tzres.dll,-212
    StandardStart                   REG_BINARY    00000B000100020000000000000000
    TimeZoneKeyName                 REG_SZ        Pacific Standard Time
    DynamicDaylightTimeDisabled     REG_DWORD     0x0
    ActiveTimeBias                  REG_DWORD     0x1a4
    RealTimeIsUniversal             REG_DWORD     0x1
```

Leap seconds

Leap seconds, introduced in 1972, are occasional one-second adjustments to UTC time to factor in irregularities in the earth's rotation in order to accommodate differences between International Atomic Time (TAI) and solar time (Ut1). To manage leap seconds on behalf of customers, we designed leap second smearing within the Amazon Time Sync Service. For more information, see Look Before You Leap – The Coming Leap Second and AWS.

Leap seconds are going away, and we are in full support of the decision made at the 27th General Conference on Weights and Measures to abandon leap seconds by or before 2035.

To support this transition, we still plan on smearing time during a leap second event when accessing the Amazon Time Sync Service over the local NTP connection or our public NTP pools (`time.aws.com`). The PTP hardware clock, however, does not provide a smeared time option. In the event of a leap second, the PTP hardware clock will add the leap second following UTC standards. Leap-smeared and leap second time sources are the same in most cases. But because they differ during a leap second event, we do not recommend using both smeared and non-smeared time sources in your time client configuration during a leap second event.

Related resources

- How the Windows Time Service Works (Microsoft)
- W32tm (Microsoft)
- How the Windows Time service treats a leap second (Microsoft)
- The story around Leap Seconds and Windows: It's likely not Y2K (Microsoft)

Set the password for a Windows instance

When you connect to a Windows instance, you must specify a user account and password that has permission to access the instance. The first time that you connect to an instance, you are prompted to specify the Administrator account and the default password.

With AWS Windows AMIs for Windows Server 2012 R2 and earlier, the EC2Config service generates the default password. With AWS Windows AMIs for Windows Server 2016 and 2019, EC2Launch generates the default password. With AWS Windows AMIs for Windows Server 2022 and later, EC2Launch v2 generates the default password.

> **Note**
>
> With Windows Server 2016 and later, `Password never expires` is disabled for the local administrator. With Windows Server 2012 R2 and earlier, `Password never expires` is enabled for the local administrator.

Change the Administrator password after connecting

When you connect to an instance the first time, we recommend that you change the Administrator password from its default value. Use the following procedure to change the Administrator password for a Windows instance.

> **Important**
>
> Store the new password in a safe place. You won't be able to retrieve the new password using the Amazon EC2 console. The console can only retrieve the default password. If you attempt to connect to the instance using the default password after changing it, you'll get a "Your credentials did not work" error.

To change the local Administrator password

1. Connect to the instance and open a command prompt.
2. Run the following command. If your new password includes special characters, enclose the password in double quotes.

```
net user Administrator "new_password"
```

3. Store the new password in a safe place.

Change a lost or expired password

If you lose your password or it expires, you can generate a new password. For password reset procedures, see Reset a lost or expired Windows administrator password.

Add Windows components using installation media

Windows Server operating systems include many optional components. Including all optional components in each Amazon EC2 Windows Server AMI is not practical. Instead, we provide you with installation media EBS snapshots that have the necessary files to configure or install components on your Windows instance.

To access and install the optional components, you must find the correct EBS snapshot for your version of Windows Server, create a volume from the snapshot, and attach the volume to your instance.

Before you begin

Use the AWS Management Console or a command line tool to get the instance ID and Availability Zone of your instance. You must create your EBS volume in the same Availability Zone as your instance.

Add Windows components using the console

Use the following procedure to use the AWS Management Console to add Windows components to your instance.

To add Windows components to your instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Snapshots**.
3. From the **Filter** bar, choose **Public snapshots**.
4. Add the **Owner Alias** filter and choose **amazon**.
5. Add the **Description** filter and enter **`Windows`**.
6. Press Enter
7. Select the snapshot that matches your system architecture and language preference. For example, select **Windows 2019 English Installation Media** if your instance is running Windows Server 2019.
8. Choose **Actions**, **Create volume from snapshot**.
9. For **Availability Zone**, select the Availability Zone that matches your Windows instance. Choose **Add tag** and enter **`Name`** for the tag key and a descriptive name for the tag value. Choose **Create volume**.
10. In the **Successfully created volume** message (green banner), choose the volume that you just created.
11. Choose **Actions**, **Attach volume**.
12. From **Instance**, select the instance ID.
13. For **Device name**, enter the name of the device for the attachment. If you need help with the device name, see Device names on Windows instances.
14. Choose **Attach volume**.
15. Connect to your instance and make the volume available. For more information, see Make an Amazon EBS volume available for use in the *Amazon EBS User Guide*.

> **Important**
>
> Do not initialize the volume.

16. Open **Control Panel**, **Programs and Features**. Choose **Turn Windows features on or off**. If you are prompted for installation media, specify the EBS volume with the installation media.
17. (Optional) When you are finished with the installation media, you can detach the volume. After you detach the volume, you can delete it.

Add Windows components using the Tools for Windows PowerShell

Use the following procedure to use the Tools for Windows PowerShell to add Windows components to your instance.

To add Windows components to your instance using the Tools for Windows PowerShell

1. Use the Get-EC2Snapshot cmdlet with the `Owner` and `description` filters to get a list of the available installation media snapshots.

```
PS C:\> Get-EC2Snapshot -Owner amazon -Filter @{ Name="description";
 Values="Windows*" }
```

2. In the output, note the ID of the snapshot that matches your system architecture and language preference. For example:

```
...
DataEncryptionKeyId :
Description         : Windows 2019 English Installation Media
Encrypted           : False
KmsKeyId            :
OwnerAlias          : amazon
OwnerId             : 123456789012
Progress            : 100%
SnapshotId          : snap-22da283e
StartTime           : 10/25/2019 8:00:47 PM
State               : completed
StateMessage        :
Tags                : {}
VolumeId            : vol-be5eafcb
VolumeSize          : 6
```

```
...
```

3. Use the New-EC2Volume cmdlet to create a volume from the snapshot. Specify the same Availability Zone as your instance.

```
PS C:\> New-EC2Volume -AvailabilityZone us-east-1a -VolumeType gp2 -
SnapshotId snap-22da283e
```

4. In the output, note the volume ID.

```
Attachments      : {}
AvailabilityZone : us-east-1a
CreateTime       : 4/18/2017 10:50:25 AM
Encrypted        : False
Iops             : 100
KmsKeyId         :
Size             : 6
SnapshotId       : snap-22da283e
State            : creating
Tags             : {}
VolumeId         : vol-06aa9e1fbf8b82ed1
VolumeType       : gp2
```

5. Use the Add-EC2Volume cmdlet to attach the volume to your instance.

```
PS C:\> Add-EC2Volume -InstanceId i-087711ddaf98f9489 -
VolumeId vol-06aa9e1fbf8b82ed1 -Device xvdh
```

6. Connect to your instance and make the volume available. For more information, see Make an Amazon EBS volume available for use in the *Amazon EBS User Guide*.

> **Important**
>
> Do not initialize the volume.

7. Open **Control Panel**, **Programs and Features**. Choose **Turn Windows features on or off**. If you are prompted for installation media, specify the EBS volume with the installation media.
8. (Optional) When you are finished with the installation media, use the Dismount-EC2Volume cmdlet to detach the volume from your instance. After you detach the volume, you can use the Remove-EC2Volume cmdlet to delete the volume.

Add Windows components using the AWS CLI

Use the following procedure to use the AWS CLI to add Windows components to your instance.

To add Windows components to your instance using the AWS CLI

1. Use the describe-snapshots command with the `owner-ids` parameter and `description` filter to get a list of the available installation media snapshots.

```
aws ec2 describe-snapshots --owner-ids amazon --filters
 Name=description,Values=Windows*
```

2. In the output, note the ID of the snapshot that matches your system architecture and language preference. For example:

```
{
    "Snapshots": [
    ...
        {
            "OwnerAlias": "amazon",
            "Description": "Windows 2019 English Installation Media",
            "Encrypted": false,
            "VolumeId": "vol-be5eafcb",
            "State": "completed",
            "VolumeSize": 6,
            "Progress": "100%",
            "StartTime": "2019-10-25T20:00:47.000Z",
            "SnapshotId": "snap-22da283e",
            "OwnerId": "123456789012"
        },
    ...
   ]
}
```

3. Use the create-volume command to create a volume from the snapshot. Specify the same Availability Zone as your instance.

```
aws ec2 create-volume --snapshot-id snap-22da283e --volume-type gp2 --availability-
zone us-east-1a
```

4. In the output, note the volume ID.

```
{
    "AvailabilityZone": "us-east-1a",
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-0c98b37f30bcbc290",
    "State": "creating",
    "Iops": 100,
    "SnapshotId": "snap-22da283e",
    "CreateTime": "2017-04-18T10:33:10.940Z",
    "Size": 6
}
```

5. Use the attach-volume command to attach the volume to your instance.

```
aws ec2 attach-volume --volume-id vol-0c98b37f30bcbc290 --instance-
id i-01474ef662b89480 --device xvdg
```

6. Connect to your instance and make the volume available. For more information, see Make an Amazon EBS volume available for use in the *Amazon EBS User Guide*.

> **Important**
>
> Do not initialize the volume.

7. Open **Control Panel**, **Programs and Features**. Choose **Turn Windows features on or off**. If you are prompted for installation media, specify the EBS volume with the installation media.
8. (Optional) When you are finished with the installation media, use the detach-volume command to detach the volume from your instance. After you detach the volume, you can use the delete-volume command to delete the volume.

Configure a secondary private IPv4 address for your Windows instance

You can specify multiple private IPv4 addresses for your instances. After you assign a secondary private IPv4 address to an instance, you must configure the operating system on the instance to recognize the secondary private IPv4 address.

Configuring the operating system on a Windows instance to recognize a secondary private IPv4 address requires the following:

Topics

- Prerequisite steps
- Step 1: Configure static IP addressing on your instance
- Step 2: Configure a secondary private IP address for your instance
- Step 3: Configure applications to Use the secondary private IP address

> **Note**
>
> These instructions are based on Windows Server 2008 R2. The implementation of these steps may vary based on the operating system of the Windows instance.

Before you begin

As a best practice, launch your Windows instances using the latest AMIs. If you are using an older Windows AMI, ensure that it has the Microsoft hot fix referenced in http://support.microsoft.com/kb/2582281.

Prerequisite steps

1. Assign the secondary private IPv4 address to the network interface for the instance. You can assign the secondary private IPv4 address when you launch the instance, or after the instance is running. For more information, see Assign a secondary private IPv4 address.
2. Allocate an Elastic IP address and associate it with the secondary private IPv4 address. For more information, see Allocate an Elastic IP address and Associate an Elastic IP address with the secondary private IPv4 address.

Step 1: Configure static IP addressing on your instance

To enable your Windows instance to use multiple IP addresses, you must configure your instance to use static IP addressing rather than a DHCP server.

> **Important**
>
> When you configure static IP addressing on your instance, the IP address must match exactly what is shown in the console, CLI, or API. If you enter these IP addresses incorrectly, the instance could become unreachable.

To configure static IP addressing on a Windows instance

1. Connect to your instance.
2. Find the IP address, subnet mask, and default gateway addresses for the instance by performing the following steps:

   - At a Command Prompt window, run the following command:

     ```
     ipconfig /all
     ```

     Review the following section in your output, and note the **IPv4 Address**, **Subnet Mask**, **Default Gateway**, and **DNS Servers** values for the network interface.

     ```
     Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . :
        Physical Address  . . . . . . . . :
        DHCP Enabled. . . . . . . . . . . :
        Autoconfiguration Enabled . . . . :
        IPv4 Address. . . . . . . . . . . : 10.0.0.131
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.0.0.1
        DNS Servers . . . . . . . . . . . : 10.1.1.10
                                            10.1.1.20
     ```

3. Open the **Network and Sharing Center** by running the following command:

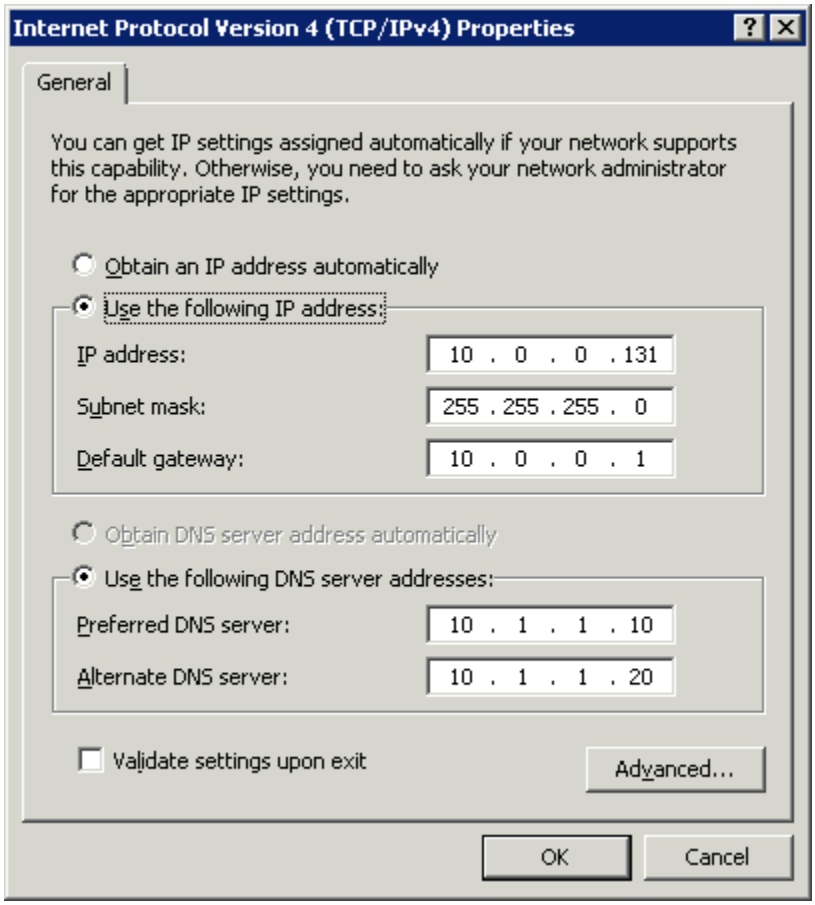
   ```
   %SystemRoot%\system32\control.exe ncpa.cpl
   ```

4. Open the context (right-click) menu for the network interface (Local Area Connection) and choose **Properties**.
5. Choose **Internet Protocol Version 4 (TCP/IPv4)**, **Properties**.

6. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, choose **Use the following IP address**, enter the following values, and then choose **OK**.

Field	Value
IP address	The IPv4 address obtained in step 2 above.
Subnet mask	The subnet mask obtained in step 2 above.
Default gateway	The default gateway address obtained in step 2 above.
Preferred DNS server	The DNS server obtained in step 2 above.
Alternate DNS server	The alternate DNS server obtained in step 2 above. If an alternate DNS server was not listed, leave this field blank.

> **Important**
>
> If you set the IP address to any value other than the current IP address, you will lose connectivity to the instance.



You will lose RDP connectivity to the Windows instance for a few seconds while the instance converts from using DHCP to static addressing. The instance retains the same IP address information as before, but now this information is static and not managed by DHCP.

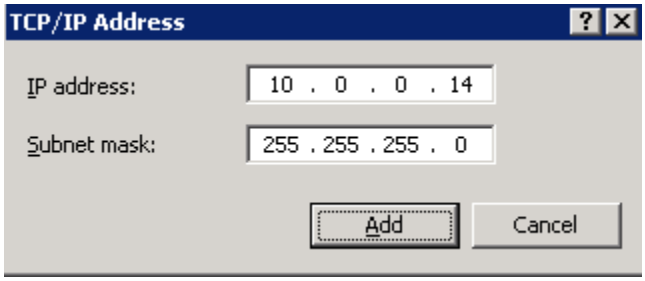
Step 2: Configure a secondary private IP address for your instance

After you have set up static IP addressing on your Windows instance, you are ready to prepare a second private IP address.

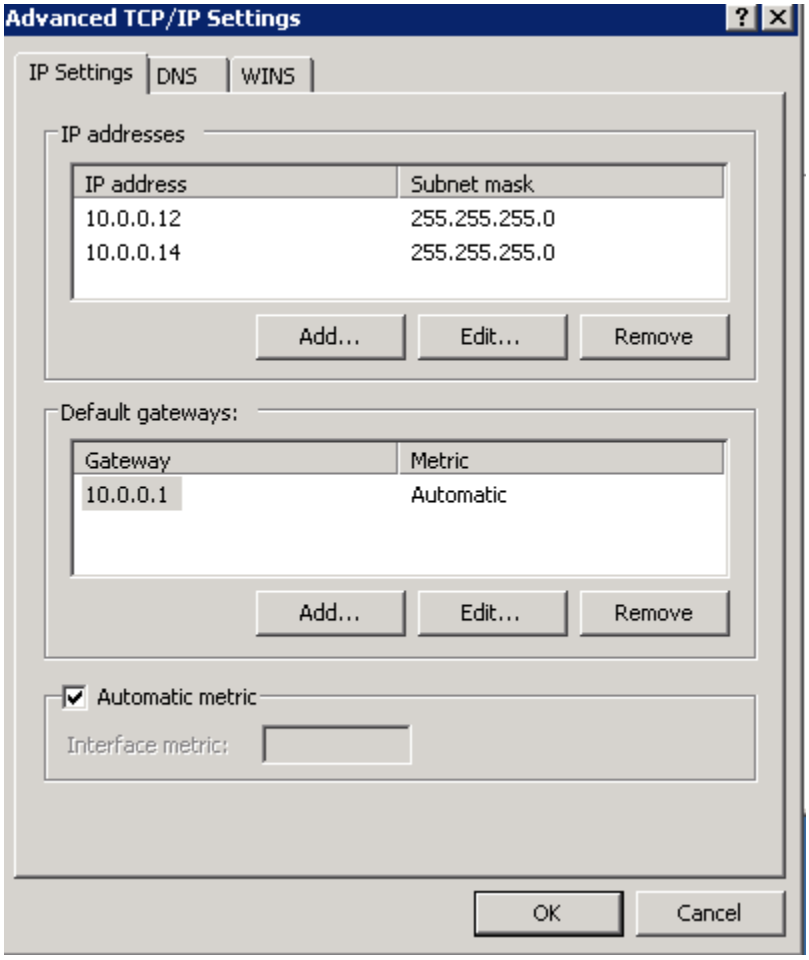
To configure a secondary IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select your instance.
3. On the **Networking**, note the secondary IP address.
4. Connect to your instance.

5. On your Windows instance, choose **Start**, **Control Panel**.
6. Choose **Network and Internet**, **Network and Sharing Center**.
7. Select the network interface (Local Area Connection) and choose **Properties**.
8. On the **Local Area Connection Properties** page, choose **Internet Protocol Version 4 (TCP/IPv4)**, **Properties**, **Advanced**.
9. Choose **Add**.
10. In the **TCP/IP Address** dialog box, type the secondary private IP address for **IP address**. For **Subnet mask**, type the same subnet mask that you entered for the primary private IP address in Step 1: Configure static IP addressing on your instance, and then choose **Add**.



11. Verify the IP address settings and choose **OK**.



12. Choose **OK**, **Close**.
13. To confirm that the secondary IP address has been added to the operating system, at a command prompt, run the command **ipconfig /all**.

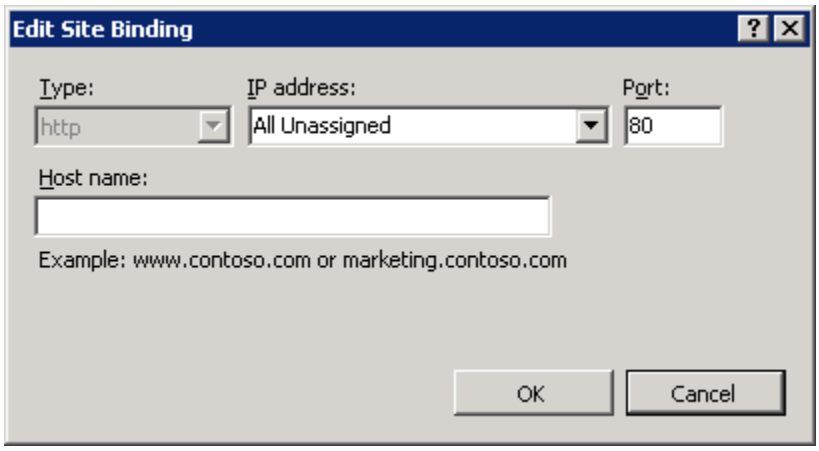
Step 3: Configure applications to Use the secondary private IP address

You can configure any applications to use the secondary private IP address. For example, if your instance is running a website on IIS, you can configure IIS to use the secondary private IP address.

To configure IIS to use the secondary private IP address

1. Connect to your instance.
2. Open Internet Information Services (IIS) Manager.
3. In the **Connections** pane, expand **Sites**.
4. Open the context (right-click) menu for your website and choose **Edit Bindings**.
5. In the **Site Bindings** dialog box, for **Type**, choose **http**, **Edit**.

6. In the **Edit Site Binding** dialog box, for **IP address**, select the secondary private IP address. (By default, each website accepts HTTP requests from all IP addresses.)



7. Choose **OK**, **Close**.

Run commands on your Windows instance at launch

When you launch a Windows instance using Amazon EC2, you can pass user data to the instance that can be used to perform automated configuration tasks, or to run scripts after the instance starts. Instance user data is treated as opaque data; it is up to the instance to interpret it. User data is processed by EC2Launch v2 on Windows Server 2022, EC2Launch on Windows Server 2016 and 2019, and EC2Config on Windows Server 2012 R2 and earlier.

For examples of the assembly of a `UserData` property in a AWS CloudFormation template, see Base64 Encoded UserData Property and Base64 Encoded UserData Property with AccessKey and SecretKey.

For information about running commands on your Linux instance at launch, see Running commands on your Linux instance at launch in the *Amazon EC2 User Guide for Linux Instances*.

For an example of running commands on an instance within an Auto Scaling that work with lifecycle hooks, see Tutorial: Configure user data to retrieve the target lifecycle state through instance metadata in the *Amazon EC2 Auto Scaling User Guide*.

Contents

- User data scripts
- User data execution
- User data and the console

- User data and the Tools for Windows PowerShell

User data scripts

For EC2Config or EC2Launch to run scripts, you must enclose the script within a special tag when you add it to user data. The tag that you use depends on whether the commands run in a Command Prompt window (batch commands) or use Windows PowerShell.

If you specify both a batch script and a Windows PowerShell script, the batch script runs first and the Windows PowerShell script runs next, regardless of the order in which they appear in the instance user data.

If you use an AWS API, including the AWS CLI, in a user data script, you must use an instance profile when launching the instance. An instance profile provides the appropriate AWS credentials required by the user data script to make the API call. For more information, see Instance profiles. The permissions you assign to the IAM role depend on which services you are calling with the API. For more information, see IAM roles for Amazon EC2.

Script type

- Syntax for batch scripts
- Syntax for Windows PowerShell scripts
- Syntax for YAML configuration scripts
- Base64 encoding

Syntax for batch scripts

Specify a batch script using the `script` tag. Separate the commands using line breaks as shown in the following example.

```
<script>
echo Current date and time >> %SystemRoot%\Temp\test.log
echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
```

By default, user data scripts run one time when you launch the instance. To run the user data scripts every time you reboot or start the instance, add `<persist>true</persist>` to the user data.

```
<script>
echo Current date and time >> %SystemRoot%\Temp\test.log
echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<persist>true</persist>
```

EC2Launch v2 agent

To run an XML user data script as a detached process with the EC2Launch v2 **executeScript** task in the `UserData` stage, add the following tag to your user data.

`<detach>true</detach>`

> **Note**
>
> The detach tag is not supported on previous launch agents.

```
<script>
echo Current date and time >> %SystemRoot%\Temp\test.log
echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<detach>true</detach>
```

Syntax for Windows PowerShell scripts

The AWS Windows AMIs include the AWS Tools for Windows PowerShell, so you can specify these cmdlets in user data. If you associate an IAM role with your instance, you don't need to specify credentials to the cmdlets, as applications that run on the instance use the role's credentials to access AWS resources (for example, Amazon S3 buckets).

Specify a Windows PowerShell script using the `<powershell>` tag. Separate the commands using line breaks. The `<powershell>` tag is case-sensitive.

For example:

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
```

By default, the user data scripts are run one time when you launch the instance. To run the user data scripts every time you reboot or start the instance, add `<persist>true</persist>` to the user data.

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

EC2Launch v2 agent

To run an XML user data script as a detached process with the EC2Launch v2 **executeScript** task in the `UserData` stage, add the following tag to your user data.

`<detach>true</detach>`

> **Note**
>
> The detach tag is not supported on previous launch agents.

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

Syntax for YAML configuration scripts

If you are using EC2Launch v2 to run scripts, you can use the YAML format. To view configuration tasks, details, and examples for EC2Launch v2, see EC2Launch v2 task configuration.

Specify a YAML script with the `executeScript` task.

Example YAML syntax to run a PowerShell script

```
version: 1.0
tasks:
- task: executeScript
  inputs:
  - frequency: always
```

```
    type: powershell
    runAs: localSystem
    content: |-
      $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
      New-Item $file -ItemType file
```

Example YAML syntax to run a batch script

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: batch
    runAs: localSystem
    content: |-
      echo Current date and time >> %SystemRoot%\Temp\test.log
      echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
```

Base64 encoding

If you're using the Amazon EC2 API or a tool that does not perform base64 encoding of the user data, you must encode the user data yourself. If not, an error is logged about being unable to find `script` or `powershell` tags to run. The following is an example that encodes using Windows PowerShell.

```
$UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

The following is an example that decodes using PowerShell.

```
$Script =
 [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData))
```

For more information about base64 encoding, see https://www.ietf.org/rfc/rfc4648.txt.

User data execution

By default, all AWS Windows AMIs have user data execution enabled for the initial launch. You can specify that user data scripts are run the next time the instance reboots or restarts. Alternatively, you can specify that user data scripts are run every time the instance reboots or restarts.

> **Note**
>
> User data is not enabled to run by default after the initial launch. To enable user data to run when you reboot or start the instance, see Subsequent reboots or starts.

User data scripts are run from the local administrator account when a random password is generated. Otherwise, user data scripts are run from the System account.

Instance launch

Scripts in the instance user data are run during the initial launch of the instance. If the `persist` tag is found, user data execution is enabled for subsequent reboots or starts. The log files for EC2Launch v2, EC2Launch, and EC2Config contain the output from the standard output and standard error streams.

EC2Launch v2

The log file for EC2Launch v2 is `C:\ProgramData\Amazon\EC2Launch\log\agent.log`.

> **Note**
>
> The `C:\ProgramData` folder might be hidden. To view the folder, you must show hidden files and folders.

The following information is logged when the user data is run:

- `Info: Converting user-data to yaml format` – If the user data was provided in XML format
- `Info: Initialize user-data state` – The start of user data execution
- `Info: Frequency is: always` – If the user data task is running on every boot
- `Info: Frequency is: once` – If the user data task is running just once
- `Stage: postReadyUserData execution completed` – The end of user data execution

EC2Launch

The log file for EC2Launch is `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\UserdataExecution.log`.

The `C:\ProgramData` folder might be hidden. To view the folder, you must show hidden files and folders.

The following information is logged when the user data is run:

- `Userdata execution begins` – The start of user data execution
- `<persist> tag was provided: true` – If the persist tag is found
- `Running userdata on every boot` – If the persist tag is found
- `<powershell> tag was provided.. running powershell content` – If the powershell tag is found
- `<script> tag was provided.. running script content` – If the script tag is found
- `Message: The output from user scripts` – If user data scripts are run, their output is logged

EC2Config

The log file for EC2Config is `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2Config.log`. The following information is logged when the user data is run:

- `Ec2HandleUserData: Message: Start running user scripts` – The start of user data execution
- `Ec2HandleUserData: Message: Re-enabled userdata execution` – If the persist tag is found
- `Ec2HandleUserData: Message: Could not find <persist> and </persist>` – If the persist tag is not found
- `Ec2HandleUserData: Message: The output from user scripts` – If user data scripts are run, their output is logged

Subsequent reboots or starts

When you update instance user data, user data scripts are not run automatically when you reboot or start the instance. However, you can enable user data execution so that user data scripts are run one time when you reboot or start the instance, or every time you reboot or start the instance.

If you choose the **Shutdown with Sysprep** option, user data scripts are run the next time the instance starts or reboots, even if you did not enable user data execution for subsequent reboots or starts. The user data scripts will not be executed on subsequent reboots or starts.

To enable user data execution with EC2Launch v2 (Preview AMIs)

- To run a task in user data on first boot, set `frequency` to `once`.
- To run a task in user data on every boot, set `frequency` to `always`.

To enable user data execution with EC2Launch (Windows Server 2016 or later)

1. Connect to your Windows instance.
2. Open a PowerShell command window and run the following command:

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

3. Disconnect from your Windows instance. To run updated scripts the next time the instance is started, stop the instance and update the user data. For more information, see View and update the instance user data.

To enable user data execution with EC2Config (Windows Server 2012 R2 and earlier)

1. Connect to your Windows instance.
2. Open `C:\Program Files\Amazon\Ec2ConfigService\Ec2ConfigServiceSetting.exe`.
3. For **User Data**, select **Enable UserData execution for next service start**.
4. Disconnect from your Windows instance. To run updated scripts the next time the instance is started, stop the instance and update the user data. For more information, see View and update the instance user data.

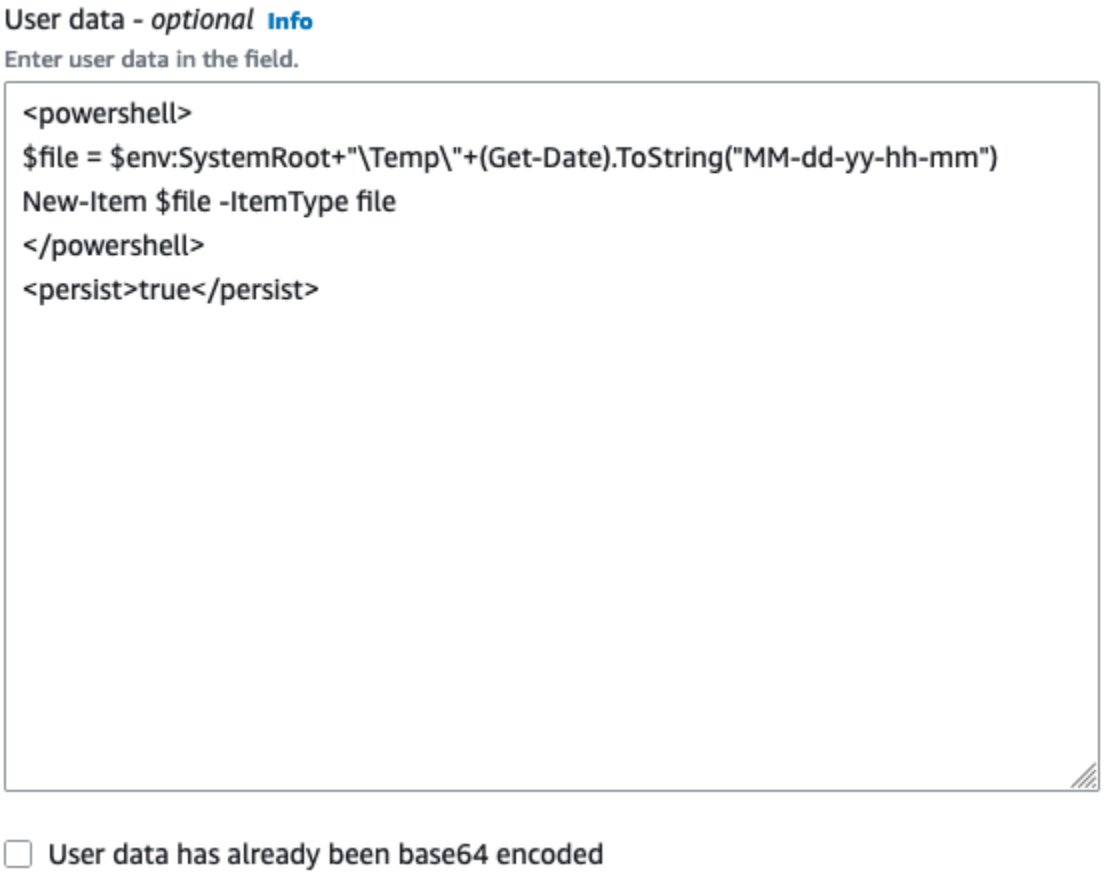
User data and the console

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 PowerShell script in the **User data** field, and then complete the instance launch procedure.

In the following screenshot of the **User data** field, the example script creates a file in the Windows temporary folder, using the current date and time in the file name. When you include `<persist>true</persist>`, the script is run every time you reboot or start the instance. If you leave the **User data has already been base64 encoded** check box empty, the Amazon EC2 console performs the base64 encoding for you.



View and update the instance user data

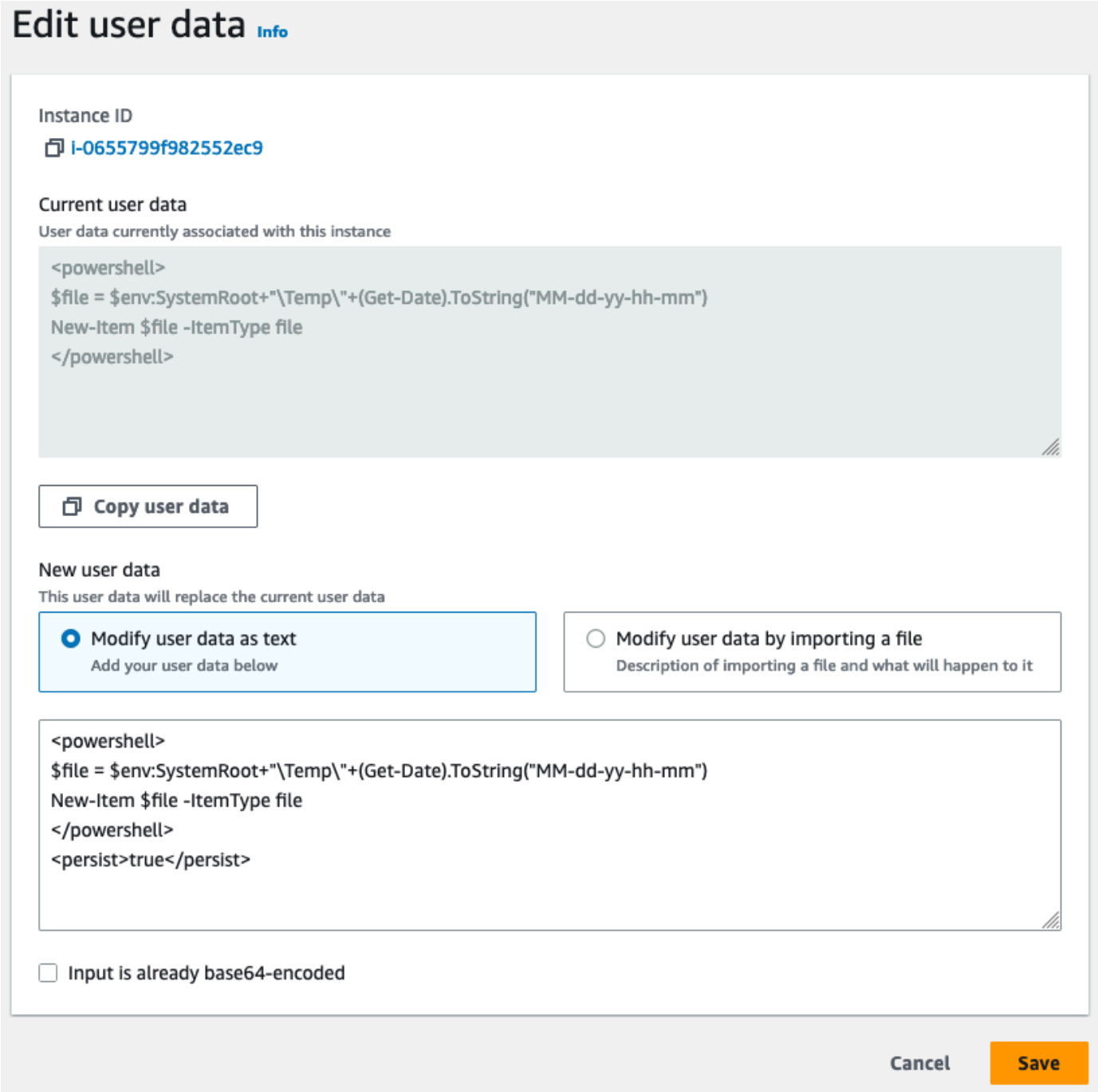
You can view the instance user data for any instance, and you can update the instance user data for a stopped instance.

To update the user data for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions**, **Instance state**, **Stop instance**.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions**, **Instance settings**, **Edit user data**. You can't change the user data if the instance is running, but you can view it.
6. In the **Edit user data** dialog box, update the user data, and then choose **Save**. To run user data scripts every time you reboot or start the instance, add `<persist>true</persist>`, as shown in the following example:



7. Start the instance. If you enabled user data execution for subsequent reboots or starts, the updated user data scripts are run as part of the instance start process.

User data and the Tools for Windows PowerShell

You can use the Tools for Windows PowerShell to specify, modify, and view the user data for your instance. For information about viewing user data from your instance using instance metadata, see

Retrieve instance user data from your instance. For information about user data and the AWS CLI, see User data and the AWS CLI in the *Amazon EC2 User Guide for Linux Instances*.

Example: Specify instance user data at launch

Create a text file with the instance user data. To run user data scripts every time you reboot or start the instance, add `<persist>true</persist>`, as shown in the following example.

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

To specify instance user data when you launch your instance, use the New-EC2Instance command. This command does not perform base64 encoding of the user data for you. Use the following commands to encode the user data in a text file named `script.txt`.

```
PS C:\> $Script = Get-Content -Raw script.txt
PS C:\> $UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

Use the `-UserData` parameter to pass the user data to the **New-EC2Instance** command.

```
PS C:\> New-EC2Instance -ImageId ami-abcd1234 -MinCount 1 -MaxCount 1 -
InstanceType m3.medium \
-KeyName my-key-pair -SubnetId subnet-12345678 -SecurityGroupIds sg-1a2b3c4d \
-UserData $UserData
```

Example: Update instance user data for a stopped instance

You can modify the user data of a stopped instance using the Edit-EC2InstanceAttribute command.

Create a text file with the new script. Use the following commands to encode the user data in the text file named `new-script.txt`.

```
PS C:\> $NewScript = Get-Content -Raw new-script.txt
PS C:\> $NewUserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($NewScript))
```

Use the `-UserData` and `-Value` parameters to specify the user data.

```
PS C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute userData -
Value $NewUserData
```

Example: View instance user data

To retrieve the user data for an instance, use the Get-EC2InstanceAttribute command.

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute
 userData).UserData
```

The following is example output. Note that the user data is encoded.

```
PHBvd2Vyc2hlbGw
+DQpSZW5hbWUtQ29tcHV0ZXIgLU5ld05hbWUgdXNlci1kYXRhLXRlc3QNCjwvcG93ZXJzaGVsbD4=
```

Use the following commands to store the encoded user data in a variable and then decode it.

```
PS C:\> $UserData_encoded = (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -
Attribute userData).UserData
PS C:
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData_encoded)
```

The following is example output.

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

Example: Rename the instance to match the tag value

You can use the Get-EC2Tag command to read the tag value, rename the instance on first boot to match the tag value, and reboot. To run this command successfully, you must have a role with `ec2:DescribeTags` permissions attached to the instance because tag information is retrieved by the API call. For more information on settings permissions by using IAM roles, see Attaching an IAM Role to an Instance.

> **Note**
>
> This script fails on Windows Server versions prior to 2008.

```
<powershell>
$instanceId = (invoke-webrequest http://169.254.169.254/latest/meta-data/instance-id -
UseBasicParsing).content
$nameValue = (get-ec2tag -filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
$pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
#Verify Name Value satisfies best practices for Windows hostnames
If ($nameValue -match $pattern)
    {Try
        {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
    Catch
        {$ErrorMessage = $_.Exception.Message
        Write-Output "Rename failed: $ErrorMessage"}}
Else
    {Throw "Provided name not a valid hostname. Please ensure Name value is between 1
 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

You can also rename the instance using tags in instance metadata, if your instance is configured to access tags from the instance metadata.

> **Note**
>
> This script fails on Windows Server versions prior to 2008.

```
<powershell>
$nameValue = Get-EC2InstanceMetadata -Path /tags/instance/Name
$pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
#Verify Name Value satisfies best practices for Windows hostnames
If ($nameValue -match $pattern)
    {Try
        {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
    Catch
        {$ErrorMessage = $_.Exception.Message
        Write-Output "Rename failed: $ErrorMessage"}}
```

```
Else
    {Throw "Provided name not a valid hostname. Please ensure Name value is between 1
 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

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.

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 Nitro System.

Contents

- Use IMDSv2
- Configure the instance metadata options
- Retrieve instance metadata
- Work with instance user data
- Retrieve dynamic data
- Instance metadata categories
- 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 Nitro System.

The following example uses a PowerShell shell script and IMDSv2 to retrieve the top-level instance metadata items. The example:

- Creates a session token lasting six hours (21,600 seconds) using the PUT request
- Stores the session token header in a variable named `token`
- Requests the top-level metadata items using the token

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

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

After you've created a token, you can reuse it until it expires. In the following example command, which gets the ID of the AMI used to launch the instance, the token that is stored in `$token` in the previous example is reused.

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
-Method GET -uri http://169.254.169.254/latest/meta-data/ami-id
```

When you use IMDSv2 to request instance metadata, the request must include the following:

1. Use a PUT request to initiate a session to the instance metadata service. The PUT request returns a token that must be included in subsequent GET requests to the instance metadata service. The token is required to access metadata using IMDSv2.
2. Include the token in all GET requests to the IMDS. When token usage is set to `required`, requests without a valid token or with an expired token receive a `401 - Unauthorized` HTTP error code.
   - The token is an instance-specific key. The token is not valid on other EC2 instances and will be rejected if you attempt to use it outside of the instance on which it was generated.
   - The PUT request must include a header that specifies the time to live (TTL) for the token, in seconds, up to a maximum of six hours (21,600 seconds). The token represents a logical session. The TTL specifies the length of time that the token is valid and, therefore, the duration of the session.
   - After a token expires, to continue accessing instance metadata, you must create a new session using another PUT.
   - You can choose to reuse a token or create a new token with every request. For a small number of requests, it might be easier to generate and immediately use a token each time you need to access the IMDS. But for efficiency, you can specify a longer duration for the token and reuse it rather than having to write a PUT request every time you need to request instance metadata. There is no practical limit on the number of concurrent tokens, each representing its own session. IMDSv2 is, however, still constrained by normal IMDS connection and throttling limits. For more information, see Query throttling.

HTTP GET and HEAD methods are allowed in IMDSv2 instance metadata requests. PUT requests are rejected if they contain an X-Forwarded-For header.

By default, the response to PUT requests has a response hop limit (time to live) of 1 at the IP protocol level. If you need a bigger hop limit, you can adjust it by using the modify-instance-

metadata-options AWS CLI command. For example, you might need a bigger hop limit for backward compatibility with container services running on the instance. For more information, see Modify instance metadata options for existing instances.

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 software packages support IMDSv2.

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:

- **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 modify-instance-metadata-defaults CLI command to set IMDSv2 is to be used.
- **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 to stay up to date with SDK releases to keep up with the latest features, security updates, and underlying dependencies. Continued use of an unsupported SDK version is not recommended and is done at your discretion. For more information, see the AWS SDKs and Tools maintenance policy in the *AWS SDKs and Tools Reference Guide*.

The following are the minimum versions that support using IMDSv2:

- AWS CLI – 1.16.289
- AWS Tools for Windows PowerShell – 4.0.1.0
- AWS SDK for .NET – 3.3.634.1
- AWS SDK for C++ – 1.7.229
- AWS SDK for Go – 1.25.38
- AWS SDK for Go v2 – 0.19.0
- AWS SDK for Java – 1.11.678
- AWS SDK for Java 2.x – 2.10.21
- AWS SDK for JavaScript in Node.js – 2.722.0
- AWS SDK for PHP – 3.147.7
- AWS SDK for Python (Boto) – 1.13.25
- AWS SDK for Python (Boto3) – 1.12.6
- AWS SDK for Ruby – 3.79.0

Configure the instance metadata options

Instance metadata is data about an instance that you can use to configure or manage the instance. You can configure instance metadata options for new or existing instances as follows:

- Require the use of IMDSv2 when requesting instance metadata
- Specify the PUT response hop limit
- Turn off access to instance metadata

You can also use IAM condition keys in an IAM policy or SCP 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

> **Note**
>
> If your PowerShell version is earlier than 4.0, you must update to Windows Management Framework 4.0 to require the use of IMDSv2.

> **Note**
>
> You should proceed cautiously and conduct careful testing before making any changes. Take note of the following:
>
> - If you enforce the use of IMDSv2, applications or agents that use IMDSv1 for instance metadata access will break.
> - If you turn off all access to instance metadata, applications or agents that rely on instance metadata access to function will break.
> - For IMDSv2, you must use `/latest/api/token` when retrieving the token.

Tasks

- Configure instance metadata options for new instances
- Modify instance metadata options for existing instances

Configure instance metadata options for new instances

You can configure the following instance metadata options.

Options

- Require the use of IMDSv2
- Configure IPv4 and IPv6 endpoints
- Turn off access to instance metadata

Require the use of IMDSv2

Use one of the following methods to require the use of IMDSv2 on your instances.

To require IMDSv2

- Configure the instance at launch
- Configure the AMI
- Use an IAM policy

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 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 `
    -Name 'my-image' `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping  (
    New-Object `
        -TypeName Amazon.EC2.Model.BlockDeviceMapping `
        -Property @{
        DeviceName = '/dev/xvda';
        EBS        = (New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property
 @{
                SnapshotId = 'snap-0123456789example;
                VolumeType = 'gp3'
                } )
        }  ) `
    -Architecture X86_64 `
    -ImdsSupport v2.0
```

To configure an existing AMI for IMDSv2

Use one of the following methods to configure an existing AMI for IMDSv2.

AWS CLI

The following modify-image-attribute example modifies an existing AMI for IMDSv2 only. Specify `v2.0` for the `imds-support` parameter so that instances launched from this AMI will require that IMDSv2 is used when requesting instance metadata.

```
aws ec2 modify-image-attribute \
    --image-id ami-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 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.

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 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
```

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 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 PowerShell cmdlets to retrieve the URI. For example, if you are running version 3.0 or later of PowerShell, use the following cmdlet.

IMDSv2

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

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

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
```

If you don't want to use PowerShell, you can install a third-party tool such as GNU Wget or cURL.

> **Important**
>
> If you install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the method of calling the HTTP and the output format might be different from what is documented here.

For the command to retrieve instance metadata from a Linux 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.

- **Create custom Windows AMIs using Sysprep.**

  If you launch a Windows instance using a custom Windows AMI, to ensure that the IMDS works on the instance, the AMI must be a standardized image created using Sysprep. Otherwise, the IMDS won't work.

- **For IMDSv2, you must use `/latest/api/token` when retrieving the token.**

  Issuing PUT requests to any version-specific path, for example `/2021-03-23/api/token`, will result in the metadata service returning 403 Forbidden errors. This behavior is intended.

- **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 Windows instance. For the commands to retrieve instance metadata from a Linux 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

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
```

```
2016-04-19
...
latest
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

Get the top-level metadata items

This example gets the top-level metadata items. For more information, see Instance metadata categories.

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
```

```
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
```

```
reservation-id
security-groups
services/
```

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

IMDSv2

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/reservation-
id
r-0efghijk987654321
```

IMDSv2

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/local-
hostname
ip-10-251-50-12.ec2.internal
```

IMDSv2

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

Get the list of available public keys

This example gets the list of available public keys.

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/ 0=my-public-key
```

Show the formats in which public key 0 is available

This example shows the formats in which public key 0 is available.

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
openssh-key
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
openssh-key
```

Get public key 0 (in the OpenSSH key format)

This example gets public key 0 (in the OpenSSH key format).

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
```

```
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

Get the subnet ID for an instance

This example gets the subnet ID for an instance.

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -
Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
```

```
subnet-be9b61d7
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/network/
interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

Get the instance tags for an instance

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

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/tags/instance
Name
Environment
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri 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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/tags/instance/Name
```

```
MyInstance
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri 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 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 Windows firewall to limit access

The following PowerShell example uses the built-in Windows firewall to prevent the Internet Information Server webserver (based on its default installation user ID of `NT AUTHORITY\IUSR`) from accessing 169.254.169.254. It uses a *deny rule* to reject all instance metadata requests (whether IMDSv1 or IMDSv2) from any process running as that user.

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount ("NT
 AUTHORITY\IUSR")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $BlockPrincipalSDDL = "D:(A;;CC;;;$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service from IIS" -Action
 block -Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $BlockPrincipalSDDL
```

Or, you can consider only allowing access to particular users or groups, by using *allow rules*. Allow rules might be easier to manage from a security perspective, because they require you to make a decision about what software needs access to instance metadata. If you use *allow rules*, it's less likely you will accidentally allow software to access the metadata service (that you did not intend to have access) if you later change the software or configuration on an instance. You can also combine group usage with allow rules, so that you can add and remove users from a permitted group without needing to change the firewall rule.

The following example prevents access to instance metadata by all processes running as an OS group specified in the variable `blockPrincipal` (in this example, the Windows group `Everyone`), except for processes specified in `exceptionPrincipal` (in this example, a group called `trustworthy-users`). You must specify both deny and allow principals because Windows Firewall, unlike the `! --uid-owner trustworthy-user` rule in Linux iptables, does not provide a shortcut mechanism to allow only a particular principal (user or group) by denying all the others.

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("Everyone")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $exceptionPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("trustworthy-users")
PS C:\> $ExceptionPrincipalSID =
 $exceptionPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $PrincipalSDDL = "O:LSD:(D;;CC;;;$ExceptionPrincipalSID)(A;;CC;;;
$BlockPrincipalSID)"
```

```
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service for
 $($blockPrincipal.Value), exception: $($exceptionPrincipal.Value)" -Action block -
Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $PrincipalSDDL
```

> **Note**
>
> To use local firewall rules, you need to adapt the preceding example commands to suit your needs.

Using netsh rules to limit access

You can consider blocking all software using `netsh` rules, but those are much less flexible.

```
C:\> netsh advfirewall firewall add rule name="Block metadata service altogether"
 dir=out protocol=TCP remoteip=169.254.169.254 action=block
```

> **Note**
>
> - To use local firewall rules, you need to adapt the preceding example commands to suit your needs.
> - `netsh` rules must be set from an elevated command prompt, and can't be set to deny or allow particular principals.

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 ceil(*n*/3)*4.
- User data must be base64-decoded when you retrieve it. If you retrieve the data using instance metadata or the console, it's decoded for you automatically.
- If you stop an instance, modify its user data, and start the instance, the updated user data is not run automatically when you start the instance. However, you can configure settings so that updated user data scripts are run one time when you start the instance or every time you reboot or start the instance.
- 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 Tools for Windows PowerShell, see the section called "User data and the Tools for Windows PowerShell".

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 Tools for Windows PowerShell, see the section called "User data and the Tools for Windows PowerShell".

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 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

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = Invoke-RestMethod
 -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
-Method PUT -Uri http://169.254.169.254/latest/api/token} -Method GET -uri
 http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

This example returns user data that was provided as a script.

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/user-data
```

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

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 Tools for Windows PowerShell, see User data and the Tools for Windows PowerShell.

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 Nitro System.

This example shows how to retrieve the high-level instance identity categories.

IMDSv2

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
```

```
pkcs7
signature
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-
identity/
document
rsa2048
pkcs7
signature
```

For more information about dynamic data and examples of how to retrieve it, see Instance identity documents.

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
`ami-id`	The AMI ID used to launch the instance.	1.0
`ami-launch-index`	If you launch multiple instances using the same `RunInstances` call, this value indicates the launch order for each instance. The value of the first instance launched is 0. If you launch instances using Auto Scaling or EC2 fleet, this value is always 0.	1.0
`ami-manifest-path`	The path to the AMI manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is `unknown`.	1.0
`ancestor-ami-ids`	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the AMI manifest file contained an `ancestor-amis` key.	2007-10-10
`autoscaling/target-lifecycle-state`	Value showing the target Auto Scaling lifecycle state that an Auto Scaling instance is transitioning to. Present when the instance transitions to one of the target lifecycle states after March 10, 2022. Possible values: `Detached` | `InService` | `Standby` | `Terminated` | `Warmed:Hibernated` |	2021-07-15

Category	Description	Version when category was released
	`Warmed:Running` | `Warmed:Stopped` | `Warmed:Terminated`. See Retrieve the target lifecycle state through instance metadata in the *Amazon EC2 Auto Scaling User Guide*.	
`block-device-mapping/ami`	The virtual device that contains the root/boot file system.	2007-12-15
`block-device-mapping/ebs` *N*	The virtual devices associated with any Amazon EBS volumes. Amazon EBS volumes are only available in metadata if they were present at launch time or when the instance was last started. The *N* indicates the index of the Amazon EBS volume (such as `ebs1` or `ebs2`).	2007-12-15

Category	Description	Version when category was released
`block-device-mapping/ephemeral` *N*	The virtual devices for any non-NVMe instance store volumes. The *N* indicates the index of each volume. The number of instance store volumes in the block device mapping might not match the actual number of instance store volumes for the instance. The instance type determines the number of instance store volumes that are available to an instance. If the number of instance store volumes in a block device mapping exceeds the number available to an instance, the additional instance store volumes are ignored.	2007-12-15
`block-device-mapping/root`	The virtual devices or partitions associated with the root devices or partitions on the virtual device, where the root (/ or C:) file system is associated with the given instance.	2007-12-15
`block-device-mapping/swap`	The virtual devices associated with `swap`. Not always present.	2007-12-15
`elastic-gpus/associations/` *`elastic-gpu-id`*	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

Category	Description	Version when category was released
`elastic-inference/associations/` *`eia-id`*	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 informati on about the events. For more information, see View scheduled events.	2018-08-17

Category	Description	Version when category was released
`events/recommendations/rebalance`	The approximate time, in UTC, when the EC2 instance rebalance recommendation notification is emitted for the instance. The following is an example of the metadata for this category: `{"noticeTime": "2020-11-05T08:22:00Z"}`. This category is available only after the notification is emitted. For more information, see EC2 instance rebalance recommendations.	2020-10-27
`hostname`	If the EC2 instance is using IP-based naming (IPBN), this is the private IPv4 DNS hostname of the instance. If the EC2 instance is using Resource-based naming (RBN), this is the RBN. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). For more information about IPBN and RBN, see Amazon EC2 instance hostname types.	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, *`role-name`* is the name of the role, and *`role-name`* contains the temporary security credentials associated with the role (for more information, see Retrieve security credentials from instance metadata). Otherwise, not present.	2012-01-12
`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 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
`local-ipv4`	The private IPv4 address of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). If this is an IPv6-only instance, this item is not set and results in an HTTP 404 response.	1.0
`mac`	The instance's media access control (MAC) address. In cases where multiple network interface s are present, this refers to the eth0 device (the device for which the device number is 0).	2011-01-01
`metrics/vhostmd`	No longer available.	2011-05-01
`network/interfaces/macs/mac/device-number`	The unique device number associated with that interface. The device number corresponds to the device name; for example, a `device-number` of 2 is for the eth2 device. This category corresponds to the `DeviceIndex` and `device-index` fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01
`network/interfaces/macs/mac/interface-id`	The ID of the network interface.	2011-01-01

Category	Description	Version when category was released
`network/interfaces/macs/mac/ipv4-associations/public-ip`	The private IPv4 addresses that are associated with each public IP address and assigned to that interface.	2011-01-01
`network/interfaces/macs/mac/ipv6s`	The IPv6 addresses assigned to the interface.	2016-06-30
`network/interfaces/macs/mac/ipv6-prefix`	The IPv6 prefix assigned to the network interface.	
`network/interfaces/macs/mac/local-hostname`	The private IPv4 DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0). If this is a IPv6-only instance, this is the resource-based name. For more information about IPBN and RBN, see 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/`*`mac`*`/network-card-index`	The index of the network card. Some instance types support multiple network cards.	2020-11-01

Category	Description	Version when category was released
`network/interfaces/macs/mac/owner-id`	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
`network/interfaces/macs/mac/public-hostname`	The interface's public DNS (IPv4). This category is only returned if the `enableDnsHostnames` attribute is set to `true`. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*. If the instance only has a public-IPv6 address and no public-IPv4 address, this item is not set and results in an HTTP 404 response.	2011-01-01
`network/interfaces/macs/mac/public-ipv4s`	The public IP address or Elastic IP addresses associated with the interface. There may be multiple IPv4 addresses on an instance.	2011-01-01
`network/interfaces/macs/mac/security-groups`	Security groups to which the network interface belongs.	2011-01-01
`network/interfaces/macs/mac/security-group-ids`	The IDs of the security groups to which the network interface 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 `enableDnsHostnames` attribute is set to `true`. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*. If the instance only has a public-IPv6 address and no public-IPv4 address, this item is not set and results in an HTTP 404 response.	2007-01-19
`public-ipv4`	The public IPv4 address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19

Category	Description	Version when category was released
`public-keys/0/openssh-key`	Public key. Only available if supplied at instance launch time.	1.0
`ramdisk-id`	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
`reservation-id`	The ID of the reservation.	1.0
`security-groups`	The names of the security groups applied to the instance. After launch, you can change the security groups of the instances. Such changes are reflected here and in network/interfaces/macs/*mac*/security-groups.	1.0
`services/domain`	The domain for AWS resources for the Region.	2014-02-25
`services/partition`	The partition that the resource is in. For standard AWS Regions, the partition is `aws`. If you have resources in other partitions, the partition is `aws-`*partition name*. For example, the partition for resources in the China (Beijing) Region is `aws-cn`.	2015-10-20

Category	Description	Version when category was released
`spot/instance-action`	The action (hibernate, stop, or terminate) and the approximate time, in UTC, when the action will occur. This item is present only if the Spot Instance has been marked for hibernate, stop, or terminate. For more information, see instance-action.	2016-11-15
`spot/termination-time`	The approximate time, in UTC, that the operating system for your Spot Instance will receive the shutdown signal. This item is present and contains a time value (for example, 2015-01-05T18:02:00Z) only if the Spot Instance has been marked for termination by Amazon EC2. The termination-time item is not set to a time if you terminated the Spot Instance yourself. For more information, see termination-time.	2014-11-05
`tags/instance`	The instance tags associated with the instance. Only available if you explicitly allow access to tags in instance metadata. For more information, see 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

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 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
`accountId`	The ID of the AWS account that launched the instance.
`architecture`	The architecture of the AMI used to launch the instance (i386 | x86_64 | arm64).
`availabilityZone`	The Availability Zone in which the instance is running.
`billingProducts`	The billing products of the instance.
`devpayProductCodes`	Deprecated.
`imageId`	The ID of the AMI used to launch the instance.
`instanceId`	The ID of the instance.
`instanceType`	The instance type of the instance.
`kernelId`	The ID of the kernel associated with the instance, if applicable.
`marketplaceProductCodes`	The AWS Marketplace product code of the AMI used to launch the instance.
`pendingTime`	The date and time that the instance was launched.

Data	Description
`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

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> (Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $Token}
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content
```

IMDSv1

```
PS C:\> (Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/
document).Content
```

The following is example output.

```
{
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
    "availabilityZone" : "us-west-2b",
    "privateIp" : "10.158.112.84",
```

```
    "version" : "2017-09-30",
    "instanceId" : "i-1234567890abcdef0",
    "billingProducts" : null,
    "instanceType" : "t2.micro",
    "accountId" : "123456789012",
    "imageId" : "ami-5fb8c835",
    "pendingTime" : "2016-11-19T16:32:11Z",
    "architecture" : "x86_64",
    "kernelId" : null,
    "ramdiskId" : null,
    "region" : "us-west-2"
}
```

Verify the instance identity document

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.

Prerequisites

This procedure requires the `System.Security` Microsoft .NET Core class. To add the class to your PowerShell session, run the following command.

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> The command adds the class to the current PowerShell session only. If you start a new session, you must run the command again.

To verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate

1. Connect to the instance.
2. Retrieve the PKCS7 signature from the instance metadata, convert it to a byte array, and add it to a variable named `$Signature`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/pkcs7).Content)
   ```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/pkcs7).Content)
```

3. Retrieve the plaintext instance identity document from the instance metadata, convert it to a byte array, and add it to a variable named `$Document`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. Find the **DSA** public certificate for your Region in AWS public certificates and add the contents to a new file named `certificate.pem`.

5. Extract the certificate from the certificate file and store it in a variable named `$Store`.

   ```
   PS C:\> $Store =
    [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
   Path certificate.pem)))
   ```

6. Verify the signature.

   ```
   PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
   ```

   ```
   PS C:\> $SignatureDocument.Decode($Signature)
   ```

   ```
   PS C:\> $SignatureDocument.CheckSignature($Store, $true)
   ```

   If the signature is valid, the command returns no output. If the signature cannot be verified, the command returns `Exception calling "CheckSignature" with "2"`

`argument(s): "Cannot find the original signer`. If your signature cannot be verified, contact AWS Support.

7. Validate the content of the instance identity document.

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

If the content of the instance identity document is valid, the command returns `True`. If instance identity document cannot be validated, contact AWS Support.

Use the base64-encoded signature to verify the instance identity document

This topic explains how to verify the instance identity document using the base64-encoded signature and the AWS RSA public certificate.

To validate the instance identity document using the base64-encoded signature and the AWS RSA public certificate

1. Connect to the instance.
2. Retrieve the base64-encoded signature from the instance metadata, convert it to a byte array, and add it to variable named `$Signature`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/signature).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/signature).Content)
   ```

3. Retrieve the plaintext instance identity document from the instance metadata, convert it to a byte array, and add it to a variable named `$Document`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. Find the **RSA** public certificate for your Region in AWS public certificates and add the contents to a new file named `certificate.pem`.
5. Verify the instance identity document.

   ```
   PS C:\> [Security.Cryptography.X509Certificates.X509Certificate2]::new((Resolve-
   Path certificate.pem)).PublicKey.Key.VerifyData($Document, 'SHA256', $Signature)
   ```

   If the signature is valid, the command returns `True`. If the signature cannot be verified, contact AWS Support.

Use the RSA-2048 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate.

Prerequisites

This procedure requires the `System.Security` Microsoft .NET Core class. To add the class to your PowerShell session, run the following command.

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> The command adds the class to the current PowerShell session only. If you start a new session, you must run the command again.

To verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate

1. Connect to the instance.
2. Retrieve the RSA-2048 signature from the instance metadata, convert it to a byte array, and add it to a variable named `$Signature`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/rsa2048).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/rsa2048).Content)
   ```

3. Retrieve the plaintext instance identity document from the instance metadata, convert it to a byte array, and add it to a variable named `$Document`. Use one of the following commands depending on the IMDS version used by the instance.

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. Find the **RSA-2048** public certificate for your Region in AWS public certificates and add the contents to a new file named `certificate.pem`.
5. Extract the certificate from the certificate file and store it in a variable named `$Store`.

```
PS C:\> $Store =
 [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
Path certificate.pem)))
```

6. Verify the signature.

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

If the signature is valid, the command returns no output. If the signature cannot be verified, the command returns `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`. If your signature cannot be verified, contact AWS Support.

7. Validate the content of the instance identity document.

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

If the content of the instance identity document is valid, the command returns `True`. If instance identity document cannot be validated, 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-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
```

```
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM07oeX4xevdMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA2MTAx
MjU4MThaGA8yMTk1MTExNDEyNTgxOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA6v6kGMnRmFDLxBEqXzP4npnL65OO0kmQ7w8YXQygSdmNIoScGSU5wfh9
mZdcvCxCdxgALFsFqPvH8fqiE9ttI0fEfuZvHOs8wUsIdKr0Zz0MjSx3cik4tKET
ch0EKfMnzKOgDBavraCDeX1rUDU0Rg7HFqNAOry3uqDmnqtk00XC9GenS3z/7ebJ
fIBEPAam5oYMVFpX6M6St77WdNE8wEU8SuerQughiMVx9kMB07imeVHBiELbMQ0N
lwSWRL/61fA02keGSTfSp/0m3u+lesf2VwVFhqIJs+JbsEscPxOkIRlzy8mGd/JV
ONb/DQpTedzUKLgXbw7KtO3HTG9iXQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU2CTGYE5fTjx7gQXzdZSGPEWAJY4wgY4GA1UdIwSBhjCBg4AU2CTG
YE5fTjx7gQXzdZSGPEWAJY6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM07oeX4xevdMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANdqkIpVypr2PveqUsAKke1wKCOSuw1UmH9k
xX1/VRoHbrI/UznrXtPQOPMmHA2LKSTedwsJuorUn3cFH6qNs8ixBDrl8pZwfKOY
IBJcTFBbI1xBEFkZoO3wczzo5+8vPQ60RVqAaYb+iCa1HFJpccC3Ovajfa4GRdNb
n6FYnluIcDbmpcQePoVQwX7W3oOYLB1QLN7fE6H1j4TBIsFdO3OuKzmaifQlwLYt
DVxVCNDabpOr6Uozd5ASm4ihPPoEoKo7Ilp0fOT6fZ41U2xWA4+HF/89UoygZSo7
K+cQ90xGxJ+gmlYbLFR5rbJOLfjrgDAb2ogbFy8LzHo2ZtSe60M=
-----END CERTIFICATE-----
```

US East (Virginia) — us-east-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
```

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

RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALFpzEAVWaQZMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU5MTJaGA8yMTk1MDExNzA4NTkxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjS2vqZu9mEOhOq+0bRpAbCUiapbZMFNQqRg7kTlr7Cf+gDqXKpHPjsng
SfNz+JHQd8WPI+pmNs+q0Z2aTe23klmf2U52KH9/j1k8RlIbap/yFibFTSedmegX
E5r447GbJRsHUmuIIfZTZ/oRlpuIIO5/Vz7SOj22tdkdY2ADp7caZkNxhSP915fk
2jJMTBUOzyXUS2rBU/ulNHbTTeePjcEkvzVYPahD30TeQ+/A+uWUu89bHSQOJR8h
Um4cFApzZgN3aD5j2LrSMu2pctkQwf9CaWyVznqrsGYjYOY66LuFzSCXwqSnFBfv
fFBAFsjCgY24G2DoMyYkF3MyZlu+rwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUrynSPp4uqSECwy+PiO4qyJ8TWSkwgY4GA1UdIwSBhjCBg4AUrynS
Pp4uqSECwy+PiO4qyJ8TWSmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALFpzEAVWaQZMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADW/s8lXijwdP6NkEoH1m9XLrvK4YTqkNfR6
er/uRRgTx2QjFcMNrx+g87gAml11z+D0crAZ5LbEhDMs+JtZYR3ty0HkDk6SJM85
haoJNAFF7EQ/zCp1EJRIkLLsC7bcDL/Eriv1swt78/BB4RnC9W9kSp/sxd5svJMg
N9a6FAplpNRsWAnbP8JBlAP93oJzblX2LQXgykTghMkQO7NaY5hg/H5o4dMPclTK
lYGqlFUCH6A2vdrxmpKDLmTn5//5pujdD2MN0df6sZWtxwZ0osljV4rDjm9Q3VpA
NWIsDEcp3GUB4proOR+C7PNkY+VGODitBOw09qBGosCBstwyEqY=
-----END CERTIFICATE-----
```


US West (N. California) — us-west-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
```

```
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJANNPkIpcyEtIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAzMDdaGA8yMTk1MDQwMzA5MDMwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApHQGvHvq3SVCzDrC7575BW7GWLzcj8CLqYcL3YY7Jffupz7OjcftO57Z
4fo5Pj0CaS8DtPzh8+8vdwUSMbiJ6cDd3ooio3MnCq6DwzmsY+pY7CiI3UVG7KcH
```

```
4TriDqr1Iii7nB5MiPJ8wTeAqX89T3SYaf6Vo+4GCb3LCDGvnkZ9TrGcz2CHkJsj
AIGwgopFpwhIjVYm7obmuIxSIUv+oNH0wXgDL029Zd98SnIYQd/njiqkzE+lvXgk
4h4Tu17xZIKBgFcTtWPky+POGu81DYFqiWVEyR2JKKm2/iR1dL1YsT39kbNg47xY
aR129sS4nB5Vw3TRQA2jL0ToTIxzhQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUgepyiONs8j+q67dmcWu+mKKDa+gwgY4GA1UdIwSBhjCBg4AUgepy
iONs8j+q67dmcWu+mKKDa+ihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJANNPkIpcyEtIMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGLFWyutf1u0xcAc+kmnMPqtc/Q6b79VIX0E
tNoKMI2KR8lcV8ZElXDb0NC6v8UeLpe1WBKjaWQtEjL1ifKg9hdY9RJj4RXIDSK7
33qCQ8juF4vep2U5TTBd6hfWxt1Izi88xudjixmbpUU4YKr8UPbmixldYR+BEx0u
B1KJi9l1lxvuc/Igy/xeHOAZEjAXzVvHp8Bne33VVwMiMxWECZCiJxE4I7+Y6fqJ
pLLSFFJKbNaFyXlDiJ3kXyePEZSc1xiWeyRB2ZbTi5eu7vMG4i3AYWuFVLthaBgu
lPfHafJpj/JDcqt2vKUKfur5edQ6j1CGdxqqjawhOTEqcN8m7us=
-----END CERTIFICATE-----
```


US West (Oregon) — us-west-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


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

```
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALZL3lrQCSTMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAxMzJaGA8yMTk1MDExNzA5MDEzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA02Y59qtAA0a6uzo7nEQcnJ26OKF+LRPwZfixBH+EbEN/Fx0gYy1jpjCP
s5+VRNg6/WbfqAsV6X2VSjUKN59ZMnMY9ALA/Ipz0n00Huxj38EBZmX/NdNqKm7C
qWu1q5kmIvYjKGiadfboU8wLwLcHo8ywvfgI6FiGGsEO9VMC56E/hL6Cohko11LW
dizyvRcvg/IidazVkJQCN/4zC9PUOVyKdhW33jXy8BTg/QH927QuNk+ZzD7HH//y
tIYxDhR6TIZsSnRjz3bOcEHxt1nsidc65mY0ejQty4hy7ioSiapw316mdbtE+RTN
fcH9FPIFKQNBpiqfAW5Ebp3Lal3/+wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU7coQx8Qnd75qA9XotSWT3IhvJmowgY4GA1UdIwSBhjCBg4AU7coQ
x8Qnd75qA9XotSWT3IhvJmqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALZL3lrQCSTMMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFZ1e2MnzRaXCaLwEC1pW/f0oRG8nHrlPZ9W
OYZEWbh+QanRgaikBNDtVTwARQcZm3z+HWSkaIx3cyb6vM0DSkZuiwzm1LJ9rDPc
aBm03SEt5v8mcc7sXWvgFjCnUpzosmky6JheCD4O1Cf8k0olZ93FQnTrbg62OK0h
83mGCDeVKU3hLH97FYoUq+3N/IliWFDhvibAYYKFJydZLhIdlCiiB99AM6Sg53rm
oukS3csyUxZyTU2hQfdjyo1nqW9yhvFAKjnnggiwxNKTTPZzstKW8+cnYwiiTwJN
```

```
QpVoZdt0SfbuNnmwRUMi+QbuccXweav29QeQ3ADqjgB0CZdSRKk=
-----END CERTIFICATE-----
```

Africa (Cape Town) — af-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7DCCAqwCCQCncbCtQbjuyzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA2MDQxMjQ4MDVaFw00
NTA2MDQxMjQ4MDVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQC12Nr1gMrHcFSZ7S/A
pQBSCMHWmn2qeoQTMVWqe50fnTd0zGFxDdIjKxUK58/8zjWG5uR4TXRzmZpGpmXB
bSufAR6BGqud2LnT/HIWGJAsnX2uOtSyNfCoJigqwhea5w+CqZ6I7iBDdnB4TtTw
qO6TlnExHFVj8LMkylZgiaE1CQIVAIhdobse4K0QnbAhCL6R2euQzloXAoGAV/21
WUuMz/79Ga0JvQcz1FNy1sT0pU9rU4TenqLQIt5iccn/7EIfNtvVO5TZKulIKq7J
gXZr0x/KIT8zsNweetLOaGehPIYRMPX0vunMMR7hN7qA7W17WZv/76adywIsnDKq
ekfe15jinaX8MsKUdyDK7Y+ifCG4PVhoM4+W2XwDgYQAAoGAIxOKbVgwLxbn6Pi2
6hBOihFv16jKxAQI0hHzXJLV0Vyv9QwnqjJJRfOCy3dB0zicLXiIxeIdYfvqJr+u
hlN8rGxEZYYJjEUKMGvsc0DW85jonXz0bNfcP0aaKH0lKKVjL+OZi5n2kn9wgdo5
F3CVnMl8BUra8A1Tr2yrrE6TVZ4wCQYHKoZIzjgEAwMvADAsAhQfa7MCJZ+/TEY5
AUr0J4wm8VzjoAIUSYZVu2NdRJ/ERPmDfhW5EsjHlCA=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAKumfZiRrNvHMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTExMjcw
NzE0MDVaGA8yMTk5MDUwMjA3MTQwNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFd571nUzVtke3rPyRkYfvs3jh0C0EMzzG72boyUNjnfw1+m0TeFraTLKb9T6F
7TuB/ZEN+vmlYqr2+5Va8U8qLbPF0bRH+FdaKjhgWZdYXxGzQzU3ioy5W5ZM1VyB
7iUsxEAlxsybC3ziPYaHI42UiTkQNahmoroNeqVyHNnBpQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAAJLylWyElEgOpW4B1XPyRVD4pAds8Guw2+krgqkY0HxLCdjosuH
RytGDGN+q75aAoXzW5a7SGpxLxk6Hfv0xp3RjDHsoeP0i1d8MD3hAC5ezxS4oukK
s5gbPOnokhKTMPXbTdRn5ZifCbWlx+bYN/mTYKvxho7b5SVg2o1La9aK
-----END CERTIFICATE-----
```

RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAIFI+O5A6/ZIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA2MDQx
MjQ4MDRaGA8yMTk4MTEwNzEyNDgwNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy7/WHBBHOrk+20aumT07g8rxrSM0UXgki3eYgKauPCG4Xx//vwQbuZwI
oeVmR9nqnhfij2wOcQdbLandh0EGtbxerete3IoXzd1KXJb11PVmzrzyu5SPBPuP
iCeV4qdjjkXo2YWM6t9YQ911hcG96YSp89TBXFYUh3KLxfqAdTVhuC0NRGhXpyii
j/czo9njofHhqhTr7UEyPun8NVS2QWctLQ86N5zWR3Q0GRoVqqMrJs0cowHTrVw2
9Qr7QBjjBOVbyYmtYxm/DtiKprYV/e6bCAVok015X1sZDd3oCOQNoGlv5XbHJe2o
JFD8GRRy2rkWO/lNwVFDcwec6zC3QwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCE
goqzjpCpmMgCpszFHwvRaSMbspKtK7wNImUjrSBOfBJsfFulyg1Zgn2nDCK7kQhx
jMJmNIvXbps3yMqQ2cHUkKcKf5t+WldfeT4Vk1Rz6HSA8sd0kgVcIesIaoy2aaXU
VEB/oQziRGyKdN1d4TGYVZXG44CkrzSDvlbmfiTq5tL+kAieznVF3bzHgPZW6hKP
EXC3G/IXrXicFEe6YyE1Rakl62VncYSXiGe/i2XvsiNH3Qlmnx5XS7W0SCN0oAxW
EH9twibauv82DVg1WOkQu8EwFw8hFde9X0Rkiu0qVcuU8lJgFEvPWMDFU5sGB6ZM
gkEKTzMvlZpPbBhg99Jl
-----END CERTIFICATE-----
```


Asia Pacific (Hong Kong) — ap-east-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQCO7MJe5Y3VLjAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDMwMjIxMjFaFw00
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDvQ9RzVvf4MAwGbqfX
blCvCoVb9957OkLGn/04CowHXJ+vTBR7eyIa6AoXltsQXBOmrJswToFKKxT4gbuw
jK7s9QQX4CmTRWcEgO2RXtZSVjOhsUQMh+yf7Ht4OVL97LWnNfGsX2cwjcRWHYgI
7lvnuBNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGKd9FAoGBAOCG
eSNmxpW4QFu4pIlAykm6EnTZKKHT87gdXkAkfoC5fAfOxxhnE2HezZHp9Ap2tMV5
8bWNvoPHvoKCQqwfm+OUBlAxC/3vqoVkKL2mG1KgUH9+hrtpMTkwO3RREnKe7I5O
x9qDimJpOihrL4I0dYvy9xUOoz+DzFAW8+ylWVYpA4GFAAKBgQDbnBAKSxWr9QHY
6Dt+EFdGz6lAZLedeBKpaP53Z1DTO34J0C55YbJTwBTFGqPtOLxnUVDlGiD6GbmC
80f3jvogPR1mSmGsydbNbZnbUEVWrRhe+y5zJ3g9qs/DWmDW0deEFvkhWVnLJkFJ
```

```
9pdOu/ibRPH1lE2nz6pK7GbOQtLyHTAJBgcqhkjOOAQDAzAAMC0CFQCoJlwGtJQC
cLoM4p/jtVFOj26xbgIUUS4pDKyHaG/eaygLTtFpFJqzWHc=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICSzCCAbQCCQDtQvkVxRvK9TANBgkqhkiG9w0BAQsFADBqMQswCQYDVQQGEwJV
UzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2VhdHRsZTEYMBYGA1UE
ChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1hem9uYXdzLmNvbTAe
Fw0xOTAyMDMwMzAwMDZaFw0yOTAyMDIwMzAwMDZaMGoxCzAJBgNVBAYTAlVTMRMw
EQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgwFgYDVQQKEw9B
bWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3MuY29tMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1kkHXYTfc7gY5Q55JJhjTieHAgacaQkiR
Pity9QPDE3b+NXDh4UdP1xdIw73JcIIG3sG9RhWiXVCHh6KkuCTqJfPUknIKk8vs
M3RXflUpBe8Pf+P92pxqPMCz1Fr2NehS3JhhpkCZVGxxwLC5gaG0Lr4rFORubjYY
Rh84dK98VwIDAQABMA0GCSqGSIb3DQEBCwUAA4GBAA6xV9f0HMqXjPHuGILDyaNN
dKcvplNFwDTydVg32MNubAGnecoEBtUPtxBsLoVYXCOb+b5/ZMDubPF9tU/vSXuo
TpYM5Bq57gJzDRaBOntQbX9bgHiUxw6XZWaTS/6xjRJDT5p3S1E0mPI3lP/eJv4o
Ezk5zb3eIf10/sqt4756
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMoxixvs3YssMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA3MjAw
ODQ0NDRaGA8yMTk3MTIyMzA4NDQ0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA4T1PNsOg0FDrGlWePoHeOSmOJTA3HCRy5LSbYD33GFU2eBrOIxoU/+SM
rInKu3GghAMfH7WxPW3etIAZiyTDDU5RLcUq2Qwdr/ZpXAWpYocNc/CEmBFtfbxF
z4uwBIN3/drM0RSbe/wP9EcgmNUGQMMZWeAji8sMtwpOblNWAP9BniUG0Flcz6Dp
uPovwDTLdAYT3TyhzlohKL3f6O48TR5yTaV+3Ran2SGRhyJjfh3FRpP4VC+z5LnT
WPQHN74Kdq35UgrUxNhJraMGCzznolUuoR/tFMwR93401GsM9fVA7SW3jjCGF81z
PSzjy+ArKyQqIpLW1YGWDFk3sf08FQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDK
2/+C3nPMgtyOFX/I3Cyk+Pui44IgOwCsIdNGwuJysdqp5VIfnjegEu2zIMWJSKGO
lMZoQXjffkVZZ97J7RNDW06oB7kj3WVE8a7U4WEOfnO/CbMUf/x99CckNDwpjgW+
K8V8SzAsQDvYZs2KaE+18GFfLVF1TGUYK2rPSZMHyX+v/TIlc/qUceBycrIQ/kke
jDFsihUMLqgmOV2hXKUpIsmiWMGrFQV4AeV0iXP8L/ZhcepLf1t5SbsGdUA3AUY1
3If8s81uTheiQjwY5t9nM0SY/1Th/tL3+RaEI79VNEVfG1FQ8mgqCK0ar4m0oZJl
tmmEJM7xeURdpBBx36Di
```

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

Asia Pacific (Hyderabad) — ap-south-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXjrQ4+XMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAJCKGBBoxIUxqBk94JHhwZZbgvbP0DAOoHENQWxp/98lI7/
YOfYJOVMJS22aCnHDurofmo5rvNIkgXi7Rztbhu
+lko9rK6DgpmpUwBU0WZtf34aZ2IWNBwHaVhHvWAQf9/46u18dMa2YucK1Wi+Vc+M
+KldrvgXmhym6ErNlzhJyMAkGByqGSM44BAMDLwAwLAIUaaPKxaOHoYvwz709xXpsQueIq+UCFFa/
GpzoDOSokl1057NU/2hnsiW4
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjwLj9CMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+sFcobrjvcAYm0PNRD8f4R1jAzvoLt2+qGeOTAyO1Httj6cmsYN3AP1hN5iYuppFiYsl2eNPa/
CD0Vg0BAfDFlV5rzjpA0j7TJabVh4kj7JvtD+xYMi6wEQA4x6SPONY4OeZ2+8o/
HS8nucpWDVdPRO6ciWUlMhjmDmwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAAy6sgTdRkTqELHBeWj69q60xHyUmsWqHAQ
TGGbYP0yP2qfMl0cCImzRI5W0gn8gogdervfeT7nH5ih0TWEy/QDWfkQ601L4erm4yh4YQq8vcqAPSkf04N
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAIVWfPw/X82fMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MDQx
NDMwMjhaGA8yMjAxMTIwODE0MzAyOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAg29QEFriG+qFEjYW/v62nN70lMJY/Hevx5TtmU/VIYBPQa3HUGTBAbbI
```

```
2Tmy8UMpa8kZeaYeI3RAfiQWt0Ws7wUrBuO2Pdp518WDPaJUH7RWEuu1BDDkyZRW
NAMNPCn3ph7Od243IFcLGku7HVekel5poqRpSfojrMasjlf+CvixUeAJbmFoxUHK
kh5unzG2sZyO4wHXcJPQkRf5a8zSTPe9YZP1kXPPEv4p/jTSggaYPxXyS6QVaTlV
zLeLFZ0fesLPMeil3KYQtV7IKLQiEA2F6dxWnxNWQlyMHtdq6PucfEmVx17i/Xza
yNBRo0azY8WUNVkEXrRhp/pU8Nh3GQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU9AO1aZk9RLXk2ZvRVoUxYvQy9uwwgY4GA1UdIwSBhjCBg4AU9AO1
aZk9RLXk2ZvRVoUxYvQy9uyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAIVWfPw/X82fMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADexluMRQRftqViahCnauEWGdMvLCBr8A+Yr
6hJq0guoxEk/lahxR137DnfMPuSbi1Rx5QKo7oBrWfG/zsgQUnF2IwHTzwD+i/2m
XCane6FiS5RpK3lGdILq8ZmlhQk+6iI8yoZLr0LCfTh+CLgIKHOknfR5lFzgzAiF
SI8/Q9mm+uvYtSTZECI6Zh57QZPoETAG/yl+9jiOy2lAelqa/k1i+Qo8gMfOc+Pm
dwY7o6fV+oucgRlsdey6VM45LeyILQqv0RXtVzjuowanzmCCFMjgqiO9oZAWu4Oh
+F3unijELo01vZJs8s2N3KGlo3/jtUFTX6RTKShZlAPLwBi5GMI=
-----END CERTIFICATE-----
```


Asia Pacific (Jakarta) — ap-southeast-3

DSA


```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXbVDEikMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAPjuiEx05N3JQ6cVwntJie67D8OuNo4jGRn
+crEtL7YO0jSVB9zGE1ga
+UgRPIaYETL293S8rTJTVgXAqdpBwfaHC6NUzre8U8iJ8FMNnlP9Gw1oUIlgQBjORyynVJexoB31TDZM
+/52g9O/bpq1QqNyKbeIgyBBlc1dAtr1QLnsMAkGByqGSM44BAMDLwAwLAIUK8E6RDIRtwK+9qnaTOBhvO/
njuQCFFocyT1OxK+UDR888oNsdgtif2Sf
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXbVDG2yMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
Vbt0gQ1ebWcur2hSO7PnJifE4OPxQ7RgSAlc4/spJp1sDP+ZrS0LO1ZJfKhXf1R9S3AUwLnsc7b
+IuVXdY5LK9RKqu64nyXP5dx170zoL8loEyCSuRR2fs+04i2QsWBVP+KFNAn7P5L1EHRjkgTO8kjNKviwRV
```

```
+OkP9ab5wIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAI4WUy6+DKh0JDSzQEZNyBgNlSoSuC2owtMxCwGB6nBfzzfcekWvs
+87w/g9lNwUnUt0ZHYyh2tuBG6hVJuUEwDJ/z3wDd6wQviLOTF3MITawt9P8siR1hXqLJNxpjRQFZrgHqi
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMtdyRcH51j9MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MDgx
MjM5MTZaGA8yMjAxMDkxMjEyMzkxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvUsKCxoH6KXRYJLeYTWAQfaBQeCwhJaR56mfUeFHJE4g8aFjWkiN4uc1
TvOyYNnIZKTHWmzmulmdinWNbwP0GiROHb/i7ro0Hhvnptyycgt8ag8affiIbx5X
7ohdwSN2KJ6G0IKflIx7f2NEI0oAMM/9k+T1eVF+MVWzpZoiDp8frLNkqp8+RAgz
ScZsbRfwv3u/if5xJAvdg2nCkIWDMSHEVPoz0lJo7v0ZuDtWWsL1LHnL5ozvsKEk
+ZJyEi23r+U1hIT1NTBdp4yoigNQexedtwCSr7q36oOdDwvZpqYlkLi3uxZ4ta+a
01pzOSTwMLgQZSbKWQrpMvsIAPrxoQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU1GgnGdNpbnL3lLF30Jomg7Ji9hYwgY4GA1UdIwSBhjCBg4AU1Ggn
GdNpbnL3lLF30Jomg7Ji9hahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMtdyRcH51j9MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACVl00qQlatBKVeiWMrhpczsJroxDxlZTOba
6wTMZk7c3akb6XMOSZFbGaifkebPZqTHEhDlrClM2j9AIlYcCx6YCrTf4cuhn2mD
gcJN33143eOWSaeRY3ee4j+V9ne98y3kO2wLz95VrRgclPFR8po2iWGzGhwUi+FG
q8dXeCH3N0DZgQsSgQWwmdNQXZZej6RHLU/8In5trHKLY0ppnLBjn/UZQbeTyW5q
RJB3GaveXjfgFUWj2qOcDuRGaikdS+dYaLsi5z9cA3FolHzWxx9MOs8io8vKqQzV
XUrLTNWwuhZy88cOlqGPxnoRbw7TmifwPw/cunNrsjUUOgs6ZTk=
-----END CERTIFICATE-----
```

Asia Pacific (Melbourne) — ap-southeast-4

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjWF7P2MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
```

```
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAPRXSsQP9E3dw8QXKlrgBgEVCprLHdK/bbrMas0XMu1EhOD
+q
+0PcTr8+iwbtoXlY5MCeatWIp1GrXQjVqsF8vQqxlEuRuYKbR3nq4mWwaeGlx9AG5EjQHRa3GQ44wWHOdof0M3NRI1MP
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjSh40SMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+qWTGAbGsPeMX4hBMjAJUKys2NIRcRZaLM/BCew2FIPVjNtlaj6Gwn9ipU4Mlz3zIwAMWi1AvGMSreppt
+wV6MRtfOjh0Dvj/veJe88aEZJMozNgkJFRS
+WFWsckQeL56tf6kY6QTlNo8V/0CsQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAF7vpPghH0FRo5gu49EArRNPrIvW1egM
wcgkqIwwuXYj+1rhlL+/
iMpQWjdVGEqIZSeXn5fLmdx50eegFCwND837r9e8XYTiQS143Sxt9+Yi6BZ7U7YD8kK9NBWoJxFqUeHdpRCs0O7COjT3
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAN4GTQ64zVs8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTMx
MzMzMDBaGA8yMjAxMTIxNzEzMzMwMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA2BYgeCr+Rk/jIAEDOHS7wJql62vc83QEwjuzk0qOFEReIZzlN1fBRNXK
g0T178Kd3gLYcE59wEFbTe/X5yOA1Lo95x1anSAo7R+Cisf9C2HQuJp+gVb+zx71
lniPF7gHziGpm0M8DdAU/IW+wkZwGbP4z7Hq9+bJ0P2ltvPJ5yxSgkFuDsI9VBHa
CLoprHsCHh2VdP8KcMgQQMmHe1NmBpyTk0ul/aLmQkCQEX6ZIRGOeq228fwlh/t+
Ho+jv87duihVKic6MrL32SlD+maX0LSDUydWdaOLLTGkh7oV7+bFuH6msrXUu+Ur
ZEP1r/MidCWMhfgrFzeTBzOHA97qxQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUcHMd1cHqzmsQ5hpUK3EMLhHdsi4wgY4GA1UdIwSBhjCBg4AUcHMd
1cHqzmsQ5hpUK3EMLhHdsi6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAN4GTQ64zVs8MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAI4PFyVN+7EGSObioiPnv0LL0f70SSzUZJ8p
XO90d4rWea7jIbgZ2AKb+ErynkU9xVg7XQQ5k6KDWgp/4jYFL2dqnt/YAY4PS0un
RSrYElawxLT0BcLn4rcSDC79vQe1xGC5//wDdV6b399COAHRAK6axWYy5w32u9PL
uw0cIp3Ch8JoNwcgTHKRRGzePmBeR4PNqhHTArG4/dJk6/aUO4OpX0WzI6L67CGY
6Nex3dau+gkLCK93dTEkrXtyXHu4wB0J9zd1w+iQ0SEa9eKc78/NjEsF/FZdGrWC
t57lIMOOXJhQ1kRgSwNeZdQWV1dRakvO6sfcvVYkfj1wAvZvvAw=
```

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

Asia Pacific (Mumbai) — ap-south-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
```

```
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPRYyD8TtmC0MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjAzMDcx
MDQ1MDFaGA8yMTk1MDgxMTEwNDUwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA0LSS5I/eCT2PM0+qusorBx67QL26BIWQHd/yF6ARtHBb/1DdFLRqE5Dj
07Xw7eENC+T79mOxOAbeWg91KaODOzw6i9I/2/HpK0+NDEdD6sPKDA1d45jRra+v
CqAjI+nV9Vw91wv7HjMk3RcjWGziM8/hw+3YNIutt7aQzZRwIWlBpcqx3/AFd8Eu
2UsRMSHgkGUW6UzUF+h/U8218XfrauKNGmNKDYUhtmyBrHT+k6J0hQ4pN7fe6h+Z
w9RVHm24BGhlLxLHLmsOIxvbrF277uX9Dxu1HfKfu5D2kimTY7xSZDNLR2dt+kNY
/+iWdIeEFpPT0PLSILt52wP6stF+3QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBI
E6w+WWC2gCfoJO6c9HMyGLMFEpqZmz1n5IcQt1h9iyO7Vkm1wkJiZsMhXpk73zXf
TPxuXEacTX3SOEa07OIMCFwkusO5f6leOyFTynHCzBgZ3U0UkRVZA3WcpbNB6Dwy
h7ysVlqyT9WZd7EOYm5j5oue2G2xdei+6etgn5UjyWm6liZGrcOF6WPTdmzqa6WG
ApEqanpkQd/HM+hUYex/ZS6zEhd4CCDLgYkIjlrFbFb3pJ1OVLztIfSN5J4Oolpu
JVCfIq5u1NkpzL7ys/Ub8eYipbzI6P+yxXiUSuF0v9b98ymczMYjrSQXIf1e8In3
OP2CclCHoZ8XDQcvvKAh
-----END CERTIFICATE-----
```


Asia Pacific (Osaka) — ap-northeast-3

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
```

```
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMn1yPk22ditMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA3MTkx
MTEyNThaGA8yMTk2MTIyMjExMTI1OFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArznEYef8IjhrJoazI0QGZkmlmHm/4rEbyQbMNifxjsDE8YWtHNwaM91z
zmyK6Sk/tKlWxcnl3g31iq305ziyFPEewe5Qbwf1iz2cMsvfNBcTh/E6u+mBPH3J
gvGanqUJt6c4IbipdEouIjjnynyVWd4D6erLl/ENijeR1OxVpaqSW5SBK7jms49E
```

```
pw3wtbchEl3qsE42Ip4IYmWxqjgaxB7vps91n4kfyzAjUmklcqTfMfPCkzmJCRgp
Vh1C79vRQhmriVKD6BXwfZ8tG3a7mijeDn7kTsQzgO07Z2SAE63PIO48JK8HcObH
tXORUQ/XF1jzi/SIaUJZT7kq3kWl8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBj
ThtO9dLvU2QmKuXAhxXjsIdlQgGG3ZGh/Vke4If1ymgLx95v2Vj9Moxk+gJuUSRL
BzFte3TT6b3jPolbECgmAorjj8NxjC17N8QAAI1d0S0gI8kqkG7V8iRyPIFekv+M
pcai1+cIv5IV5qAz8QOMGYfGdYkcoBjsgiyvMJu/2N2UbZJNGWvcEGkdjGJUYYOO
NaspCAFm+6HA/K7BD9zXB1IKsprLgqhiIUgEaW3UFEbThJT+z8UfHG9fQjzzfN/J
nT6vuY/0RRu1xAZPyh2gr5okN/s6rnmh2zmBHU1n8cbCc64MVfXe2g3EZ9Glq/9n
izPrI09hMypJDP04ugQc
-----END CERTIFICATE-----
```


Asia Pacific (Seoul) — ap-northeast-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
```

```
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANuCgCcHtOJhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTQx
NTU3NDRaGA8yMTk1MDIxNzE1NTc0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA66iNv6pJPmGM20W8HbVYJSlKcAg2vUGx8xeAbzZIQdpGfkabVcUHGB6m
Gy59VXDMDlrJckDDk6dxUOhmcX9z785TtVZURq1fua9QosdbTzX4kAgHGdp4xQEs
mO6QZqg5qKjBP6xr3+PshfQ1rB8Bmwg0gXEm22CC7o77+7N7Mu2sWzWbiUR7vil4
9FjWS8XmMNwFTlShp4l1TDTevDWW/uYmC30RThM9S4QPvTZ0rAS18hHVam8BCTxa
LHaVCH/Yy52rsz0hM/FlghnSnK105ZKj+b+KIp3adBL8OMCjgc/Pxi0+j3HQLdYE
32+FaXWU84D2iP2gDT28evnstzuYTQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC1
mA4q+12pxy7By6g3nBk1s34PmWikNRJBwOqhF8ucGRv8aiNhRRye9lokcXomwo8r
KHbbqvtK85l0xUZp/Cx4sm4aTgcMvfJP29jGLclDzeqADIvkWEJ4+xncxSYVlS9x
+78TvF/+8h9U2LnSl64PXaKdxHy2IsHIVRN4GtoaP2Xhpa1S0M328Jykq/571nfN
1WRD1c/fQf1edgzRjhQ4whcAhv7WRRF+qTbfQJ/vDxy8lkiOsvU9XzUaZ0fZSfXX
wXxZamQbONvFcxVHY/0PSiM8nQoUmkkBQuKleDwRWvkoJKYKyr3jvXK7HIWtMrO4
jmXe0aMy3thyK6g5sJVg
-----END CERTIFICATE-----
```


Asia Pacific (Singapore) — ap-southeast-1

DSA


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

```
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


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

```
MIIEEjCCAvqgAwIBAgIJAJVMGw5SHkcvMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
ODU3MTlaGA8yMTk1MDQwMzA4NTcxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAlaSSLfBl7OgmikjLReHuNhVuvM20dCsVzptUyRbut+KmIEEc24wd/xVy
2RMIrydGedkW4tUjkUyOyfET5OAyT43jTzDPHZTkRSVkYjBdcYbe9o/0Q4P7IVS3
XlvwrUu0qo9nSID0mxMnOoF1l8KAqnn10tQ0W+lNSTkasW7QVzcb+3okPEVhPAOq
MnlY3vkMQGI8zX4iOKbEcSVIzf6wuIffXMGHVC/JjwihJ2USQ8fq6oy686g54P4w
ROg415kLYcodjqThmGJPNUpAZ7MOc5Z4pymFuCHgNAZNvjhZDA842Ojecqm62zcm
Tzh/pNMNeGCRYq2EQX0aQtYOIj7bOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU6SSB+3qALorPMVNjToM1Bj3oJMswgY4GA1UdIwSBhjCBg4AU6SSB
+3qALorPMVNjToM1Bj3oJMuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAJVMGw5SHkcvMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAF/0dWqkIEZKg5rca8o0P0VS+tolJJE/FRZO
atHOeaQbWzyac6NEwjYeeV2kY63skJ+QPuYbSuIBLM8p/uTRIvYM4LZYImLGUvoO
IdtJ8mAzq8CZ3ipdMs1hRqF5GRp8lg4w2QpX+PfhnW47iIOBiqSAUkIr3Y3BDaDn
EjeXF6qS4iPIvBaQQ0cvdddNh/pE33/ceghbkZNTYkrwMyBkQlRTTVKXFN7pCRUV
+L9FuQ9y8mP0BYZa5e1sdkwebydU+eqVzsil98ntkhpjvRkaJ5+Drs8TjGaJWlRw
5WuOr8unKj7YxdL1bv7//RtVYVVi296ldoRUYv4SCvJF11z0OdQ=
-----END CERTIFICATE-----
```


Asia Pacific (Sydney) — ap-southeast-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
```

```
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL2bOgb+dq9rMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAwNTdaGA8yMTk1MDQwMzA5MDA1N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpviBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAmRcyLWraysQS8yDC1b5Abs3TUaJabjqWu7d5gHik5Icd6dKl8EYpQSeS
vz6pLhkgO4xBbCRGlgE8LS/OijcZ5HwdrxBiKbicR1YvIPaIyEQQvF5sX6UWkGYw
Ma5IRGj4YbRmJkBybw+AAV9Icb5LJNOMWPi34OWM+2tMh+8L234v/JA6ogpdPuDr
sM6YFHMZ0NWo58MQ0FnEj2D7H58Ti//vFPl0TaaPWaAIRF85zBiJtKcFJ6vPidqK
f2/SDuAvZmyHC8ZBHg1moX9bR5FsU3QazfbW+c+JzAQWHj2AaQrGSCITxCMlS9sJ
l51DeoZBjnx8cnRe+HCaC4YoRBiqIQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU/wHIo+r5U31VIsPoWoRVsNXGxowwgY4GA1UdIwSBhjCBg4AU/wHI
```

```
o+r5U31VIsPoWoRVsNXGxoyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL2bOgb+dq9rMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACobLvj8IxlQyORTz/9q7/VJL509/p4HAeve
92riHp6+Moi0/dSEYPeFTgdWB9W3YCNc34Ss9TJq2D7t/zLGGlbI4wYXU6VJjL0S
hCjWeIyBXUZOZKFCb0DSJeUElsTRSXSFuVrZ9EAwjLvHni3BaC9Ve34iP71ifr75
8Tpk6PEj0+JwiijFH8E4GhcV5chB0/iooU6ioQqJrMwFYnwo1cVZJD5v6D0mu9bS
TMIJLJKv4QQQqPsNdjiB7G9bfkB6trP8fUVYLHLsVlIy5lGx+tgwFEYkG1N8IOO/
2LCawwaWm8FYAFd3IZl04RImNs/IMG7VmH1bf4swHOBHgCN1uYo=
-----END CERTIFICATE-----
```


Asia Pacific (Tokyo) — ap-northeast-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
```

```
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL9KIB7Fgvg/MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAwMjVaGA8yMTk1MDExNzA5MDAyNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAz0djWUcmRW85C5CiCKPFiTIvj6y2OuopFxNE5d3Wtab10bm06vnXVKXu
tz3AndG+Dg0zIL0gMlU+QmrSR0PH2PfV9iejfLak9iwdm1WbwRrCEAj5VxPe0Q+I
KeznOtxzqQ5Wo5NLE9bA61sziUAFNVsTFUzphEwRohcekYyd3bBC4v/RuAjCXHVx
40z6AIksnAOGN2VABMlTeMNvPItKOCIeRLlllSqXX1gbtL1gxSW40JWdF3WPB68E
e+/1U3F7OEr7XqmNODOL6yh92QqZ8fHjG+afOL9Y2Hc4g+P1nk4w4iohQOPABqzb
MPjK7B2Rze0f9OEc51GBQu13kxkWWQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU5DS5IFdU/QwYbikgtWvkU3fDwRgwgY4GA1UdIwSBhjCBg4AU5DS5
IFdU/QwYbikgtWvkU3fDwRihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL9KIB7Fgvg/MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAG/N7ua8IE9IMyno0n5T57erBvLTOQ79fIJN
Mf+mKRM7qRRsdg/eumFft0rLOKo54pJ+Kim2cngCWNhkzctRHBV567AJNt4+ZDG5
hDgV0IxWO1+eaLE4qzqWP/9VrO+p3reuumgFZLVpvVpwXBBeBFUf2drUR14aWfI2
L/6VGINXYs7uP8v/2VBS7r6XZRnPBUy/R4hv5efYXnjwA9gq8+a3stC2ur8m5ySl
faKSwE4H320yAyaZWH4gpwUdbUlYgPHtm/ohRtiWPrN7KEG5Wq/REzMIjZCnxOfS
6KR6PNjlhxBsImQhmBvz6j5PLQxOxBZIpDoiK278e/1Wqm9LrBc=
-----END CERTIFICATE-----
```

Canada (Central) — ca-central-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
```

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

RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJNKhJhaJOuMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA3Mjkx
MTM3MTdaGA8yMTk2MDEwMjExMzcxN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAhDUh6j1ACSt057nSxAcwMaGr8Ez87VA2RW2HyY8l9XoHndnxmP50Cqld
+26AJtltlqHpI1YdtnZ6OrVgVhXcVtbvte0lZ3ldEzC3PMvmISBhHs6A3SWHA9ln
InHbToLX/SWqBHLOX78HkPRaG2k0COHpRy+fG9gvz8HCiQaXCbWNFDHZev9OToNI
xhXBVzIa3AgUnGMalCYZuh5AfVRCEeALG60kxMMC8IoAN7+HG+pMdqAhJxGUcMO0
LBvmTGGeWhi04MUZWfOkwn9JjQZuyLg6B1OD4Y6s0LB2P1MovmSJKGY4JcF8Qu3z
xxUbl7Bh9pvzFR5gJN1pjM2n3gJEPwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAJ
UNKM+gIIHNk0G0tzv6vZBT+o/vt+tIp8lEoZwaPQh1121iw/I7ZvhMLAigx7eyvf
IxUt9/nf8pxWaeGzi98RbSmbap+uxYRynqe1p5rifTamOsguuPrhVpl12OgRWLcT
rjg/K60UMXRsmg2w/cxV45pUBcyVb5h6Op5uEVAVq+CVns13ExiQL6kk3guG4+Yq
LvP1p4DZfeC33a2Rfre2IHLsJH5D4SdWcYqBsfTpf3FQThH0l0KoacGrXtsedsxs
9aRd7OzuSEJ+mBxmzxSjSwM84Ooh78DjkdpQgv967p3d+8NiSLt3/n7MgnUy6WwB
KtDujDnB+ttEHwRRngX7
-----END CERTIFICATE-----
```


Canada West (Calgary) — ca-west-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAYPouptUMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAMITzTJUa6cBsIfdHN69zW/
aHjUB4r1ZfKb1FMhIp9EZtEf5n+06oXjUG2+dKRSlFQeEK333ehNZsPd6uqey6TYKtHpFb5XRLS8BpqB
+7gnbAd0CBZM5o4NWesSQ1GLnTdQcGZkYG/
QESkbadoCXQTifCujJE682hTDLIVt1d4ewwCQYHKoZIzjgEAwMvADAsAhRJc4gRS/HWTkCR2MESaQEe/
jOMNQIUNoTwLvuPrmGPupPlGiHe0veZiO8=
```

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

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAYPou9weMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
v4XBVH13ZCMgqlRHMqV8AWI5iO6gFn2A9sN3AZXTMqwtZeiDdebq3k6Wt7ieYvpXTg0qvgsjQIovRZWaBDBJy9x8C2hW
+w9lMQjFHkJ7Jy/
PHCJ69EzebQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAGe9Snkz1A6rHBH6/5kDtYvtPYwhx2sXNxztbhkXErFk40Nw5l4
gvDVtWG7qyb6fAqgoisyAbk8K9LzxSim2S1nmT9vD84B/t/VvwQBylc
+ej8kRxMH7fquZLp7IXfmtBzyUqu6Dpbne+chG2
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALyTn5IHrIZjMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMzEyMDcx
NTM3MDFaGA8yMjAzMDUxMzE1MzcwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1GP5os424BJmGPCKOSgOclP7lzUiB85du03M4hfjzSOszsBpmBGFDLz1
owYHtIxlq3+Vi1Lt5Qlx3id/ov1QyaBPFWXVeklHVXy9vieCcI3TdjGjTl1W/8MM
m3X26QPcsnHM/Kk2wJ7s186MrqmdSsp3SCPpxv4vEG2Q9yR2bXY41hpc2rWlW8qU
D0JGXluvmmAdFnto20l1XWZ6xFen1h60DRugek/ufCbN+ljKy0xLqPoavH0Ybjsb
UpsAsBs7phaoN+X/5hIERfbp5LfVnqq54pNG5KNu4KynfW9+kA/WS4cJ6FTTN5t+
y0P1HvcL+BL2RuDy6T2bB21xw5WqtQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQURTvu/Dd4zDnmS5G5CfVLnmUBNOswgY4GA1UdIwSBhjCBg4AURTvu
/Dd4zDnmS5G5CfVLnmUBNOuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALyTn5IHrIZjMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFt523A3Aug6/F8xxyITgA8gkU0btFhlXNSP
U4U2OQ9n0tWI9WqnKNWH3KBxwY5EPitU6b3LM4xc9lDWpz7h2Pto+WhxP9LVKe6f
r8r7teTLCVZ7cfYZHzHg+flZjVpAgzE5BVfrRlj3QKpv0hYT3J1wMtI++Vorq5Nf
aPjzedehJLhmZVALwnfqfLrgv6/gmraP9Vmoa8U4D6AljNiQGYaLwyoPoRm3bUs2
v1Mh9GkEQ1b9+1pFXcqgzJJTGRuiPCyPbECI79FAnx5JM/CkGJV8HlOmjIW1qkK1
Y2qT7wzErrKLJyB53Pw15BdIM1onbDAQreZb0yZQLdoEl/tx7Uk=
-----END CERTIFICATE-----
```

Europe (Frankfurt) — eu-central-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
```

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

RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAKD+v6LeR/WrMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTA4MTlaGA8yMTk1MDExNzA5MDgxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAka8FLhxs1cSJGK+Q+q/vTf8zVnDAPZ3U6oqppOW/cupCtpwMAQcky8DY
Yb62GF7+C6usniaq/9W6xPn/3o//wti0cNt6MLsiUeHqNl5H/4U/Q/fR+GA8pJ+L
npqZDG2tFi1WMvvGhGgIbScrjR4VO3TuKy+rZXMYvMRk1RXZ9gPhk6evFnviwHsE
jV5AEjxLz3duD+u/SjPp1vloxe2KuWnyC+EKInnka909sl4ZAUh+qIYfZK85DAjm
GJP4W036E9wTJQF2hZJrzsiB1MGyC1WI9veRISd30izZZL6VVXLXUtHwVHnVASrS
zZDVpzj+3yD5hRXsvFigGhY0FCVFnwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUxC2l6pvJaRflgu3MUdN6zTuP6YcwgY4GA1UdIwSBhjCBg4AUxC2l
6pvJaRflgu3MUdN6zTuP6YehYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAKD+v6LeR/WrMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAIK+DtbUPppJXFqQMv1f2Gky5/82ZwgbbfXa
HBeGSii55b3tsyC3ZW5ZlMJ7Dtnr3vUkiWbV1EUaZGOUlndUFtXUMABCb/coDndw
CAr53XTv7UwGVNe/AFO/6pQDdPxXn3xBhF0mTKPrOGdvYmjZUtQMSVb9lbMWCFfs
w+SwDLnm5NF4yZchIcTs2fdpoyZpOHDXy0xgxO1gWhKTnYbaZOxkJvEvcckxVAwJ
obF8NyJla0/pWdjhlHafEXEN8lyxyTTyOa0BGTuYOBD2cTYYynauVKY4fqHUkr3v
Z6fboaHEd4RFamShM8uvSu6eEFD+qRmvqlcodbpsSOhuGNLzhOQ=
-----END CERTIFICATE-----
```


Europe (Ireland) — eu-west-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
```

```
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOrmqHuaUt0vMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTA2MTlaGA8yMTk1MDQwMzA5MDYxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjE7nVu+aHLtzp9FYV25Qs1mvJ1JXD7J0iQ1Gs/RirW9a5ZECCtc4ssnf
zQHq2JRVr0GRchvDrbm1HaP/avtFQR/Thvfltwu9AROVT22dUOTvERdkNzveoFCy
```

```
hf52Rqf0DMrLXG8ZmQPPXPDFAv+sVMWCDftcChxRYZ6mP9O+TpgYNT1krD5PdvJU
7HcXrkNHDYqbsg8A+Mu2hzl0QkvUET83Csg1ibeK54HP9w+FsD6F5W+6ZSHGJ88l
FI+qYKs7xsjJQYgXWfEt6bbckWs1kZIaIOyMzYdPF6ClYzEec/UhIe/uJyUUNfpT
VIsI5OltBbcPF4c7Y20jOIwwI2SgOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUF2DgPUZivKQR/Zl8mB/MxIkjZDUwgY4GA1UdIwSBhjCBg4AUF2Dg
PUZivKQR/Zl8mB/MxIkjZDWhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOrmqHuaUt0vMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGm6+57W5brzJ3+T8/XsIdLTuiBSe5ALgSqI
qnO5usUKAeQsa+kZIJPyEri5i8LEodh46DAF1RlXTMYgXXxl0YggX88XPmPtok17
l4hib/D9/lu4IaFIyLzYNSzsETYWKWoGVe7ZFz60MTRTwY2u8YgJ5dec7gQgPSGj
avB0vTIgoW41G58sfw5b+wjXCsh0nROon79RcQFFhGnvup0MZ+JbljyhZUYFzCli
31jPZiKzqWa87xh2DbAyvj2KZrZtTe2LQ48Z4G8wWytJzxEeZdREe4NoETf+Mu5G
4CqoaPR05KWkdNUdGNwXewydb3+agdCgfTs+uAjeXKNdSpbhMYg=
-----END CERTIFICATE-----
```


Europe (London) — eu-west-2

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


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

```
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANBx0E2bOCEPMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA4MTEx
NDU2NDJaGA8yMTk2MDExNTE0NTY0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpviBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArYS3mJLGaMrh2DmiPLbqr4Z+xWXTzBWCjOwpsuHE9H6dWUUyl2Bgnu+Z
d8QvW306Yleec45M4F2RA3J4hWHtShzsMlOJVRt+YulGeTf9OCPr26QmIFfs5nD4
fgsJQEry2MBSGA9Fxq3Cw6qkWcrOPsCR+bHOU0XykdKl0MnIbpBf0kTfciAupQEA
dEHnM2J1L2iI0NTLBgKxy5PXLH9weX20BFauNmHH9/J07OpwL20SN5f8TxcM9+pj
Lbk8h1V4KdIwVQpdWkbDL9BCGlYjyadQJxSxz1J343NzrnDM0M4h4HtVaKOS7bQo
Bqt2ruopLRCYgcuFHck/1348iAmbRQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBG
wujwU1Otpi3iBgmhjMClgZyMMn0aQIxMigoFNqXMUNx1Mq/e/Tx+SNaOEAu0n2FF
aiYjvY0/hXOx75ewzZvM7/zJWIdLdsgewpUqOBH4DXFhbSk2TxggSPb0WRqTBxq5
Ed7F7+7GRIeBbRzdLqmISDnfqey8ufW0ks51XcQNomDIRG5s9XZ5KHviDCar8FgL
HngBCdFI04CMagM+pwTO9XN1Ivt+NzUj208ca3oP1IwEAd5KhIhPLcihBQA5/Lpi
h1s3170z1JQ1HZbDrH1pgp+8hSI0DwwDVb3IIH8kPR/J0Qn+hvOl2HOpaUg2Ly0E
pt1RCZe+W7/dF4zsbqwK
-----END CERTIFICATE-----
```

Europe (Milan) — eu-south-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqwCCQCMElHPdwG37jAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA0MjkyMDM1MjJaFw00
NTA0MjkyMDM1MjJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQDAkoL4YfdMI/MrQ0oL
NPfeEk94eiCQA5xNOnU7+2eVQtEqjFbDADFENh1p3sh9Q9OoheLFH8qpSfNDWn/0
ktCS909ApTY6Esx1ExjGSeQq/U+SC2JSuuTT4WFMKJ63a/czMtFkEPPnVIjJJJmT
HJSKSsVUgpdDIRvJXuyB0zdB+wIVALQ3OLaVGdlPMNfS1nD/Yyn+32wnAoGAPBQ3
7XHg5NLOS4326eFRUT+4ornQFjJjP6dp3pOBEzpImNmZTtkCNNUKE4Go9hv5T4lh
R0pODvWv0CBupMAZVBP9ObplXPCyEIZtuDqVa7ukPOUpQNgQhLLAqkigTyXVOSmt
ECBj9tu5WNP/x3iTZTHJ+g0rhIqpgh012UwJpKADgYQAAoGAV1OEQPYQUg5/M3xf
6vE7jKTxxyFWEyjKfJK7PZCzOIGrE/swgACy4PYQW+AwcUweSlK/Hx2OaZVUKzWo
wDUbeu65DcRdw2rSwCbBTU342sitFo/iGCV/Gjf+BaiAJtxniZze7J1ob8vOBeLv
uaMQmgOYeZ5e0fl04GtqPl+lhcQwCQYHKoZIzjgEAwMwADAtAhQdoeWLrkm0K49+
AeBK+j6m2h9SKQIVAIBNhS2a8cQVABDCQXVXrc0tOmO8
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAOZ3GEIaDcugMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTEwMjQx
NTE5MDlaGA8yMTk5MDMyOTE1MTkwOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCjiPgW3vsXRj4JoA16WQDyoPc/eh3QBARaApJEc4nPIGoUolpAXcjFhWplo2O+
ivgfCsc4AU9OpYdAPha3spLey/bhHPRi1JZHRNqScKP0hzsCNmKhfnZTIEQCFvsp
DRp4zr91/WS06/flJFBYJ6JHhp0KwM81XQG59lV6kkoW7QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAGLLrY3P+HH6C57dYgtJkuGZGT2+rMkk2n81/abzTJvsqRqGRrWv
XRKRXlKdM/dfiuYGokDGxiC0Mg6TYy6wvsR2qRhtXW1OtZkiHWcQCnOttz+8vpew
wx8JGMvowtuKB1iMsbwyRqZkFYLcvH+Opfb/Aayi20/ChQLdI6M2R5VU
-----END CERTIFICATE-----
```


RSA-2048


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

```
MIIDOzCCAiOgAwIBAgIJAO/+DgYF78KwMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA0Mjky
MDM1MjJaGA8yMTk4MTAwMjIwMzUyMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAv1ZLV+Z/P6INq+R1qLkzETBg7sFGKPiwHekbpuB6lrRxKHhj8V9vaReM
lnv1Ur5LAPpMPYDsuJ4WoUbPYAqVqyMAo7ikJHCCM1cXgZJefgN6z9bpS+uA3YVh
V/0ipHh/X2hc2S9wvxKWiSHu6Aq9GVpqL035tJQD+NJuqFd+nXrtcw4yGtmvA6wl
5Bjn8WdsP3xOTKjrByYY1BhXpP/f1ohU9jE9dstsRXLa+XTgTPWcWdCS2oRTWPGR
c5Aeh47nnDsyQfP9gLxHeYeQItV/BD9kU/2Hn6mnRg/B9/TYH8qzlRTzLapXp4/5
iNwusrTNexGl8BgvAPrfhjDpdgYuTwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQB7
5ya11K/hKgvaRTvZwVV8GlVZt0CGPtNvOi4AR/UN6TMm51BzUB5nurB4z0R2MoYO
Uts9sLGvSFALJ4otoB77hyNpH3drttU1CVVwal/yK/RQLSon/IoUkaGEbqalu+mH
nYad5IG4tEbmepX456XXcO58MKmnczNbPyw3FRzUZQtI/sf94qBwJ1Xo6XbzPKMy
xjL57LHIZCssD+XPifXay69OFlsCIgLim11HgPkRIHEOXLSf3dsW9r+4CjoZqB/Z
jj/P4TLCxbYCLkvglwaMjgEWF40Img0fhx7yT2X92MiSrs3oncv/IqfdVTiN8OXq
jgnq1bf+EZEZKvb6UCQV
-----END CERTIFICATE-----
```


Europe (Paris) — eu-west-3

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALWSfgHuT/ARMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA1MzEx
MTE4MTZaGA8yMTk2MTEwMzExMTgxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy5V7KDqnEvF3DrSProFcgu/oL+QYD62b1U+Naq8aPuljJe127Sm9WnWA
EBdOSASkOaQ9fzjCPoG5SGgWKxYoZjsevHpmzjVv9+Ci+F57bSuMbjgUbvbRIFUB
bxQojVoXQPHgK5v433ODxkQ4sjRyUbf4YV1AFdfU7zabC698YgPVOExGhXPlTvco
8mlc631ubw2g52j0lzaozUkHPSbknTomhQIvO6kUfX0e0TDMH4jLDG2ZIrUB1L4r
OWKG4KetduFrRZyDHF6ILZu+s6ywiMicUd+2UllDFC6oas+a8D11hmO/rpWU/ieV
jj4rWAFrsebpn+Nhgy96iiVUGS2LuQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDE
iYv6FQ6knXCg+svlcaQG9q59xUC5z8HvJZ1+SxzPKKC4PKQdKvIIfE8GxVXqlZG1
cl5WKTFDMapnzb9RV/DTaVzWx3cMYT77vm1Hl1XGjhx611CGcENH1egI3lOTILsa
+KfopuJEQQ9TDMAIkGjhA+KieU/U5Ctv9fdej6d0GC6OEuwKkTNzPWue6UMq8d4H
2xqJboWsE1t4nybEosvZfQJcZ8jyIYcYBnsG13vCLM+ixjuU5MVVQNMY/gBJzqJB
V+U0QiGiuT5cYgY/QihxdHt99zwGaE0ZBC7213NKrlNuLSrqhDI2NLu8NsExqOFy
OmY0v/xVmQUQl26jJXaM
```

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

Europe (Spain) — eu-south-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCAq
+gAwIBAgIGAXjwLk46MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAGG2m8EKmaf5qQqj3Z
+rzSaTaXE3B/R/4A2VuGqRYR7MljPtwdmU6/3CPjCACcZmTIcOAKbFiDHqadQgBZXfzGpzw8Zo
+eYmmk5fXycgnj57PYH1dIWU6I7mCbAah5MZMcmHaTmIsomGrhcnWB8d8qOU7oZ0UWK4lbiAQs1MihoUwCQYHKoZIzjg
WmbaU7YM5GwCFCvIJOesO5hZ8PHC52dAR8WWC6oe
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjwLkiaMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
VvR1+45Aey5zn3vPk6xBm5o9grSDL6D2iAuprQnfVXn8CIbSDbWFhA3fi5ippjKkh3sl8VyCvCOUXKdOaNrYBrPRkrdH
+3m/
rxIUZ2IK1fDlC6sWAjddf6sBrV2w2a78H0H8EwuwiSgttURBjwJ7KPPJCqaqrQIDAQABMA0GCSqGSIb3DQEBBQUAA4GB
+FzqQDzun/
iMMzcFucmLMl5BxEblrFXOz7IIuOeiGkndmrqUeDCykztLku45s7hxdNy4ltTuVAaE5aNBdw5J8U1mRvsKvHLy2ThH6h
+hBgiphYp84DUbWVYeP8YqLEJSqscKscWC
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALWSmO6DvSpQMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
MzU4NDNaGA8yMjAxMTIyMjEzNTg0M1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
```

```
CgKCAQEAuAAhuSpsHC00/fD2zNlBDpNLRndi9qbHsNeuz3WqN7Samj2aSrM2hS+i
hUxx0BspZj0tZCOsbpPZ+i74NOEQtFeqQoEGvKhB1nJiF4y5I81HDhs5qHvoIivm
7rbbik3zgm1PqS/DmDjVQaXPcD31Rd9ILwBmWEwJqHigyNVlxYtCzTQcrlBrvNZM
dnNgCDAdX/HBEFxx9O12xeu0bSt0s+PJWZ1RTbYrNe7LIH6ntUqHxP/ziQ5trXEZ
uqy7aWk1L8uK4jmyNph0lbaqBa3Y6pYmU1nC27UE4i3fnPB0LSiAr+SrwVvX1g4z
ilo8kr+tbIF+JmcgYLBvO8Jwp+EUqQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUwvGzKJL9A5LReJ4Fxo5K6I20xcowgY4GA1UdIwSBhjCBg4AUwvGz
KJL9A5LReJ4Fxo5K6I20xcqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALWSmO6DvSpQMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAJAZd31jyoTGLawAD2+v/vQsaB9vZIx5EImi
G8YGkd61uFWeNhAmtrwyE/i6FDSIphDrMHBkvw/D3BsqK+Ev/JOK/VYuaYDx/8fp
H4cwp9jC57CXzdIDREWNf6M9PsHFg2WA9XNNtClOZL5WJiJwel8eDSg+sqJUxE0l
MW+QChq/20F6niyaRK4bXrZq14as7h+F9u3A9xHEOVP7Zk9C2ehrBXzCMLSDt3GV
fEuMea2RxMhozwz34Hkdb6j18qoCfygubulovRNQjKw/cEmgPR16KfZPP5caILVt
9qkYPvePmbiVswZDee73cDymJYxLqILpOZwyXvUH8StiH42FHZQ=
-----END CERTIFICATE-----
```


Europe (Stockholm) — eu-north-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALc/uRxg++EnMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
NDAwMTFaGA8yMTk3MDkxMzE0MDAxMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzwCGJEJIxqtr2PD2a1mA6LhRzKhTBa1AZsg3eYfpETXIVlrpojMfvVoN
qHvGshWLgrGTT6os/3gsaADheSaJKavxwX3X6tJA8fvEGqr3a1C1MffH9hBWbQqC
LbfUTAbkwis4GdTUwOwPjT1Cm3u9R/VzilCNwkj7iQ65AFAI8Enmsw3UGldEsop4
yChKB3KW3WI0FTh0+gD0YtjrqqYJxpGOYBpJp5vwdd3fZ4t1vidmDMs7liv4f9Bx
p0oSmUobU4GUlFhBchK1DukICVQdnOVzdMonYm7s+HtpFbVHR8yf6QoixBKGdSal
mBf7+y0ixjCn0pnC0VLVooGo4mi17QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
4ONZiixgk2sjJctwbyD5WKLTH6+mxYcDw+3y/F0fWz561YORhP2FNnPOmEkf0Sl/
Jqk4svzJbCbQeMzRoyaya/46d7UioXMHRZam5IaGBhOdQbi97R4VsQjwQj0RmQsq
yDueDyuKTwWLK9KnvI+ZA6e6bRkdNGflK4N8GGKQ+fBhPwVELkbT9f16OJkezeeN
S+F/gDADGJgmPXfjogICb4Kvshq0H5Lm/xZlDULF2g/cYhyNY6EOI/eS5m1I7R8p
D/m6WoyZdpInxJfxW616OMkxQMRVsruLTNGtby3u1g6ScjmpFtvAMhYejBSdzKG4
FEyxIdEjoeO1jhTsck3R
```

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

Europe (Zurich) — eu-central-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXiKJnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAYNjaCNg/
cfgQOl1BUj5ClUulqwZ9Q+SfDzPZh9D2C0VbiRANiZoxrV8RdgmzzC5T7VcriVwjwvta2Ch//
b+sZ86E5h0XWWr+BeEjD9cu3eDj12XB5sWEbNHNx49p5Tmtu5r2LDtlL8X/
Rpfalu2Z2OJgjFJWGf7hRwxe456n
+lowCQYHKoZIzjgEAwMvADAsAhRChsLcj4U5CVb2cp5M0RE1XbXmhAIUeGSnH+aiUQIWmPEFja+itWDufIk=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjSGFGiMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
opKZAUusJx2hpgU3pUHhlp9ATh/VeVD582jTd9IY
+8t5MDa6Z3fGliByEiXz0LEHdi8MBacLREu1TwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAILlpoE3k9o7KdALAxsFJNit
+g3RMzdbiFM+7MA63Nv5fsf+0xgcjSNBElvPCDKFvTJl4QQhToy056llO5GvdS9RK
+H8xrP2mrqngApoKTApv93vHBixgFSn5KrczRO0YSm3OjkqbydU7DFlmkXXR7GYE+5jbHvQHYiT1J5sMu
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALvT012pxTxNMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
NTEyMDdaGA8yMjAxMTIyMjE1MTIwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAyn+Lsnq1ykrfYlZkk6aAAYNReNd9Iw8AUwCBkgOr2eBiBBepYxHwU85N
++moQ+j0EV2VaahBeTLShGZZS1HsyK8+cYT2QzpghIoamcYhrPXyIxlWiRQlaqSg
```

```
OFiE9bsqL3rCF5Vz+tOiTe5W/7ojfOFls6++g7ZpobwJlpMbuJepqyeHMPyjvO5A
age81lJewc4bxo2ntaW0HCqNksqfYB78j6X6kn3PFpX7FaYAwZA+Xx6C7UCY7rNi
UdQzfAo8htfJi4chz7frpUdQ9kl3IOQrsLshBB5fFUjl09NiFipCGBwi+8ZMeSn1
5qwBI01BWXPfG7WX6Owyjhmh6JtE1wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU8HN4vvJrsZgPQeksMBgJb9xR1yYwgY4GA1UdIwSBhjCBg4AU8HN4
vvJrsZgPQeksMBgJb9xR1yahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALvT012pxTxNMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGlHYDtcHpfBvdHx9HeQE8HgNugJUPdEqxun
t9U33p8VFrs+uLPtrOd9HDJEGvvs5h84EUie/oGJxRt7V1Vlid1PvHf6cRmpjgqY
YdggAVkZtY/PnFVmzf2bMVlSQPrqCl7U0zaw2Kvnj4zgX0rZyCetgrRZSUSxotyp
978Wy9ccXwVSeYG/YAr5rJpS6ZH7eRQvUY0IzwFNeaOPgOTEVpcjWlV6+MQEvsEx
W85q+s6AVr49eppEx8SLJsl0C23yB+L+t32tAveQImRWtJMpzZ5cxh/sYgDVeoC0
85H1NK/7H9fAzT1cPu1oHSnB0xYzzHGOAmXmusMfwUk8fL1RQkE=
-----END CERTIFICATE-----
```


Israel (Tel Aviv) — il-central-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq+gAwIBAgIGAX0QPi
+9MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYX
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAbazCL5XXyPmcw3+oMYQUF5/9YogW6D0FZbYuyPgjOoUwWdl6fjlzWca
pq+l1ezuK2DFOzNTEyPEwwCQYHKoZIzjgEAwMvADAsAhRt1jkpXsvrS
+xTo2M9h2s2uLAhEQIUOZ2FcnTSrshF2EIdixZZwtNv66Q=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAX0QQGVLMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
+S8vOy5hpLoRe4RkOrY0cM3bN07GdEMlin5mU0y1t8y3ct4YewvmkgT42kTyMM
+t1K4S0xsqjXxxS716uGYh7eWtkxrCihj8AbXN/6pa095h
+7TZyl2n83keiNUzM2KoqQVMwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBADwA6VVEIIZD2YL00F12po40xDLzIc9XvqFPS
FmU7H8s62/jD6cOR1A1cClIyZUe1yT1ZbPySCs43J+Thr8i8FSRxzDBSZZi5foW
```

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

RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOVp1h2I9wW7MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTUx
MjQ0MTJaGA8yMjAxMTIxOTEyNDQxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13PkyWvl61iV/SYfO1UFO76UpDfPm2SF/Rz/o33cm699X++EYPxTnoEc
vmWeS0I7eDXc4OCUiToG0sEx0klE0CX1Z1tK6qJ+zgWQLZ9SZEC9H0NsSA6LhrHu
Nq0dzeK3LjhdfcX46/4GqdiptpdTuM4m/hOQ5yx4JMQ/n1sdpv4M5VLRWwWW9Lem
ufb79Id7O9SispxgRsz1KXIjp7N9S4BY7itSXz97uSyzTqEjWZ6mDUhTu3t21GKC
6flALGTTrG2yghEhz53rkvLsvwzjPSSlT6LIf0mrRPzHaf+EdaKoasElE1SHh+ZH
9mI81HywpE+HZ+W+5hBCvjYp90Y1fwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU58tN2JO+yEGq5JbIXxGi4vRVPyIwgY4GA1UdIwSBhjCBg4AU58tN
2JO+yEGq5JbIXxGi4vRVPyKhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOVp1h2I9wW7MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANBN0e1EqNy4+IU2yQzMJ+Wy5ZIOtTP6GSBR
7muVY1bDeAwtNTEOpwgrZVlC7/xq5Q0LC1y0Z70hHXEf8au7qStaAoUtxzvhTAZI
NC0lwoFU56UFw4N0vZII17iqEfoqRC4PpI3OxqEJHFy0VLlvAzJoKB4QLLqDAYVA
LXCiOLoVT+y9tRYsxw5MyO0Bi6fxQIIADl2bE9xkunTN1Jkkwqo3LxNy/ryz4QWR
8K7jHUItifv4h/hxBKpHEquN8CkdvM9oeGl7I8PFrSFEpGr1euDXY0euZzzYiDBV
m6GpTJgzpVsEuIX52dPcPemwQncoIfZyhWDW85MJUnby2WTEcFo=
-----END CERTIFICATE-----
```


Middle East (Bahrain) — me-south-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCVWIgSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDcwojQfgWdV1QliO0B
8n6cLZ38VE7ZmrjZ9OQV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q
PHlP1WGL8IZ34BUgRTtG4TVolvp0smjkMvyRu5hIdKtzjV93Ccx15gVgyk+o1IEG
```

```
fZ2Kbw/Dd8JfoPS7KaSCmJKxXQIVAIZbIaDFRGa2qcMkW2HWASyNDl7bAoGBANtz
IdhfMq+l2I5iofY2oj3HI21Kj3LtZrWEg3W+/4rVhL3lTm0Nne1rl9yGujrjQwy5
Zp9V4A/w9w2O10Lx4K6hj34Eefy/aQnZwNdNhv/FQP7Az0fju+Yl6L13OOHQrL0z
Q+9cF7zEosekEnBQx3v6psNknKgD3Shgx+GO/LpCA4GFAAKBgQCVS7m77nuNAlZ8
wvUqcooxXMPkxJFl54NxAsAul9KP9KN4svm0O3Zrb7t2FOtXRM8zU3TqMpryq1o5
mpMPsZDg6RXo9BF7Hn0DoZ6PJTamkFA6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr
12AztQ8bFWsrTgTzPE3p6U5ckcgV1TAJBgcqhkjOOAQDAy8AMCwCFB2NZGWm5EDl
86ayV3c1PEDukgQIAhQow38rQkN/VwHVeSW9DqEshXHjuQ==
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDPDCCAqWgAwIBAgIJAMl6uIV/zqJFMA0GCSqGSIb3DQEBCwUAMHIxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0dGxlMSAw
HgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwRZWMyLmFt
YXpvbmF3cy5jb20wIBcNMTkwNDI2MTQzMjQ3WhgPMjE5ODA5MjkxNDMyNDdaMHIx
CzAJBgNVBAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0
dGxlMSAwHgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwR
ZWMyLmFtYXpvbmF3cy5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALVN
CDTZEnIeoX1SEYqq6k1BV0ZlpY5y3KnoOreCAE589TwS4MX5+8Fzd6AmACmugeBP
Qk7Hm6b2+g/d4tWycyxLaQlcq81DB1GmXehRkZRgGeRge1ePWd1TUA0I8P/QBT7S
gUePm/kANSFU+P7s7u1NNl+vynyi0wUUrw7/wIZTAgMBAAGjgdcwgdQwHQYDVR0O
BBYEFILtMd+T4YgH1cgc+hVsVOV+480FMIGkBgNVHSMEgZwwgZmAFILtMd+T4YgH
1cgc+hVsVOV+480FoXakdDByMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGlu
Z3RvbjEQMA4GA1UEBwwHU2VhdHRsZTEgMB4GA1UECgwXQW1hem9uIFdlYiBTZXJ2
aWNlcyBMTEMxGjAYBgNVBAMMEWVjMi5hbWF6b25hd3MuY29tggkAyXq4hX/OokUw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOBgQBhkNTBIFgWFd+ZhC/LhRUY
4OjEiykmbEp6hlzQ79T0Tfbn5A4NYDI2icBP0+hmf6qSnIhwJF6typyd1yPK5Fqt
NTpxxcXmUKquX+pHmIkK1LKDO8rNE84jqxrxRsfDi6by82fjVYf2pgjJW8R1FAw+
mL5WQRFexbfB5aXhcMo0AA==
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANZkFlQR2rKqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTAyMDUx
MzA2MjBaGA8yMTk4MDcxMTEzMDYyMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpviBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy4Vnit2eBpEjKgOKBmyupJzJAiT4fr74tuGJNwwa+Is2vH12jMZn9Il1
```

```
UpvvEUYTIboIgISpf6SJ5LmV5rCv4jT4a1Wm0kjfNbiIlkUi8SxZrPypcw24m6ke
BVuxQZrZDs+xDUYIZifTmdgD50u5YE+TLg+YmXKnVgxBU6WZjbuK2INohi71aPBw
2zWUR7Gr/ggIpf635JLU3KIBLNEmrkXCVSnDFlsK4eeCrB7+UNak+4BwgpuykSGG
Op9+2vsuNqFeU1l9daQeG9roHR+4rIWSPa0opmMxv5nctgypOrE6zKXx2dNXQldd
VULv+WH7s6Vm4+yBeG8ctPYH5GOo+QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBs
ZcViiZdFdpcXESZP/KmZNDxB/kktlIEIhsQ+MNn29jayE5oLmtGjHj5dtA3XNKlr
f6PVygVTKbtQLQqunRT83e8+7iCZMKI5ev7pITUQVvTUwI+Fc01JkYZxRFlVBuFA
WGZO+98kxCS4n6tTwVt+nSuJr9BJRVC17apfHBgSS8c5OWna0VU/Cc9ka4eAfQR4
7pYSDU3wSRE01cs30q34lXZ629IyFirSJ5TTOIc0osNL7vwMQYj8HOn4OBYqxKy8
ZJyvfXsIPh0Na76PaBIs6ZlqAOflLrjGzxBPiwRM/XrGmF8ze4KzoUqJEnK13O6A
KHKgfiigQZ1+gv5FlyXH
-----END CERTIFICATE-----
```


Middle East (UAE) — me-central-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXhqnnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAW+csuHsWp/7/
pv8CTKFwxsYudxuR6rbWaHCykIeAydXL9AWnphK6yp1ODEMBFl68Xq8Hp23sOWyf8moOhqCom9+0+ovuUFdpvCie86bp
TOZU568Ty1ff3dDWbdRzeNQRHodRG+XEQSizMkAreeWt4kBa+PUwCQYHKoZIzjgEAwMvADAsAhQD3Z
+XGmzKmgaLgGcVX/Qf1+Tn4QIUH1cgksBSVKbWj81tovBMJeKgdYo=
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjRrnDjMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
KyA6zyruJQrYy00a6wqLA7eeUzk3bMiTkLsTeDQfrkaZMfBAjGaaOymRo1C3qzE4rIenmahvUplu9ZmLwL1idWXMRX2R
+d2SeoKOKQWoc2UOFZMHYxDue7zkyk1CIRaBukTeY13/
RIrlc6X61zJ5BBtZXlHwayjQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBABTqTy3R6RXKPW45FA+cgo7YZEj/
Cnz5YaoUivRRdX2A83BHuBTvJE2+WX00FTEj4hRVjameE1nENoO8Z7fUVloAFDlDo69fhkJeSvn51D1WRrPnoWGgEfr1
B+Wqm3kVEz/QNcz6npmA6
-----END CERTIFICATE-----
```

RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM4h7b1CVhqqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MTEx
MDE1MDNaGA8yMjAxMDkxNTEwMTUwM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApybTWfMOhSoMpqPo72eqAmnn1dXGZM+G8EoZXzwHwT/+IHEXNB4q5N6k
tudYLre1bJxuzEw+iProSHjmb9bB9YscRTofjVhBlt35Fc+i8BaMeH94SR/eE8QO
mll8gnLNW3d62lyuhzuyv1e5wVlRqzYw+X2zRH4/wRDOCOpzjKoHIgyPKsMgwsw5
aTZhNMsGxZN9dbkf0iCGeQLDytwU/JTh/HqvSr3VfUOapTJJiyAxoCtZWgp1/7wC
RvOCSMRJobpUqxZgl/VsttwNkikSFz1wGkcYeSQvk+odbnYQckA8tdddoVI56eD4
qtREQvfpMAX5v7fcqLexl5d5vH8uZQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUOadrbTs+OhzwoAgUJ7RqQNdwufkwgY4GA1UdIwSBhjCBg4AUOadr
bTs+OhzwoAgUJ7RqQNdwufmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM4h7b1CVhqqMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAICTdAOGEOnII8HaGCpCB8us/hGFaLptJaAf
D5SJAyVy66/mdfjGzE1BKkKxnbxemEVUIzbRidOnyilB+pKwN3edAjTZtWdpVA0V
R/G/qQPmcVljtycBz4VC6SuOUYflGzLH1GZ6GJWbuDtFzw8r7HGdRN1wrEPe3UF2
sMpuVezqnRUdvVRoVQP4jFgNsE7kNvtn2NiPhb/CtrxpcwIQ7r6YeoHcBSheuV1Z
xZDHynC3KUprQGx1+Z9QqPrDf180MaoqAlTl4+W6Pr2NJYrVUFGS/ivYshMg574l
CPU6r4wWZSKwEUXq4BInYX6z6iclp/p/J5QnJp2mAwyi6M+I13Y=
-----END CERTIFICATE-----
```


South America (São Paulo) — sa-east-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
```

```
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDIjCCAougAwIBAgIJAKnL4UEDMN/FMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgw
FgYDVQQKEw9BbWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3Mu
Y29tMB4XDTE0MDYwNTE0MjgwMloXDTI0MDYwNTE0MjgwMlowajELMAkGA1UEBhMC
VVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGDAWBgNV
BAoTD0FtYXpvbi5jb20gSW5jLjEaMBgGA1UEAxMRZWMyLmFtYXpvbmF3cy5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAIe9GN//SRK2knbjySG0ho3yqQM3
e2TDhWO8D2e8+XZqck754gFSo99AbT2RmXClambI7xsYHZFapbELC4H91ycihvrD
jbST1ZjkLQgga0NE1q43eS68ZeTDccScXQSNivSlzJZS8HJZjgqzBlXjZftjtdJL
XeE4hwvo0sD4f3j9AgMBAAGjgc8wgcwwHQYDVR0OBBYEFCXWzAgVyrbwnFncFFIs
77VBdlE4MIGcBgNVHSMEgZQwgZGAFCXWzAgVyrbwnFncFFIs77VBdlE4oW6kbDBq
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2Vh
dHRsZTEYMBYGA1UEChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1h
em9uYXdzLmNvbYIJAKnL4UEDMN/FMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAFYcz1OgEhQBXIwIdsgCOS8vEtiJYF+j9uO6jz7VOmJqO+pRlAbRlvY8T
C1haGgSI/A1uZUKs/Zfnph0oEI0/hu1IIJ/SKBDtN5lvmZ/IzbOPIJWirlsllQIQ
7zvWbGd9c9+Rm3p04oTvhup99la7kZqevJK0QRdD/6NpCKsqP/0=
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMcyoxx4U0xxMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU4MDJaGA8yMTk1MDExNzA4NTgwMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAw45IhGZVbQcy1fHBqzROhO8CsrDzxj/WP4cRbJo/2DAnimVrCCDs5O86
FA39Zo1xsDuJHDlwMKqeXYXkJXHYbcPWc6EYYAnR+PlLG+aNSOGUzsy202S03hT0
B20hWPCqpPp39itIRhG4id6nbNRJOzLm6evHuepMAHR4/OV7hyGOiGaV/v9zqiNA
pMCLhbh2xk0PO35HCVBuWt3HUjsgeks2eEsu9Ws6H3JXTCfiqp0TjyRWapM29OhA
```

```
cRJfJ/d/+wBTz1fkWOZ7TF+EWRIN5ITEadlDTPnF1r8kBRuDcS/lIGFwrOOHLo4C
cKoNgXkhTqDDBDu6oNBb2rS0K+sz3QIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUqBy7D847Ya/w321Dfr+rBJGsGTwwgY4GA1UdIwSBhjCBg4AUqBy7
D847Ya/w321Dfr+rBJGsGTyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMcyoxx4U0xxMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACOoWSBf7b9AlcNrl4lr3QWWSc7k90/tUZal
PlT0G3Obl2x9T/ZiBsQpbUvs0lfotG0XqGVVHcIxF38EbVwbw9KJGXbGSCJSEJkW
vGCtc/jYMHXfhx67Szmftm/MTYNvnzsyQQ3v8y3Rdah+xe1NPdpFrwmfL6xe3pFF
cY33KdHA/3PNLdn9CaEsHmcmj3ctaaXLFIzZhQyyjtsrgGfTLvXeXRokktvsLDS/
YgKedQ+jFjzVJqgr4NjfY/Wt7/8kbbdhzaqlB5pCPjLLzv0zp/XmO6k+JvOePOGh
JzGk5t1QrSju+MqNPFk3+1O7o910Vrhqw1QRB0gr1ExrviLbyfU=
-----END CERTIFICATE-----
```


China (Beijing) — cn-north-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```


RSA


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

```
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAOtrM5XLDSjCMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQx
MDAxNDJaGA8yMTk1MDExNzEwMDE0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvVBz+WQNdPiM9S+aUULOQEriTmNDUrjLWLr7SfaOJScBzis5D5ju0jh1
+qJdkbuGKtFX5OTWTm8pWhInX+hIOoS3exC4BaANoa1A3o6quoG+Rsv72qQf8LLH
sgEi6+LMlCN9TwnRKOToEabmDKorss4zFl7VSsbQJwcBSfOcIwbdRRaW9Ab6uJHu
79L+mBR3Ea+G7vSDrVIA8goAPkae6jY9WGw9KxsOrcvNdQoEkqRVtHo4bs9fMRHU
Etphj2gh4ObXlFN92VtvzD6QBs3CcoFWgyWGvzg+dNG5VCbsiiuRdmii3kcijZ3H
Nv1wCcZoEAqH72etVhsuvNRC/xAP8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA8
ezx5LRjzUU9EYWYhyYIEShFlP1qDHs7F4L46/5lc4pL8FPoQm5CZuAF31DJhYi/b
fcV7i3n++/ymQbCLC6kAg8DUB7NrcROll5ag8d/JXGzcTCnlDXLXx1905fPNa+jI
0q5quTmdmiSi0taeaKZmyUdhrB+a7ohWdSdlokEIOtbH1P+g5yll3bI2leYE6Tm8
LKbyfK/532xJPqO9abx4Ddn89ZEC6vvWVNDgTsxERg992Wi+/xoSw3XxkgAryIv1
zQ4dQ6irFmXwCWJqc6kHg/M5W+z60S/94+wGTXmp+19U6Rkq5jVMLh16XJXrXwHe
4KcgIS/aQGVgjM6wivVA
-----END CERTIFICATE-----
```

China (Ningxia) — cn-northwest-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
```

```
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPu4ssY3BlzcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEyMDMy
MTI5MzJaGA8yMTk1MDUwODIxMjkzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsOiGi4A6+YTLzCdIyP8b8SCT2M/6PGKwzKJ5XbSBoL3gsnSWiFYqPg9c
uJPNbiy9wSA9vlyfWMd90qvTfiNrT6vewP813QdJ3EENZOx4ERcf/Wd22tV72kxD
yw1Q3I1OMH4bOItGQAxU5OtXCjBZEEUZooOkU8RoUQOU2Pql4NTiUpzWacNutAn5
HHS7MDc4lUlsJqbN+5QW6fFrcNG/0Mrib3JbwdFUNhrQ5j+Yq5h78HarnUivnX/3
Ap+oPbentv1qd7wvPJu556LZuhfqI0TohiIT1Ah+yUdN5osoaMxTHKKtf/CsSJ1F
w3qXqFJQA0VWsqjFyHXFI32I/GOupwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCn
Um00QHvUsJSN6KATbghowLynHn3wZSQsuS8E0COpcFJFxP2SV0NYkERbXu0n/Vhi
yq5F8v4/bRA2/xpedLWmvFs7QWlomuXhSnYFkd33Z5gnXPb9vRkLwiMSw4uXls35
qQraczUJ9EXDhrv7VmngIk9H3YsxYrlDGEqh/oz4Ze4ULOgnfkauanHikk+BUEsg
/jsTD+7e+niEzJPihHdsvKFDlud5pakEzyxovHwNJ1GS2I//yxrJFIL91mehjqEk
RLPdNse7N6UvSnuXcOokwu6l6kfzigGkJBxkcq4gre3szZFdCQcUioj7Z4xtuTL8
YMqfiDtN5cbD8R8ojw9Y
-----END CERTIFICATE-----
```


AWS GovCloud (US-East) — us-gov-east-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
```

```
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAIe9Hnq82O7UMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMTA3MTQx
NDI3NTdaFw0yNDA3MTMxNDI3NTdaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
qaIcGFFTx/SO1W5G91jHvyQdGP25n1Y91aXCuOOWAUTvSvNGpXrI4AXNrQF+CmIO
C4beBASnHCx082jYudWBBl9Wiza0psYc9flrczSzVLMmN8w/c78F/95NfiQdnUQP
pvgqcMeJo82cgHkLR7XoFWgMrZJqrcUK0gnsQcb6kakCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFNWV53gWJz72F5B1ZVY4O/dfFYBPMIGOBgNVHSME
gYYwgYOAFNWV53gWJz72F5B1ZVY4O/dfFYBPoWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQCHvR56vNju1DASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBACrKjWj460GUPZCGm3/z0dIz
M2BPuH769wcOsqfFZcMKEysSFK91tVtUb1soFwH4/Lb/T0PqNrvtEwD1Nva5k0h2
xZhNNRmDuhOhW1K9wCcnHGRBwY5t4lYL6hNV6hcrqYwGMjTjcAjBG2yMgznSNFle
Rwi/S3BFXISixNx9cILu
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALPB6hxFhay8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
MjMyNDlaGA8yMTk3MDkxMzEyMzI0OVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAva9xsI9237KYb/SPWmeCVzi7giKNron8hoRDwlwwMC9+uHPd53UxzKLb
pTgtJWAPkZVxEdl2Gdhwr3SULoKcKmkqE6ltVFrVuPT33La1UufguT9k8ZDDuO9C
hQNHUdSVEuVrK3bLjaSsMOS7Uxmnn7lYT990IReowvnBNBsBlcabfQTBV04xfUG0
/m0XUiUFjOxDBqbNzkEIblW7vK7ydSJtFMSljga54UAVXibQt9EAIF7B8k9l2iLa
```

```
mu9yEjyQy+ZQICTuAvPUEWe6va2CHVY9gYQLA31/zU0VBKZPTNExjaqK4j8bKs1/
7dOV1so39sIGBz21cUBec1o+yCS5SwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBt
hO2W/Lm+Nk0qsXW6mqQFsAou0cASc/vtGNCyBfoFNX6aKXsVCHxq2aq2TUKWENs+
mKmYu1lZVhBOmLshyllh3RRoL3Ohp3jCwXytkWQ7ElcGjDzNGc0FArzB8xFyQNdK
MNvXDi/ErzgrHGSpcvmGHiOhMf3UzChMWbIr6udoDlMbSIO7+8F+jUJkh4Xl1lKb
YeN5fsLZp7T/6YvbFSPpmbn1YoE2vKtuGKxObRrhU3h4JHdp1Zel1pZ6lh5iM0ec
SD11SximGIYCjfZpRqI3q50mbxCd7ckULz+UUPwLrfOds4VrVVSj+x0ZdY19Plv2
9shw5ez6Cn7E3IfzqNHO
-----END CERTIFICATE-----
```


AWS GovCloud (US-West) — us-gov-west-1

DSA


```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```


RSA


```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAIe9Hnq82O7UMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMTA3MTQx
NDI3NTdaFw0yNDA3MTMxNDI3NTdaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
```

```
qaIcGFFTx/SO1W5G91jHvyQdGP25n1Y91aXCuOOWAUTvSvNGpXrI4AXNrQF+CmIO
C4beBASnHCx082jYudWBBl9Wiza0psYc9flrczSzVLMmN8w/c78F/95NfiQdnUQP
pvgqcMeJo82cgHkLR7XoFWgMrZJqrcUK0gnsQcb6kakCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFNWV53gWJz72F5B1ZVY4O/dfFYBPMIGOBgNVHSME
gYYwgYOAFNWV53gWJz72F5B1ZVY4O/dfFYBPoWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQCHvR56vNju1DASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBACrKjWj460GUPZCGm3/z0dIz
M2BPuH769wcOsqfFZcMKEysSFK91tVtUb1soFwH4/Lb/T0PqNrvtEwD1Nva5k0h2
xZhNNRmDuhOhW1K9wCcnHGRBwY5t4lYL6hNV6hcrqYwGMjTjcAjBG2yMgznSNFle
Rwi/S3BFXISixNx9cILu
-----END CERTIFICATE-----
```


RSA-2048


```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANCOF0Q6ohnuMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTAx
OTQyNDdaGA8yMTk1MDIxMzE5NDI0N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzIcGTzNqie3f1olrrqcfzGfbymSM2QfbTzDIOG6xXXeFrCDAmOq0wUhi
3fRCuoeHlKOWAPu76B9os71+zgF22dIDEVkpqHCjBrGzDQZXXUwOzhm+PmBUI8Z1
qvbVD4ZYhjCujWWzrsX6Z4yEK7PEFjtf4M4W8euw0RmiNwjy+knIFa+VxK6aQv94
lW98URFP2fD84xedHp6ozZlr3+RZSIFZsOiyxYsgiwTbesRMI0Y7LnkKGCIHQ/XJ
OwSISWaCddbu59BZeADnyhl4f+pWaSQpQQ1DpXvZAVBYvCH97J1oAxLfH8xcwgSQ
/se3wtn095VBt5b7qTVjOvy6vKZazwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA/
S8+a9csfASkdtQUOLsBynAbsBCH9Gykq2m8JS7YE4TGvqlpnWehz78rFTzQwmz4D
fwq8byPkl6DjdF9utqZ0JUo/Fxelxom0h6oievtBlSkmZJNbgc2WYm1zi6ptViup
Y+4S2+vWZyg/X1PXD7wyRWuETmykk73uEyeWFBYKCHWsO9sI+62O4Vf8Jkuj/cie
1NSJX8fkervfLrZSHBYhxLbL+actVEo00tiyZz8GnhgWx5faCY38D/k4Y/j5Vz99
7lUX/+fWHT3+lTL8ZZK7fOQWh6NQpI0wTP9KtWqfOUwMIbgFQPoxkP00TWRmdmPz
WOwTObEf9ouTnjG9OZ20
-----END CERTIFICATE-----
```


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** – EKS Runtime Monitoring uses the instance identity role to allow the runtime agent to send security telemetry to the GuardDuty VPC endpoint.
- **AWS Security Token Service (AWS STS)** – Instance identity role credentials can be used with the AWS STS `GetCallerIdentity` action.
- **AWS Systems Manager** – When using Default Host Management Configuration, AWS Systems Manager uses the identity provided by the instance identity role to register EC2 instances. After identifying your instance, Systems Manager can pass your `AWSSystemsManagerDefaultEC2InstanceManagementRole` IAM role to your instance.

Instance identity roles can't be used with other AWS services or features because they do not have an integration with instance identity roles.

Instance identity role ARN

The instance identity role ARN takes the following format:

```
arn:aws-partition:iam::account-number:assumed-role/aws:ec2-instance/instance-id
```

For example:

```
arn:aws:iam::0123456789012:assumed-role/aws:ec2-instance/i-0123456789example
```

For more information about ARNs, see Amazon Resource Names (ARNs) in the *IAM User Guide*.

Best practices and recommendations for SQL Server clustering in Amazon EC2

For information about SQL Server clustering in Amazon EC2, see Best practices and recommendations for SQL Server clustering on Amazon EC2 in the *Microsoft SQL Server on Amazon EC2 User Guide*.

Install WSL on your Windows instance

Windows Subsystem for Linux (WSL) is a free download that you can install on your Windows instance. By installing WSL, you can run native Linux command line tools directly on your Windows instance and use the Linux tools for scripting, alongside your traditional Windows desktop. You can easily swap between Linux and Windows on a single Windows instance, which you might find useful in a development environment.

For more information about WSL, see the Windows Subsystem for Linux Documentation on the *Microsoft Build* website.

Limitations

- WSL is available in two versions: WSL 1 and WSL 2.
  - For `.metal` EC2 instances, you can install either WSL 1 or WSL 2.
  - For virtualized EC2 instances, you must install WSL 1.
- For Windows Server operating systems, WSL can only be installed on instances running the following:
  - Windows Server 2019
  - Windows Server 2022

Install WSL

The following instructions install WSL on an EC2 instance running Windows Server 2022. For the instructions to install WSL on an EC2 instance running Windows Server 2019, see Install WSL on previous versions of Windows Server on the Microsoft website. After you've followed those instructions, you can use step 3 in the instructions below to configure WSL to use WSL 1.

To install WSL 1

1. To install WSL, run the following standard installation command on your EC2 instance, but make sure to enable WSL 1 by including `--enable-wsl1`. By default, WSL 2 is installed. If your instance was launched using a virtualized instance type, you must complete step 3 in this procedure to set the version to WSL 1.

   ```
   wsl --install --enable-wsl1
   ```

2. Restart your EC2 instance.
3. To configure WSL to use WSL 1, run the following command on your instance. For more information about setting the WSL version, see Manual installation steps for older versions of WSL on the *Microsoft Build* website.

   ```
   wsl --set-default-version 1
   ```

To install WSL 2

- To install WSL, run the following standard installation command on your EC2 instance. By default, WSL 2 is installed. If you are installing WSL on a `.metal` instance, then this is the only step to perform.

  ```
  wsl --install
  ```

  For more information, see Install Linux on Windows with WSL on the *Microsoft Build* website.

Upgrade an Amazon EC2 Windows instance to a newer version of Windows Server

There are two methods to upgrade an earlier version of Windows Server running on an instance: in-place upgrade and migration (also called side-by-side upgrade). An in-place upgrade upgrades the operating system files while your personal settings and files are intact. A migration involves capturing settings, configurations, and data and porting these to a newer operating system on a fresh Amazon EC2 instance.

Microsoft has traditionally recommended migrating to a newer version of Windows Server instead of upgrading. Migrating can result in fewer upgrade errors or issues, but can take longer than an in-place upgrade because of the need to provision a new instance, plan for and port applications, and adjust configurations settings on the new instance. An in-place upgrade can be faster, but software incompatibilities can produce errors.

Contents

- Perform an in-place upgrade
- Perform an automated upgrade
- Migrate to latest generation instance types
- Windows to Linux replatforming assistant for Microsoft SQL Server databases
- Troubleshoot an upgrade

Perform an in-place upgrade

Before you perform an in-place upgrade, you must determine which network drivers the instance is running. PV network drivers enable you to access your instance using Remote Desktop. Instances use either AWS PV, Intel Network Adapter, or the Enhanced Networking drivers. For more information, see Paravirtual drivers for Windows instances.

Before you begin an in-place upgrade

Complete the following tasks and note the following important details before you begin your in-place upgrade.

- Read the Microsoft documentation to understand the upgrade requirements, known issues, and restrictions. Also review the official instructions for upgrading.
  - Upgrade Options for Windows Server 2012
  - Upgrade Options for Windows Server 2012 R2
  - Upgrade and conversion options for Windows Server 2016
  - Upgrade and conversion options for Windows Server 2019
  - Upgrade and conversion options for Windows Server 2022
  - Windows Server Upgrade Center
- We recommend performing an operating system upgrade on instances with at least 2 vCPUs and 4GB of RAM. If needed, you can change the instance to a larger size of the same type (t2.small

to t2.large, for example), perform the upgrade, and then resize it back to the original size. If you are required to retain the instance size, you can monitor the progress using the instance console screenshot. For more information, see Change the instance type.

- Verify that the root volume on your Windows instance has enough free disk space. The Windows Setup process might not warn you of insufficient disk space. For information about how much disk space is required to upgrade a specific operating system, see the Microsoft documentation. If the volume does not have enough space, it can be expanded. For more information, see Amazon EBS Elastic Volumes in the *Amazon EBS User Guide*.
- Determine your upgrade path. You must upgrade the operating system to the same architecture. For example, you must upgrade a 32-bit system to a 32-bit system. Windows Server 2008 R2 and later are 64-bit only.
- Disable antivirus and anti-spyware software and firewalls. These types of software can conflict with the upgrade process. Re-enable antivirus and anti-spyware software and firewalls after the upgrade completes.
- Update to the latest drivers as described in Migrate to latest generation instance types.
- The Upgrade Helper Service only supports instances running Citrix PV drivers. If the instance is running Red Hat drivers, you must manually upgrade those drivers first.

Upgrade an instance in-place with AWS PV, Intel Network Adapter, or the Enhanced Networking drivers

Use the following procedure to upgrade a Windows Server instance using the AWS PV, Intel Network Adapter, or the Enhanced Networking network drivers.

To perform the in-place upgrade

1. Create an AMI of the system you plan to upgrade for either backup or testing purposes. You can then perform the upgrade on the copy to simulate a test environment. If the upgrade completes, you can switch traffic to this instance with little downtime. If the upgrade fails, you can revert to the backup. For more information, see Create a custom Windows AMI.
2. Ensure that your Windows Server instance is using the latest network drivers.
   a. To update your AWS PV driver, see Upgrade PV drivers on Windows instances.
   b. To update your ENA driver, see Install or upgrade Elastic Network Adapter (ENA) driver.
   c. To update your Intel drivers, see Enable enhanced networking with the Intel 82599 VF interface on Windows instances

3. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
4. In the navigation pane, choose **Instances**. Locate the instance. Make a note of the instance ID and Availability Zone for the instance. You need this information later in this procedure.
5. If you are upgrading from Windows Server 2012 or 2012 R2 to Windows Server 2016, 2019, or 2022, perform the following on your instance before proceeding:
   a. Uninstall the EC2Config service. For more information, see Stop, restart, delete, or uninstall EC2Config.
   b. Install EC2Launch v1 or the EC2Launch v2 agent. For more information, see Configure a Windows instance using EC2Launch and Configure a Windows instance using EC2Launch v2.
   c. Install the AWS Systems Manager SSM Agent. For more information, see Working with SSM Agent in the *AWS Systems Manager User Guide*.
6. Create a new volume from a Windows Server installation media snapshot.
   a. In the left navigation pane, under **Elastic Block Store**, choose **Snapshots**.
   b. From the filter bar, choose **Public snapshots**.
   c. In the search bar, specify the following filters:
      - Choose **Owner Alias**, then **=**, then **amazon**.
      - Choose **Description**, and then start typing **`Windows`**. Select the Windows filter that matches the system architecture and language preference you're upgrading to. For example, choose **Windows 2019 English Installation Media** to upgrade to Windows Server 2019.
   d. Select the check box next to the snapshot that matches the system architecture and language preference you're upgrading to, and then choose **Actions**, **Create volume from snapshot**.
   e. On the **Create volume** page, choose the Availability Zone that matches your Windows instance, and then choose **Create volume**.
7. In the **Successfully created volume vol-*`1234567890example`*** banner at the top of the page, choose the ID of the volume that you just created.
8. Choose **Actions**, **Attach volume**.
9. On the **Attach volume** page, for **Instance**, select the instance ID of your Windows instance, and then choose **Attach volume**.

10. Make the new volume available for use by following the steps at Make an Amazon EBS volume available for use on Windows.

> **Important**
>
> Do not initialize the disk because doing so will delete the existing data.

11. In Windows PowerShell, switch to the new volume drive. Begin the upgrade by opening the installation media volume you attached to the instance.

    a. If you are upgrading to Windows Server 2016 or later, run the following:

    ```
    .\setup.exe /auto upgrade /dynamicupdate disable
    ```

    > **Note**
    >
    > Running the setup.exe with the `/dynamicupdate` option set to disabled prevents Windows from installing updates during the Windows Server upgrade process, as installing updates during the upgrade can cause failures. You can install updates with Windows Update after the upgrade completes.

    If you are upgrading to an earlier version of Windows Server, run the following:

    ```
    Sources\setup.exe
    ```

    b. For **Select the operating system you want to install**, select the full installation SKU for your Windows Server instance, and choose **Next**.

    c. For **Which type of installation do you want?**, choose **Upgrade**.

    d. Complete the wizard.

Windows Server Setup copies and processes files. After several minutes, your Remote Desktop session closes. The time it takes to upgrade depends on the number of applications and server roles running on your Windows Server instance. The upgrade process could take as little as 40 minutes or several hours. The instance fails status check 1 of 2 during the upgrade process. When the upgrade completes, both status checks pass. You can check the system log for console output

or use Amazon CloudWatch metrics for disk and CPU activity to determine whether the upgrade is progressing.

> **Note**
>
> If upgrading to Windows Server 2019, after the upgrade is complete you can change the desktop background manually to remove the previous operating system name if desired.

If the instance has not passed both status checks after several hours, see Troubleshoot an upgrade.

Post upgrade tasks

1. Log in to the instance to initiate an upgrade for the .NET Framework and reboot the system when prompted.
2. If you haven't already done so in a prior step, install the EC2Launch v1 or EC2Launch v2 agent. For more information, see Configure a Windows instance using EC2Launch and Configure a Windows instance using EC2Launch v2.
3. If you upgraded to Windows Server 2012 R2, we recommend that you upgrade the PV drivers to AWS PV drivers. If you upgraded on a Nitro-based instance, we recommend that you install or upgrade the NVME and ENA drivers. For more information, see Windows Server 2012 R2, Install or upgrade AWS NVMe drivers using PowerShell, or Enable enhanced networking on Windows.
4. Re-enable antivirus and anti-spyware software and firewalls.

Perform an automated upgrade

You can perform an automated upgrade of your Windows and SQL Server instances on AWS with AWS Systems Manager Automation runbooks.

Contents

- Related services
- Execution options
- Upgrade Windows Server
- Upgrade SQL Server

Related services

The following AWS services are used in the automated upgrade process:

- **AWS Systems Manager**. AWS Systems Manager is a powerful, unified interface for centrally managing your AWS resources. For more information, see the *AWS Systems Manager User Guide*.
- AWS Systems Manager Agent (SSM Agent) is Amazon software that can be installed and configured on an Amazon EC2 instance, an on-premises server, or a virtual machine (VM). SSM Agent makes it possible for Systems Manager to update, manage, and configure these resources. The agent processes requests from the Systems Manager service in the AWS Cloud, and then runs them as specified in the request. For more information, see Working with SSM Agent in the *AWS Systems Manager User Guide*.
- **AWS Systems Manager SSM runbooks**. An SSM runbook defines the actions that Systems Manager performs on your managed instances. SSM runbooks use JavaScript Object Notation (JSON) or YAML, and they include steps and parameters that you specify. This topic uses two Systems Manager SSM runbooks for automation. For more information, see AWS Systems Manager Automation runbook reference in the *AWS Systems Manager User Guide*.

Execution options

When you select **Automation** on the Systems Manager console, select **Execute**. After you select an Automation document, you are then prompted to choose an automation execution option. You choose from the following options. In the steps for the paths provided later in this topic, we use the **Simple execution** option.

Simple execution

Choose this option if you want to update a single instance but do not want to go through each automation step to audit the results. This option is explained in further detail in the upgrade steps that follow.

Rate control

Choose this option if you want to apply the upgrade to more than one instance. You define the following settings.

- **Parameter**

This setting, which is also set in Multi-Account and Region settings, defines how your automation branches out.

- **Targets**

  Select the target to which you want to apply the automation. This setting is also set in Multi-Account and Region settings.

- **Parameter Values**

  Use the values defined in the automation document parameters.

- **Resource Group**

  In AWS, a resource is an entity you can work with. Examples include Amazon EC2 instances, AWS CloudFormation stacks, or Amazon S3 buckets. If you work with multiple resources, it might be useful to manage them as a group as opposed to moving from one AWS service to another for every task. In some cases, you may want to manage large numbers of related resources, such as EC2 instances that make up an application layer. In this case, you will likely need to perform bulk actions on these resources at one time.

- **Tags**

  Tags help you categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This categorization is useful when you have many resources of the same type. You can quickly identify a specific resource using the assigned tags.

- **Rate Control**

  Rate Control is also set in Multi-Account and Region settings. When you set the rate control parameters, you define how many of your fleet to apply the automation to, either by target count or by percentage of the fleet.

Multi-Account and Region

In addition to the parameters specified under Rate Control that are also used in the Multi-Account and Region settings, there are two additional settings:

- **Accounts and organizational units (OUs)**

  Specify multiple accounts on which you want to run the automation.

- **AWS Regions**

Specify multiple AWS Regions where you want to run the automation.

Manual execution

This option is similar to **Simple execution**, but allows you to step through each automation step and audit the results.

Upgrade Windows Server

The `AWSEC2-CloneInstanceAndUpgradeWindows` runbook creates an Amazon Machine Image (AMI) from a Windows Server instance in your account and upgrades this AMI to a supported version of your choice. This multi-step process can take up to two hours to complete.

To upgrade your Windows Server 2008 R2 instance to Windows Server 2016, 2019, or 2022, an in-place upgrade is performed twice, first from Windows Server 2008 R2 to Windows Server 2012 R2, and then from Windows Server 2012 R2 to Windows Server 2016, 2019, or 2022. Directly upgrading Windows Server 2008 R2 to Windows Server 2016, 2019, or 2022 is not supported.

In this workflow, the automation creates an AMI from the instance and then launches the new AMI in the subnet you provide. The automation workflow performs an in-place upgrade from Windows Server 2008 R2, 2016, 2019 to the selected version (Windows Server 2012 R2, 2016, 2019, or 2022). It also updates or installs the AWS drivers required by the upgraded instance. After the upgrade is complete, the workflow creates a new AMI and terminates the upgraded instance. If you upgrade from Windows Server 2008 R2 to Windows Server 2016, 2019, or 2022, the automation creates two AMIs because the in-place upgrade is performed twice.

There are two AMIs included in the automated upgrade process:

- **Current running instance**. The first AMI is the current running instance, which is not upgraded. This AMI is used to launch another instance to run the in-place upgrade. When the process is complete, this AMI is deleted from your account, unless you specifically request to keep the original instance. This setting is handled by the parameter `KeepPreUpgradeImageBackUp` (default value is `false`, which means the AMI is deleted by default).
- **Upgraded AMI**. This AMI is the outcome of the automation process.

The final result is one AMI, which is the upgraded instance of the AMI.

When the upgrade is complete, you can test your application functionality by launching the new AMI in your Amazon VPC. After testing, and before you perform another upgrade, schedule application downtime before completely switching to the upgraded instance.

Windows Server automated upgrade paths

The Systems Manager Automation runbook AWSEC2-CloneInstanceAndUpgradeWindows supports the following upgrade paths:

- Windows Server 2008 R2 to Windows Server 2012 R2
- Windows Server 2012 R2 to Windows Server 2016
- Windows Server 2012 R2 to Windows Server 2019
- Windows Server 2012 R2 to Windows Server 2022
- Windows Server 2016 to Windows Server 2019
- Windows Server 2016 to Windows Server 2022
- Windows Server 2019 to Windows Server 2022

Prerequisites

In order to automate your Windows Server upgrade with the AWS Systems Manager Automation document, you must perform the following tasks:

- Create an IAM role with the specified IAM policies to allow Systems Manager to perform automation tasks on your Amazon EC2 instances and verify that you meet the prerequisites to use Systems Manager. For more information, see Creating a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.
- Select the option for how you want the automation to be run. The options for execution are **Simple execution**, **Rate control**, **Multi-account and Region**, and **Manual execution**. For more information about these options, see Execution options.
- Verify that SSM Agent is installed on your instance. For more information see Installing and configuring SSM Agent on Amazon EC2 instances for Windows Server.
- Windows PowerShell 3.0 or later must be installed on your instance.
- For instances that are joined to a Microsoft Active Directory domain, we recommend specifying a `SubnetId` that does not have connectivity to your domain controllers to help avoid hostname conflicts.

- The instance subnet must have outbound connectivity to the internet, which provides access to AWS services such as Amazon S3 and access to download patches from Microsoft. This requirement is met if either the subnet is a public subnet and the instance has a public IP address, or if the subnet is a private subnet with a route that sends internet traffic to a public NAT device.
- This Automation works with instances running Windows Server 2008 R2, Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019.
- Verify that the instance has 20 GB of free disk space in the boot disk.
- If the instance does not use a Windows license provided by AWS, then specify an Amazon EBS snapshot ID that includes Windows Server 2012 R2 installation media. To do this:
  1. Verify that the Amazon EC2 instance is running Windows Server 2012 or later.
  2. Create a 6 GB Amazon EBS volume in the same Availability Zone where the instance is running. Attach the volume to the instance. Mount it, for example, as drive D.
  3. Right-click the ISO and mount it to an instance as, for example, drive E.
  4. Copy the content of the ISO from drive E:\ to drive D:\
  5. Create an Amazon EBS snapshot of the 6 GB volume created in step 2 above.

Windows Server upgrade limitations

This automation doesn't support upgrading Windows domain controllers, clusters, or Windows desktop operating systems. In addition, this automation doesn't support Amazon EC2 instances for Windows Server with the following roles installed:

- Remote Desktop Session Host (RDSH)
- Remote Desktop Connection Broker (RDCB)
- Remote Desktop Virtualization Host (RDVH)
- Remote Desktop Web Access (RDWA)

Steps to perform an automated upgrade of Windows Server

Follow these steps to upgrade your Windows Server instance using the AWSEC2-CloneInstanceAndUpgradeWindows automation runbook.

1. Open Systems Manager from the **AWS Management Console**.
2. From the left navigation pane, under **Change Management**, choose **Automation**.

3. Choose **Execute automation**.
4. Search for the automation document called `AWSEC2-CloneInstanceAndUpgradeWindows`.
5. When the document name appears, select it. When you select it, the document details appear.
6. Choose **Execute automation** to input the parameters for this document. Leave **Simple execution** selected at the top of the page.
7. Enter the requested parameters based on the following guidance.

   - `InstanceID`

     Type: String

     (Required) The instance running Windows Server 2008 R2, 2012 R2, 2016, or 2019 with the SSM agent installed.

   - `InstanceProfile`.

     Type: String

     (Required) The IAM instance profile. This is the IAM role used to perform the Systems Manager automation against the Amazon EC2 instance and AWS AMIs. For more information, see Create an IAM Instance Profile for Systems Manager in the *AWS Systems Manager User Guide*.

   - `TargetWindowsVersion`

     Type: String

     (Required) Select the target Windows version.

   - `SubnetId`

     Type: String

     (Required) This is the subnet for the upgrade process and where your source EC2 instance resides. Verify that the subnet has outbound connectivity to AWS services, including Amazon S3, and also to Microsoft (in order to download patches).

   - `KeepPreUpgradedBackUp`

     Type: String

(Optional) If this parameter is set to `true`, the automation retains the image created from the instance. The default setting is `false`.

- `RebootInstanceBeforeTakingImage`

  Type: String

  (Optional) The default is `false` (no reboot). If this parameter is set to `true`, Systems Manager reboots the instance before creating an AMI for the upgrade.

8. After you have entered the parameters, choose **Execute**. When the automation begins, you can monitor the execution progress.
9. When the automation completes, you will see the AMI ID. You can launch the AMI to verify that the Windows OS is upgraded.

> **Note**
>
> It is not necessary for the automation to run all of the steps. The steps are conditional based on the behavior of the automation and instance. Systems Manager might skip some steps that are not required.
> Additionally, some steps may time out. Systems Manager attempts to upgrade and install all of the latest patches. Sometimes, however, patches time out based on a definable timeout setting for the given step. When this happens, the Systems Manager automation continues to the next step to ensure that the internal OS is upgraded to the target Windows Server version.

10. After the automation completes, you can launch an Amazon EC2 instance using the AMI ID to review your upgrade. For more information about how to create an Amazon EC2 instance from an AWS AMI, see How do I launch an EC2 instance from a custom Amazon Machine Image (AMI)?

Upgrade SQL Server

The `AWSEC2-CloneInstanceAndUpgradeSQLServer` script creates an AMI from an Amazon EC2 instance running SQL Server in your account, and then upgrades the AMI to a later version of SQL Server. This multi-step process can take up to two hours to complete.

In this workflow, the automation creates an AMI from the instance and then launches the new AMI in the subnet you provide. The automation then performs an in-place upgrade of SQL Server.

After the upgrade is complete, the automation creates a new AMI before terminating the upgraded instance.

There are two AMIs included in the automated upgrade process:

- **Current running instance**. The first AMI is the current running instance, which is not upgraded. This AMI is used to launch another instance to run the in-place upgrade. When the process is complete, this AMI is deleted from your account, unless you specifically request to keep the original instance. This setting is handled by the parameter `KeepPreUpgradeImageBackUp` (default value is `false`, which means the AMI is deleted by default).
- **Upgraded AMI**. This AMI is the outcome of the automation process.

The final result is one AMI, which is the upgraded instance of the AMI.

When the upgrade is complete, you can test your application functionality by launching the new AMI in your Amazon VPC. After testing, and before you perform another upgrade, schedule application downtime before completely switching to the upgraded instance.

SQL Server automated upgrade paths

The AWSEC2-CloneInstanceAndUpgradeSQLServer automation runbook supports the following upgrade paths:

- SQL Server 2008 to SQL Server 2017, 2016, or 2014
- SQL Server 2008 R2 to SQL Server 2017, 2016, or 2014
- SQL Server 2012 to SQL Server 2019, 2017, 2016, or 2014
- SQL Server 2014 to SQL Server 2019, 2017, or 2016
- SQL Server 2016 to SQL Server 2019 or 2017
- SQL Server 2017 to SQL Server 2019

Prerequisites

In order to automate your SQL Server upgrade with the AWS Systems Manager Automation document, you must perform the following tasks:

- Create an IAM role with the specified IAM policies to allow Systems Manager to perform automation tasks on your Amazon EC2 instances and verify that you meet the prerequisites to

use Systems Manager. For more information, see Creating a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.

- Select the option for how you want the automation to be run. The options for execution are **Simple execution**, **Rate control**, **Multi-account and Region**, and **Manual execution**. For more information about these options, see Execution options.
- The Amazon EC2 instance must use Windows Server 2008 R2 or later and SQL Server 2008 or later.
- Verify that SSM Agent is installed on your instance. For more information, see Working with SSM Agent on Amazon EC2 instances for Windows Server.
- Verify that the instance has enough free disk space:
  - If you are upgrading from Windows Server 2008 R2 to 2012 R2, or from Windows Server 2012 R2 to a later operating system, verify that you have 20 GB of free disk space in the instance boot disk.
  - If you are upgrading from Windows Server 2008 R2 to 2016 or later, verify that the instance has 40 GB of free disk space in the instance boot disk.
- For instances that use a Bring Your Own License (BYOL) SQL Server version, the following additional prerequisites apply:
  - Provide an Amazon EBS snapshot ID that includes the target SQL Server installation media. To do this:
    1. Verify that the Amazon EC2 instance is running Windows Server 2008 R2 or later.
    2. Create a 6 GB Amazon EBS volume in the same Availability Zone where the instance is running. Attach the volume to the instance. Mount it, for example, as drive D.
    3. Right-click the ISO and mount it to an instance as, for example, drive E.
    4. Copy the content of the ISO from drive E:\ to drive D:\
    5. Create an Amazon EBS snapshot of the 6 GB volume created in step 2.

SQL Server automated upgrade limitations

The following limitations apply when using the AWSEC2-CloneInstanceAndUpgradeSQLServer runbook to perform an automated upgrade:

- The upgrade can be performed on only a SQL Server using Windows authentication.
- Verify that no security patch updates are pending on the instances. Open **Control Panel**, then choose **Check for updates**.

- SQL Server deployments in HA and mirroring mode are not supported.

Steps to perform an automated upgrade of SQL Server

Follow these steps to upgrade your SQL Server using the AWSEC2-CloneInstanceAndUpgradeSQLServer automation runbook.

1. If you haven't already, download the SQL Server 2016 .iso file and mount it to the source server.
2. After the .iso file is mounted, copy all of the component files and place them on any volume of your choice.
3. Take an Amazon EBS snapshot of the volume and copy the snapshot ID onto a clipboard for later use. For more information about creating an EBS snapshot, see Create Amazon EBS snapshots.
4. Attach the instance profile to the Amazon EC2 source instance. This allows Systems Manager to communicate with the EC2 instance and run commands on it after it is added to the AWS Systems Manager service. For this example, we named the role `SSM-EC2-Profile-Role` with the `AmazonSSMManagedInstanceCore` policy attached to the role. See Create an IAM instance profile for Systems Manager in the *AWS Systems Manager User Guide*.
5. In the AWS Systems Manager console, in the left navigation pane, choose **Managed Instances**. Verify that your EC2 instance is in the list of managed instance. If you don't see your instance after a few minutes, see Where Are My Instances? in the *AWS Systems Manager User Guide*.
6. In the left navigation pane, under **Change Management** choose **Automation**.
7. Choose **Execute automation**.
8. Search for the automation document called `AWSEC2-CloneInstanceAndUpgradeSQLServer`.
9. Choose the `AWSEC2-CloneInstanceAndUpgradeSQLServer` SSM document, and then choose **Next**.
10. Ensure that the **Simple execution** option is selected.
11. Enter the requested parameters based on the following guidance.

    - `InstanceId`

      Type: String

      (Required) The instance running SQL Server 2008 R2 (or later).

- `IamInstanceProfile`

  Type: String

  (Required) The IAM instance profile.

- `SQLServerSnapshotId`

  Type: String

  (Required) The Snapshot ID for the target SQL Server installation media. This parameter is not required for SQL Server license-included instances.

- `SubnetId`

  Type: String

  (Required) This is the subnet for the upgrade process and where your source EC2 instance resides. Verify that the subnet has outbound connectivity to AWS services, including Amazon S3, and also to Microsoft (in order to download patches).

- `KeepPreUpgradedBackUp`

  Type: String

  (Optional) If this parameter is set to `true`, the automation retains the image created from the instance. The default setting is `false`.

- `RebootInstanceBeforeTakingImage`

  Type: String

  (Optional) The default is `false` (no reboot). If this parameter is set to `true`, Systems Manager reboots the instance before creating an AMI for the upgrade.

- `TargetSQLVersion`

  Type: String

  (Optional) The target SQL Server version. The default is `2016`.

12. After you have entered the parameters, choose **Execute**. When the automation begins, you can monitor the execution progress.
13. When **Execution status** shows **Success**, expand **Outputs** to view the AMI information. You can use the AMI ID to launch your SQL Server instance for the VPC of your choice.

14. Open the Amazon EC2 console. In the left navigation pane, choose **AMIs**. You should see the new AMI.
15. To verify that the new SQL Server version has been successfully installed, choose the new AMI and choose **Launch**.
16. Choose the type of instance that you want for the AMI, the VPC and subnet that you want to deploy to, and the storage that you want to use. Because you're launching the new instance from an AMI, the volumes are presented to you as an option to include within the new EC2 instance you are launching. You can remove any of these volumes, or you can add volumes.
17. Add a tag to help you identify your instance.
18. Add the security group or groups to the instance.
19. Choose **Launch Instance**.
20. Choose the tag name for the instance and select **Connect** under the **Actions** dropdown.
21. Verify that the new SQL Server version is the database engine on the new instance.

Migrate to latest generation instance types

The AWS Windows AMIs are configured with the default settings used by the Microsoft installation media, with some customizations. The customizations include drivers and configurations that support the latest generation instance types, which are instances built on the Nitro System, such as an M5 or C5.

When migrating to Nitro-based instances, including bare metal instances, we recommend that you follow the steps in this topic in the following cases:

- If you are launching instances from custom Windows AMIs
- If you are launching instances from Windows AMIs provided by Amazon that were created before August 2018

For more information, see Amazon EC2 Update — Additional Instance Types, Nitro System, and CPU Options.

> **Note**
>
> The following migration procedures can be performed on Windows Server version 2008 R2 and later. To migrate Linux instances to the latest generation instance types, see Change the instance type.

Contents

- Part 1: Install and upgrade AWS PV drivers
- Part 2: Install and upgrade ENA
- Part 3: Upgrade AWS NVMe drivers
- Part 4: Update EC2Config and EC2Launch
- Part 5: Install the serial port driver for bare metal instances
- Part 6: Update power management settings
- Part 7: Update Intel chipset drivers for new instance types
- (Alternative) Upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager
- Migrate to Xen instance types from Nitro instance types

> **Note**
>
> Alternatively, you can use the `AWSSupport-UpgradeWindowsAWSDrivers` automation document to automate the procedures described in Part 1, Part 2, and Part 3. If you choose to use the automated procedure, see (Alternative) Upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager, and then continue with Part 4 and Part 5.

Before you begin

This procedure assumes that you are currently running on a previous generation Xen-based instance type, such as an M4 or C4, and you are migrating to an instance based on the Nitro System, such as an M5 or C5.

You must use PowerShell version 3.0 or later to successfully perform the upgrade.

> **Note**
>
> When migrating to the latest generation instances, the static IP or custom DNS network settings on the existing ENI may be lost as the instance will default to a new Enhanced Networking Adapter device.

Before following the steps in this procedure, we recommend that you create a backup of the instance. From the EC2 console, choose the instance that requires the migration, open the context (right-click) menu, and choose **Instance State**, **Stop**.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To preserve data on instance store volumes, ensure that you back up the data to persistent storage.

Open the context (right-click) menu for the instance in the EC2 console, choose **Image**, and then choose **Create Image**.

> **Note**
>
> Parts 4 and 5 of these instructions can be completed after you migrate or change the instance type to the latest generation, such as M5 or C5. However, we recommend that you complete them before you migrate if you are migrating specifically to an EC2 Bare Metal instance type.

Part 1: Install and upgrade AWS PV drivers

Though AWS PV drivers are not used in the Nitro system, you should still upgrade them if you are on previous versions of either Citrix PV or AWS PV. The latest AWS PV drivers resolve bugs in previous versions of the drivers that may appear while you are on a Nitro system, or if you need to migrate back to a Xen-based instance. As a best practice, we recommend always updating to the latest drivers for Windows instances on AWS.

Use the following procedure to perform an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers on Windows Server 2008 R2, Windows Server 2012,

Windows Server 2012 R2, Windows Server 2016, or Windows Server 2019. For more information, see Upgrade PV drivers on Windows instances.

To upgrade a Domain Controller, see Upgrade a domain controller (AWS PV upgrade).

To perform an upgrade of or to AWS PV drivers

1. Connect to the instance using Remote Desktop and prepare the instance for upgrade. Take all non-system disks offline before you perform the upgrade. If you are performing an in-place update of AWS PV drivers, this step is not required. Set non-essential services to **Manual** start-up in the Services console.
2. Download the latest driver package to the instance.
3. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and upgrades the driver. The instance may not be available for up to 15 minutes.

After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see AWS PV driver package history.

Part 2: Install and upgrade ENA

Upgrade to the latest Elastic Network Adapter driver to ensure that all network features are supported. If you launched your instance and it does not have enhanced networking already enabled, you must download and install the required network adapter driver on your instance. Then, set the enaSupport instance attribute to **activate enhanced networking**. You can only enable this attribute on supported instance types and only if the ENA driver is installed. For more information, see Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances.

1. Download the latest driver to the instance.
2. Extract the zip archive.
3. Install the driver by running the `install.ps1` PowerShell script from the extracted folder.

> **Note**
>
> To avoid installation errors, run the `install.ps1` script as an administrator.

4. Check if your AMI has enaSupport activated. If not, continue by following the documentation at Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances.

Part 3: Upgrade AWS NVMe drivers

AWS NVMe drivers are used to interact with Amazon EBS and SSD instance store volumes that are exposed as NVMe block devices in the Nitro system for better performance.

> **Important**
>
> The following instructions are modified specifically for when you install or upgrade AWS NVMe on a previous generation instance with the intention to migrate the instance to the latest generation instance type.

1. Download the latest driver package to the instance.
2. Extract the zip archive.
3. Install the driver by running `dpinst.exe`.
4. Open a **PowerShell** session and run the following command:

```
PS C:\> start rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
```

> **Note**
>
> To apply the command, you must run the PowerShell session as an administrator. PowerShell (x86) versions will result in an error.
> This command only runs sysprep on the device drivers. It does not run the full sysprep preparation.

5. For Windows Server 2008 R2 and Windows Server 2012, shut down the instance, change the instance type to a latest generation instance and start it, then proceed to Part 4. If you start the instance again on a previous generation instance type before migrating to a latest

generation instance type, it will not boot. For other supported Windows AMIs, you can change the instance type anytime after the device sysprep.

Part 4: Update EC2Config and EC2Launch

For Windows instances, the latest EC2Config and EC2Launch utilities provide additional functionality and information when running on the Nitro system, including on EC2 Bare Metal. By default, the EC2Config service is included in AMIs prior to Windows Server 2016. EC2Launch replaces EC2Config on Windows Server 2016 and later AMIs.

When the EC2Config and EC2Launch services are updated, new Windows AMIs from AWS include the latest version of the service. However, you must update your own Windows AMIs and instances with the latest version of EC2Config and EC2Launch.

To install or update EC2Config

1. Download and unzip the EC2Config Installer.
2. Run `EC2Install.exe`. For a complete list of options, run `EC2Install` with the `/?` option. By default, setup displays prompts. To run the command with no prompts, use the `/quiet` option.

For more information, see Install the latest version of EC2Config.

To install or update EC2Launch

1. If you have already installed and configured EC2Launch on an instance, make a backup of the EC2Launch configuration file. The installation process does not preserve changes in this file. By default, the file is located in the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.
2. Download EC2-Windows-Launch.zip to a directory on the instance.
3. Download install.ps1 to the same directory where you downloaded `EC2-Windows-Launch.zip`.
4. Run `install.ps1`.

> **Note**
>
> To avoid installation errors, run the `install.ps1` script as an administrator.

5. If you made a backup of the EC2Launch configuration file, copy it to the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.

For more information, see Configure a Windows instance using EC2Launch.

Part 5: Install the serial port driver for bare metal instances

The `i3.metal` instance type uses a PCI-based serial device rather than an I/O port-based serial device. The latest Windows AMIs automatically use the PCI-based serial device and have the serial port driver installed. If you are not using an instance launched from an Amazon-provided Windows AMI dated 2018.04.11 or later, you must install the Serial Port Driver to enable the serial device for EC2 features such as Password Generation and Console Output. The latest EC2Config and EC2Launch utilities also support i3.metal and provide additional functionality. Follow the steps in Part 4, if you have not yet done so.

To install the serial port driver

1. Download the serial driver package to the instance.
2. Extract the contents of the folder, open the context (right-click) menu for `aws_ser.INF`, and choose **install**.
3. Choose **Okay**.

Part 6: Update power management settings

The following update to power management settings sets displays to never turn off, which allows for graceful OS shutdowns on the Nitro system. All Windows AMIs provided by Amazon as of 2018.11.28 already have this default configuration.

1. Open a command prompt or PowerShell session.
2. Run the following commands:

```
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f71556f20b4a 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
```

Part 7: Update Intel chipset drivers for new instance types

The `u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal` instance types use hardware that requires chipset drivers that were not previously installed on Windows AMIs. If you are not using an instance launched from an Amazon-provided Windows AMI dated 2018.11.19 or later, you must install the drivers using the Intel Chipset INF Utility.

To install the chipset drivers

1. Download the chipset utility to the instance.
2. Extract the files.
3. Run `SetupChipset.exe`.
4. Accept the Intel software license agreement and install the chipset drivers.
5. Reboot the instance.

(Alternative) Upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager

The `AWSSupport-UpgradeWindowsAWSDrivers` automation document automates the steps described in Part 1, Part 2, and Part 3. This method can also repair an instance where the driver upgrades have failed.

The `AWSSupport-UpgradeWindowsAWSDrivers` automation document upgrades or repairs storage and network AWS drivers on the specified EC2 instance. The document attempts to install the latest versions of AWS drivers online by calling the AWS Systems Manager Agent (SSM Agent). If SSM Agent is not contactable, the document can perform an offline installation of the AWS drivers if explicitly requested.

> **Note**
>
> This procedure will fail on a domain controller. To update drivers on a domain controller, see Upgrade a domain controller (AWS PV upgrade).

To automatically upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager

1. Open the Systems Manager console at https://console.aws.amazon.com/systems-manager.
2. Choose **Automation**, **Execute Automation**.

3. Search for and then select the **AWSSupport-UpgradeWindowsAWSDrivers** automation document, and then choose **Execute automation**.
4. In the **Input Parameters** section, configure the following options:

   Instance ID

   Enter the unique ID of the instance to upgrade.

   AllowOffline

   (Optional) Choose one of the following options:

   - `True` — Choose this option to perform an offline installation. The instance is stopped and restarted during the upgrade process.

     > **Warning**
     > When you stop an instance, the data on any instance store volumes is erased. To preserve data on instance store volumes, ensure that you back up the data to persistent storage.

   - `False` — (Default) To perform an online installation, leave this option selected. The instance is restarted during the upgrade process.

     > **Important**
     > Online and offline upgrades create an AMI before attempting the upgrade operations. The AMI persists after the automation completes. Secure your access to the AMI, or delete it if it is no longer needed.

   SubnetId

   (Optional) Enter one of the following values:

   - `SelectedInstanceSubnet` — (Default) The upgrade process launches the *helper* instance into the same subnet as the instance that is to be upgraded. The subnet must allow communication to the Systems Manager endpoints (`ssm.*`).
   - `CreateNewVPC` — The upgrade process launches the *helper* instance into a new VPC. Use this option if you're not sure whether the target instance's subnet allows communication to the `ssm.*` endpoints. Your user must have permission to create a VPC.

- A specific subnet ID — Specify the ID of a specific subnet into which to launch the *helper* instance. The subnet must be in the same Availability Zone as the instance that is to be upgraded, and it must allow communication with the `ssm.*` endpoints.

5. Choose **Execute**.
6. Allow the upgrade to complete. It could take up to 10 minutes to complete an online upgrade, and up to 25 minutes to complete an offline upgrade.

Migrate to Xen instance types from Nitro instance types

The following procedure assumes that you are currently running on a Nitro-based instance type, such as M5 or C5, and that you are migrating to an instance based on the Xen System, such as M4 or C4. For instance type specifications, see Amazon EC2 Instance Types. Perform the following steps before the migration to avoid errors during the booting process.

1. AWS PV drivers must be installed and upgraded on a Nitro instance before you migrate to a Xen instance. For steps to install and upgrade AWS PV drivers, see Part 1: Install and upgrade AWS PV drivers.
2. Update to the latest EC2Launch v2 version. See Migrate to EC2Launch v2 for steps.
3. Open a PowerShell session and run the following command as an administrator to sysprep the device drivers. Running sysprep ensures that early boot storage drivers required for booting on Xen instances are properly registered with Windows.

   > **Note**
   >
   > Running the command using PowerShell (x86) versions will result in an error. This command adds only the boot-critical device drivers to the critical device database. It does not run the full sysprep preparation.

   ```
   Start-Process rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
   ```

4. Perform the migration to a Xen instance type when the sysprep process completes.

Windows to Linux replatforming assistant for Microsoft SQL Server databases

For information about replatforming Microsoft SQL Server databases from Windows to Linux, see Windows to Linux replatforming assistant for Microsoft SQL Server Databases in the *Microsoft SQL Server on Amazon EC2 User Guide*.

Troubleshoot an upgrade

AWS provides upgrade support for issues or problems with the Upgrade Helper Service, an AWS utility that helps you perform in-place upgrades involving Citrix PV drivers.

After the upgrade, the instance might temporarily experience higher than average CPU utilization while the .NET Runtime Optimization service optimizes the .NET framework. This is expected behavior.

If the instance has not passed both status checks after several hours, check the following.

- If you upgraded to Windows Server 2008 and both status checks fail after several hours, the upgrade may have failed and be presenting a prompt to **Click OK** to confirm rolling back. Because the console is not accessible at this state, there is no way to click the button. To get around this, perform a reboot via the Amazon EC2 console or API. The reboot takes ten minutes or more to initiate. The instance might become available after 25 minutes.
- Remove applications or server roles from the server and try again.

If the instance does not pass both status checks after removing applications or server roles from the server, do the following.

- Stop the instance and attach the root volume to another instance. For more information, see the description of how to stop and attach the root volume to another instance in "Waiting for the metadata service".
- Analyze Windows Setup log files and event logs for failures.

For other issues or problems with an operating system upgrade or migration, we recommend reviewing the articles listed in Before you begin an in-place upgrade.

Identify EC2 Windows instances

You might need to determine whether your application is running on an EC2 instance.

For information about identifying Linux instances, see Identify EC2 Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

Inspect the instance identity document

For a definitive and cryptographically verified method of identifying an EC2 instance, check the instance identity document, including its signature. These documents are available on every EC2 instance at the local, non-routable address `http://169.254.169.254/latest/dynamic/instance-identity/`. For more information, see Instance identity documents.

Inspect the system UUID

You can get the system UUID and look for the presence of the characters "EC2" in the beginning octet of the UUID. This method to determine whether a system is an EC2 instance is quick but potentially inaccurate because there is a small chance that a system that is not an EC2 instance could have a UUID that starts with these characters. Furthermore, EC2 instances using SMBIOS 2.4 might represent the UUID in little-endian format, therefore the "EC2" characters do not appear at the beginning of the UUID.

Example : Get the UUID using WMI or Windows PowerShell

Use the Windows Management Instrumentation command line (WMIC) as follows:

```
wmic path win32_computersystemproduct get uuid
```

Alternatively, if you're using Windows PowerShell, use the **Get-WmiObject** cmdlet as follows:

```
PS C:\> Get-WmiObject -query "select uuid from Win32_ComputerSystemProduct" | Select
 UUID
```

In the following example output, the UUID starts with "EC2", which indicates that the system is probably an EC2 instance.

```
EC2AE145-D1DC-13B2-94ED-012345ABCDEF
```

For instances using SMBIOS 2.4, the UUID might be represented in little-endian format; for example:

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

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 Windows

You can create a sample application to retrieve the virtual machine generation identifier from your instances running Windows. For more information, see Obtaining the virtual machine generation identifier in the Microsoft documentation.

Tutorial: Set up a Windows HPC cluster on Amazon EC2

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

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

- Step 1: Create your security groups
- Step 2: Set up your Active Directory domain controller
- Step 3: Configure your head node
- Step 4: Set up the compute node
- Step 5: Scale your HPC compute nodes (optional)

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

Prerequisites

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

Step 1: Create your security groups

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

To create the security groups

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

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

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

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

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

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

To add rules to the security groups

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

```
PS C:\> $sg_dm = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\> $sg_dm.GroupId = "sg-12345678
PS C:\> $r1 = @{ IpProtocol="UDP"; FromPort="123"; ToPort="123"; UserIdGroupPairs=
$sg_dm }
```

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

2. Use the Grant-EC2SecurityGroupIngress cmdlet to add the rules to the domain controller security group.

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

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

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

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

4. Use the Grant-EC2SecurityGroupIngress cmdlet to add the rules to the domain member security group.

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

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

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

```
PS C:\> $r9 = @{ IpProtocol="TCP"; FromPort="5999"; ToPort="5999";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r10 = @{ IpProtocol="TCP"; FromPort="6729"; ToPort="6730";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r11 = @{ IpProtocol="TCP"; FromPort="7997"; ToPort="7997";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r12 = @{ IpProtocol="TCP"; FromPort="8677"; ToPort="8677";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r13 = @{ IpProtocol="TCP"; FromPort="9087"; ToPort="9087";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r14 = @{ IpProtocol="TCP"; FromPort="9090"; ToPort="9092";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r15 = @{ IpProtocol="TCP"; FromPort="9100"; ToPort="9163";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r16 = @{ IpProtocol="TCP"; FromPort="9200"; ToPort="9263";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r17 = @{ IpProtocol="TCP"; FromPort="9794"; ToPort="9794";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r18 = @{ IpProtocol="TCP"; FromPort="9892"; ToPort="9893";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r19 = @{ IpProtocol="UDP"; FromPort="9893"; ToPort="9893";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r20 = @{ IpProtocol="TCP"; FromPort="6498"; ToPort="6498";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r21 = @{ IpProtocol="TCP"; FromPort="7998"; ToPort="7998";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r22 = @{ IpProtocol="TCP"; FromPort="8050"; ToPort="8050";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r23 = @{ IpProtocol="TCP"; FromPort="5051"; ToPort="5051";
 UserIdGroupPairs=$sg_hpc }
PS C:\> $r24 = @{ IpProtocol="TCP"; FromPort="3389"; ToPort="3389";
 IpRanges="203.0.113.25/32" }
```

6. Use the Grant-EC2SecurityGroupIngress cmdlet to add the rules to the HPC cluster security group.

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

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

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

Step 2: Set up your Active Directory domain controller

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

To launch a domain controller for your HPC cluster

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the console dashboard, choose **Launch Instance**.
3. On the **Choose an AMI** page, select an AMI for Windows Server, and choose **Select**.
4. On the next page of the wizard, select an instance type, then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from **Network** and a subnet from **Subnet**. On the next page of the wizard, you can specify additional storage for your instance.
6. On the **Add Tags** page, enter `Domain Controller` as the value for the `Name` tag for the instance, and then choose **Next: Configure Security Group**.
7. On the **Configure Security Group** page, choose **Select an existing security group**, choose the `SG - Domain Controller` security group, and then choose **Review and Launch**.
8. Choose **Launch**.
9. In the navigation pane, choose **Elastic IPs**.
10. Choose **Allocate new address**. Choose **Allocate**. Choose **Close**.
11. Select the Elastic IP address you created, and choose **Actions**, **Associate address**. For **Instance**, choose the domain controller instance. Choose **Associate**.

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

To configure your instance as a domain controller

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

Step 3: Configure your head node

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

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

To configure an instance as your head node

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

```
IPConfig /all
```

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

To install the HPC Pack

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

To configure your HPC cluster on the head node

1. Start **HPC Cluster Manager**.
2. In the **Deployment To-Do List**, select **Configure your network**.
   a. In the wizard, select the default option (5), and then choose **Next**.

   b. Complete the wizard accepting default values on all screens, and choose how you want to update the server and participate in customer feedback.
   c. Choose **Configure**.
3. Select **Provide Network Credentials**, then provide the `hpc.local\hpcuser` credentials.
4. Select **Configure the naming of new nodes**, and then choose **OK**.
5. Select **Create a node template**.
   a. Select the **Compute node template**, and then choose **Next**.
   b. Select **Without operating system**, and then continue with the defaults.
   c. Choose **Create**.

Step 4: Set up the compute node

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

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

To configure an instance for your compute node

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

   ```
   IPConfig /all
   ```

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

To install the HPC Pack on the compute node

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

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

To add the compute node to your cluster

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

Step 5: Scale your HPC compute nodes (optional)

To scale your compute nodes

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

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

5. Save `sysprep2008.xml`.
6. Choose **Start**, **All Programs**, **EC2ConfigService Settings**.
   a. Choose the **General** tab, and clear the **Set Computer Name** check box.
   b. Choose the **Bundle** tab, and then choose **Run Sysprep and Shutdown Now**.
7. Open the Amazon EC2 console.
8. In the navigation pane, choose **Instances**.
9. Wait for the instance status to show **Stopped**.
10. Select the instance, choose **Actions**, **Image and templates**, **Create image**.

11. Specify an image name and image description, and then choose **Create image** to create an AMI from the instance.
12. Start the original `HPC-Compute` instance that was shut down.
13. Connect to the head node using the `hpc.local\hpcuser` account.
14. From **HPC Cluster Manager**, delete the old node that now appears in an error state.
15. In the Amazon EC2 console, in the navigation pane, choose **AMIs**.
16. Use the AMI you created to add additional nodes to the cluster.

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

EC2 Fleet and Spot Fleet

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.

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. As an added benefit, you can access a deeper Amazon EC2 capacity pool when compared to fleets in a single Availability Zone.

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.

As a general best practice, we recommend launching fleets of Spot and On-Demand Instances with Amazon EC2 Auto Scaling because it provides additional features you can use to manage your fleet. The list of additional features includes automatic health check replacements for both Spot and On-Demand Instances, application-based health checks, and an integration with Elastic Load Balancing to ensure an even distribution of application traffic to your healthy instances. You can also use Auto Scaling groups when you use AWS services such as Amazon ECS, Amazon EKS (self-managed node groups), and Amazon VPC Lattice. For more information, see the *Amazon EC2 Auto Scaling User Guide*.

EC2 Fleet is a good option if you require more 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.

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

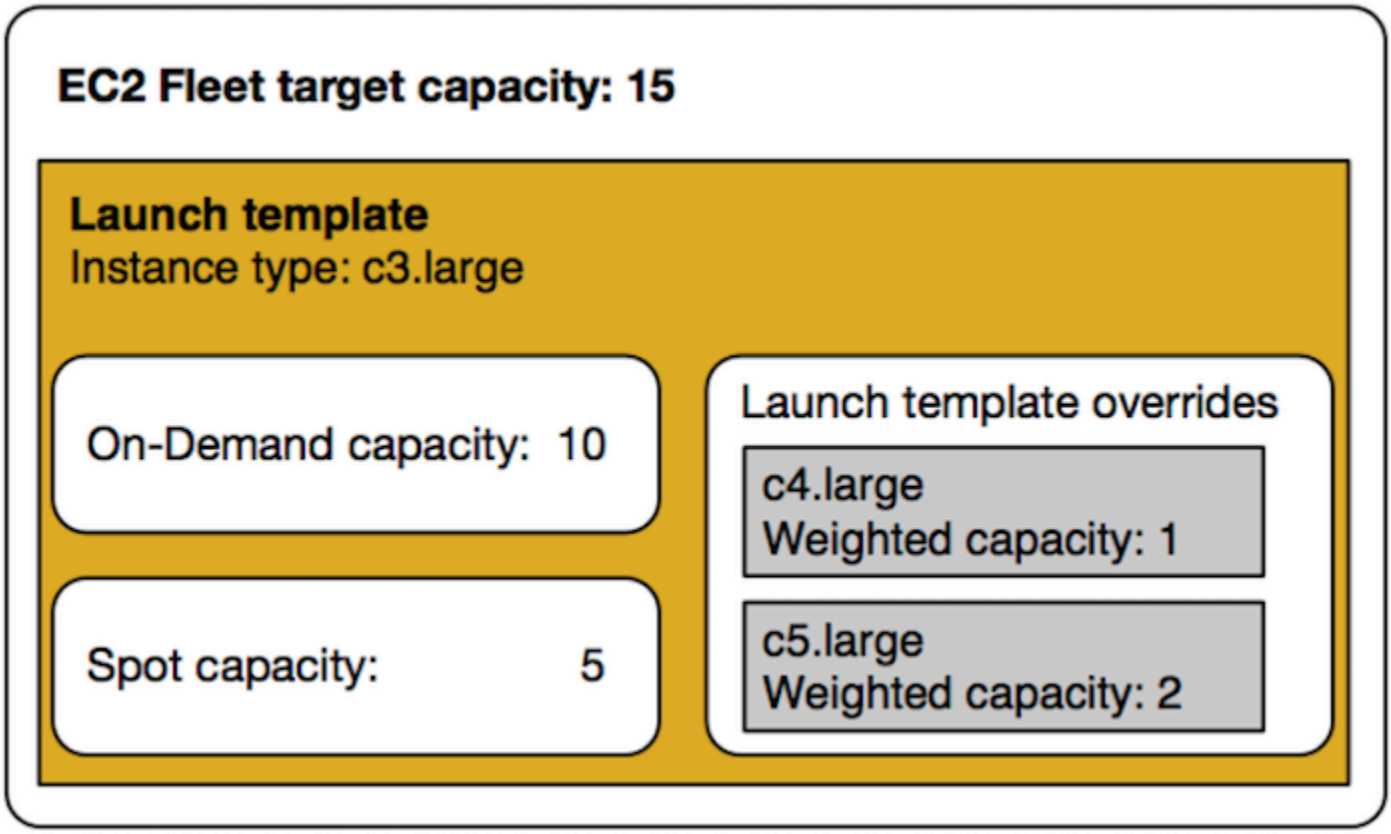
EC2 Fleet

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-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
```

```
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
```

```
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 2: Launch a single Spot Instance with the capacity-optimized allocation strategy

You can optimally launch one Spot Instance at a time by making multiple EC2 Fleet API calls of type `instant`, by setting the TotalTargetCapacity to 1.

The following example specifies the parameters required in an EC2 Fleet of type instant: a launch template, target capacity, default purchasing option, and launch template overrides. The launch template is identified by its launch template name and version number. The 12 launch template overrides have 4 different instance types and 3 different subnets, each in a separate Availability Zone. The target capacity for the fleet is 1 instance, and the default purchasing option is spot, which results in the fleet attempting to launch a Spot Instance from one of the 12 Spot capacity pools based on the capacity-optimized allocation strategy, to launch a Spot Instance from the most-available capacity pool.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
```

```
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
   ],
```

```
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 3: Launch Spot Instances using instance weighting

The following examples use instance weighting, which means that the price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight based on how many units of the workload can run on the instance assuming a unit of the workload requires a 15 GB of memory and 4 vCPUs. For example an m5.xlarge (4 vCPUs and 16 GB of memory) can run one unit and is weighted 1, m5.2xlarge (8 vCPUs and 32 GB of memory) can run 2 units and is weighted 2, and so on. The total target capacity is set to 40 units. The default purchasing option is spot, and the allocation strategy is capacity-optimized, which results in either 40 m5.xlarge (40 divided by 1), 20 m5.2xlarge (40 divided by 2), 10 m5.4xlarge (40 divided by 4), 5 m5.8xlarge (40 divided by 8), or a mix of the instance types with weights adding up to the desired capacity based on the capacity-optimized allocation strategy.

For more information, see EC2 Fleet instance weighting.

```
{
   "SpotOptions":{
      "AllocationStrategy":"capacity-optimized"
   },
   "LaunchTemplateConfigs":[
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":1
            },
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":1
```

```
            },
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":1
            },
            {
               "InstanceType":"m5.2xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":2
            },
            {
               "InstanceType":"m5.2xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":2
            },
            {
               "InstanceType":"m5.2xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":2
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":4
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":4
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":4
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":8
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-e7188bab",
```

```
                "WeightedCapacity":8
            },
            {
                "InstanceType":"m5.8xlarge",
                "SubnetId":"subnet-49e41922",
                "WeightedCapacity":8
            }
        ]
    }
],
"TargetCapacitySpecification":{
    "TotalTargetCapacity":40,
    "DefaultTargetCapacityType":"spot"
},
"Type":"instant"
}
```

Example 4: Launch Spot Instances within single Availability zone

You can configure a fleet to launch all instances in a single Availability Zone by setting the Spot options SingleAvailabilityZone to true.

The 12 launch template overrides have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity is 20 instances, the default purchasing option is spot, and the Spot allocation strategy is capacity-optimized. The EC2 Fleet launches 20 Spot Instances all in a single AZ, from the Spot capacity pool(s) with optimal capacity using the launch specifications.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
```

```
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
```

```
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 5: Launch Spot Instances of single instance type within single Availability zone

You can configure a fleet to launch all instances of the same instance type and in a single Availability Zone by setting the SpotOptions SingleInstanceType to true and SingleAvailabilityZone to true.

The 12 launch template overrides have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity is 20 instances, the default purchasing option is spot, the Spot allocation strategy is capacity-optimized. The EC2 Fleet launches 20 Spot Instances of the same instance type all in a single AZ from the Spot Instance pool with optimal capacity using the launch specifications.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
```

```
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
   ],
```

```
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 6: Launch Spot Instances only if minimum target capacity can be launched

You can configure a fleet to launch instances only if the minimum target capacity can be launched by setting the Spot options MinTargetCapacity to the minimum target capacity you want to launch together.

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-9xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":36
            },
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":36
            },
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":36
            }
         ]
      },
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-4xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.4xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":16
            },
            {
               "InstanceType":"c5n.4xlarge",
```

```
                "SubnetId":"subnet-e7188bab",
                "WeightedCapacity":16
            },
            {
                "InstanceType":"c5n.4xlarge",
                "SubnetId":"subnet-49e41922",
                "WeightedCapacity":16
            }
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 144,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 9: Launch Spot Instance with a base of On-Demand Instances

The following example specifies the total target capacity of 20 instances for the fleet, and a target capacity of 5 On-Demand Instances. The default purchasing option is spot. The fleet launches 5 On-Demand Instance as specified, but needs to launch 15 more instances to fulfill the total target capacity. The purchasing option for the difference is calculated as TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType, which results in the fleet launching 15 Spot Instances form one of the 12 Spot capacity pools based on the capacity-optimized allocation strategy.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
        "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
        },
        "Overrides":[
            {
                "InstanceType":"c5.large",
                "SubnetId":"subnet-fae8c380"
```

```
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
```

```
            }
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

Example 10: Launch Spot Instances using capacity-optimized allocation strategy with a base of On-Demand Instances using Capacity Reservations and the prioritized allocation strategy

You can configure a fleet to use On-Demand Capacity Reservations first when launching a base of On-Demand Instances with the default target capacity type as spot by setting the usage strategy for Capacity Reservations to use-capacity-reservations-first. And if multiple instance pools have unused Capacity Reservations, the chosen On-Demand allocation strategy is applied. In this example, the On-Demand allocation strategy is prioritized.

In this example, there are 6 available unused Capacity Reservations. This is less than the fleet's target On-Demand capacity of 10 On-Demand Instances.

The account has the following 6 unused Capacity Reservations in 2 pools. The number of Capacity Reservations in each pool is indicated by AvailableInstanceCount.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
```

```
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is prioritized, and the usage strategy for Capacity Reservations is use-capacity-reservations-first. The Spot allocation strategy is capacity-optimized. The total target capacity is 20, the On-Demand target capacity is 10, and the default target capacity type is spot.

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "OnDemandOptions":{
       "CapacityReservationOptions": {
         "UsageStrategy": "use-capacity-reservations-first"
       },
       "AllocationStrategy":"prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 3.0
            },
            {
```

```
                "InstanceType":"c5d.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 4.0
            },
            {
                "InstanceType":"c5d.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 5.0
            },
            {
                "InstanceType":"c5d.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 6.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 7.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 8.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 9.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 10.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 11.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 12.0
            }
```

```
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 10,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

After you create the instant fleet using the preceding configuration, the following 20 instances are launched to meet the target capacity:

- 7 c5.large On-Demand Instances in us-east-1a – c5.large in us-east-1a is prioritized first, and there are 3 available unused c5.large Capacity Reservations. The Capacity Reservations are used first to launch 3 On-Demand Instances plus 4 additional On-Demand Instances are launched according to the On-Demand allocation strategy, which is prioritized in this example.
- 3 m5.large On-Demand Instances in us-east-1a – m5.large in us-east-1a is prioritized second, and there are 3 available unused c3.large Capacity Reservations.
- 10 Spot Instances from one of the 12 Spot capacity pools that has the optimal capacity according to the capacity-optimized allocation strategy.

After the fleet is launched, you can run describe-capacity-reservations 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 you 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.

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
```

```
                            }
                        }
                    }
                }
            ]
        }
    ],
     "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
  }
}
```

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 (10 divided by 8, result rounded up)	$0.10	$0.0125 (.10 divided by 8)

Use EC2 Fleet instance weighting as follows to provision the target capacity that you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your EC2 Fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the price per unit.
3. For each launch specification, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance weighting example

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest price per unit (price for `r3.2xlarge` per instance hour divided by 6), the EC2 Fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest price per unit (price for `c3.xlarge` per instance hour divided by 5), the EC2 Fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 30 Spot Instances
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The EC2 Fleet would launch four instances (30 divided by 8, result rounded up). With the `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=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.

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 `instant` fleets in a single request. If you exceed this number, no `instant` fleets are deleted and an error is returned. There is no restriction on the number of fleets of type `maintain` or `request` that can be deleted in a single request.
- Up to 1000 instances can be terminated in a single request to delete `instant` fleets.

To delete an EC2 Fleet and terminate its instances (AWS CLI)

Use the delete-fleets command and the `--terminate-instances` parameter to delete the specified EC2 Fleet and terminate 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
KnSMMiq5s6cGqjjPtEDpsnGHzzyHasFHOaRYJpaDVravoW25azn6KNkUQQlFwhJyujt2dtNCdduJfrqcFYAjlEiRMkfDHt7
BHturzDK6A560Y2nDSUiMmAB1y9UNtqaZJ9SNe5sNxKMqZaqKtjRbk02RZu5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtJ76
VPiU5v2s-
UgZ7h0p2yth6ysUdhlONg6dBYu8_y_HtEI54invCj4CoK0qawqzMNe6rcmCQHvtCxtXsbkgyaEbcwmrm2m01-
EMhekLFZeJLr
DtYOpYcEl4_nWFX1wtQDCnNNCmxnJZAoJvb3VMDYpDTsxjQv1PxODZuqWHs23YXWVywzgnLtHeRf2o4lUhGBw17mXsS07k7
PT9vrHtQiILor5VVTsjSPWg7edj__1rsnXhwPSu8gI48ZLRGrPQqFq0RmKO_QIE8N8s6NWzCK4yoX-9gDcheurOGpkprPIC
</Message></Error></Errors><RequestID>89b1215c-7814-40ae-a8db-41761f43f2b0</
RequestID></Response>
```


To resolve the error, you must add the `ec2:TerminateInstances` action to the IAM policy, as shown in the following example.

```
{
       "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.

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	`aws:SourceArn`
RequestSpotFleet	AWS STS (`AssumeRole`)	Limit the `AssumeRole` capability on `aws-ec2-spot-fleet-tagging-role` to spot-fleet-requests in the specified account.	`arn:aws:ec2:*:`*`123456789012`*`:spot-fleet-request/sfr-*`
RequestSpotFleet	AWS STS (`AssumeRole`)	Limit the `AssumeRole` capability on `aws-ec2-spot-fleet-tagging-role` to spot-fleet-requests in the specified account and specified Region. Note that this role will not be usable in other Regions.	`arn:aws:ec2:`*`us-east-1`*`:`*`123456789012`*`:spot-fleet-request/sfr-*`
RequestSpotFleet	AWS STS (`AssumeRole`)	Limit the `AssumeRole` capability on `aws-ec2-spot-fleet-tagging-role` to only actions affecting the fleet sfr-11111111-1111-1111-1111-111111111111. Note that this role may not be usable for other Spot Fleets.	`arn:aws:ec2:`*`us-east-1`*`:`*`123456789012`*`:spot-fleet-request/sfr-`*`11111111-1111-1111-1111-111111111111`*

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 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.

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
```

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
`BidsSubmittedForCapacity`	The capacity for which Amazon EC2 has submitted Spot Fleet requests.

Metric	Description
	Units: Count
`EligibleInstancePoolCount`	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. Units: Count
`FulfilledCapacity`	The capacity that Amazon EC2 has fulfilled. Units: Count
`MaxPercentCapacityAllocation`	The maximum value of `PercentCapacityAllocation` across all Spot Fleet pools specified in the Spot Fleet request. Units: Percent
`PendingCapacity`	The difference between `TargetCapacity` and `FulfilledCapacity`. Units: Count
`PercentCapacityAllocation`	The capacity allocated for the Spot capacity pool for the specified dimensions. To get the maximum value recorded across all Spot capacity pools, use `MaxPercentCapacityAllocation`. Units: Percent

Metric	Description
`TargetCapacity`	The target capacity of the Spot Fleet request. Units: Count
`TerminatingCapacity`	The capacity that is being terminated because the provision ed capacity is greater than the target capacity. Units: Count

If the unit of measure for a metric is `Count`, the most useful statistic is `Average`.

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.

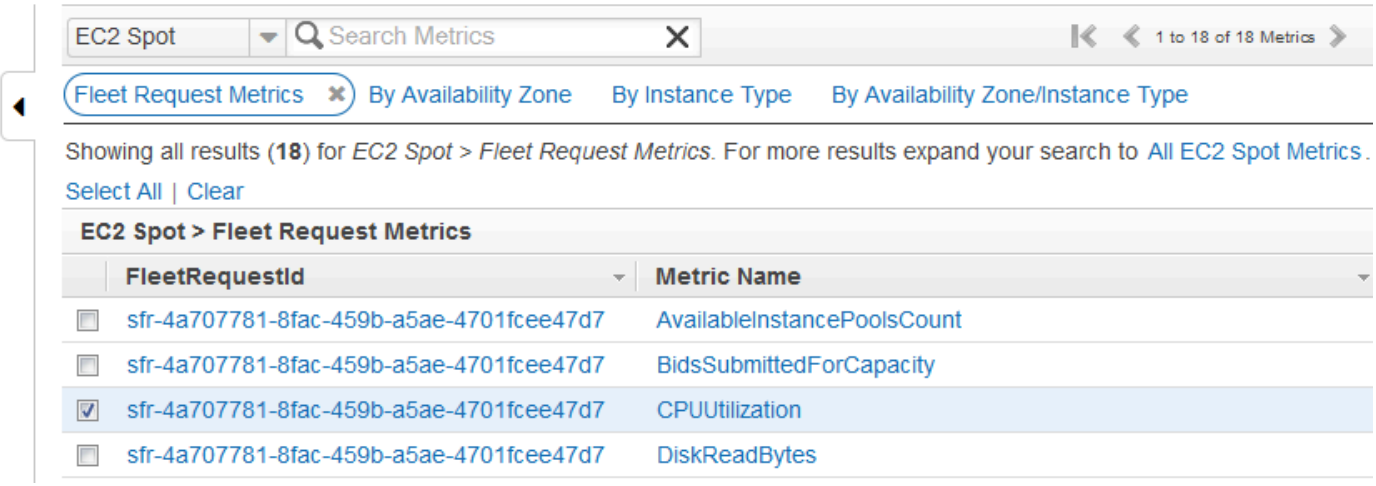
Metrics are grouped first by namespace, and then by the various combinations of dimensions within each namespace. For example, you can view all Spot Fleet metrics or Spot Fleet metrics groups by Spot Fleet request ID, instance type, or Availability Zone.

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.



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-6b0809bfff0a"
    ],
    "detail": {
        "instance-id": "i-0c594155dd5ff1829",
        "description": "{\"instanceType\":\"c5.large\",\"image\":\"ami-6057e21a\",
\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":\"us-east-1d\"}",
        "sub-type": "launched"
    }
}
```

The possible values for `sub-type` are:

`launched`

A new instance was launched.

`terminated`

The instance was terminated.

`termination_notified`

An instance termination notification was sent when a Spot Instance was terminated by Amazon EC2 during scale-down, when the target capacity of the fleet was modified down, for example, from a target capacity of 4 to a target capacity of 3.

EC2 Fleet Information

EC2 Fleet sends an `EC2 Fleet Information` event to Amazon EventBridge when there is an error during fulfillment. The information event does not block the fleet from attempting to fulfil its target capacity.

The following is example data for this event.

```
{
    "version": "0",
    "id": "76529817-d605-4571-7224-d36cc1b2c0c4",
    "detail-type": "EC2 Fleet Information",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T08:17:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-8becf5fe-
bb9e-415d-8f54-3fa5a8628b91"
    ],
    "detail": {
        "description": "c4.xlarge, ami-0947d2ba12ee1ff75, Linux/UNIX, us-east-1a,
 Spot price in either SpotFleetRequestConfigData or SpotFleetLaunchSpecification or
 LaunchTemplate or LaunchTemplateOverrides is less than Spot market price $0.0619",
        "sub-type": "launchSpecUnusable"
    }
}
```

The possible values for `sub-type` are:

`fleetProgressHalted`

The price in every launch specification is not valid because it is below the Spot price (all the launch specifications have produced `launchSpecUnusable` events). A launch specification might become valid if the Spot price changes.

`launchSpecTemporarilyBlacklisted`

The configuration is not valid and several attempts to launch instances have failed. For more information, see the description of the event.

`launchSpecUnusable`

The price in a launch specification is not valid because it is below the Spot price.

`registerWithLoadBalancersFailed`

An attempt to register instances with load balancers failed. For more information, see the description of the event.

EC2 Fleet Error

EC2 Fleet sends an `EC2 Fleet Error` event to Amazon EventBridge when there is an error during fulfillment. The error event blocks the fleet from attempting to fulfil its target capacity.

The following is example data for this event.

```
{
    "version": "0",
    "id": "69849a22-6d0f-d4ce-602b-b47c1c98240e",
    "detail-type": "EC2 Fleet Error",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-10-07T01:44:24Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-9bb19bc6-60d3-4fd2-ae47-
d33e68eafa08"
    ],
    "detail": {
        "description": "m3.large, ami-00068cd7555f543d5, Linux/UNIX: IPv6 is not
 supported for the instance type 'm3.large'. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
```

```
    }
}
```

The possible values for `sub-type` are:

`iamFleetRoleInvalid`

The EC2 Fleet does not have the required permissions to either launch or terminate an instance.

`allLaunchSpecsTemporarilyBlacklisted`

None of the configurations are valid, and several attempts to launch instances have failed. For more information, see the description of the event.

`spotInstanceCountLimitExceeded`

You've reached the limit on the number of Spot Instances that you can launch.

`spotFleetRequestConfigurationInvalid`

The configuration is not valid. For more information, see the description of the event.

Spot Fleet event types

There are five Spot Fleet event types. For each event type, there are several sub-types.

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.

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 then 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 in the `active`, `deleted_running` , and `cancelled_running` states	1,000 [1] [2] [3] [4]
The number of Spot capacity pools (unique combination of instance type and subnet)	300 [1] [4]
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.

[4] This quota only applies to fleets of type `request` or `maintain`. This quota does not apply to `instant` fleets.

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**.

Amazon Elastic Graphics

> **Important**
>
> Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.

Amazon Elastic Graphics provides flexible, low-cost, and high performance graphics acceleration for your Windows instances. Elastic Graphics accelerators come in multiple sizes and are a low-cost alternative to using GPU graphics instance types (such as G2 and G3). You have the flexibility to choose an instance type that meets the compute, memory, and storage needs of your application. Then, choose the accelerator for your instance that meets the graphics requirements of your workload.

Elastic Graphics is suited for applications that require a small or intermittent amount of additional graphics acceleration, and that use OpenGL graphics support. If you need access to full, directly attached GPUs and use of DirectX, CUDA, or Open Computing Language (OpenCL) parallel computing frameworks, use an accelerated computing instance type instance instead. For more information, see Windows accelerated computing instances.

Contents

- Elastic Graphics basics
- Pricing for Elastic Graphics
- Elastic Graphics limitations
- Work with Elastic Graphics
- Elastic Graphics maintenance
- Use CloudWatch metrics to monitor Elastic Graphics
- Troubleshoot

Elastic Graphics basics

To use Elastic Graphics, launch a Windows instance and specify an accelerator type for the instance during launch. AWS finds available Elastic Graphics capacity and establishes a network connection between your instance and the Elastic Graphics accelerator.

> **Note**
>
> Bare metal instances are not supported.

Elastic Graphics accelerators are available in the following AWS Regions: `us-east-1`, `us-east-2`, `us-west-2`, `ap-northeast-1`, `ap-southeast-1`, `ap-southeast-2`, `eu-central-1`, and `eu-west-1`.

The following instance types support Elastic Graphics accelerators:

- **General purpose:** M3, M4, M5, M5d, M5dn, M5n, T2, T3

  > **Note**
  >
  > Only `t2.medium` and larger and `t3.medium` and larger are supported.

- **Compute optimized:** C3, C4, C5, C5a, C5ad, C5d, C5n
- **Memory optimized:** R3, R4, R5, R5d, R5dn, R5n, X1, X1e, z1d
- **Storage optimized:** D2, D3, D3en, H1, I3, I3en
- **Accelerated computing:** P2, P3, P3dn

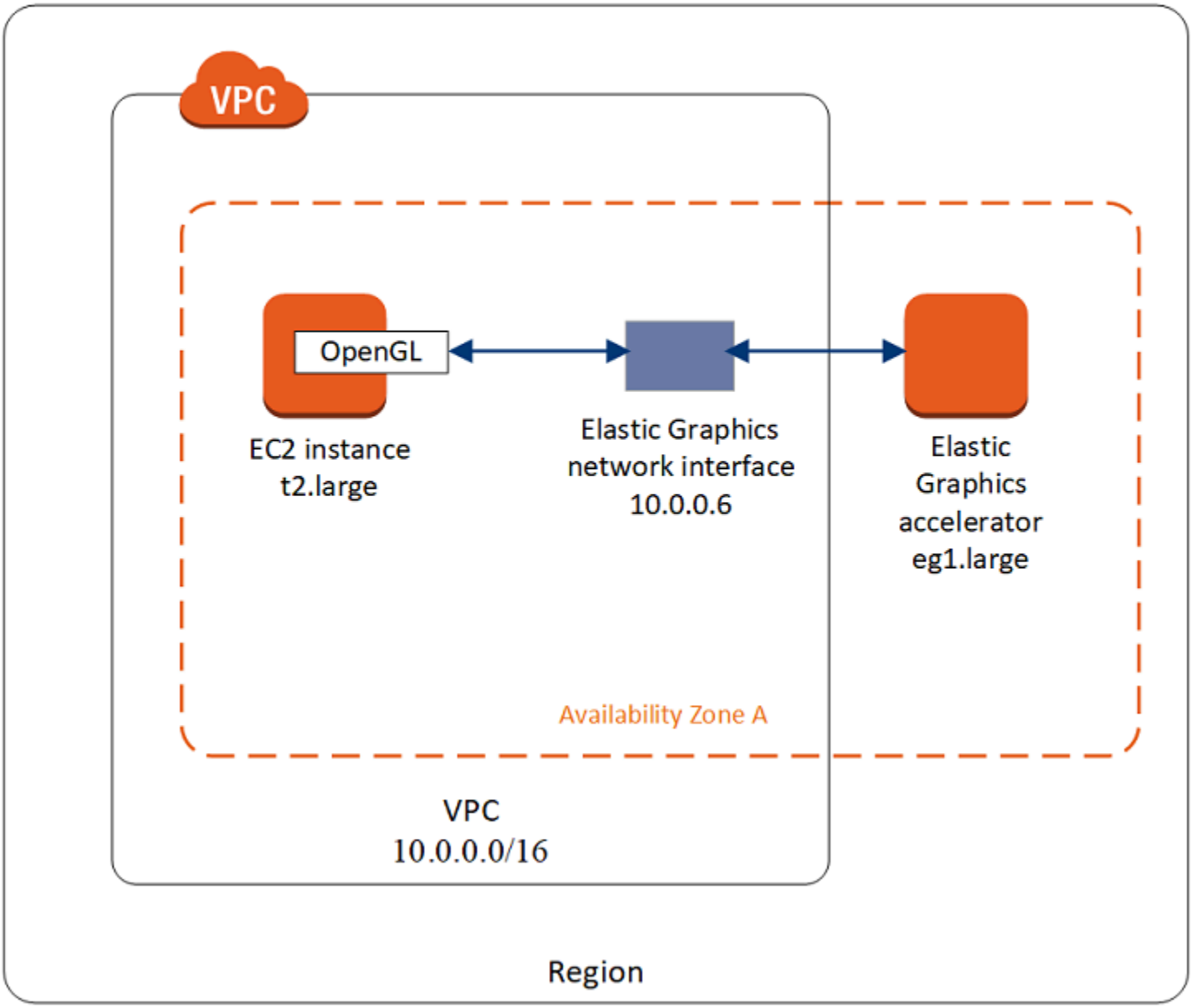
The following Elastic Graphics accelerators are available. You can attach any Elastic Graphics accelerator to any supported instance type.

Elastic Graphics accelerator	Graphics memory (GB)
`eg1.medium`	1
`eg1.large`	2

Elastic Graphics accelerator	Graphics memory (GB)
`eg1.xlarge`	4
`eg1.2xlarge`	8

An Elastic Graphics accelerator does not form part of the hardware of your instance. Instead, it is network-attached through a network interface, known as the *Elastic Graphics network interface*. When you launch or restart an instance with graphics acceleration, the Elastic Graphics network interface is created in your VPC for you.

The Elastic Graphics network interface is created in the same subnet and VPC as your instance and is assigned a private IPv4 address from that subnet. The accelerator attached to your Amazon EC2 instance is allocated from a pool of available accelerators in the same Availability Zone as your instance.



Elastic Graphics accelerators support the API standards for OpenGL 4.3 API and earlier, which can be used for batch applications or 3D-graphics acceleration. An Amazon-optimized OpenGL library on your instance detects the attached accelerator. It directs OpenGL API calls from your instance to the accelerator, which then processes the requests and returns the results. Traffic between the instance and the accelerator uses the same bandwidth as the instance's network traffic so we recommend that you have adequate network bandwidth available. Consult your software vendor for any OpenGL compliance and version questions.

By default, the default security group for your VPC is associated with the Elastic Graphics network interface. The Elastic Graphics network traffic uses the TCP protocol and port 2007. Ensure that the security group for your instance allows for this. For more information, see Configure your security groups.

Pricing for Elastic Graphics

You are charged for each second that an Elastic Graphics accelerator is attached to an instance in the `running` state when the accelerator is in the `Ok` state. You are not charged for an accelerator attached to an instance that is in the `pending`, `stopping`, `stopped`, `shutting-down`, or `terminated` state. You are also not charged when an accelerator is in the `Unknown` or `Impaired` state.

Pricing for accelerators is available at On-Demand rates only. You can attach an accelerator to a Reserved Instance or Spot Instance, however, the On-Demand price for the accelerator applies.

For more information, see Amazon Elastic Graphics Pricing.

Elastic Graphics limitations

Before you start using Elastic Graphics accelerators, be aware of the following limitations:

- You can attach accelerators only to Windows instances with Microsoft Windows Server 2012 R2 or later. Linux instances are currently not supported.
- You can attach one accelerator to an instance at a time.
- You can attach an accelerator only during instance launch. You cannot attach an accelerator to an existing instance.
- You can't hibernate an instance with an attached accelerator.
- You can't share an accelerator between instances.
- You can't detach an accelerator from an instance or transfer it to another instance. If you no longer require an accelerator, you must terminate your instance. To change the accelerator type, create an AMI from your instance, terminate the instance, and launch a new instance with a different accelerator specification.
- The only supported versions of the OpenGL API are 4.3 and earlier. DirectX, CUDA, and OpenCL are not supported.
- The Elastic Graphics accelerator is not visible or accessible through the device manager of your instance.

- You can't reserve or schedule accelerator capacity.

Work with Elastic Graphics

> **Important**
>
> Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.

You can launch an instance and associate it with an Elastic Graphics accelerator during launch. You must then manually install the necessary libraries on your instance that enable communication with the accelerator. For limitations, see Elastic Graphics limitations.

Tasks

- Configure your security groups
- Launch an instance with an Elastic Graphics accelerator
- Install the required software for Elastic Graphics
- Verify Elastic Graphics functionality on your instance
- View Elastic Graphics information
- Submit feedback

Configure your security groups

Elastic Graphics requires a self-referencing security group that allows inbound and outbound traffic to and from the security group itself. The security group must include the following inbound and outbound rules.

Inbound

Type	Protocol	Port	Source
Elastic Graphics	TCP	2007	The security group ID (its own resource ID)

Outbound

Type	Protocol	Port range	Destination
Elastic Graphics	TCP	2007	The security group ID (its own resource ID)

If you use the Amazon EC2 console to launch your instance with an Elastic Graphics accelerator, you can either allow the launch instance wizard to automatically create the required security group rules, or you can select a security that you created previously.

If you are launching your instance using the AWS CLI or an SDK, you must specify a security group that you created previously.

To create a security group for Elastic Graphics

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups** and then choose **Create security group**.
3. In the **Create security group** window, do the following:
   a. For **Security group name**, enter a descriptive name for the security group, such as `Elastic Graphics security group`.
   b. (Optional) For **Description**, enter a brief description of the security group.
   c. For **VPC**, select the VPC into which you intend to use Elastic Graphics.
   d. Choose **Create security group**.
4. In the navigation pane, choose **Security Groups**, select the security group that you just created, and on the **Details** tab, copy the **Security group ID**.
5. On the **Inbound rules** tab, choose **Edit inbound rules** and then do the following:
   a. Choose **Add rule**.
   b. For **Type**, choose **Elastic Graphics**.
   c. For **Source type**, choose **Custom**.
   d. For **Source**, paste the security group ID that you copied previously.
   e. Choose **Save rules**.
6. On the **Outbound rules** tab, choose **Edit outbound rules** and then do the following:

a. Choose **Add rule**.

b. For **Type**, choose **Elastic Graphics**.

c. For **Destination type**, choose **Custom**.

d. For **Destination**, paste the security group ID that you copied previously.

e. Choose **Save rules**.

For more information, see Amazon EC2 security groups for Windows instances.

Launch an instance with an Elastic Graphics accelerator

You can associate an Elastic Graphics accelerator to an instance during launch. If the launch fails, the following are possible reasons:

- Insufficient Elastic Graphics accelerator capacity
- Exceeded limit on Elastic Graphics accelerators in the Region
- Not enough private IPv4 addresses in your VPC to create a network interface for the accelerator

For more information, see Elastic Graphics limitations.

To associate an Elastic Graphics accelerator during instance launch (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the dashboard, choose **Launch Instance**.
3. Select a Windows AMI and a supported instance type. For more information, see Elastic Graphics basics.
4. On the **Configure Instance Details** page, select a VPC and subnet in which to launch your instance.
5. Choose **Add Graphics Acceleration**, and select an Elastic Graphics accelerator type.
6. (Optional) On the **Add Storage** and **Add Tags** pages, add volumes and tags as needed.
7. On the **Configure Security Group** page, you can let the console create a security group for you with the required inbound and outbound rules, or you can use the security group that you created manually in Configure your security groups. Add additional security groups as needed.
8. Choose **Review and Launch** to review your instance options and then choose **Launch**.

To associate an Elastic Graphics accelerator during instance launch (AWS CLI)

You can use the run-instances AWS CLI command with the following parameter:

```
--elastic-gpu-specification Type=eg1.medium
```

For the `--security-group-ids` parameter, you must include a security group that has the required inbound and outbound rules. For more information, see Configure your security groups.

To associate an Elastic Graphics accelerator during instance launch (Tools for Windows PowerShell)

Use the New-EC2Instance Tools for Windows PowerShell command.

Install the required software for Elastic Graphics

If you launched your instance using a current AWS Windows AMI, the required software is installed automatically during the first boot. If you launched your instance using Windows AMIs that do not automatically install the required software, you must install the required software on the instance manually.

To install the required software for Elastic Graphics (if necessary)

1. Connect to the instance.
2. Download the Elastic Graphics installer and open it. The installation manager connects to the Elastic Graphics endpoint and downloads the latest version of the required software.

   > **Note**
   >
   > If the download link does not work, try a different browser, or copy the link address and paste it into a new browser tab.

3. Reboot the instance to verify.

Verify Elastic Graphics functionality on your instance

The Elastic Graphics packages on your instance include tools that you can use to view the status of the accelerator, and to verify that OpenGL commands from your instance to the accelerator are functional.

If your instance was launched with an AMI that does not have the Elastic Graphics packages pre-installed, you can download and install them yourself. For more information, see Install the required software for Elastic Graphics.

You can use one of the following methods to verify Elastic Graphics functionality on your instance.

> **Note**
>
> If the Elastic Graphics status monitor or command line tool returns an unexpected result, see Resolve unhealthy status issues.

Elastic Graphics status monitor

You can use the status monitor tool to view information about the status of an attached Elastic Graphics accelerator. By default, this tool is available in the notification area of the taskbar in your Windows instance and shows the status of the graphics accelerator. The following are the possible values.

Healthy

The Elastic Graphics accelerator is enabled and healthy.

Updating

The status of the Elastic Graphics accelerator is currently updating. It might take a few minutes to display the status.

Out of service

The Elastic Graphics accelerator is out of service. To get more information about the error, choose **Read More**.

Elastic Graphics command line tool

You can use the Elastic Graphics command line tool, `egcli.exe`, to check the status of the accelerator. If there is a problem with the accelerator, the tool returns an error message.

To launch the tool, open a command prompt from within your instance and run the following command:

```
C:\Program Files\Amazon\EC2ElasticGPUs\manager\egcli.exe
```

The tool also supports the following parameters:

`--json`, `-j`

Indicates whether to show the JSON message. The possible values are `true` and `false`. The default is `true`.

`--imds`, `-i`

Indicates whether to check the instance metadata for the availability of the accelerator. The possible values are `true` and `false`. The default is `true`.

The following is example output. A status of `OK` indicates that the accelerator is enabled and healthy.

```
EG Infrastructure is available.
Instance ID egpu-f6d94dfa66df4883b284e96db7397ee6
Instance Type eg1.large
EG Version 1.0.0.885 (Manager) / 1.0.0.95 (OpenGL Library) / 1.0.0.69 (OpenGL
 Redirector)
EG Status: Healthy
JSON Message:
{
  "version": "2016-11-30",
  "status": "OK"
}
```

The following are the possible values for `status`:

`OK`

The Elastic Graphics accelerator is enabled and healthy.

`UPDATING`

The Elastic Graphics driver is being updated.

`NEEDS_REBOOT`

The Elastic Graphics driver has been updated and a reboot of the Amazon EC2 instance is required.

`LOADING_DRIVER`

The Elastic Graphics driver is being loaded.

`CONNECTING_EGPU`

The Elastic Graphics driver is verifying the connectivity with the Elastic Graphics accelerator.

`ERROR_UPDATE_RETRY`

An error occurred while updating the Elastic Graphics driver, an update will be retried soon.

`ERROR_UPDATE`

An unrecoverable error occurred while updating the Elastic Graphics driver.

`ERROR_LOAD_DRIVER`

An error occurred loading the Elastic Graphics driver.

`ERROR_EGPU_CONNECTIVITY`

The Elastic Graphics accelerator is unreacheable.

View Elastic Graphics information

You can view information about the Elastic Graphics accelerator attached to your instance.

To view information about an Elastic Graphics accelerator (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select your instance.
3. On the **Details** tab, find **Elastic Graphics ID**. Choose the ID to view the following information about the Elastic Graphics accelerator:
   - **Attachment State**
   - **Type**
   - **Health status**

To view information about an Elastic Graphics accelerator (AWS CLI)

You can use the describe-elastic-gpus AWS CLI command:

```
aws ec2 describe-elastic-gpus
```

You can use the describe-network-interfaces AWS CLI command and filter by owner ID to view information about the Elastic Graphics network interface.

```
aws ec2 describe-network-interfaces --filters "Name=attachment.instance-owner-
id,Values=amazon-elasticgpus"
```

To view information about an Elastic Graphics accelerator (Tools for Windows PowerShell)

Use the following commands:

- Get-EC2ElasticGpu
- Get-EC2NetworkInterface

To view information about an Elastic Graphics accelerator using instance metadata

1. Connect to your Windows instance that is using an Elastic Graphics accelerator.
2. Do one of the following:
   - From PowerShell, use the following cmdlet:

     ```
     PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/elastic-
     gpus/associations/egpu-f6d94dfa66df4883b284e96db7397ee6
     ```

   - From your web browser, paste the following URL into the address field:

     ```
     http://169.254.169.254/latest/meta-data/elastic-gpus/associations/egpu-
     f6d94dfa66df4883b284e96db7397ee6
     ```

Submit feedback

You can submit feedback about your experience with Elastic Graphics so that the team can make further improvements.

To submit feedback using the Elastic Graphics Status Monitor

1. In the notification area of the taskbar in your Windows instance, open the Elastic Graphics Status Monitor.
2. In the lower left corner, choose **Feedback**.
3. Enter your feedback and choose **Submit**.

Elastic Graphics maintenance

> **Important**
>
> Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.

AWS might determine that an Elastic Graphics accelerator is in an *unhealthy* state if:

- A security or infrastructure update is needed
- A software update is needed
- There is an issue with the underlying host

When AWS determines that an Elastic Graphics accelerator is in an unhealthy state, it schedules the accelerator for retirement. AWS notifies you of the accelerator's pending retirement, and provides you with the remedial steps that you need to take.

Topics

- How will I be notified?
- What do I need to do?
- What happens when an accelerator reaches its retirement date?

How will I be notified?

When AWS schedules an Elastic Graphics accelerator for retirement, it sends an accelerator retirement notice to your AWS Health Dashboard. AWS also sends an email to the email address that is associated with your AWS account. This is the same email address that you use to log in to the AWS Management Console.

> **Note**
>
> If you use an email account that you don't check regularly, use the AWS Health Dashboard to determine if any of your Elastic Graphics accelerators are scheduled for retirement. You

can also change the contact information for your AWS account on the Account Settings page.

The retirement notice provides the following:

- The ID of instance to which the accelerator is attached
- Information about the issue impacting the accelerator
- The retirement date for the accelerator
- The remedial steps that you should take

What do I need to do?

When you are notified that your Elastic Graphics accelerator is scheduled for retirement, you must stop and start the instance to which the accelerator is attached for the old, *unhealthy* accelerator to be replaced with a new, *healthy* accelerator.

We recommend that you close graphic applications running on the instance before you stop and restart the instance.

Important

If you do not stop and start your instance before the scheduled retirement date, the accelerator associated with your instance is automatically stopped, which might cause your applications to stop working.
You must stop and start the instance. Rebooting the instance will not replace the unhealthy accelerator with a healthy one.

What happens when an accelerator reaches its retirement date?

When an unhealthy Elastic Graphics accelerator reaches its scheduled retirement date, AWS permanently terminates it. To receive a replacement for your unhealthy accelerator, either before or after the retirement date, you must stop and start the instance to which the accelerator is attached.

If you do not stop and start your instance before the scheduled retirement date, the accelerator associated with your instance is automatically stopped, which might cause your applications to stop working.

Use CloudWatch metrics to monitor Elastic Graphics

> **Important**
>
> Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.

You can monitor your Elastic Graphics accelerator using Amazon CloudWatch, which collects metrics about your accelerator performance. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your service is performing.

By default, Elastic Graphics accelerators send metric data to CloudWatch in 5-minute periods.

For more information about Amazon CloudWatch, see the Amazon CloudWatch User Guide.

Elastic Graphics metrics

The `AWS/ElasticGPUs` namespace includes the following metrics for Elastic Graphics.

Metric	Description
GPUConnectivityCheckFailed	Reports whether connectivity to the Elastic Graphics accelerator is active or has failed. A value of zero (0) indicates that the connection is active. A value of one (1) indicates a connectivity failure. Units: Count
GPUHealthCheckFailed	Reports whether the Elastic Graphics accelerator has passed a status health check in the last minute. A value of zero (0) indicates that the status check

Metric	Description
	passed. A value of one (1) indicates a status check failure. Units: Count
GPUMemoryUtilization	The GPU memory used. Units: MiB

Elastic Graphics dimensions

You can filter the metrics data for your Elastic Graphics accelerators using the following dimensions.

Dimension	Description
`EGPUId`	Filters the data by the Elastic Graphics accelerator.
`InstanceId`	Filters the data by the instance to which the Elastic Graphics accelerator is attached.

View CloudWatch metrics for Elastic Graphics

Metrics are grouped first by the service namespace, and then by the supported dimensions. You can use the following procedures to view the metrics for your Elastic Graphics accelerators.

To view Elastic Graphics metrics using the CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the Region. From the navigation bar, select the Region where your Elastic Graphics accelerator resides. For more information, see Regions and Endpoints.
3. In the navigation pane, choose **Metrics**.
4. For **All metrics**, select **Elastic Graphics**, **Elastic Graphics Metrics**.

To view Elastic Graphics metrics (AWS CLI)

Use the following list-metrics command:

```
aws cloudwatch list-metrics --namespace "AWS/ElasticGPUs"
```

Create CloudWatch alarms to monitor Elastic Graphics

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify, and sends a notification to an Amazon SNS topic based on the value of the metric relative to a given threshold over a number of time periods.

For example, you can create an alarm that monitors the health of an Elastic Graphics accelerator and sends a notification when the graphics accelerator fails a status health check for three consecutive 5-minute periods.

To create an alarm for an Elastic Graphics accelerator health status

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Alarms**, **Create Alarm**.
3. Choose **Select metric**, **Elastic Graphics**, **Elastic Graphics Metrics**.
4. Select the **GPUHealthCheckFailed** metric and choose **Select metric**.
5. Configure the alarm as follows:
   a. For **Alarm details**, type a name and description for your alarm. For **Whenever**, choose **>=** and type 1.
   b. For **Actions**, select an existing notification list or choose **New list**.
   c. Choose **Create Alarm**.

Troubleshoot

> **Important**
>
> Amazon Elastic Graphics reached end of life on January 8, 2024. For workloads that require graphics acceleration, we recommend that you use Amazon EC2 G4ad, G4dn, or G5 instances.

The following are common errors and troubleshooting steps.

Contents

- Investigate application performance issues
  - OpenGL rendering performance issues
  - Remote access performance issues
- Resolve unhealthy status issues
  - Check the instance configuration
  - Stop and start the instance
  - Verify the installed components
  - Check the Elastic Graphics logs
- Why am I seeing multiple ENIs?

Investigate application performance issues

Elastic Graphics uses the instance network to send OpenGL commands to a remotely attached graphics card. In addition, a desktop running an OpenGL application with an Elastic Graphics accelerator is usually accessed using a remote access technology. It is important to distinguish between a performance problem related to the OpenGL rendering or the desktop remote access technology.

OpenGL rendering performance issues

The OpenGL rendering performance is determined by the number of OpenGL commands and frames generated on the remote instance.

Rendering performance may vary depending on the following factors:

- Elastic Graphics accelerator performance
- Network performance
- CPU performance
- Rendering model, scenario complexity
- OpenGL application behavior

An easy way to evaluate performance is to display the number of rendered frames on the remote instance. Elastic Graphics accelerators display a maximum of 25 FPS on the remote instance to achieve the best perceived quality while reducing network usage.

To show the number of frames produced

1. Open the following file in a text editor. If the file does not exist, create it.

```
C:\Program Files\Amazon\EC2ElasticGPUs\conf\eg.conf
```

2. Identify the `[Application]` section, or add it if it is not present, and add the following configuration parameter:

```
[Application]
show_fps=1
```

3. Restart the application and check the FPS again.

If the FPS reaches 15-25 FPS when updating the rendered scene, then the Elastic Graphics accelerator is performing at peak. Any other performance problems you experience are likely related to the remote access to the instance desktop. If that is the case, see the Remote Access Performance Issues section.

If the FPS number is lower than 15, you can try the following:

- Improve Elastic Graphics accelerator performance by selecting a more powerful graphics accelerator type.
- Improve overall network performance by using these tips:
  - Check the amount of incoming and outgoing bandwidth to and from the Elastic Graphics accelerator endpoint. The Elastic Graphics accelerator endpoint can be retrieved with the following PowerShell command:

```
PS C:\> (Invoke-WebRequest http://169.254.169.254/latest/meta-data/elastic-gpus/
associations/[ELASTICGPU_ID]).content
```

  - The network traffic from the instance to the Elastic Graphics accelerator endpoint relates to the volume of commands the OpenGL application is producing.
  - The network traffic from the Elastic Graphics accelerator endpoint to the instance relates to the number of frames generated by the graphics accelerator.

  - If you see the network usage reaching the instances maximum network throughput, try using an instance with a higher network throughput allowance.
- Improve CPU performance:
  - Applications may require a lot of CPU resources in addition to what the Elastic Graphics accelerator requires. If Windows Task Manager is reporting a high usage of CPU resources, try using an instance with more CPU power.

Remote access performance issues

An instance with an attached Elastic Graphics accelerator can be accessed using different remote access technologies. Performance and quality may vary depending on:

- The remote access technology
- Instance performance
- Client performance
- Network latency and bandwidth between the client and the instance

Possible choices for the remote access protocol include:

- Microsoft Remote Desktop Connection
- NICE DCV
- VNC

For more information about optimization, see the specific protocol.

Resolve unhealthy status issues

If the Elastic Graphics accelerator is in an unhealthy state, use the following troubleshooting steps to resolve the issue.

Check the instance configuration

If the Elastic Graphics command line tool, `egcli.exe`, returns output similar to the following, ensure that your security group is properly configured and that you launched the instance with Instance Metadata Service enabled.

```
EG Version 1.0.7.4240 (Manager) / N/A (OpenGL Library) / N/A (OpenGL Redirector)
EG Status: Out Of Service
Something prevented the EG Infrastructure to work properly.
```

Stop and start the instance

If your Elastic Graphics accelerator is in an unhealthy state, stopping the instance and starting it again is the simplest option. For more information, see Manually stop and start an instances.

> **Warning**
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Verify the installed components

Open the Windows Control Panel and confirm that the following components are installed:

- Amazon Elastic Graphics Manager
- Amazon Elastic Graphics OpenGL Library
- Amazon EC2 Elastic GPUs OpenGL Redirector

If any of these items are missing, you must install them manually. For more information, see Install the required software for Elastic Graphics.

Check the Elastic Graphics logs

Open the Windows Event Viewer, expand the **Application and Services Logs** section, and search for errors in the following event logs:

- EC2ElasticGPUs
- EC2ElasticGPUs GUI

Why am I seeing multiple ENIs?

When calling StartInstances on an EC2 instance with an Elastic Graphics accelerator, a new Elastic Network Interface (ENI) is created on the instance to allow OpenGL commands to be sent to the remotely attached graphics card.

If you call StartInstances many times in a short period of time (a few seconds or less) on the same EC2 instance, a new network interface is created on each call. However:

- Only one network interface will be used by the Elastic Graphics accelerator.
- Extra network interfaces don't incur any charges and will be automatically released in 24 hours.

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		[Linux and Windows Server instances] Collect Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent [Migration from previous CloudWatch Logs agent on Windows Server instances] Migrate Windows Server Instance Log Collection to the CloudWatch Agent

Automated and manual monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

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
- During instance reboot or while a Windows instance store-backed instance is being bundled, an instance status check reports a failure until the instance becomes available again.

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

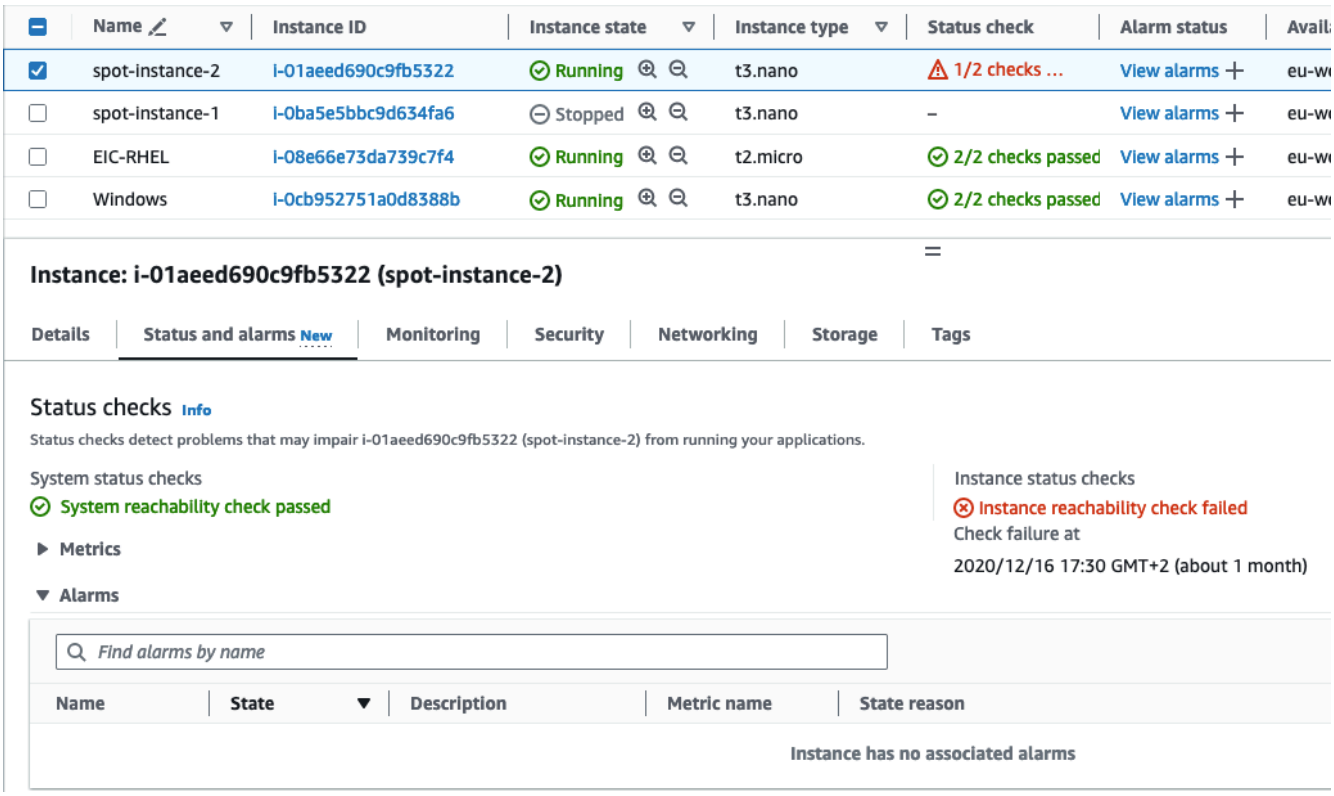
To view status checks, use one of the following methods.

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.



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).

5. To review the CloudWatch metrics for status checks, on the **Status and alarms** tab, expand **Metrics** to see the graphs for the following metrics:

   - **Status check failed for system**
   - **Status check failed for instance**

   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)

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.

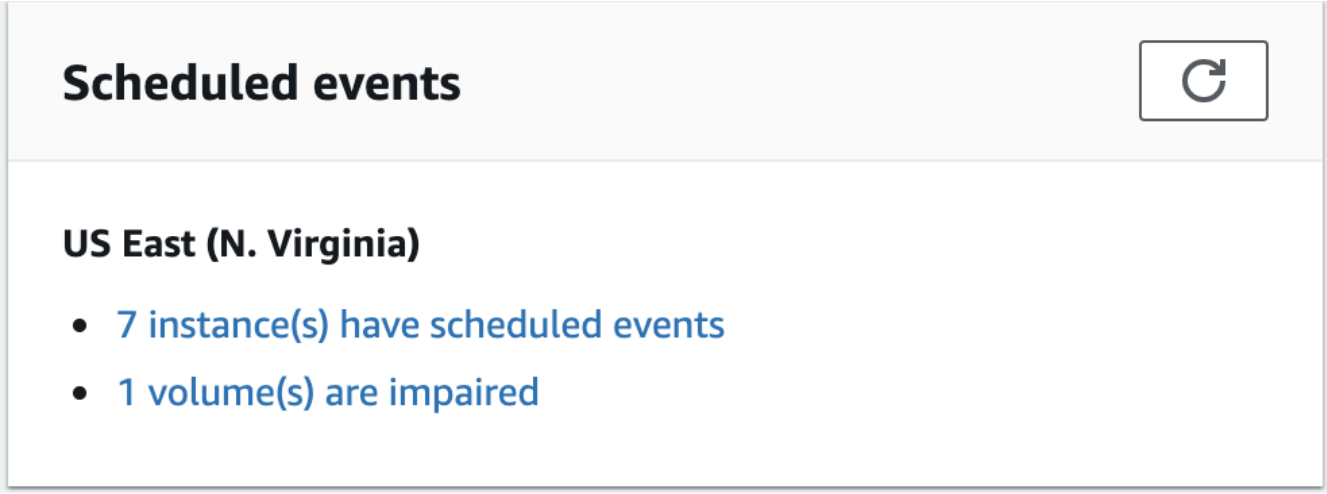
View scheduled events

In addition to receiving notification of scheduled events in email, you can check for scheduled events by using one of the following methods.

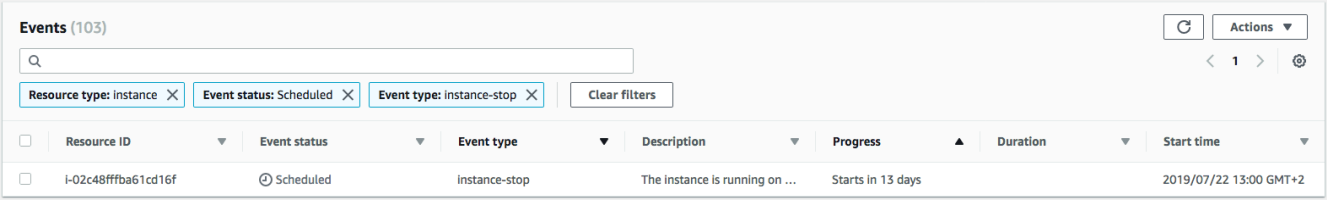
Console

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.



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 your instance.

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 your instance.

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"
                }
               ]
]
```

2. To reschedule the event, use the modify-instance-event-start-time command. Specify the new event start time by using the `not-before` parameter. The new event start time must fall before the `NotBeforeDeadline`.

```
aws ec2 modify-instance-event-start-time \
    --instance-id i-1234567890abcdef0 \
    --instance-event-id instance-event-0d59937288b749b32 \
    --not-before 2019-03-25T10:00:00.000
```

It might take a minute or 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
- 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 Amazon CloudWatch alarms for your instances on the **Instances** screen in the Amazon EC2 console.

Costs for ListMetrics API calls

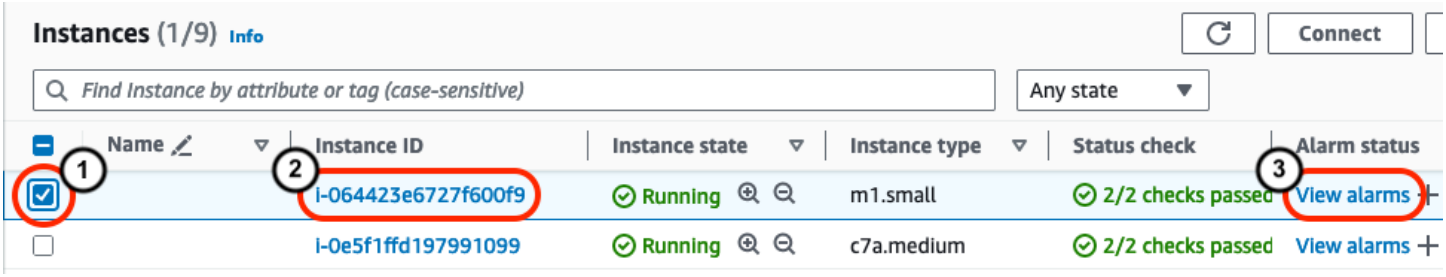
For every 1,000 `ListMetrics` API requests, you might incur a $0.01 cost, depending on whether you're still within the AWS Free Tier. Under the Free Tier, you get 1 million free CloudWatch API requests (excluding `GetMetricData`, `GetInsightRuleReport`, and `GetMetricWidgetImage`, which are always charged). For more information, see **Free Tier** on the Amazon CloudWatch Pricing page.

When you perform the following actions in the EC2 console, Amazon EC2 makes a CloudWatch `ListMetrics` API request:

- When you select the check box for an instance in the **Instances** table (indicated by **1** in the screenshot below).
- When you select an instance by choosing its ID in the **Instances** table (indicated by **2** in the screenshot below).
- When you choose **View alarms** in the **Instances** table to open the **Alarm details for i-*1234567890example*** window (indicated by **3** in the screenshot below).

> **Note**
>
> When you choose **View alarms**, the check box for the instance (indicated by **1** in the screenshot below) is automatically selected, which results in another `ListMetrics` API request.

The following screenshot indicates the console controls, numbered **1**, **2**, and **3**, that when selected, invoke the `ListMetrics` API.



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 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.	You are charged per metric that is sent to CloudWatch. You are not charged for data storage. For more information, see **Paid tier** and **Example 1 - EC2 Detailed Monitoring** on the Amazon CloudWatch pricing page.

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**, **Monitoring**, **Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, select the **Enable** check box.
5. Choose **Save**.

To enable detailed monitoring when launching an instance

When launching an instance using the 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**, **Monitoring**, **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`	The percentage of physical CPU time that Amazon EC2 uses to run the EC2 instance, which includes time spent to run both the user code and the Amazon EC2 code. At a very high level, `CPUUtilization` is the sum of guest `CPUUtilization` and hypervisor `CPUUtilization` . Tools in your operating system can show a different percentage than CloudWatch due to factors such as legacy device simulation, configuration of non-legacy devices, interrupt -heavy workloads, live migration, and live update.	Percent	• Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`DiskReadOps`	Completed read operations from all instance store volumes available to the instance in a specified period of time. To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Count	• Sum • Average • Minimum • Maximum
`DiskWriteOps`	Completed write operations to all instance store volumes available to the instance in a specified period of time. To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`DiskReadBytes`	Bytes read from all instance store volumes available to the instance. This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application. The number reported is the number of bytes received during the period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `DiskReadBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`DiskWriteBytes`	Bytes written to all instance store volumes available to the instance. This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application. The number reported is the number of bytes received during the period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `DiskWriteBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more informati on about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*. If there are no instance store volumes, either the value is 0 or the metric is not reported.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`MetadataNoToken`	The number of times the Instance Metadata Service (IMDS) was successfully accessed using a method that does not use a token. This metric is used to determine if there are any processes accessing instance metadata that are using Instance Metadata Service Version 1 (IMDSv1), which does not use a token. If all requests use token-backed sessions, i.e., Instance Metadata Service Version 2 (IMDSv2), the value is 0. For more information, see Transition to using Instance Metadata Service Version 2.	Count	• Sum • Percentiles
`MetadataNoTokenRejected`	The number of times an IMDSv1 call was attempted after IMDSv1 was disabled. If this metric appears, it indicates that an IMDSv1 call was attempted and rejected. You can either re-enable IMDSv1 or make sure all of your calls use IMDSv2. For more information, see Transition to using Instance Metadata Service Version 2.	Count	• Sum • Percentiles

Metric	Description	Unit	Meaningful statistics
`NetworkIn`	The number of bytes received by the instance on all network interfaces. This metric identifies the volume of incoming network traffic to a single instance. The number reported is the number of bytes received during the period. If you are using basic (5-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring and the statistic is Sum, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `NetworkIn` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`NetworkOut`	The number of bytes sent out by the instance on all network interfaces. This metric identifies the volume of outgoing network traffic from a single instance. The number reported is the number of bytes sent during the period. If you are using basic (5-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (1-minute) monitoring and the statistic is Sum, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `NetworkOut` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`NetworkPacketsIn`	The number of packets received by the instance on all network interfaces. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only (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 `DIFF_TIME` to find the packets per second. For example, if you have graphed `NetworkPacketsIn` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in packets/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`NetworkPacketsOut`	The number of packets sent out by the instance on all network interfaces. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance. This metric is available for basic 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 `DIFF_TIME` to find the packets per second. For example, if you have graphed `NetworkPacketsOut` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in packets/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Count	• Sum • Average • Minimum • Maximum

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)	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
	of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilizati on for two minutes). CPU credit metrics are available at a 5-minute frequency only. If you specify a period greater than five minutes, use the `Sum` statistic instead of the `Average` statistic.		

Metric	Description	Unit	Meaningful statistics
`CPUCreditBalance`	The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the `CPUCreditBalance` also includes the number of launch credits that have been accrued. Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded . For T2 Standard, launch credits do not count towards the limit. The credits in the `CPUCreditBalance` are available for the instance to spend to burst beyond its baseline CPU utilization. When an instance is running, credits in the `CPUCreditBalance` do not expire. When a T3 or T3a instance stops, the `CPUCreditBalance` value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the `CPUCreditBalance` value does not persist, and all accrued credits are lost. CPU credit metrics are available at a 5-minute frequency only.	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`CPUSurplusCreditBalance`	The number of surplus credits that have been spent by an `unlimited` instance when its `CPUCreditBalance` value is zero. The `CPUSurplusCreditBalance` value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge. CPU credit metrics are available at a 5-minute frequency only.	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum
`CPUSurplusCreditsCharged`	The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge. Spent surplus credits are charged when any of the following occurs: • The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. • The instance is stopped or terminated. • The instance is switched from `unlimited` to `standard`. CPU credit metrics are available at a 5-minute frequency only.	Credits (vCPU-minutes)	• Sum • Average • Minimum • Maximum

Dedicated Host metrics

The `AWS/EC2` namespace includes the following metrics for T3 Dedicated Hosts.

Metric	Description	Unit	Meaningful statistics
`DedicatedHostCPUUtilization`	The percentage of allocated compute capacity that is currently in use by the instances running on the Dedicated Host.	Percent	• Sum • Average • Minimum • Maximum

Amazon EBS metrics for Nitro-based instances

The `AWS/EC2` namespace includes additional Amazon EBS metrics for volumes that are attached to Nitro-based instances that are not bare metal instances.

Metric	Description	Unit	Meaningful statistics
`EBSReadOps`	Completed read operations from all Amazon EBS volumes attached to the instance in a specified period of time. To calculate the average read I/O operations per second (Read IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to calculate the Read IOPS. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the operations per second. For example, if you have graphed `EBSReadOps` in CloudWatch as `m1`, the	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
	metric math formula `m1/(DIFF_TIME(m1))` returns the metric in operations/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.		
`EBSWriteOps`	Completed write operations to all EBS volumes attached to the instance in a specified period of time. To calculate the average write I/O operations per second (Write IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to calculate the Write IOPS. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the operations per second. For example, if you have graphed `EBSWriteOps` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in operations/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Count	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`EBSReadBytes`	Bytes read from all EBS volumes attached to the instance in a specified period of time. The number reported is the number of bytes read during the period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to find Read Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `EBSReadBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Bytes	• Sum • Average • Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`EBSWriteBytes`	Bytes written to all EBS volumes attached to the instance in a specified period of time. The number reported is the number of bytes written during the period. If you are using basic (5-minute) monitoring, you can divide this number by 300 to find Write Bytes/second. If you have detailed (1-minute) monitoring, divide it by 60. You can also use the CloudWatch metric math function `DIFF_TIME` to find the bytes per second. For example, if you have graphed `EBSWriteBytes` in CloudWatch as `m1`, the metric math formula `m1/(DIFF_TIME(m1))` returns the metric in bytes/second. For more information about `DIFF_TIME` and other metric math functions, see Use metric math in the *Amazon CloudWatch User Guide*.	Bytes	• Sum • Average • Minimum • Maximum
`EBSIOBalance%`	Provides information about the percentage of I/O credits remaining in the burst bucket. This metric is available for basic monitoring only. This metric is available only for some `*.4xlarge` instance sizes and smaller that burst to their maximum performance for only 30 minutes at least once every 24 hours. The `Sum` statistic is not applicable to this metric.	Percent	• Minimum • Maximum

Metric	Description	Unit	Meaningful statistics
`EBSByteBalance%`	Provides information about the percentage of throughput credits remaining in the burst bucket. This metric is available for basic monitoring only. This metric is available only for some `*.4xlarge` instance sizes and smaller that burst to their maximum performance for only 30 minutes at least once every 24 hours. The `Sum` statistic is not applicable to this metric.	Percent	• Minimum • Maximum

For information about the metrics provided for your EBS volumes, see Metrics for Amazon EBS volumes in the *Amazon EBS User Guide*. For information about the metrics provided for your 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`	Reports whether the instance has passed both the instance status check and the system status check in the last minute. This metric can be either `0` (passed) or `1` (failed).	Count	• Sum • Average

Metric	Description	Unit	Meaningful statistics
	By default, this metric is available at a 1-minute frequency at no charge.		
`StatusCheckFailed_Instance`	Reports whether the instance has passed the instance status check in the last minute. This metric can be either `0` (passed) or `1` (failed). By default, this metric is available at a 1-minute frequency at no charge.	Count	• Sum • Average
`StatusCheckFailed_System`	Reports whether the instance has passed the system status check in the last minute. This metric can be either `0` (passed) or `1` (failed). By default, this metric is available at a 1-minute frequency at no charge.	Count	• Sum • Average
`StatusCheckFailed_AttachedEBS`	Reports whether the instance has passed the attached EBS status check in the last minute. This metric can be either `0` (passed) or `1` (failed). By default, this metric is available at a 1-minute frequency at no charge.	Count	• Sum • Average

The `AWS/EBS` namespace includes the following status check metric.

Metric	Description	Unit	Meaningful statistics
`VolumeStalledIOCheck`	**Note:** For Nitro instances only. Not published for volumes attached to Amazon ECS and AWS Fargate tasks. Reports whether a volume has passed or failed a *stalled IO check* in the last minute. This metric can be either `0` (passed) or `1` (failed).	Count	• 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 *Auto Scaling group* is a collection of instances you define if you're using Auto Scaling. This dimension is available only for Amazon EC2 metrics when the instances are in such an Auto Scaling group. Available for instances with Detailed or Basic Monitoring enabled.
`ImageId`	This dimension filters the data you request for all instances running this Amazon EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.

Dimension	Description
`InstanceId`	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
`InstanceType`	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

Amazon EC2 usage metrics

You can use CloudWatch usage metrics to provide visibility into your account's usage of resources. Use these metrics to visualize your current service usage on CloudWatch graphs and dashboards.

Amazon EC2 usage metrics correspond to AWS service quotas. You can configure alarms that alert you when your usage approaches a service quota. For more information about CloudWatch integration with service quotas, see AWS usage metrics in the *Amazon CloudWatch User Guide*.

Amazon EC2 publishes the following metrics in the `AWS/Usage` namespace.

Metric	Description
`ResourceCount`	The number of the specified resources running in your account. The resources are defined by the dimensions associated with the metric. The most useful statistic for this metric is `MAXIMUM`, which represents the maximum number of resources used during the 1-minute period.

The following dimensions are used to refine the usage metrics that are published by Amazon EC2.

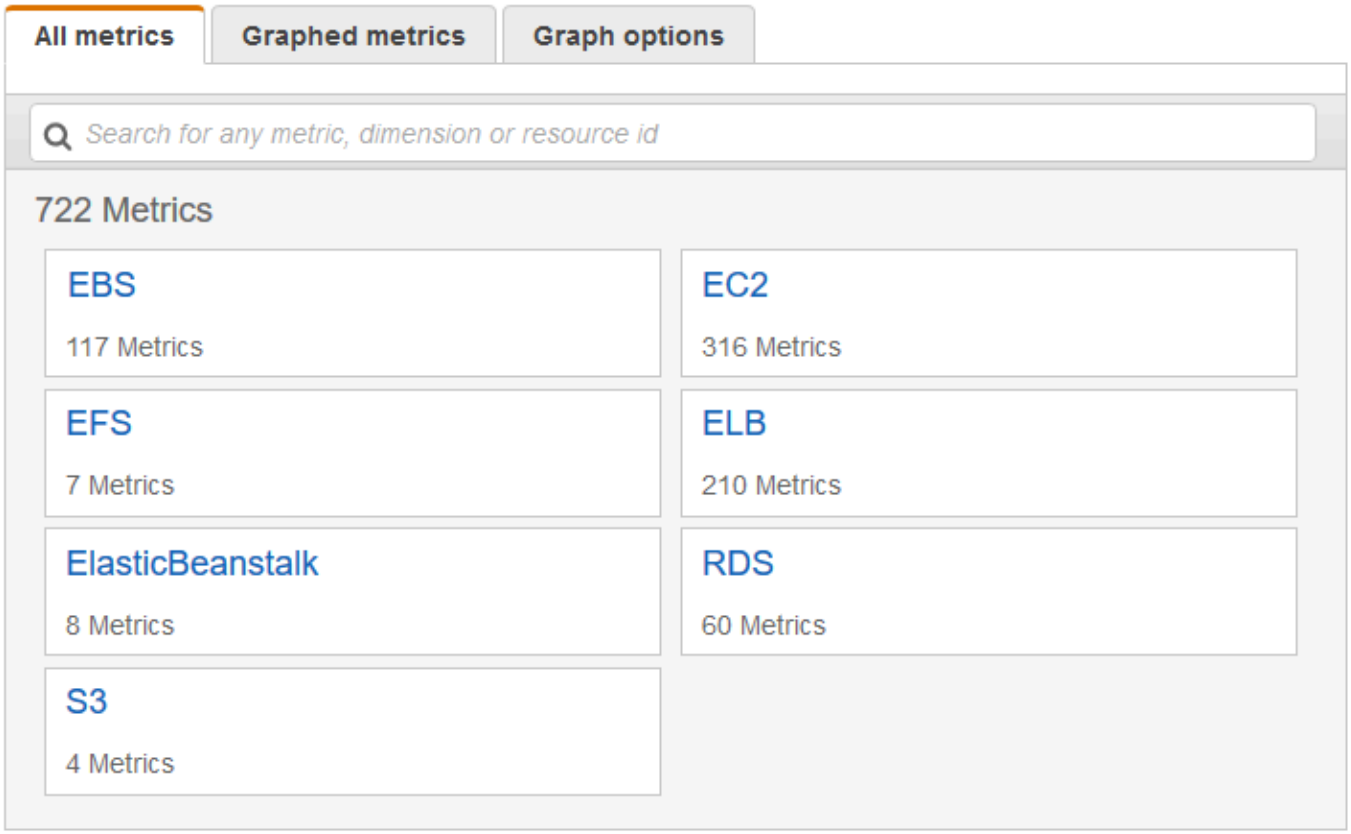
Dimension	Description
`Service`	The name of the AWS service containing the resource. For Amazon EC2 usage metrics, the value for this dimension is `EC2`.
`Type`	The type of entity that is being reported. Currently, the only valid value for Amazon EC2 usage metrics is `Resource`.
`Resource`	The type of resource that is running. Currently, the only valid value for Amazon EC2 usage metrics is `vCPU`, which returns information on instances that are running.
`Class`	The class of resource being tracked. For Amazon EC2 usage metrics with `vCPU` as the value of the `Resource` dimension, the valid values are `Standard/OnDemand` , `F/OnDemand` , `G/OnDemand` , `Inf/OnDemand` , `P/OnDemand` , and `X/OnDemand` . The values for this dimension define the first letter of the instance types that are reported by the metric. For example, `Standard/OnDemand` returns information about all running instances with types that start with A, C, D, H, I, M, R, T, and Z, and `G/OnDemand` returns information about all running instances with types that start with G.

List metrics using the console

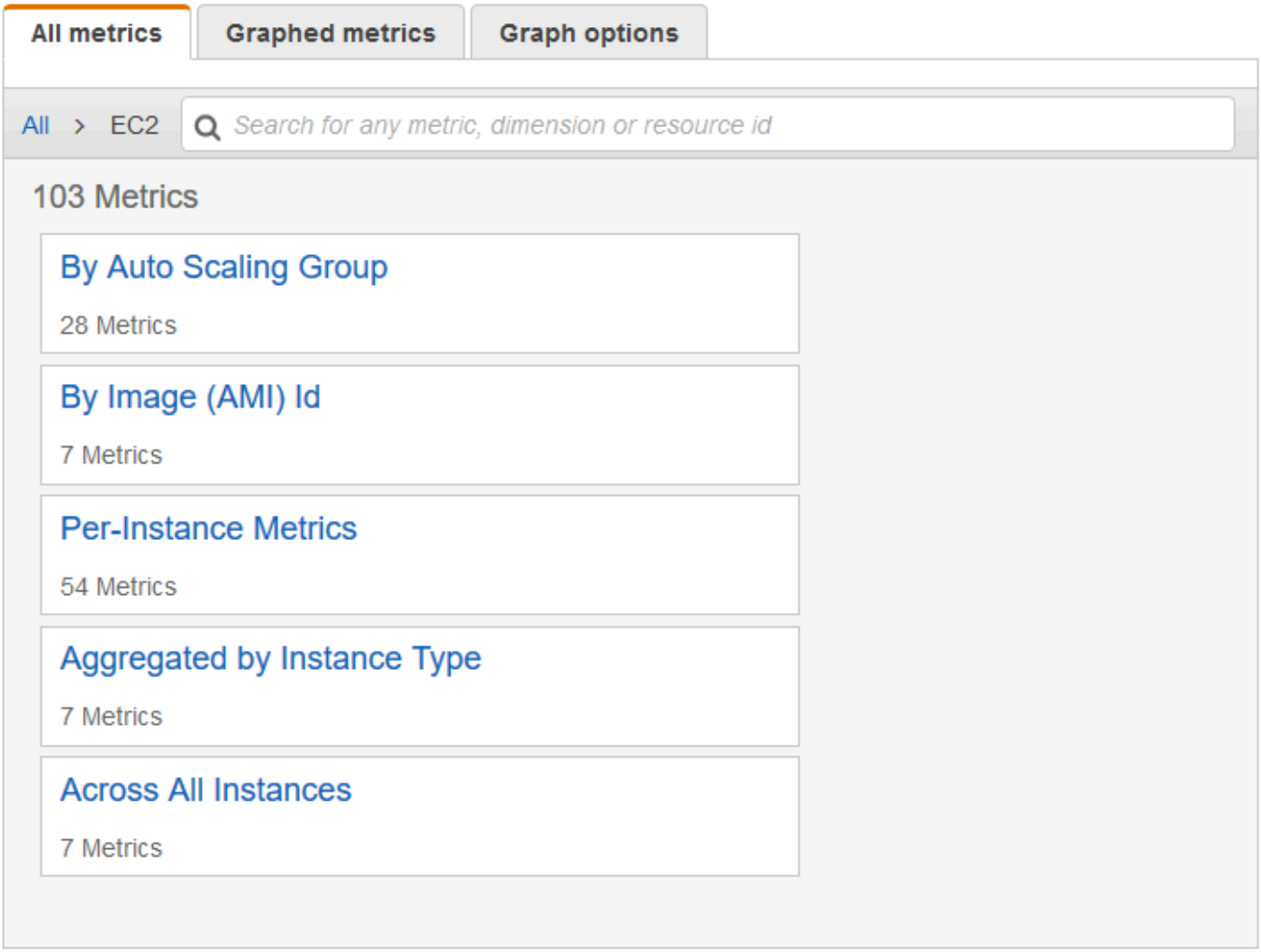
Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all metrics provided by Amazon EC2, or metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling group.

To view available metrics by category (console)

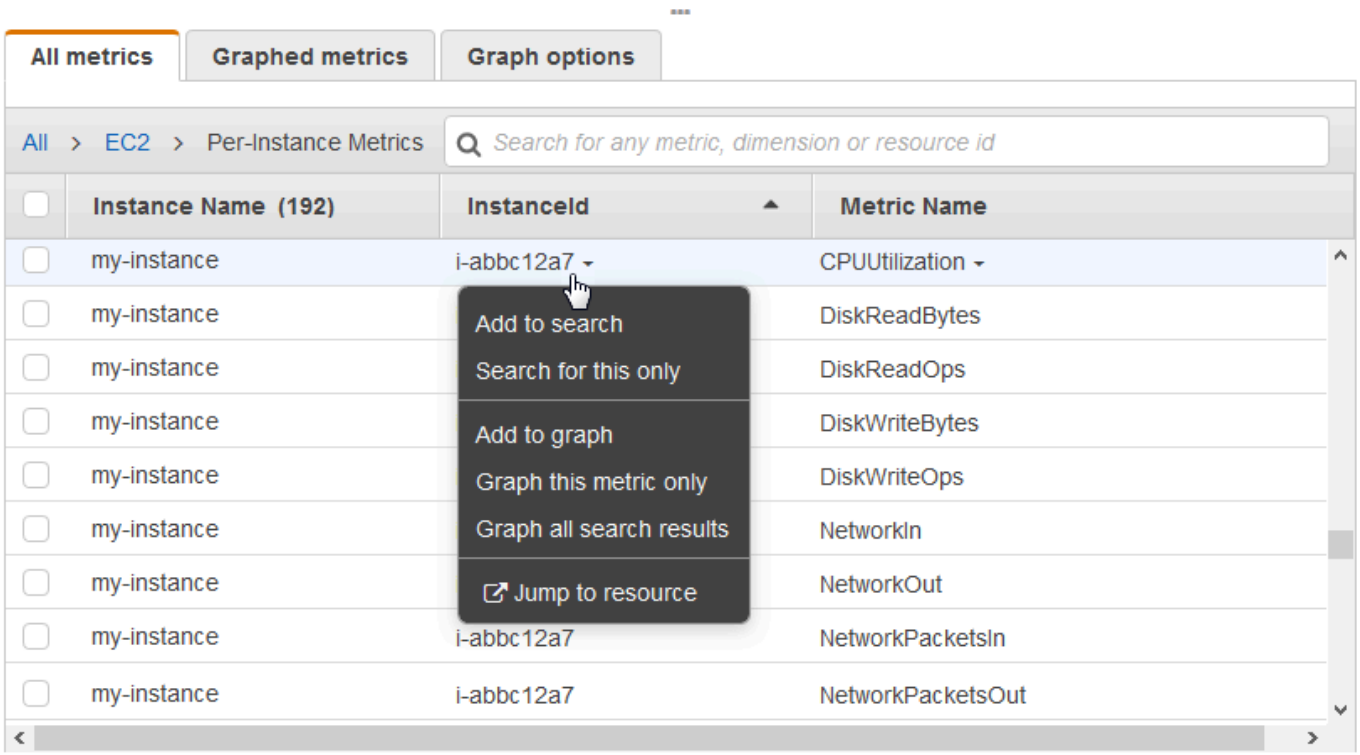
1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.



4. Select a metric dimension (for example, **Per-Instance Metrics**).



5. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric. To filter by resource, choose the resource ID and then choose **Add to search**. To filter by metric, choose the metric name and then choose **Add to search**.



List metrics using the AWS CLI

Use the list-metrics command to list the CloudWatch metrics for your instances.

To list all the available metrics for Amazon EC2 (AWS CLI)

The following example specifies the `AWS/EC2` namespace to view all the metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:

```
{
  "Metrics": [
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
```

```
        ],
        "MetricName": "NetworkOut"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkIn"
    },
    ...
  ]
}
```

To list all the available metrics for an instance (AWS CLI)

The following example specifies the `AWS/EC2` namespace and the `InstanceId` dimension to view the results for the specified instance only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

To list a metric across all instances (AWS CLI)

The following example specifies the `AWS/EC2` namespace and a metric name to view the results for the specified metric only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Get statistics for metrics for your instances

You can get statistics for the CloudWatch metrics for your instances.

Contents

- Statistics overview
- 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.
`Maximum`	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
`Sum`	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
`Average`	The value of `Sum` / `SampleCount` during the specified period. By comparing this statistic with the `Minimum` and `Maximum`, you can determine the full scope of a metric and how close the average use is to the `Minimum` and `Maximum`. This comparison helps you to know when to increase or decrease your resources as needed.

Statistic	Description
`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

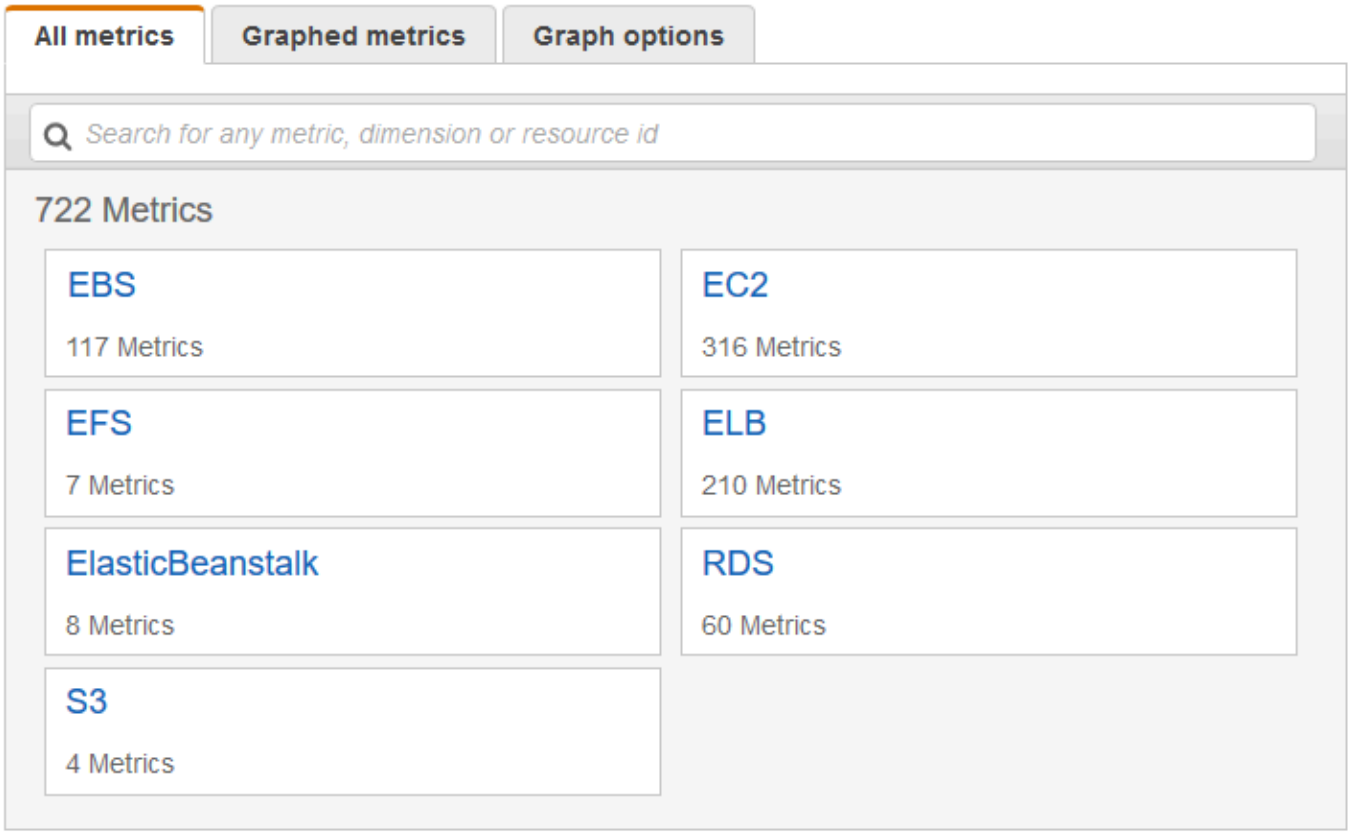
The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

Requirements

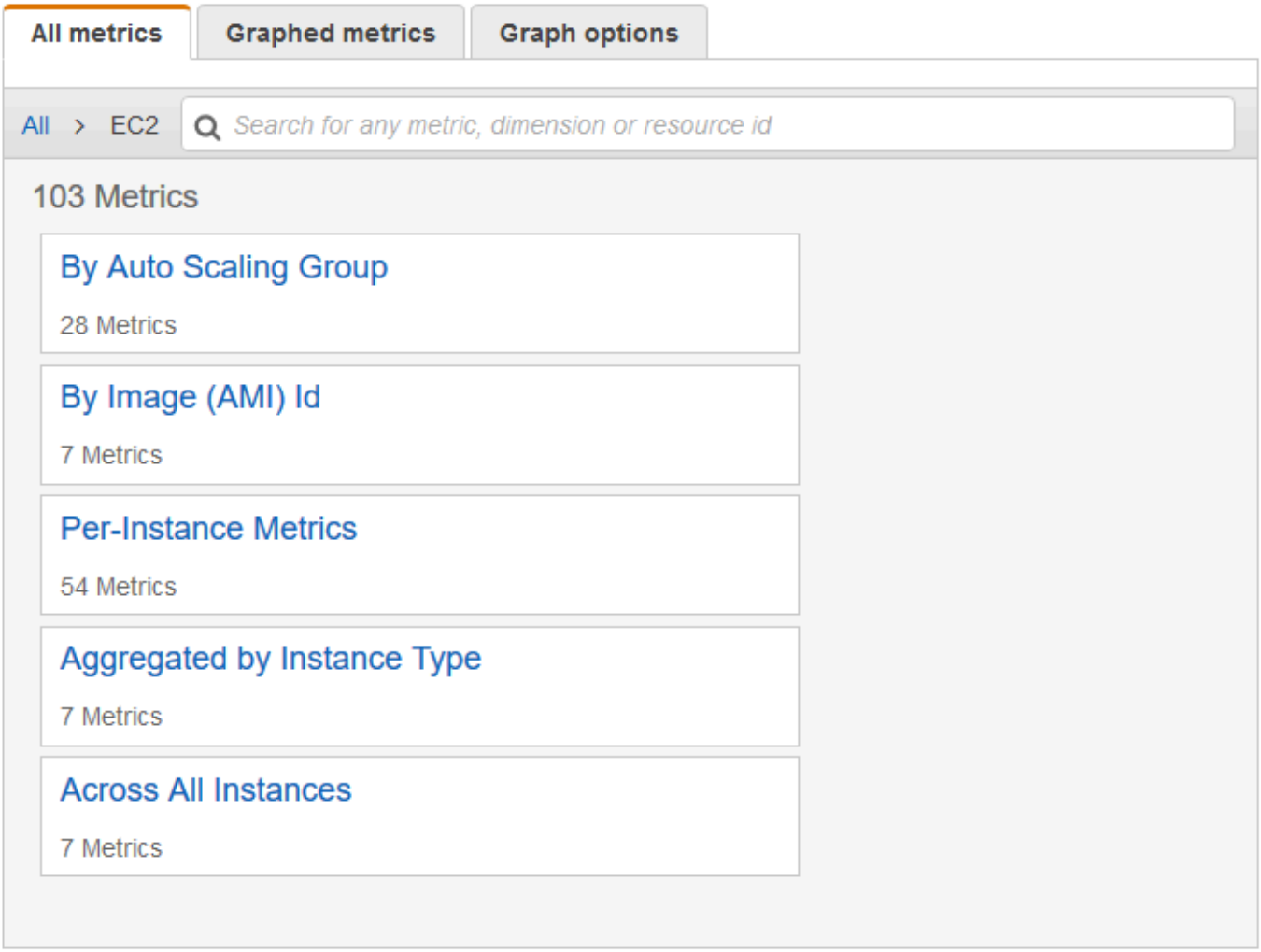
- You must have the ID of the instance. You can get the instance ID using the AWS Management Console or the describe-instances command.
- By default, basic monitoring is enabled, but you can enable detailed monitoring. For more information, see Enable or turn off detailed monitoring for your instances.

To display the CPU utilization for a specific instance (console)

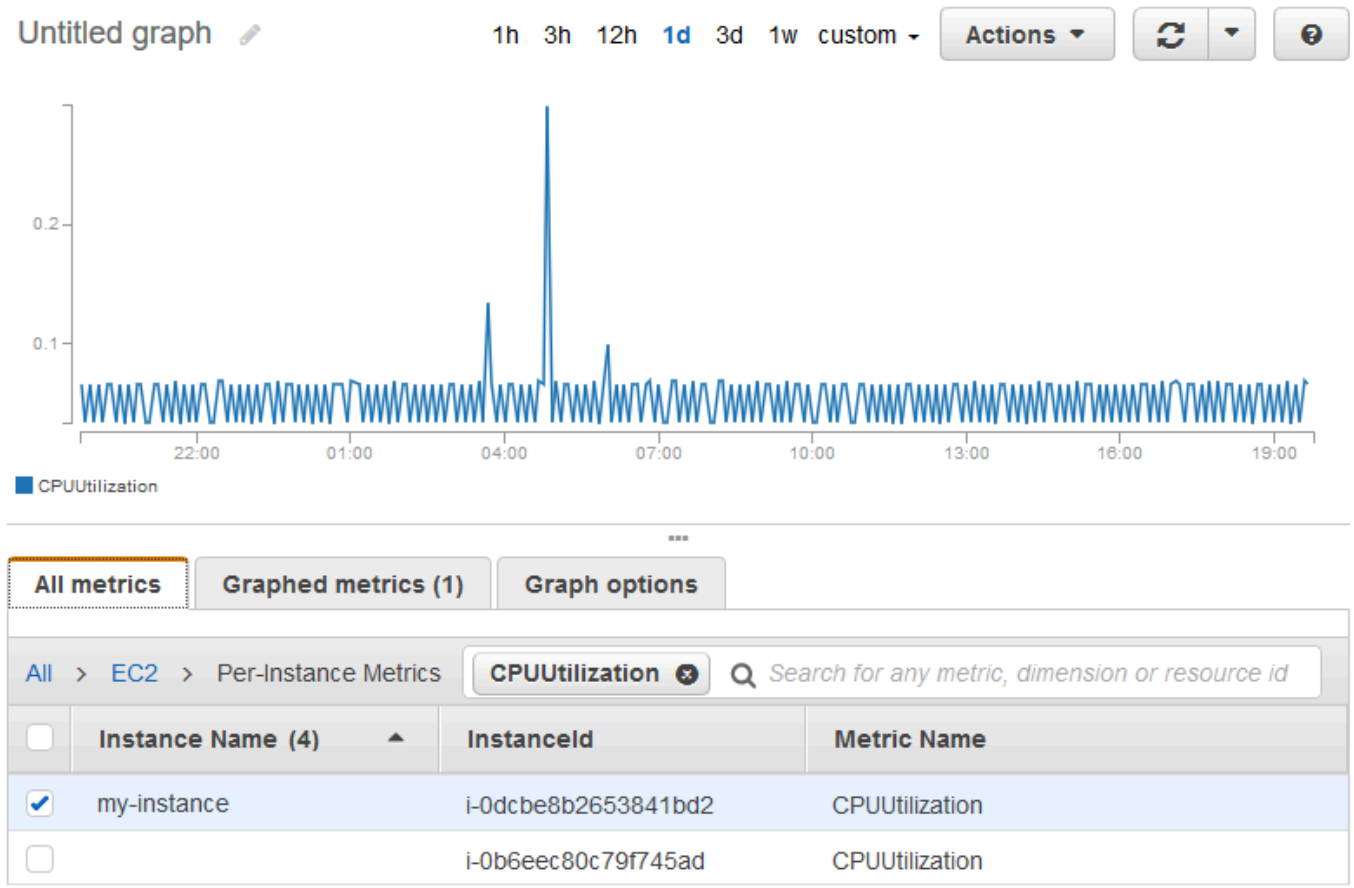
1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.



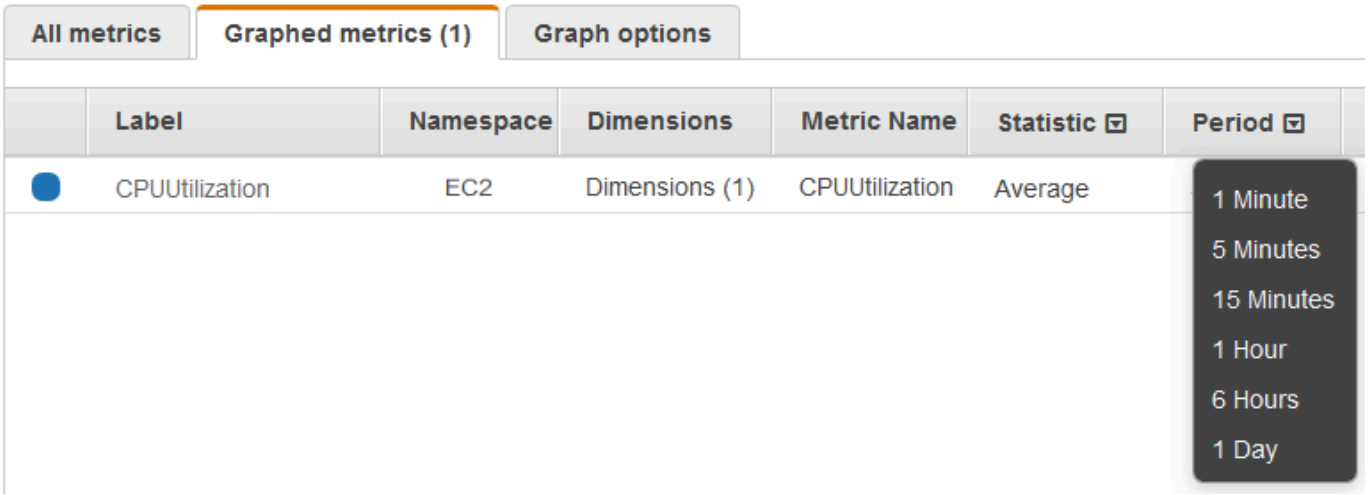
4. Choose the **Per-Instance Metrics** dimension.



5. In the search field, enter **`CPUUtilization`** and press Enter. Choose the row for the specific instance, which displays a graph for the **CPUUtilization** metric for the instance. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



6. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.



To get the CPU utilization for a specific instance (AWS CLI)

Use the following get-metric-statistics command to get the **CPUUtilization** metric for the specified instance, using the specified period and time interval:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --period 3600 \
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
--start-time 2022-10-18T23:18:00 --end-time 2022-10-19T23:18:00
```

The following is example output. Each value represents the maximum CPU utilization percentage for a single EC2 instance.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Aggregate statistics across instances

Aggregate statistics are available for instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. Before you can get statistics

aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

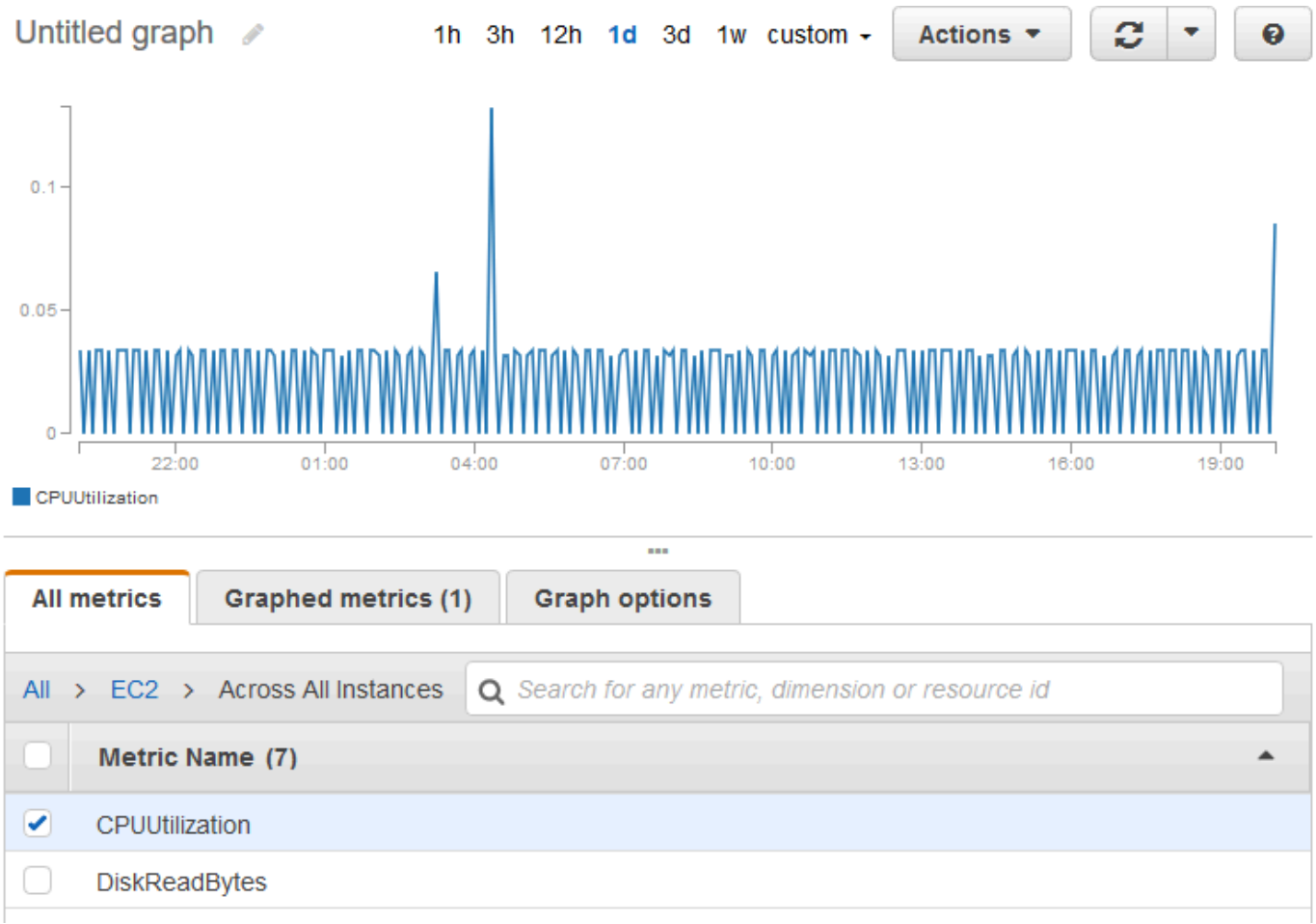
Note that Amazon CloudWatch cannot aggregate data across AWS Regions. Metrics are completely separate between Regions.

This example shows you how to use detailed monitoring to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the `AWS/EC2` namespace.

> **Important**
>
> This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

To display average CPU utilization across your instances (console)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **Across All Instances**.
4. Choose the row that contains **CPUUtilization**, which displays a graph for the metric for all your EC2 instances. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get average CPU utilization across your instances (AWS CLI)

Use the get-metric-statistics command as follows to get the average of the **CPUUtilization** metric across your instances.

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/EC2 \
    --metric-name CPUUtilization \
    --period 3600  --statistics "Average" "SampleCount" \
    --start-time 2022-10-11T23:18:00 \
    --end-time 2022-10-12T23:18:00
```

The following is example output:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2022-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Aggregate statistics by Auto Scaling group

You can aggregate statistics for the EC2 instances in an Auto Scaling group. Note that Amazon CloudWatch cannot aggregate data across AWS Regions. Metrics are completely separate between Regions.

This example shows you how to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for 1-minute periods for a 24-hour interval across all EC2 instances in the specified Auto Scaling group.

To display DiskWriteBytes for the instances in an Auto Scaling group (console)

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Auto Scaling Group**.

4. Choose the row for the **DiskWriteBytes** metric and the specific Auto Scaling group, which displays a graph for the metric for the instances in the Auto Scaling group. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To display DiskWriteBytes for the instances in an Auto Scaling group (AWS CLI)

Use the get-metric-statistics command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
 --period 360 \
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --
start-time 2022-10-16T23:18:00 --end-time 2022-10-18T23:18:00
```

The following is example output:

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2022-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2022-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

Aggregate statistics by AMI

You can aggregate statistics 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.

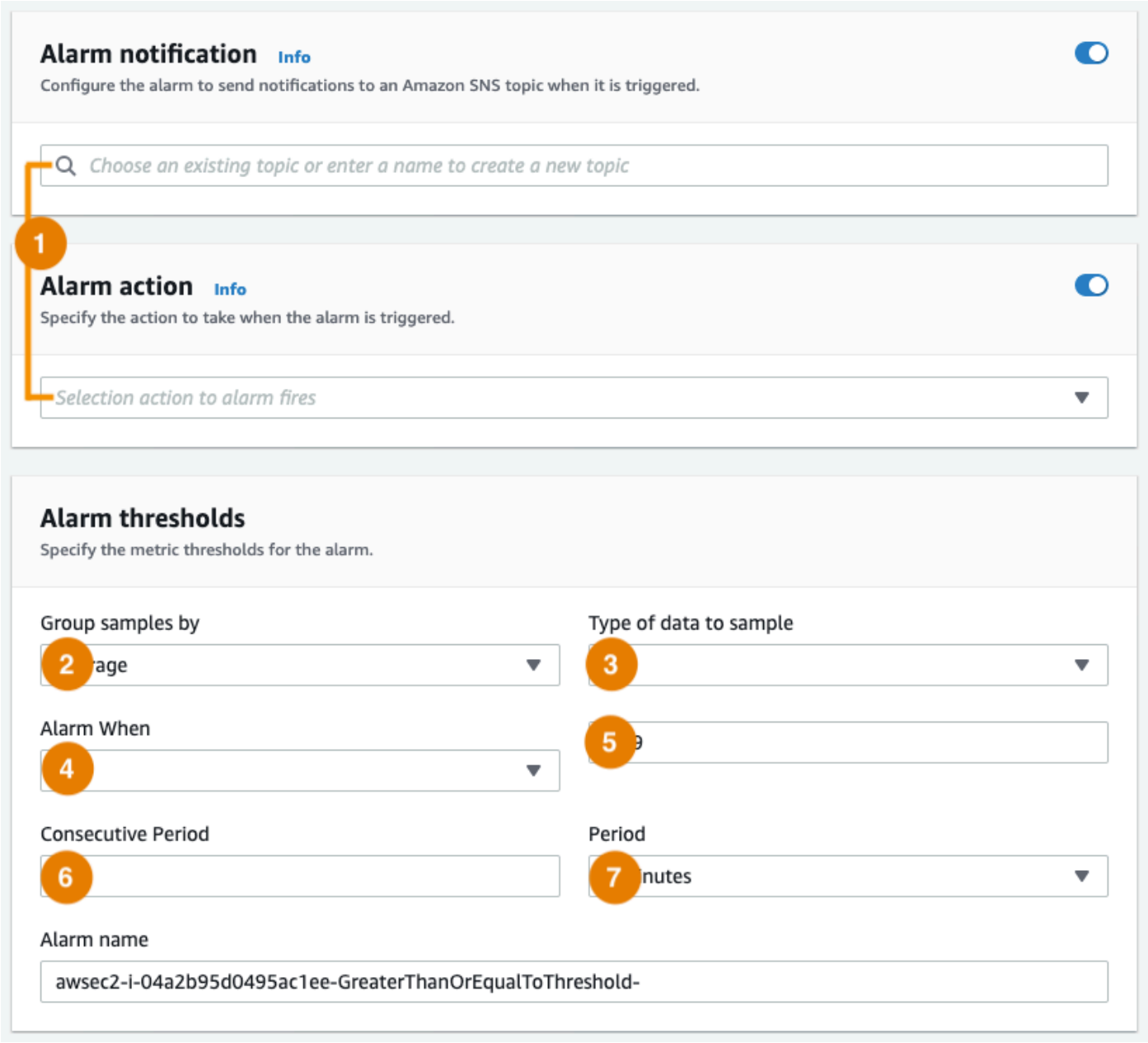
To view the history of triggered alarms and actions (CloudWatch console)

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.

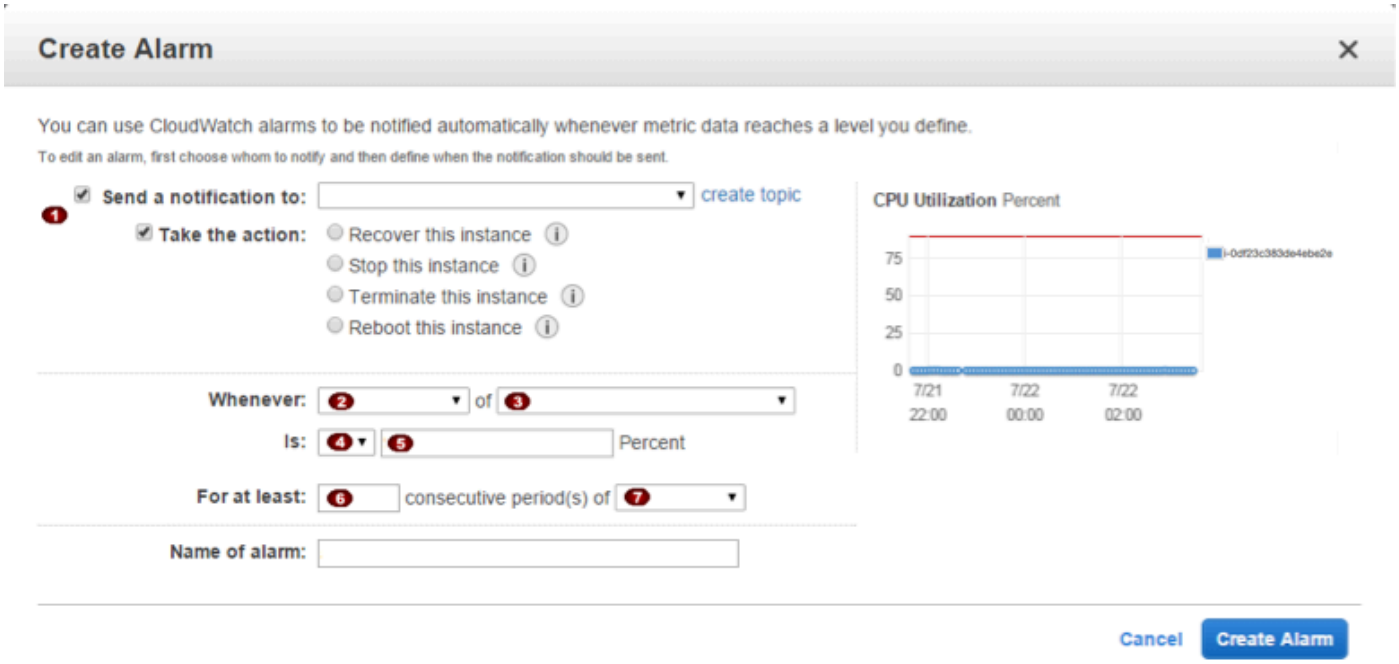
Amazon CloudWatch alarm action scenarios

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.
Choose an existing topic or enter a name to create a new topic
1
Alarm action Info
Specify the action to take when the alarm is triggered.
Selection action to alarm fires
Alarm thresholds
Specify the metric thresholds for the alarm.
Group samples by
2
Type of data to sample
3
Alarm When
4
5
Consecutive Period
6
Period
7
Alarm name
awsec2-i-04a2b95d0495ac1ee-GreaterThanOrEqualToThreshold-

Old console



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 Fast Launch State-change Notification
- EC2 Fleet Error
- EC2 Fleet Information
- EC2 Fleet Instance Change
- EC2 Fleet Spot Instance Request Change
- EC2 Fleet State Change
- EC2 Instance Rebalance Recommendation
- EC2 Instance State-change Notification
- EC2 Spot Fleet Error
- EC2 Spot Fleet Information
- EC2 Spot Fleet Instance Change
- EC2 Spot Fleet Spot Instance Request Change

- EC2 Spot Fleet State Change
- EC2 Spot Instance Interruption Warning
- EC2 Spot Instance Request Fulfillment
- EC2 ODCR Underutilization Notification

For information about the event types supported by Amazon EBS, see 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": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
        "userName": "IAM-friendly-name",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-09-21T21:37:58Z"}
        }
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey ",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
```

```
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNO01234567890EXAMPLE"
        }
     },
    "responseElements": null,
    "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",
    "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",
    "eventType": "AwsApiCall",
    "recipientAccountId": "0987654321"
}
```

If you have configured your AWS account to collect CloudTrail events in an S3 bucket, you can download and audit the information programmatically. For more information, see Getting and viewing your CloudTrail log files in the *AWS CloudTrail User Guide*.

Monitor your .NET and SQL Server applications with CloudWatch Application Insights

CloudWatch Application Insights helps you monitor your .NET and SQL Server applications that use Amazon EC2 instances along with other AWS application resources. It identifies and sets up key metrics logs, and alarms across your application resources and technology stack (for example, your Microsoft SQL Server database, web (IIS) and application servers, OS, load balancers, and queues). It continuously monitors the metrics and logs to detect and correlate anomalies and errors. When errors and anomalies are detected, Application Insights generates CloudWatch Events that you can use to set up notifications or take actions. To aid with troubleshooting, it creates automated dashboards for the detected problems, which include correlated metric anomalies and log errors, along with additional insights to point you to the potential root cause. The automated dashboards help you to take swift remedial actions to keep your applications healthy and to prevent impact to the end users of your application.

To view a complete list of supported logs and metrics, see Logs and Metrics Supported by Amazon CloudWatch Application Insights.

Information provided about detected problems:

- A short summary of the problem
- The start time and date of the problem
- The problem severity: High/Medium/Low
- The status of the detected problem: In-progress/Resolved

- Insights: Automatically generated insights on the detected problem and possible root cause
- Feedback on insights: Feedback you have provided about the usefulness of the insights generated by CloudWatch Application Insights for .NET and SQL Server
- Related observations: A detailed view of the metric anomalies and error snippets of relevant logs related to the problem across various application components

Feedback

You can provide feedback on automatically generated insights on detected problems by designating them useful or not useful. Your feedback on the insights, along with your application diagnostics (metric anomalies and log exceptions), are used to improve the future detection of similar problems.

For more information, see the CloudWatch Application Insights documentation in the *Amazon CloudWatch User Guide*.

Networking in Amazon EC2

Amazon VPC enables you to launch AWS resources, such as Amazon EC2 instances, into a virtual network dedicated to your AWS account, known as a virtual private cloud (VPC). When you launch an instance, you can select a subnet from the VPC. The instance is configured with a primary network interface, which is a logical virtual network card. The instance receives a primary private IP address from the IPv4 address of the subnet, and it is assigned to the primary network interface.

You can control whether the instance receives a public IP address from Amazon's pool of public IP addresses. The public IP address of an instance is associated with your instance only until it is stopped or terminated. If you require a persistent public IP address, you can allocate an Elastic IP address for your AWS account and associate it with an instance or a network interface. An Elastic IP address remains associated with your AWS account until you release it, and you can move it from one instance to another as needed. You can bring your own IP address range to your AWS account, where it appears as an address pool, and then allocate Elastic IP addresses from your address pool.

To increase network performance and reduce latency, you can launch instances in a placement group. You can get significantly higher packet per second (PPS) performance using enhanced networking. You can accelerate high performance computing and machine learning applications using an Elastic Fabric Adapter (EFA), which is a network device that you can attach to a supported instance type.

Features

- Regions and Zones
- 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 Windows
- Amazon EC2 instance topology
- Placement groups
- Network maximum transmission unit (MTU) for your EC2 instance

- Virtual private clouds
- Ports and Protocols for Windows Amazon Machine Images (AMIs)

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*.

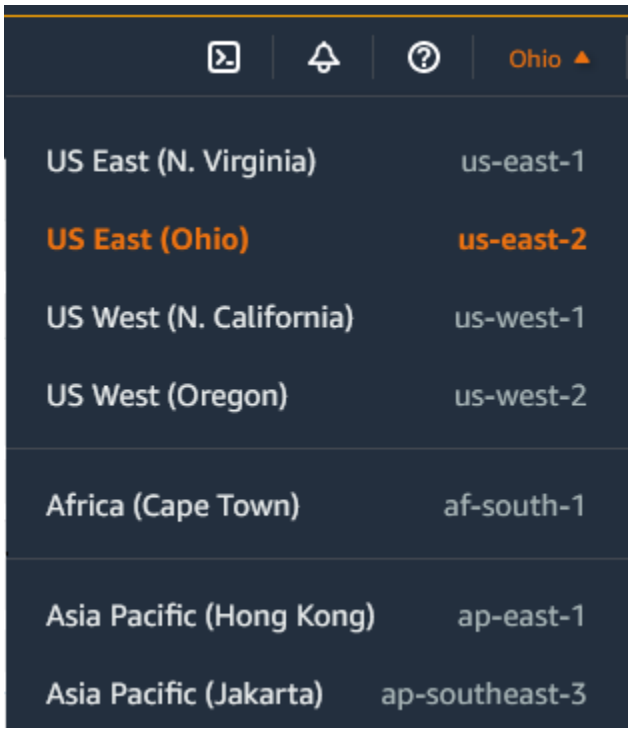
For more information about endpoints and protocols in AWS GovCloud (US-West), see Service Endpoints in the *AWS GovCloud (US) User Guide*.

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
```

Tools for PowerShell

If it is not installed, install the **AWS.Tools.SimpleSystemsManagement** module to Tools for PowerShell by running `Install-AWSToolsModule AWS.Tools.SimpleSystemsManagement -CleanUp`.

```
$parameterPath = "/aws/service/global-infrastructure/regions/af-south-1"
$substringToMatch = "longName"
$filteredParameters = Get-SSMParametersByPath -Path $parameterPath `
| Where-Object { $_.Name -like "*$substringToMatch*" } `
| ForEach-Object { Write-Output $_.Value }
$filteredParameters
```

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

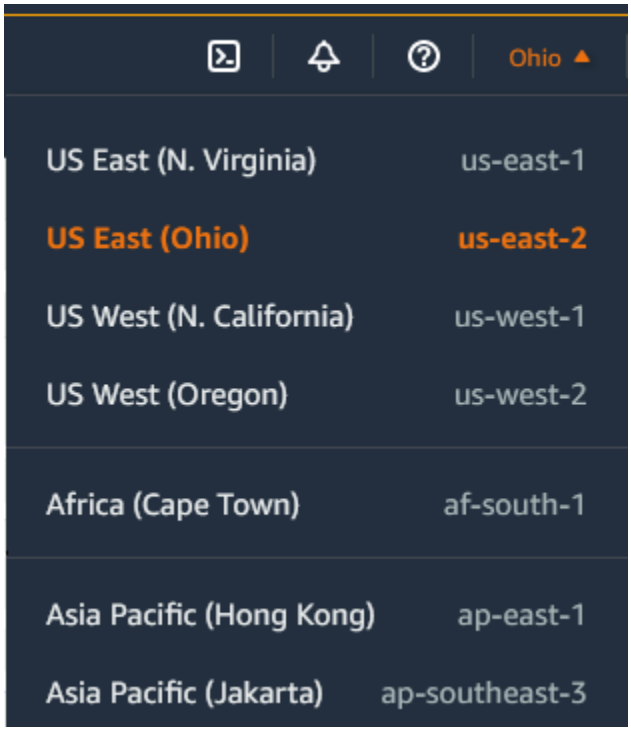
Every time you create an Amazon EC2 resource, you can specify the Region for the resource. You can specify the Region for a resource using the AWS Management Console or the command line.

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`.

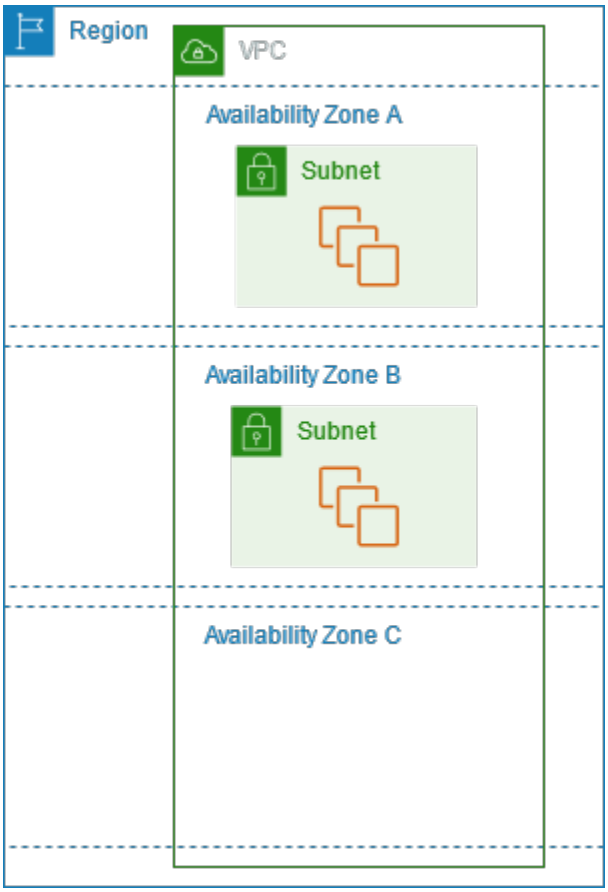
For more information about the endpoints for Amazon EC2, see Amazon EC2 endpoints and quotas in the *AWS General Reference*.

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

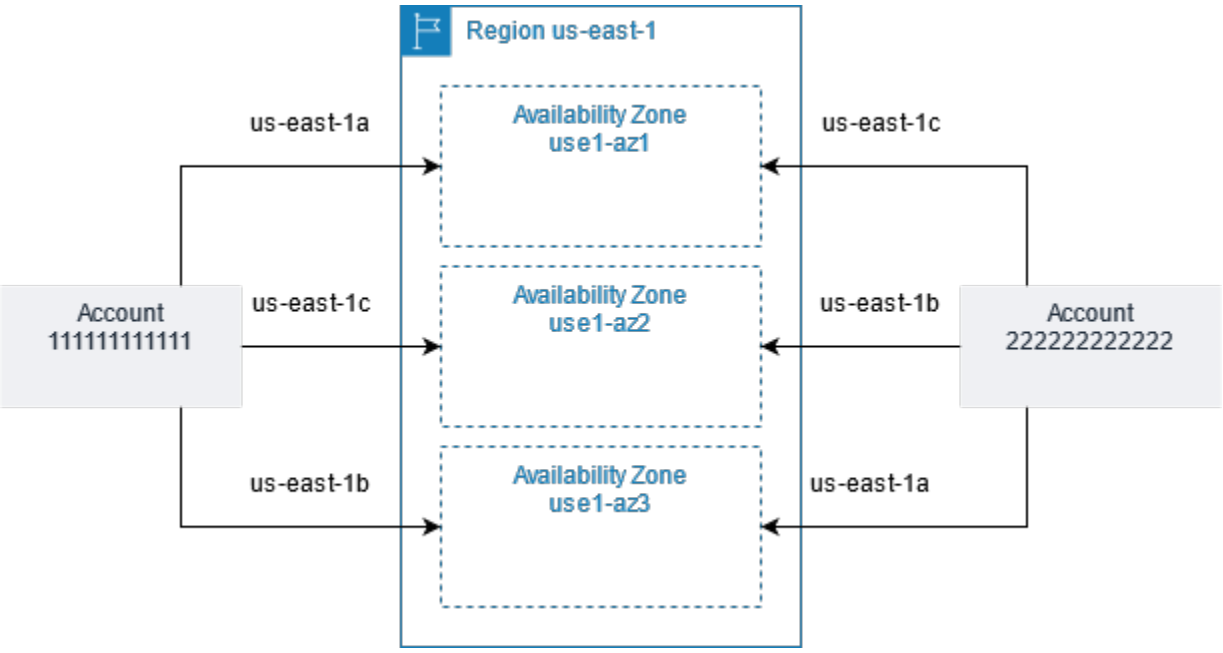
- AZ IDs
- Describe your Availability Zones
- Launch instances in an Availability Zone
- Migrate an instance to another Availability Zone

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. For example, the Availability Zone `us-east-1a` for your AWS account might not be the same physical location as `us-east-1a` for another AWS account.

To coordinate Availability Zones across accounts, you must use the *AZ ID*, which is a unique and consistent identifier for an Availability Zone. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it has the same 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

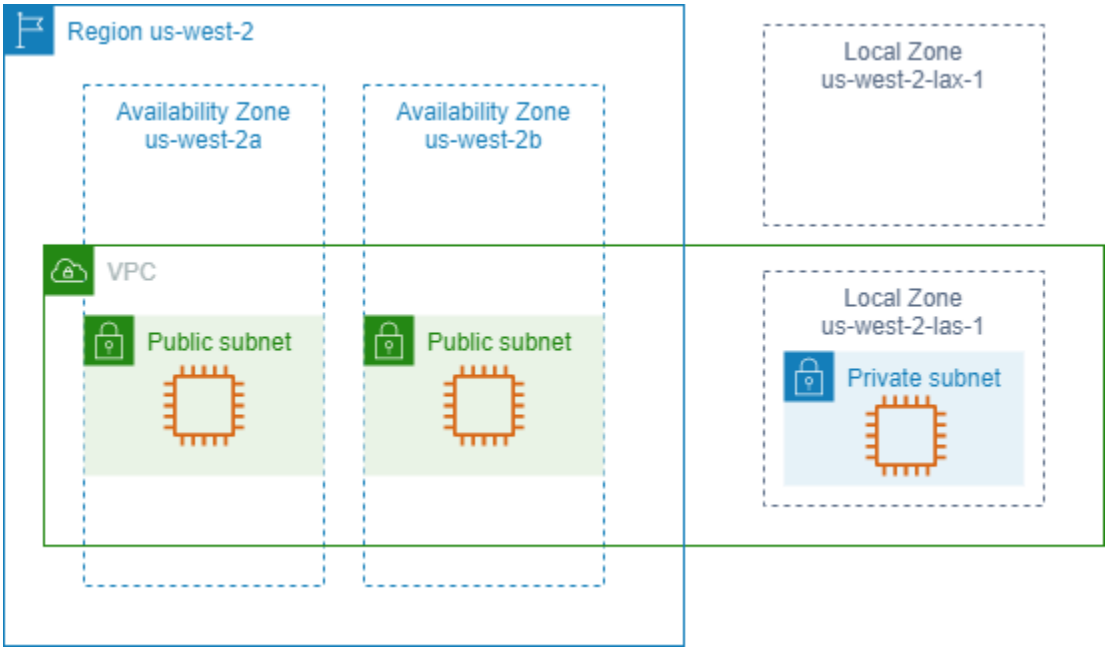
1. Create an AMI from the instance. The procedure depends on your operating system and the type of root device volume for the instance. For more information, see the documentation that corresponds to your operating system and root device volume:
   - Create an Amazon EBS-backed Linux AMI
   - Create 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 AWS Local Zones.

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. For the Local Zone to enable, choose **Manage**.
7. For **Zone group**, choose **Enabled**.
8. Choose **Update zone group**.

To opt in to Local Zones using the AWS CLI

Use the modify-availability-zone-group command.

Launch instances in a Local Zone

When you launch an instance, you can specify a subnet that is in a Local Zone. You also allocate an IP address from a network border group. A network border group is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, `us-west-2-lax-1a`.

You can allocate the following IP addresses from a network border group:

- Amazon-provided Elastic IPv4 addresses
- Amazon-provided IPv6 VPC addresses (available only in the Los Angeles zones)

For more information about how to launch an instance in a Local Zone, see Getting started with AWS Local Zones in the *AWS Local Zones User Guide*.

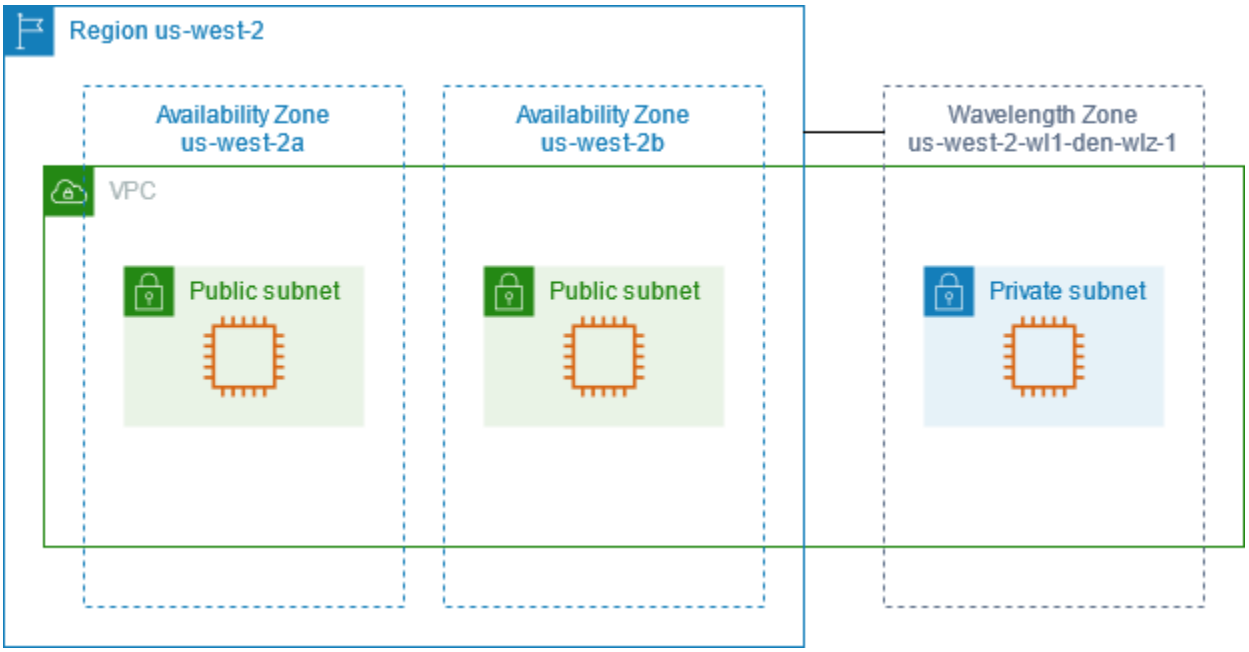
Wavelength Zones

AWS Wavelength enables developers to build applications that deliver ultra-low latencies to mobile devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of telecommunication carriers' 5G networks. Developers can extend a virtual private cloud (VPC) to one or more Wavelength Zones, and then use AWS resources like Amazon EC2 instances to run applications that require ultra-low latency and a connection to AWS services in the Region.

A Wavelength Zone is an isolated zone in the carrier location where the Wavelength infrastructure is deployed. Wavelength Zones are tied to a Region. A Wavelength Zone is a logical extension of a Region, and is managed by the control plane in the Region.

The code for a Wavelength Zone is its Region code followed by an identifier that indicates the physical location. For example, `us-east-1-wl1-bos-wlz-1` in Boston.

The following diagram illustrates the AWS Region `us-west-2`, two of its Availability Zones, and a Wavelength Zone. The VPC spans the Availability Zones and the Wavelength Zone. Each zone in the VPC has one subnet, and each subnet has an instance.



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. Under **Wavelength Zones**, choose **Manage** for the Wavelength Zone.
7. Choose **Enable**.
8. Choose **Update zone group**.

To enable Wavelength Zones using the AWS CLI

Use the modify-availability-zone-group command.

Launch instances in a Wavelength Zone

When you launch an instance, you can specify a subnet which is in a Wavelength Zone. You also allocate a carrier IP address from a network border group, which is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, `us-east-1-wl1-bos-wlz-1`.

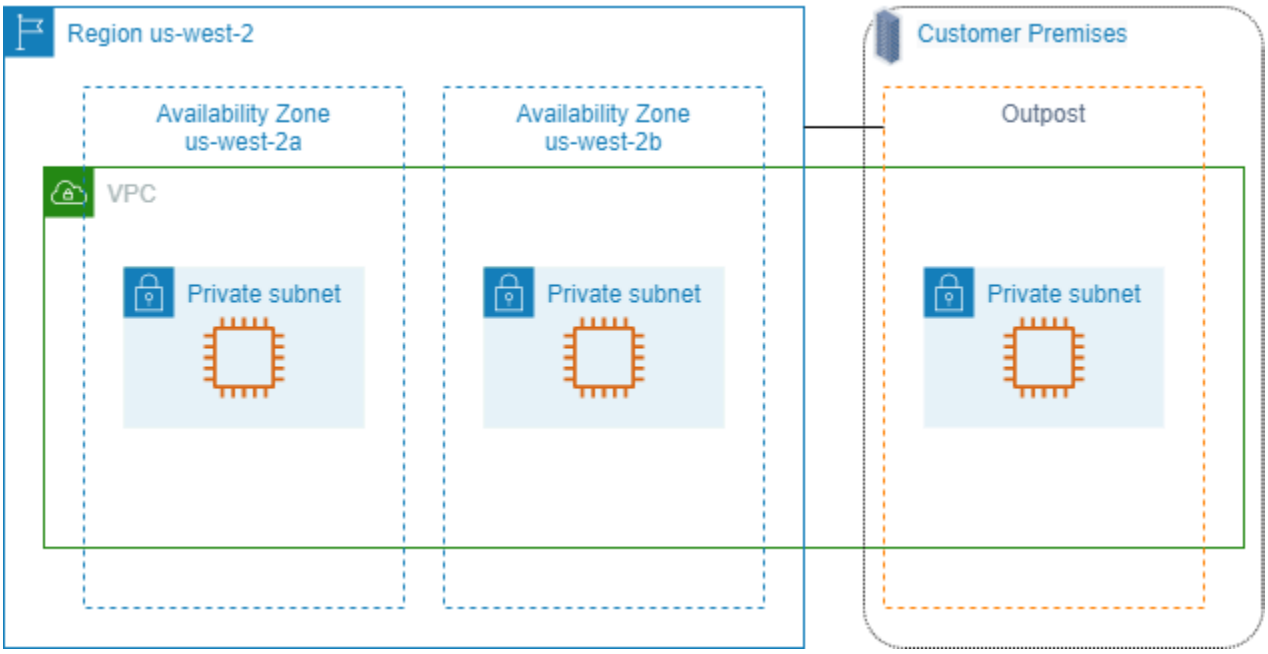
For information about how to launch an instance in a Wavelength Zone, see Get started with AWS Wavelength in the *AWS Wavelength Developer Guide*.

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.

A public IP address is assigned to your instance from Amazon's pool of public IPv4 addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IPv4 address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP (IPv4) address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release your instance's public IP address when it is stopped, hibernated, or terminated. Your stopped or hibernated instance receives a new public IP address when it is started.
- We release your instance's public IP address when you associate an Elastic IP address with it. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.
- If your instance's public IP address is released while it has a secondary private IP address that is associated with an Elastic IP address, the instance does not receive a new public IP address.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see Elastic IP addresses.

> **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 Windows instance.
2. Use the following command to access the private IP address:

```
PS C:\> Invoke-RestMethod http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

```
PS C:\> Invoke-RestMethod http://169.254.169.254/latest/meta-data/public-ipv4
```

   If an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Assign a public IPv4 address during instance launch

Each subnet has an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IPv4 addressing feature is also available for you to control whether your instance is assigned a public IPv4 address; you can override the default behavior of the subnet's IP addressing attribute. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

Considerations

- You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see Public IPv4 addresses. If you require a persistent public IP address that you can associate or

disassociate at will, assign an Elastic IP address to the instance after launch instead. For more information, see Elastic IP addresses.

- You cannot auto-assign a public IP address if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IP feature if you specify an existing network interface for eth0.
- The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see Elastic IP addresses. 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 Windows instance.
2. Use the following command to view the IPv6 address (you can get the MAC address from `http://169.254.169.254/latest/meta-data/network/interfaces/macs/`).

```
PS C:\> Invoke-RestMethod http://169.254.169.254/latest/meta-data/network/
interfaces/macs/mac-address/ipv6s
```

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.

For information about configuring a Windows instance, see Configure a secondary private IPv4 address for your Windows instance.

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 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 Linux instances, switch to this page in the *Amazon EC2 User Guide for Linux 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.

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

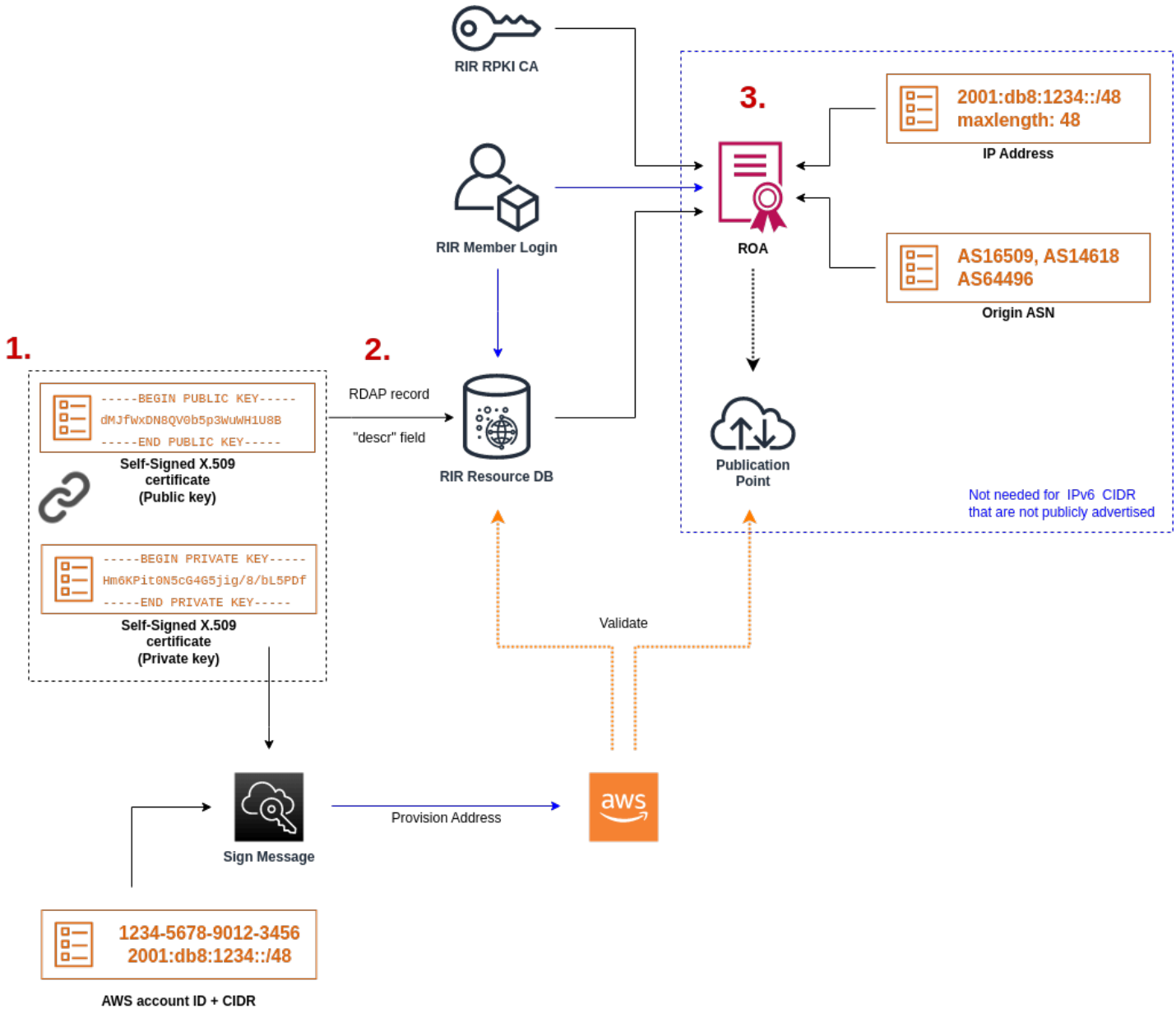
1. Create a private key and use it to generate a self-signed X.509 certificate for authentication purposes. This certificate is only used during the provisioning phase.

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-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDFBXHRI4HVKAhh
3seiciooizCRTbJe1+YsxNTja4XyKypVGIFWDGhZs44FCHlPOOSVJ+NqP74w96oM
7DPS3xo9kaQyZBFn2YEp2EBq5vf307KHNRmZZUmkn0zHOSEpNmY2fMxISBxewlxR
FAniwmSd/8TDvHJMY9FvAIvWuTsv5l0tJKk+a91K4+tO3UdDR7Sno5WXExfsBrW3
g1ydo3TBsx8i5/YiVOcNApy7ge2/FiwY3aCXJB6r6nuF6H8mRgI4r4vkMRsOlAhJ
DnZPNeweboo+K3Q3lwbgbmOKD/z9svk8N/+hUTBtIX0fRtbG+PLIw3xWRHGrMSn2
BzsPVuDLAgMBAAECggEACiJUj2hfJkKv47Dc3es3Zex67A5uDVjXmxfox2Xhdupn
fAcNqAptV6fXt0SPUNbhUxbBKNbshoJGufFwXPli1SXnpzvkdU4Hyco4zgbhXFsE
RNYjYfOGzTPwdBLpNMB6k3Tp4RHse6dNrlH0jDhpioL8cQEBdBJyVF5X0wymEbmV
mC0jgH/MxsBAPWW6ZKicg9ULMlWiAZ3MRAZPjHHgpYkAAsUWKAbCBwVQcVjGO59W
jfZjzTX5pQtVVH68ruciH88DTZCwjCkjBhxg+OIkJBLE5wkh82jIHSivZ63flwLw
z+E0+HhELSZJrn2MY6Jxmik3qNNUOF/Z+3msdj2luQKBgQDjwlC/3jxp8zJy6P8o
JQKv7TdvMwUj4VSWOHZBHLv4evJaaia0uQjIo1UDa8AYitqhX1NmCCehGH8yuXj/
v6V3CzMKDkmRr1NrONnSz5QsndQ04Z6ihAQlPmJ96g4wKtgoC7AYpyP0g1a+4/sj
b1+o3YQI4pD/F71c+qaztH7PRwKBgQDdc23yNmT3+Jyptf0fKjEvONK+xwUKzi9c
L/OzBq5yOIC1Pz2T85gOe1i8kwZws+xlpG6uBT6lmIJELd0k59FyupNu4dPvX5SD
6GGqdx4jk9KvI74usGeOBohmF0phTHkrWKBxXiyT0oS8zjnJlEn8ysIpGgO28jjr
LpaHNZ/MXQKBgQDfLNcnS0LzpsS2aK0tzyZU8SMyqVHOGMxj7quhneBq2T6FbiLD
T9TVlYaGNZ0j71vQaLI19qOubWymbautH0Op5KV8owdf4+bf1/NJaPIOzhDUSIjD
Qo01WW31Z9XDSRhKFTnWzmCjBdeIcajyzf10YKsycaAW9lItu8aBrMndnQKBgQDb
nNp/JyRwqjOrNljk7DHEs+SD39kHQzzCfqd+dnTPv2sc06+cpym3yulQcbokULpy
fmRo3bin/pvJQ3aZX/Bdh9woTXqhXDdrrSwWInVYMQPyPk8f/D9mIOJp5FUWMwHD
U+whIZSxsEeE+jtixlWtheKRYkQmzQZXbWdIhYyI3QKBgD+F/6wcZ85QW8nAUykA
3WrSIx/3cwDGdm4NRGct8ZOZjTHjiy9ojMOD1L7iMhRQ/3k3hUsin5LDMp/ryWGG
x4uIaLat40kiC7T4I66DM7P59euqdz3w0PD+VU+h7GSivvsFDdySUt7bNK0AUVLh
dMJfWxDN8QV0b5p3WuWH1U8B
-----END PRIVATE KEY-----
Private-Key: (2048 bit)
modulus:
    00:c5:05:71:d1:23:81:d5:28:08:61:de:c7:a2:72:
    2a:28:8b:30:91:4d:b2:5e:d7:e6:2c:c4:d4:e3:6b:
    85:f2:2b:2a:55:18:81:56:0c:68:59:b3:8e:05:08:
    79:4f:38:e4:95:27:e3:6a:3f:be:30:f7:aa:0c:ec:
    33:d2:df:1a:3d:91:a4:32:64:11:67:d9:81:29:d8:
    40:6a:e6:f7:f7:d3:b2:87:35:19:99:65:49:a4:9f:
    4c:c7:39:21:29:36:66:36:7c:cc:48:48:1c:5e:c2:
    5c:51:14:09:e2:c2:64:9d:ff:c4:c3:bc:72:4c:63:
    d1:6f:00:8b:d6:b9:3b:2f:e6:5d:2d:24:a9:3e:6b:
    dd:4a:e3:eb:4e:dd:47:43:47:b4:a7:a3:95:97:13:
    17:ec:06:b5:b7:83:5c:9d:a3:74:c1:b3:1f:22:e7:
    f6:22:54:e7:0d:02:9c:bb:81:ed:bf:16:2c:18:dd:
    a0:97:24:1e:ab:ea:7b:85:e8:7f:26:46:02:38:af:
    8b:e4:31:1b:0e:94:08:49:0e:76:4f:35:ec:1e:6e:
```

```
    8a:3e:2b:74:37:97:06:e0:6e:63:8a:0f:fc:fd:b2:
    f9:3c:37:ff:a1:51:30:6d:21:7d:1f:46:d6:c6:f8:
    f2:c8:c3:7c:56:44:71:ab:31:29:f6:07:3b:0f:56:
    e0:cb
publicExponent: 65537 (0x10001)
privateExponent:
    0a:22:54:8f:68:5f:26:42:af:e3:b0:dc:dd:eb:37:
    65:ec:7a:ec:0e:6e:0d:58:d7:9b:17:e8:c7:65:e1:
    76:ea:67:7c:07:0d:a8:0a:6d:57:a7:d7:b7:44:8f:
    50:d6:e1:53:16:c1:28:d6:ec:86:82:46:b9:f1:70:
    5c:f9:62:d5:25:e7:a7:3b:e4:75:4e:07:c9:ca:38:
    ce:06:e1:5c:5b:04:44:d6:23:61:f3:86:cd:33:f0:
    74:12:e9:34:c0:7a:93:74:e9:e1:11:ec:7b:a7:4d:
    ae:51:f4:8c:38:69:8a:82:fc:71:01:01:74:12:72:
    54:5e:57:d3:0c:a6:11:b9:95:98:2d:23:80:7f:cc:
    c6:c0:40:3d:65:ba:64:a8:9c:83:d5:0b:32:55:a2:
    01:9d:cc:44:06:4f:8c:71:e0:a5:89:00:02:c5:16:
    28:06:c2:07:05:50:71:58:c6:3b:9f:56:8d:f6:63:
    cd:35:f9:a5:0b:55:54:7e:bc:ae:e7:22:1f:cf:03:
    4d:90:b0:8c:29:23:06:1c:60:f8:e2:24:24:12:c4:
    e7:09:21:f3:68:c8:1d:28:af:67:ad:df:97:02:f0:
    cf:e1:34:f8:78:44:2d:26:49:ae:7d:8c:63:a2:71:
    9a:29:37:a8:d3:54:38:5f:d9:fb:79:ac:76:3d:a5:
    b9
prime1:
    00:e3:c2:50:bf:de:3c:69:f3:32:72:e8:ff:28:25:
    02:af:ed:37:6f:33:05:23:e1:54:96:38:76:41:1c:
    bb:f8:7a:f2:5a:6a:26:b4:b9:08:c8:a3:55:03:6b:
    c0:18:8a:da:a1:5f:53:66:08:27:a1:18:7f:32:b9:
    78:ff:bf:a5:77:0b:33:0a:0e:49:91:af:53:6b:38:
    d9:d2:cf:94:2c:9d:d4:34:e1:9e:a2:84:04:25:3e:
    62:7d:ea:0e:30:2a:d8:28:0b:b0:18:a7:23:f4:83:
    56:be:e3:fb:23:6f:5f:a8:dd:84:08:e2:90:ff:17:
    bd:5c:fa:a6:b3:b4:7e:cf:47
prime2:
    00:dd:73:6d:f2:36:64:f7:f8:9c:a9:b5:fd:1f:2a:
    31:2f:38:d2:be:c7:05:0a:ce:2f:5c:2f:f3:b3:06:
    ae:72:38:80:b5:3f:3d:93:f3:98:0e:7b:58:bc:93:
    06:70:b3:ec:65:a4:6e:ae:05:3e:a5:98:82:44:2d:
    dd:24:e7:d1:72:ba:93:6e:e1:d3:ef:5f:94:83:e8:
    61:aa:77:1e:23:93:d2:af:23:be:2e:b0:67:8e:06:
    88:66:17:4a:61:4c:79:2b:58:a0:71:5e:2c:93:d2:
    84:bc:ce:39:c9:94:49:fc:ca:c2:29:1a:03:b6:f2:
    38:eb:2e:96:87:35:9f:cc:5d
```

```
exponent1:
    00:df:2c:d7:27:4b:42:f3:a6:c4:b6:68:ad:2d:cf:
    26:54:f1:23:32:a9:51:ce:18:cc:63:ee:ab:a1:9d:
    e0:6a:d9:3e:85:6e:22:c3:4f:d4:d5:95:86:86:35:
    9d:23:ef:5b:d0:68:b2:35:f6:a3:ae:6d:6c:a6:6d:
    ab:ad:1f:43:a9:e4:a5:7c:a3:07:5f:e3:e6:df:d7:
    f3:49:68:f2:0e:ce:10:d4:48:88:c3:42:8d:35:59:
    6d:f5:67:d5:c3:49:18:4a:15:39:d6:ce:60:a3:05:
    d7:88:71:a8:f2:cd:fd:74:60:ab:32:71:a0:16:f6:
    52:2d:bb:c6:81:ac:c9:dd:9d
exponent2:
    00:db:9c:da:7f:27:24:70:aa:33:ab:36:58:e4:ec:
    31:c4:b3:e4:83:df:d9:07:43:3c:c2:7e:a7:7e:76:
    74:cf:bf:6b:1c:d3:af:9c:a7:29:b7:ca:e9:50:71:
    ba:24:50:ba:72:7e:64:68:dd:b8:a7:fe:9b:c9:43:
    76:99:5f:f0:5d:87:dc:28:4d:7a:a1:5c:37:6b:ad:
    2c:16:22:75:58:31:03:f2:3e:4f:1f:fc:3f:66:20:
    e2:69:e4:55:16:33:01:c3:53:ec:21:21:94:b1:b0:
    47:84:fa:3b:62:c6:55:ad:85:e2:91:62:44:26:cd:
    06:57:6d:67:48:85:8c:88:dd
coefficient:
    3f:85:ff:ac:1c:67:ce:50:5b:c9:c0:53:29:00:dd:
    6a:d2:23:1f:f7:73:00:c6:76:6e:0d:44:67:2d:f1:
    93:99:8d:31:e3:8b:2f:68:8c:c3:83:d4:be:e2:32:
    14:50:ff:79:37:85:4b:22:9f:92:c3:32:9f:eb:c9:
    61:86:c7:8b:88:68:b6:ad:e3:49:22:0b:b4:f8:23:
    ae:83:33:b3:f9:f5:eb:aa:77:3d:f0:d0:f0:fe:55:
    4f:a1:ec:64:a2:be:fb:05:0d:dc:92:52:de:db:34:
    ad:00:51:52:e1:74:c2:5f:5b:10:cd:f1:05:74:6f:
    9a:77:5a:e5:87:d5:4f:01
```

Keep your private key in a secure location when it is not in use.

2. Generate an X.509 certificate using the private key created in the previous step. In this example, the certificate expires in 365 days, after which time it cannot be trusted. Be sure to set the expiration appropriately. The `tr -d "\n"` command strips newline characters (line breaks) from the output. You need to provide a Common Name when prompted, but the other fields can be left blank.

```
$ openssl req -new -x509 -key private-key.pem -days 365 | tr -d "\n" >
 certificate.pem
```

This results in output similar to the following:

```
Enter pass phrase for private-key.pem: xxxxxxx
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:example.com
Email Address []:
```

> **Note**
>
> The Common Name is not needed for AWS provisioning. It can be any internal or public domain name.

You can inspect the certificate with the following command:

```
$ cat certificate.pem
```

The output should be a long, PEM-encoded string without line breaks, prefaced by `-----BEGIN CERTIFICATE-----` and followed by `-----END CERTIFICATE-----`.

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 not compliant, 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
   wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
   NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
   rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
   VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
   W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
   R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
   prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
   mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
   Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
   xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
   ```

```
3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

For RIPE, use `whois -r -h whois.ripe.net` *`2001:0DB8:7269::/48`* to look up the RDAP record for your address range. Check the `descr` section for the `inetnum` object (network range) in the command output. The certificate should be added as a new `descr` field for the address range.

You can inspect the `descr` containing the certificate using the following command:

```
whois -r -h whois.ripe.net 2001:0DB8:7269::/48 | grep descr | grep BEGIN
```

This returns output with the contents of the key, which should be similar to the following:

```
descr:
-----BEGIN CERTIFICATE-----MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8
RDAHSP+I1TowDQYJKoZIhvcNAQELBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAg
MCEF1Y2tsYW5kMREwDwYDVQQHDAhBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIF
dlYiBTZXJ2aWNlczETMBEGA1UECwwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPS
VAgRGVtbzAeFw0yMTEyMDcyMDI0NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNV
BAYTAk5aMREwDwYDVQQIDAhBdWNrbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDA
aBgNVBAoME0FtYXpvbiBXZWIgU2VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW
8xEzARBgNVBAMMCkJZT0lQIERlbW8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQCfmacvDp0wZ0ceiXXcR/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanA
EskgAseyFypwEEQr4CJijI/5hp9prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3
stsI5QesHVRwOaXUdprAnndaTugmDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq
35wU/x+wXlAqBXg4MZK2KoUu27kYt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp
1ZnVIc7NqnhdeIW48QaYjhMlUEfxdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2r
GlHWkJsbhr0VEUyAGu1bwkgcdww3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBS
tFyujN6SYBr2glHpGt0XGF7GbGTAfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0
XGF7GbGTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6Y
Lhz521lfyVfxY0t6o3410bQAeAF08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyL
xngwMYN0XY5tVhDQqk4/gmDNEKSZy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9
wySL507XQz76Uk5cFypBOzbnk35UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8
mBGqVpPpey+dXpzzzv1iBKN/VY4ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGC
```

```
vRDl/qdO/GIDJi77dmZWkh/ic90MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoN
PyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

For APNIC, use `whois -h whois.apnic.net` *`2001:0DB8:6170::/48`* to look up the RDAP record for your BYOIP address range. Check the `remarks` section for the `inetnum` object (network range) in the command output. The certificate should be added as a new `remarks` field for the address range.

You can inspect the `remarks` containing the certificate using the following command:

```
whois -h whois.apnic.net 2001:0DB8:6170::/48 | grep remarks | grep BEGIN
```

This returns output with the contents of the key, which should be similar to the following:

```
remarks:
-----BEGIN CERTIFICATE-----
MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

2. Validate the creation of a ROA object

Validate the successful creation of the ROA objects using the RIPEstat Data API. Be sure to test your address range against the Amazon ASNs 16509 and 14618, plus the ASNs that are currently authorized to advertise the address range.

You can inspect the ROA objects from different Amazon ASNs with your address range by using the following command:

```
curl --location --request GET "https://stat.ripe.net/data/rpki-validation/data.json?
resource=ASN&prefix=CIDR
```

In this example output, the response has a result of `"status": "valid"` for the Amazon ASN 16509. This indicates the ROA object for the address range was created successfully:

```
{
    "messages": [],
    "see_also": [],
    "version": "0.3",
    "data_call_name": "rpki-validation",
    "data_call_status": "supported",
    "cached": false,
    "data": {
        "validating_roas": [
            {
                "origin": "16509",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "valid"
            },
            {
                "origin": "14618",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            },
            {
                "origin": "64496",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            }
        ],
```

```
        "status": "valid",
        "validator": "routinator",
        "resource": "16509",
        "prefix": "2001:0DB8::/32"
    },
    "query_id": "20230224152430-81e6384e-21ba-4a86-852a-31850787105f",
    "process_time": 58,
    "server_id": "app116",
    "build_version": "live.2023.2.1.142",
    "status": "ok",
    "status_code": 200,
    "time": "2023-02-24T15:24:30.773654"
}
```

A status of “`unknown`” indicates the ROA object for the address range has not been created. A status of “`invalid_asn`” indicates that the ROA object for the address range was not created successfully.

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.

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. We impose a small hourly charge for Elastic IP addresses that are not associated with a running instance.
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS host name of the instance changes to match the Elastic IP address.

- We resolve a public DNS host name to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.
- 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
- 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*.

IP addresses per network interface per instance type

The following tables list the maximum number of network interfaces per instance type, and the maximum number of private IPv4 addresses and IPv6 addresses per network interface. The limit for IPv6 addresses is separate from the limit for private IPv4 addresses per network interface. Not all instance types support IPv6 addressing.

Contents

- General purpose
- Compute optimized
- Memory optimized
- Storage optimized
- Accelerated computing
- High-performance computing

General purpose

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
`m1.small`	2	4	IPv6 not supported
`m1.medium`	2	6	IPv6 not supported
`m1.large`	3	10	IPv6 not supported

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m1.xlarge	4	15	IPv6 not supported
m2.xlarge	4	15	IPv6 not supported
m2.2xlarge	4	30	IPv6 not supported
m2.4xlarge	8	30	IPv6 not supported
m3.medium	2	6	IPv6 not supported
m3.large	3	10	IPv6 not supported
m3.xlarge	4	15	IPv6 not supported
m3.2xlarge	4	30	IPv6 not supported
m4.large	2	10	10
m4.xlarge	4	15	15
m4.2xlarge	4	15	15
m4.4xlarge	8	30	30
m4.10xlarge	8	30	30
m4.16xlarge	8	30	30
m5.large	3	10	10
m5.xlarge	4	15	15
m5.2xlarge	4	15	15
m5.4xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m5.8xlarge	8	30	30
m5.12xlarge	8	30	30
m5.16xlarge	15	50	50
m5.24xlarge	15	50	50
m5.metal	15	50	50
m5a.large	3	10	10
m5a.xlarge	4	15	15
m5a.2xlarge	4	15	15
m5a.4xlarge	8	30	30
m5a.8xlarge	8	30	30
m5a.12xlarge	8	30	30
m5a.16xlarge	15	50	50
m5a.24xlarge	15	50	50
m5ad.large	3	10	10

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m5ad.xlarge	4	15	15
m5ad.2xlarge	4	15	15
m5ad.4xlarge	8	30	30
m5ad.8xlarge	8	30	30
m5ad.12xlarge	8	30	30
m5ad.16xlarge	15	50	50
m5ad.24xlarge	15	50	50
m5d.large	3	10	10
m5d.xlarge	4	15	15
m5d.2xlarge	4	15	15
m5d.4xlarge	8	30	30
m5d.8xlarge	8	30	30
m5d.12xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m5d.16xlarge	15	50	50
m5d.24xlarge	15	50	50
m5d.metal	15	50	50
m5dn.large	3	10	10
m5dn.xlarge	4	15	15
m5dn.2xlarge	4	15	15
m5dn.4xlarge	8	30	30
m5dn.8xlarge	8	30	30
m5dn.12xlarge	8	30	30
m5dn.16xlarge	15	50	50
m5dn.24xlarge	15	50	50
m5dn.metal	15	50	50
m5n.large	3	10	10
m5n.xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m5n.2xlarge	4	15	15
m5n.4xlarge	8	30	30
m5n.8xlarge	8	30	30
m5n.12xlarge	8	30	30
m5n.16xlarge	15	50	50
m5n.24xlarge	15	50	50
m5n.metal	15	50	50
m5zn.large	3	10	10
m5zn.xlarge	4	15	15
m5zn.2xlarge	4	15	15
m5zn.3xlarge	8	30	30
m5zn.6xlarge	8	30	30
m5zn.12xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m5zn.metal	15	50	50
m6a.large	3	10	10
m6a.xlarge	4	15	15
m6a.2xlarge	4	15	15
m6a.4xlarge	8	30	30
m6a.8xlarge	8	30	30
m6a.12xlarge	8	30	30
m6a.16xlarge	15	50	50
m6a.24xlarge	15	50	50
m6a.32xlarge	15	50	50
m6a.48xlarge	15	50	50
m6a.metal	15	50	50
m6i.large	3	10	10
m6i.xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m6i.2xlarge	4	15	15
m6i.4xlarge	8	30	30
m6i.8xlarge	8	30	30
m6i.12xlarge	8	30	30
m6i.16xlarge	15	50	50
m6i.24xlarge	15	50	50
m6i.32xlarge	15	50	50
m6i.metal	15	50	50
m6id.large	3	10	10
m6id.xlarge	4	15	15
m6id.2xlarge	4	15	15
m6id.4xlarge	8	30	30
m6id.8xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m6id.12xlarge	8	30	30
m6id.16xlarge	15	50	50
m6id.24xlarge	15	50	50
m6id.32xlarge	15	50	50
m6id.metal	15	50	50
m6idn.large	3	10	10
m6idn.xlarge	4	15	15
m6idn.2xlarge	4	15	15
m6idn.4xlarge	8	30	30
m6idn.8xlarge	8	30	30
m6idn.12xlarge	8	30	30
m6idn.16xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m6idn.24xlarge	15	50	50
m6idn.32xlarge	14	50	50
m6idn.metal	14	50	50
m6in.large	3	10	10
m6in.xlarge	4	15	15
m6in.2xlarge	4	15	15
m6in.4xlarge	8	30	30
m6in.8xlarge	8	30	30
m6in.12xlarge	8	30	30
m6in.16xlarge	15	50	50
m6in.24xlarge	15	50	50
m6in.32xlarge	14	50	50
m6in.metal	14	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m7a.medium	2	4	4
m7a.large	3	10	10
m7a.xlarge	4	15	15
m7a.2xlarge	4	15	15
m7a.4xlarge	8	30	30
m7a.8xlarge	8	30	30
m7a.12xlarge	8	30	30
m7a.16xlarge	15	50	50
m7a.24xlarge	15	50	50
m7a.32xlarge	15	50	50
m7a.48xlarge	15	50	50
m7a.metal-48xl	15	50	50
m7i.large	3	10	10
m7i.xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m7i.2xlarge	4	15	15
m7i.4xlarge	8	30	30
m7i.8xlarge	8	30	30
m7i.12xlarge	8	30	30
m7i.16xlarge	15	50	50
m7i.24xlarge	15	50	50
m7i.48xlarge	15	50	50
m7i.metal-24xl	15	50	50
m7i.metal-48xl	15	50	50
m7i-flex.large	3	10	10
m7i-flex.xlarge	4	15	15
m7i-flex.2xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m7i-flex.4xlarge	8	30	30
m7i-flex.8xlarge	8	30	30
t1.micro	2	2	IPv6 not supported
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	3	4	4
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15
t2.2xlarge	3	15	15
t3.nano	2	2	2
t3.micro	2	2	2
t3.small	3	4	4
t3.medium	3	6	6
t3.large	3	12	12
t3.xlarge	4	15	15
t3.2xlarge	4	15	15
t3a.nano	2	2	2

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
t3a.micro	2	2	2
t3a.small	2	4	4
t3a.medium	3	6	6
t3a.large	3	12	12
t3a.xlarge	4	15	15
t3a.2xlarge	4	15	15

Compute optimized

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c1.medium	2	6	IPv6 not supported
c1.xlarge	4	15	IPv6 not supported
c3.large	3	10	10
c3.xlarge	4	15	15
c3.2xlarge	4	15	15
c3.4xlarge	8	30	30
c3.8xlarge	8	30	30
c4.large	3	10	10
c4.xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c4.2xlarge	4	15	15
c4.4xlarge	8	30	30
c4.8xlarge	8	30	30
c5.large	3	10	10
c5.xlarge	4	15	15
c5.2xlarge	4	15	15
c5.4xlarge	8	30	30
c5.9xlarge	8	30	30
c5.12xlarge	8	30	30
c5.18xlarge	15	50	50
c5.24xlarge	15	50	50
c5.metal	15	50	50
c5a.large	3	10	10
c5a.xlarge	4	15	15
c5a.2xlarge	4	15	15
c5a.4xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c5a.8xlarge	8	30	30
c5a.12xlarge	8	30	30
c5a.16xlarge	15	50	50
c5a.24xlarge	15	50	50
c5ad.large	3	10	10
c5ad.xlarge	4	15	15
c5ad.2xlarge	4	15	15
c5ad.4xlarge	8	30	30
c5ad.8xlarge	8	30	30
c5ad.12xlarge	8	30	30
c5ad.16xlarge	15	50	50
c5ad.24xlarge	15	50	50
c5d.large	3	10	10

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c5d.xlarge	4	15	15
c5d.2xlarge	4	15	15
c5d.4xlarge	8	30	30
c5d.9xlarge	8	30	30
c5d.12xlarge	8	30	30
c5d.18xlarge	15	50	50
c5d.24xlarge	15	50	50
c5d.metal	15	50	50
c5n.large	3	10	10
c5n.xlarge	4	15	15
c5n.2xlarge	4	15	15
c5n.4xlarge	8	30	30
c5n.9xlarge	8	30	30
c5n.18xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c5n.metal	15	50	50
c6a.large	3	10	10
c6a.xlarge	4	15	15
c6a.2xlarge	4	15	15
c6a.4xlarge	8	30	30
c6a.8xlarge	8	30	30
c6a.12xlarge	8	30	30
c6a.16xlarge	15	50	50
c6a.24xlarge	15	50	50
c6a.32xlarge	15	50	50
c6a.48xlarge	15	50	50
c6a.metal	15	50	50
c6i.large	3	10	10
c6i.xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c6i.2xlarge	4	15	15
c6i.4xlarge	8	30	30
c6i.8xlarge	8	30	30
c6i.12xlarge	8	30	30
c6i.16xlarge	15	50	50
c6i.24xlarge	15	50	50
c6i.32xlarge	15	50	50
c6i.metal	15	50	50
c6id.large	3	10	10
c6id.xlarge	4	15	15
c6id.2xlarge	4	15	15
c6id.4xlarge	8	30	30
c6id.8xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c6id.12xlarge	8	30	30
c6id.16xlarge	15	50	50
c6id.24xlarge	15	50	50
c6id.32xlarge	15	50	50
c6id.metal	15	50	50
c6in.large	3	10	10
c6in.xlarge	4	15	15
c6in.2xlarge	4	15	15
c6in.4xlarge	8	30	30
c6in.8xlarge	8	30	30
c6in.12xlarge	8	30	30
c6in.16xlarge	15	50	50
c6in.24xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c6in.32xlarge	14	50	50
c6in.metal	14	50	50
c7a.medium	2	4	4
c7a.large	3	10	10
c7a.xlarge	4	15	15
c7a.2xlarge	4	15	15
c7a.4xlarge	8	30	30
c7a.8xlarge	8	30	30
c7a.12xlarge	8	30	30
c7a.16xlarge	15	50	50
c7a.24xlarge	15	50	50
c7a.32xlarge	15	50	50
c7a.48xlarge	15	50	50
c7a.metal-48xl	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c7i.large	3	10	10
c7i.xlarge	4	15	15
c7i.2xlarge	4	15	15
c7i.4xlarge	8	30	30
c7i.8xlarge	8	30	30
c7i.12xlarge	8	30	30
c7i.16xlarge	15	50	50
c7i.24xlarge	15	50	50
c7i.48xlarge	15	50	50
c7i.metal-24xl	15	50	50
c7i.metal-48xl	15	50	50

Memory optimized

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r3.large	3	10	10
r3.xlarge	4	15	15
r3.2xlarge	4	15	15
r3.4xlarge	8	30	30
r3.8xlarge	8	30	30
r4.large	3	10	10
r4.xlarge	4	15	15
r4.2xlarge	4	15	15
r4.4xlarge	8	30	30
r4.8xlarge	8	30	30
r4.16xlarge	15	50	50
r5.large	3	10	10
r5.xlarge	4	15	15
r5.2xlarge	4	15	15
r5.4xlarge	8	30	30
r5.8xlarge	8	30	30
r5.12xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r5.16xlarge	15	50	50
r5.24xlarge	15	50	50
r5.metal	15	50	50
r5a.large	3	10	10
r5a.xlarge	4	15	15
r5a.2xlarge	4	15	15
r5a.4xlarge	8	30	30
r5a.8xlarge	8	30	30
r5a.12xlarge	8	30	30
r5a.16xlarge	15	50	50
r5a.24xlarge	15	50	50
r5ad.large	3	10	10
r5ad.xlarge	4	15	15
r5ad.2xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r5ad.4xlarge	8	30	30
r5ad.8xlarge	8	30	30
r5ad.12xlarge	8	30	30
r5ad.16xlarge	15	50	50
r5ad.24xlarge	15	50	50
r5b.large	3	10	10
r5b.xlarge	4	15	15
r5b.2xlarge	4	15	15
r5b.4xlarge	8	30	30
r5b.8xlarge	8	30	30
r5b.12xlarge	8	30	30
r5b.16xlarge	15	50	50
r5b.24xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r5b.metal	15	50	50
r5d.large	3	10	10
r5d.xlarge	4	15	15
r5d.2xlarge	4	15	15
r5d.4xlarge	8	30	30
r5d.8xlarge	8	30	30
r5d.12xlarge	8	30	30
r5d.16xlarge	15	50	50
r5d.24xlarge	15	50	50
r5d.metal	15	50	50
r5dn.large	3	10	10
r5dn.xlarge	4	15	15
r5dn.2xlarge	4	15	15
r5dn.4xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r5dn.8xlarge	8	30	30
r5dn.12xlarge	8	30	30
r5dn.16xlarge	15	50	50
r5dn.24xlarge	15	50	50
r5dn.metal	15	50	50
r5n.large	3	10	10
r5n.xlarge	4	15	15
r5n.2xlarge	4	15	15
r5n.4xlarge	8	30	30
r5n.8xlarge	8	30	30
r5n.12xlarge	8	30	30
r5n.16xlarge	15	50	50
r5n.24xlarge	15	50	50
r5n.metal	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r6a.large	3	10	10
r6a.xlarge	4	15	15
r6a.2xlarge	4	15	15
r6a.4xlarge	8	30	30
r6a.8xlarge	8	30	30
r6a.12xlarge	8	30	30
r6a.16xlarge	15	50	50
r6a.24xlarge	15	50	50
r6a.32xlarge	15	50	50
r6a.48xlarge	15	50	50
r6a.metal	15	50	50
r6i.large	3	10	10
r6i.xlarge	4	15	15
r6i.2xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r6i.4xlarge	8	30	30
r6i.8xlarge	8	30	30
r6i.12xlarge	8	30	30
r6i.16xlarge	15	50	50
r6i.24xlarge	15	50	50
r6i.32xlarge	15	50	50
r6i.metal	15	50	50
r6idn.large	3	10	10
r6idn.xlarge	4	15	15
r6idn.2xlarge	4	15	15
r6idn.4xlarge	8	30	30
r6idn.8xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r6idn.12xlarge	8	30	30
r6idn.16xlarge	15	50	50
r6idn.24xlarge	15	50	50
r6idn.32xlarge	14	50	50
r6idn.metal	14	50	50
r6in.large	3	10	10
r6in.xlarge	4	15	15
r6in.2xlarge	4	15	15
r6in.4xlarge	8	30	30
r6in.8xlarge	8	30	30
r6in.12xlarge	8	30	30
r6in.16xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r6in.24xlarge	15	50	50
r6in.32xlarge	14	50	50
r6in.metal	14	50	50
r6id.large	3	10	10
r6id.xlarge	4	15	15
r6id.2xlarge	4	15	15
r6id.4xlarge	8	30	30
r6id.8xlarge	8	30	30
r6id.12xlarge	8	30	30
r6id.16xlarge	15	50	50
r6id.24xlarge	15	50	50
r6id.32xlarge	15	50	50
r6id.metal	15	50	50
r7a.medium	2	4	4

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r7a.large	3	10	10
r7a.xlarge	4	15	15
r7a.2xlarge	4	15	15
r7a.4xlarge	8	30	30
r7a.8xlarge	8	30	30
r7a.12xlarge	8	30	30
r7a.16xlarge	15	50	50
r7a.24xlarge	15	50	50
r7a.32xlarge	15	50	50
r7a.48xlarge	15	50	50
r7a.metal-48xl	15	50	50
r7i.large	3	10	10
r7i.xlarge	4	15	15
r7i.2xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r7i.4xlarge	8	30	30
r7i.8xlarge	8	30	30
r7i.12xlarge	8	30	30
r7i.16xlarge	15	50	50
r7i.24xlarge	15	50	50
r7i.48xlarge	15	50	50
r7i.metal-24xl	15	50	50
r7i.metal-48xl	15	50	50
r7iz.large	3	10	10
r7iz.xlarge	4	15	15
r7iz.2xlarge	4	15	15
r7iz.4xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r7iz.8xlarge	8	30	30
r7iz.12xlarge	8	30	30
r7iz.16xlarge	15	50	50
r7iz.32xlarge	15	50	50
r7iz.metal-16xl	15	50	50
r7iz.metal-32xl	15	50	50
u-3tb1.56xlarge	8	30	30
u-6tb1.56xlarge	15	50	50
u-6tb1.112xlarge	15	50	50
u-6tb1.metal	5	30	30
u-9tb1.112xlarge	15	50	50
u-9tb1.metal	5	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
u-12tb1.112xlarge	15	50	50
u-12tb1.metal	5	30	30
u-18tb1.112xlarge	15	50	50
u-18tb1.metal	15	50	50
u-24tb1.112xlarge	15	50	50
u-24tb1.metal	15	50	50
x1.16xlarge	8	30	30
x1.32xlarge	8	30	30
x2idn.16xlarge	15	50	50
x2idn.24xlarge	15	50	50
x2idn.32xlarge	15	50	50
x2idn.metal	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
x2iedn.xlarge	4	15	15
x2iedn.2xlarge	4	15	15
x2iedn.4xlarge	8	30	30
x2iedn.8xlarge	8	30	30
x2iedn.16xlarge	15	50	50
x2iedn.24xlarge	15	50	50
x2iedn.32xlarge	15	50	50
x2iedn.metal	15	50	50
x2iezn.2xlarge	4	15	15
x2iezn.4xlarge	8	30	30
x2iezn.6xlarge	8	30	30
x2iezn.8xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
x2iezn.12xlarge	15	50	50
x2iezn.metal	15	50	50
x1e.xlarge	3	10	10
x1e.2xlarge	4	15	15
x1e.4xlarge	4	15	15
x1e.8xlarge	4	15	15
x1e.16xlarge	8	30	30
x1e.32xlarge	8	30	30
z1d.large	3	10	10
z1d.xlarge	4	15	15
z1d.2xlarge	4	15	15
z1d.3xlarge	8	30	30
z1d.6xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
z1d.12xlarge	15	50	50
z1d.metal	15	50	50

Storage optimized

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
d2.xlarge	4	15	15
d2.2xlarge	4	15	15
d2.4xlarge	8	30	30
d2.8xlarge	8	30	30
d3.xlarge	4	3	3
d3.2xlarge	4	5	5
d3.4xlarge	4	10	10
d3.8xlarge	3	20	20
d3en.xlarge	4	3	3
d3en.2xlarge	4	5	5
d3en.4xlarge	4	10	10

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
d3en.6xlarge	4	15	15
d3en.8xlarge	4	20	20
d3en.12xlarge	3	30	30
h1.2xlarge	4	15	15
h1.4xlarge	8	30	30
h1.8xlarge	8	30	30
h1.16xlarge	8	50	50
i2.xlarge	4	15	15
i2.2xlarge	4	15	15
i2.4xlarge	8	30	30
i2.8xlarge	8	30	30
i3.large	3	10	10
i3.xlarge	4	15	15
i3.2xlarge	4	15	15
i3.4xlarge	8	30	30
i3.8xlarge	8	30	30
i3.16xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
i3.metal	15	50	50
i3en.large	3	10	10
i3en.xlarge	4	15	15
i3en.2xlarge	4	15	15
i3en.3xlarge	4	15	15
i3en.6xlarge	8	30	30
i3en.12xlarge	8	30	30
i3en.24xlarge	15	50	50
i3en.metal	15	50	50
i4i.large	3	10	10
i4i.xlarge	4	15	15
i4i.2xlarge	4	15	15
i4i.4xlarge	8	30	30
i4i.8xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
i4i.12xlarge	8	30	30
i4i.16xlarge	15	50	50
i4i.24xlarge	15	30	30
i4i.32xlarge	15	50	50
i4i.metal	15	50	50

Accelerated computing

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
g2.2xlarge	4	15	IPv6 not supported
g2.8xlarge	8	30	IPv6 not supported
g3.4xlarge	8	30	30
g3.8xlarge	8	30	30
g3.16xlarge	15	50	50
g4ad.xlarge	2	4	4
g4ad.2xlarge	2	4	4

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
g4ad.4xlarge	3	10	10
g4ad.8xlarge	4	15	15
g4ad.16xlarge	8	30	30
g4dn.xlarge	3	10	10
g4dn.2xlarge	3	10	10
g4dn.4xlarge	3	10	10
g4dn.8xlarge	4	15	15
g4dn.12xlarge	8	30	30
g4dn.16xlarge	4	15	15
g4dn.metal	15	50	50
g5.xlarge	4	15	15
g5.2xlarge	4	15	15
g5.4xlarge	8	30	30
g5.8xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
g5.12xlarge	15	50	50
g5.16xlarge	8	30	30
g5.24xlarge	15	50	50
g5.48xlarge	7	50	50
p2.xlarge	4	15	15
p2.8xlarge	8	30	30
p2.16xlarge	8	30	30
p3.2xlarge	4	15	15
p3.8xlarge	8	30	30
p3.16xlarge	8	30	30
p3dn.24xlarge	15	50	50

High-performance computing

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
hpc6id.32xlarge	2	50	50
hpc7a.12xlarge	4	50	50
hpc7a.24xlarge	4	50	50
hpc7a.48xlarge	4	50	50
hpc7a.96xlarge	4	50	50

You can use the describe-instance-types AWS CLI command to display information about an instance type, such as the supported network interfaces and IP addresses per interface. The following example displays this information for all 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.

> **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.

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. Select a network interface. If the instance supports multiple network cards, you can choose a network card.
6. Choose **Attach**.

Network Interfaces page

To attach a network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface.
4. Choose **Actions**, **Attach**.
5. Choose an instance. If the instance supports multiple network cards, you can choose a network card.
6. Choose **Attach**.

To attach a network interface to an instance using the command line

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

> **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. For **IPv6 addresses**, modify the IP addresses as needed. To assign an IPv6 address, choose **Assign new IP address** and then specify an IPv6 address from the subnet range or let AWS choose one for you. To unassign an IPv6 address, choose **Unassign** next to the address.
8. (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.
9. 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. If you need to use multiple network interfaces, you must configure the network interfaces to use static routing.

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).

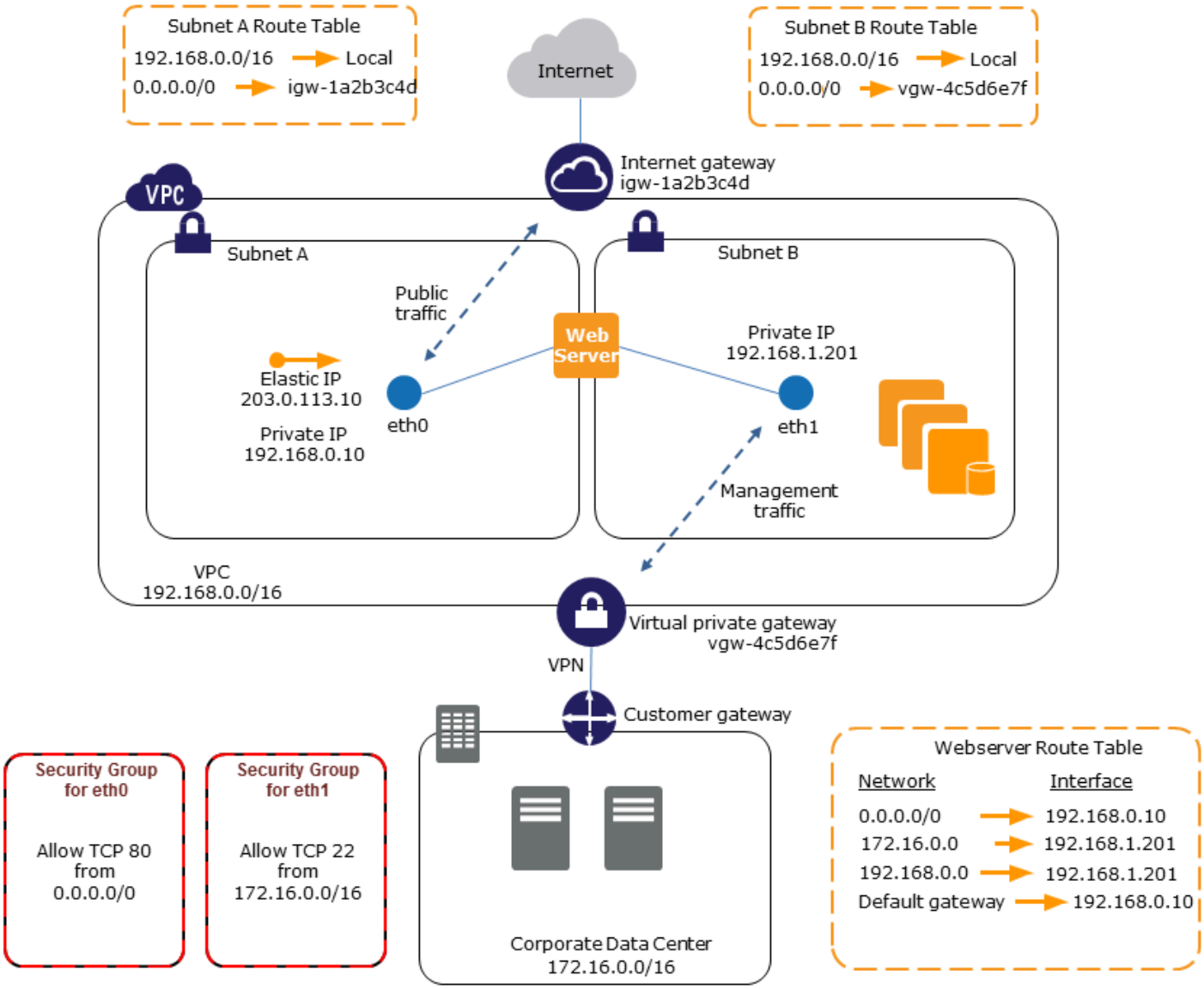
Criteria

- 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 RDP 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 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.

```
C:\> 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": "AIDAIV5AJI5LXF5XXDPCO",
        "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.

```
C:\> aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv6 automatic example" \
--ipv6-prefix-count 1
```

Example output

```
{
    "NetworkInterface": {
        "AvailabilityZone": "us-west-2a",
        "Description": "IPv6 automatic example",
        "Groups": [
```

```
            {
                "GroupName": "default",
                "GroupId": "sg-044c2de2c4EXAMPLE"
            }
        ],
        "InterfaceType": "interface",
        "Ipv6Addresses": [],
        "MacAddress": "02:bb:e4:31:fe:09",
        "NetworkInterfaceId": "eni-006edbcfa4EXAMPLE",
        "OwnerId": "123456789012",
        "PrivateIpAddress": "10.0.0.73",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateIpAddress": "10.0.0.73"
            }
        ],
        "Ipv6Prefixes": [
            {
                "Ipv6Prefix": "2600:1f13:fc2:a700:1768::/80"
            }
        ],
        "RequesterId": "AIDAIV5AJI5LXF5XXDPCO",
        "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`.

```
C:\> 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": "AIDAIV5AJI5LXF5XXDPCO",
            "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`.

```
C:\> 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": "AIDAIV5AJI5LXF5XXDPCO",
            "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.

```
C:\> 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.

```
C:\> 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`.

```
C:\> 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`.

```
C:\> aws ec2 assign-ipv6-addresses \
--network-interface-id eni-00d577338cEXAMPLE \
--ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80
```

Example output

```
{
    "NetworkInterfaceId": "eni-00d577338cEXAMPLE",
    "AssignedIpv6Prefixes": [
        {
            "Ipv6Prefix": "2600:1f13:fc2:a700:18bb::/80"
        }
    ]
}
```

Configure your operating system for network interfaces with prefixes

Amazon Linux AMIs might contain additional scripts installed by AWS, known as `ec2-net-utils`. These scripts optionally automate the configuration of your network interfaces. They are available for Amazon Linux only.

If you are not using Amazon Linux, you can use a Container Network Interface (CNI) for Kubernetes 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.

```
C:\> aws ec2 describe-network-interfaces
```

Example output

```
{
    "NetworkInterfaces": [
        {
            "AvailabilityZone": "us-west-2a",
            "Description": "IPv4 automatic example",
            "Groups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-044c2de2c4EXAMPLE"
                }
            ],
            "InterfaceType": "interface",
            "Ipv6Addresses": [],
            "MacAddress": "02:98:65:dd:18:47",
            "NetworkInterfaceId": "eni-02b80b4668EXAMPLE",
            "OwnerId": "123456789012",
            "PrivateIpAddress": "10.0.0.62",
            "PrivateIpAddresses": [
                {
                    "Primary": true,
                    "PrivateIpAddress": "10.0.0.62"
                }
```

```
            ],
            "Ipv4Prefixes": [
                {
                    "Ipv4Prefix": "10.0.0.208/28"
                }
            ],
            "Ipv6Prefixes": [],
            "RequesterId": "AIDAIV5AJI5LXF5XXDPCO",
            "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": "AIDAIV5AJI5LXF5XXDPCO",
            "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.

```
C:\> 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.

```
C:\> 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 following documentation describes the network performance for all instances, plus the baseline network bandwidth available for instances that can use burst bandwidth.

- General purpose instances
- Compute optimized instances
- Memory optimized instances
- Storage optimized instances
- Accelerated computing instances

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 Windows

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 Windows instances
- Improve network performance with ENA Express on Windows instances
- Enable enhanced networking with the Intel 82599 VF interface on Windows instances
- Operating system optimizations
- Monitor network performance for your EC2 instance
- Troubleshoot the Elastic Network Adapter (ENA) Windows driver

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 instance based on the 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.

For a summary of the enhanced networking mechanisms by instance type, see Summary of networking and storage features.

Enable enhanced networking on your instance

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances.

If your instance type supports the Intel 82599 VF interface for enhanced networking, follow the procedures in Enable enhanced networking with the Intel 82599 VF interface on Windows instances.

Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA). To use enhanced networking, you must install the required ENA module and enable ENA support.

Contents

- Requirements
- Enhanced networking performance
- Test whether enhanced networking is enabled
- Enable enhanced networking on Windows
- Install or upgrade Elastic Network Adapter (ENA) driver
- Amazon ENA driver versions
- Subscribe to notifications

Requirements

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Launch an instance based on the Nitro system.
- If the instance is running Windows Server 2008 R2 SP1, ensure that is has the SHA-2 code signing support update.
- Ensure that the instance has internet connectivity.
- Use AWS CloudShell from the AWS Management Console, or install and configure the AWS CLI or the AWS Tools for Windows PowerShell on any computer you choose, preferably your local desktop or laptop. For more information, see Access Amazon EC2 or the AWS CloudShell User Guide. Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `enaSupport` attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Enhanced networking performance

The following documentation provides a summary of the network performance for the instance types that support ENA enhanced networking:

- Network Performance for Accelerated Computing Instances
- Network Performance for Compute Optimized Instances
- Network Performance for General Purpose Instances

- Network Performance for Memory Optimized Instances
- Network Performance for Storage Optimized Instances

Test whether enhanced networking is enabled

To test whether enhanced networking is already enabled, verify that the driver is installed on your instance and that the `enaSupport` attribute is set.

Instance attribute (enaSupport)

To check whether an instance has the enhanced networking `enaSupport` attribute set, use one of the following commands. If the attribute is set, the response is true.

- describe-instances (AWS CLI/AWS CloudShell)

```
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
```

Enable enhanced networking on Windows

If you launched your instance and it does not have enhanced networking enabled already, you must download and install the required network adapter driver on your instance, and then set the `enaSupport` instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types and only if the ENA driver is installed. For more information, see Enhanced networking support.

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.
2. [Windows Server 2016 and 2019 only] Run the following EC2Launch PowerShell script to configure the instance after the driver is installed.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

3. From the instance, install the driver as follows:
   a. Download the latest driver to the instance.
   b. Extract the zip archive.
   c. Install the driver by running the `install.ps1` PowerShell script.

   > **Note**
   >
   > If you get an execution policy error, set the policy to `Unrestricted` (by default it is set to `Restricted` or `RemoteSigned`). In a command line, run `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`, and then run the `install.ps1` PowerShell script again.

4. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI/AWS CloudShell), Stop-EC2Instance (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.
5. Enable ENA support on your instance as follows:

a. From your local computer, check the EC2 instance ENA support attribute on your instance by running one of the following commands. If the attribute is not enabled, the output will be "[]" or blank. `EnaSupport` is set to `false` by default.

- 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
```

b. To enable ENA support, run one of the following commands:

- modify-instance-attribute (AWS CLI/AWS CloudShell)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

If you encounter problems when you restart the instance, you can also disable ENA support using one of the following commands:

- modify-instance-attribute (AWS CLI/AWS CloudShell)

```
aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $false
```

c. Verify that the attribute has been set to `true` using **describe-instances** or **Get-EC2Instance** as shown previously. You should now see the following output:

```
[
 true
```

```
]
```

6. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: start-instances (AWS CLI/AWS CloudShell), Start-EC2Instance (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance using the AWS OpsWorks console so that the instance state remains in sync.
7. On the instance, validate that the ENA driver is installed and enabled as follows:
   a. Right-click the network icon and choose **Open Network and Sharing Center**.
   b. Choose the Ethernet adapter (for example, **Ethernet 2**).
   c. Choose **Details**. For **Network Connection Details**, check that **Description** is **Amazon Elastic Network Adapter**.
8. (Optional) Create an AMI from the instance. The AMI inherits the `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with ENA enabled by default. For more information, see Create a custom Windows AMI.

Install or upgrade Elastic Network Adapter (ENA) driver

If your instance isn't based on one of the latest Windows Amazon Machine Images (AMIs) that Amazon provides, use the following procedure to install the current ENA driver on your instance. You should perform this update at a time when it’s convenient to reboot your instance. If the install script doesn’t automatically reboot your instance, we recommend that you reboot the instance as the final step.

If you use an instance store volume to store data while the instance is running, that data is erased when you stop the instance. Before you stop your instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

Prerequisites

To install or upgrade the ENA driver, your Windows instance must meet the following prerequisites:

- Have PowerShell version 3.0 or later installed

Step 1: Back up your data

We recommend that you create a backup AMI, in case you're not able to roll back your changes through the **Device Manager**. To create a backup AMI with the AWS Management Console, follow these steps:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires the driver upgrade, and choose **Stop instance** from the **Instance state** menu.
4. After the instance is stopped, select the instance again. To create your backup, choose **Image and templates** from the **Actions** menu, then choose **Create image**.
5. To restart your instance, choose **Start instance** from the **Instance state** menu.

Step 2: Install or upgrade your ENA driver

You can install or upgrade your ENA driver with AWS Systems Manager Distributor, or with PowerShell cmdlets. For further instructions, select the tab that matches the method that you want to use.

Systems Manager Distributor

You can use the Systems Manager Distributor feature to deploy packages to your Systems Manager managed nodes. With Systems Manager Distributor, you can install the ENA driver package once, or with scheduled updates. For more information about how to install the ENA driver package (`AwsEnaNetworkDriver`) with Systems Manager Distributor, see Install or update packages in the *AWS Systems Manager User Guide*.

PowerShell

This section covers how to download and install ENA driver packages on your instance with PowerShell cmdlets.

Option 1: Download and extract the latest version

1. Connect to your instance and log in as the local administrator.
2. Use the **invoke-webrequest** cmdlet to download the latest driver package:

```
PS C:\> invoke-webrequest https://ec2-windows-drivers-
downloads.s3.amazonaws.com/ENA/Latest/AwsEnaNetworkDriver.zip -
outfile $env:USERPROFILE\AwsEnaNetworkDriver.zip
```

> **Note**
>
> Alternatively, you can download the latest driver package from a browser window on your instance.

3. Use the **expand-archive** cmdlet to extract the zip archive that you downloaded to your instance:

```
PS C:\> expand-archive $env:userprofile\AwsEnaNetworkDriver.zip -
DestinationPath $env:userprofile\AwsEnaNetworkDriver
```

Option 2: Download and extract a specific version

1. Connect to your instance and log in as the local administrator.
2. Download the ENA driver package for the specific version you want from the version link in the Amazon ENA driver versions table.
3. Extract the zip archive to your instance.

Install the ENA driver with PowerShell

The install steps are the same whether you've downloaded the latest driver or a specific version. To install the ENA driver, follow these steps.

1. To install the driver, run the `install.ps1` PowerShell script from the `AwsEnaNetworkDriver` directory on your instance. If you get an error, make sure that you're using PowerShell 3.0 or later.
2. If the installer doesn't automatically reboot your instance, run the **Restart-Computer** PowerShell cmdlet.

```
PS C:\> Restart-Computer
```

Step 3 (optional): Verify the ENA driver version after installation

To ensure that the ENA driver package was successfully installed on your instance, you can verify the new version as follows:

1. Connect to your instance and log in as the local administrator.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.

   > **Note**
   >
   > ENA adapters all use the same driver. If you have multiple ENA adapters, you can select any one of them to update the driver for all of the ENA adapters.

6. To verify the current version that's installed, open the **Driver** tab and check the **Driver Version**. If the current version doesn't match your target version, see Troubleshoot the Elastic Network Adapter (ENA) Windows driver.

Roll back an ENA driver installation

If anything goes wrong with the installation, you might need to roll back the driver. Follow these steps to roll back to the previous version of the ENA driver that was installed on your instance.

1. Connect to your instance and log in as the local administrator.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.

> **Note**
>
> ENA adapters all use the same driver. If you have multiple ENA adapters, you can select any one of them to update the driver for all of the ENA adapters.

6. To roll back the driver, open the **Driver** tab and choose **Roll Back Driver**. This opens the **Driver Package rollback** window.

> **Note**
>
> If the **Driver** tab doesn't show the **Roll Back Driver** action, or if the action is unavailable, it means that the Driver Store on your instance doesn't contain the previously installed driver package. To troubleshoot this issue, see Troubleshooting scenarios, and expand the **Unexpected ENA driver version installed** section. For more information about the device driver package selection process, see How Windows selects a driver package for a device on the *Microsoft documentation website*.

Amazon ENA driver versions

Windows AMIs include the Amazon ENA driver to enable enhanced networking.

The following table shows the corresponding ENA driver version to download for each Windows Server version.

Windows Server version	ENA driver version
Windows Server 2022	2.4.0 and later
Windows Server 2019	latest
Windows Server 2016	latest
Windows Server 2012 R2	2.6.0 and earlier
Windows Server 2012	2.6.0 and earlier

Windows Server version	ENA driver version
Windows Server 2008 R2	2.2.3 and earlier

The following table summarizes the changes for each release.

Driver version	Details	Release date

2.6.0	New Features
	• Adds the following network performance metrics for instance types that support ENA Express.
	  • `ena_srd_mode`
	  • `ena_srd_tx_pkts`
	  • `ena_srd_eligible_tx_pkts`
	  • `ena_srd_rx_pkts`
	  • `ena_srd_resource_utilization`
	• Adds `conntrack_allowance_available` network performance metric for Nitro based instance types.
	• Adds new adapter reset reason due to detection of RX data corruption.
	• Updates driver logging infrastructure.
	
	Bug Fixes
	•
		June 20, 2023

Driver version	Details	Release date
	Prevents adapter reset in the event that CPU starvation causes a network performance metrics update to fail. • Prevents false detection of an interruption to the device heartbeat. • Fixes driver installation script to support downgrade operation. • Fixes the receive error count statistic.	
2.5.0	**Announcement** ENA Windows driver version 2.5.0 has been rolled back due to failure to initialize on the Windows domain controller. Windows Client and Windows Server are unaffected.	February 17, 2023

Driver version	Details	Release date
2.4.0	New Features • Adds support for Windows Server 2022. • Removes support for Windows Server 2008 R2. • Sets Low Latency Queuing (LLQ) to always on to improve performance on sixth generation Amazon EC2 instances. Bug Fix • Fixes a failure to publish network performance metrics to the Performance Counters for Windows (PCW) system. • Fixes a memory leak during the registry key reading operation. • Prevents an infinite reset loop in the event of an unrecoverable error during the adapter reset process.	April 28, 2022

Driver version	Details	Release date

2.2.4	**Announcement**
	
	ENA Windows driver version 2.2.4 has been rolled back due to potential performance degradation on the sixth generation EC2 instances. We recommend that you downgrade the driver, using one of the following methods:
	
	• **Install the previous version**
	
	1. Download the previous version package from the link in this table (version 2.2.3).
	2. Run the **install.ps1** PowerShell installation script.
	
	For more details for pre- and post-installation steps see Enable enhanced networking on Windows.
	
	Use Amazon EC2 Systems Manager for a bulk update
	
	• Perform a bulk update via SSM document `AWS-ConfigureAWSPackage`, with the following parameters:
	  • **Name**: AwsEnaNetworkDriver
	  • **Version**: 2.2.3
		October 26, 2021

Driver version	Details	Release date
2.2.3	New Feature • Adds support for new Nitro cards with up to 400 Gbps instance networking. Bug Fix • Fixes race condition between system time change and system time query by the ENA driver, which causes false-positive detection of HW unresponsiveness. Windows ENA driver version 2.2.3 is the final version that supports Windows Server 2008 R2. Currently available instance types that use ENA will continue to be supported on Windows Server 2008 R2, and the drivers are available by download. No future instance types will support Windows Server 2008 R2, and you cannot launch, import, or migrate Windows Server 2008 R2 images to future instance types.	March 25, 2021
2.2.2	New Feature • Adds support to query network adapter performan ce metrics with CloudWatch and the Performance Counters for Windows consumers. Bug Fix • Fixes performance issues on bare metal instances.	December 21, 2020

Driver version	Details	Release date
2.2.1	New Feature • Adds a method to allow the host to query the Elastic Network Adapter for network performance metrics.	October 1, 2020
2.2.0	New Features • Adds support for next generation hardware types. • Improves instance start time after resuming from stop-hibernate, and eliminates false positive ENA error messages. Performance Optimizations • Optimizes processing of inbound traffic. • Improves shared memory management in low resource environment. Bug Fix • Avoids system crash upon ENA device removal in rare scenario where driver fails to reset.	August 12, 2020
2.1.5	Bug Fix • Fixes occasional network adapter initialization failure on bare metal instances.	June 23, 2020

Driver version	Details	Release date
2.1.4	Bug Fixes • Prevent connectivity issues caused by corrupted LSO packet metadata arriving from the network stack. • Prevent system crash caused by a rare race condition that results in accessing an already released packet memory.	November 25, 2019
2.1.2	New Feature • Added support for vendor ID report to allow OS to generate MAC-based UUIDs. Bug Fixes • Improved DHCP network configuration performance during initialization. • Properly calculate L4 checksum on inbound IPv6 traffic when the maximum transmission unit (MTU) exceeds 4K. • General improvements to driver stability and minor bug fixes.	November 4, 2019

Driver version	Details	Release date
2.1.1	Bug Fixes • Prevent drops of highly fragmented TCP LSO packets arriving from operating system. • Properly handle Encapsulating Security Payload (ESP) protocol within the IPSec in IPv6 networks.	September 16, 2019

Driver version	Details	Release date

2.1.0	ENA Windows driver v2.1 introduces new ENA device capabilities, provides a performance boost, adds new features, and includes multiple stability improvements.
	• New features
	  • Use standardized Windows registry key for Jumbo frames configuration.
	  • Allow VLAN ID setting via the ENA driver propertie s GUI.
	  • Improved Recovery flows
	  • Improved failure identification mechanism.
	  • Added support for tunable recovery parameters.
	  • Support up to 32 I/O queues for newer EC2 instances that have more than 8 vCPUs.
	  • ~90% reduction of driver memory footprint.
	• Performance optimizations
	  • Reduced transmit path latency.
	  • Support for receive checksum offload.
	  • Performance optimization for heavily loaded system (optimized usage of locking mechanisms).
	  • Further enhancements to reduce CPU utilization and improve system responsiveness under load.
	• Bug Fixes
		July 1, 2019

Driver version	Details	Release date
	• Fix crash due to invalid parsing of non-contiguous Tx headers. • Fix driver v1.5 crash during the elastic network interface detach on Bare Metal instances. • Fix LSO pseudo-header checksum calculation error over IPv6. • Fix potential memory resource leak upon initialization failure. • Disable TCP/UDP checksum offload for IPv4 fragments. • Fix for VLAN configuration. VLAN was incorrectly disabled when only VLAN priority should have been disabled. • Enable correct parsing of custom driver messages by the event viewer. • Fix failure to initialize driver due to invalid timestamp handling. • Fix race condition between data processing and ENA device disabling.	
1.5.0	• Improved stability and performance fixes. • Receive Buffers can now be configured up to a value of 8192 in Advanced Properties of the ENA NIC. • Default Receive Buffers of 1k.	October 4, 2018

Driver version	Details	Release date
1.2.3	Includes reliability fixes and unifies support for Windows Server 2008 R2 through Windows Server 2016.	February 13, 2018
1.0.8	The initial release. Included in AMIs for Windows Server 2008 R2, Windows Server 2012 RTM, Windows Server 2012 R2, and Windows Server 2016.	July 2016

Subscribe to notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2 notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select this Region because the SNS notifications that you are subscribing to are in this Region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
   a. For **TopicARN**, copy the following Amazon Resource Name (ARN):

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers

   b. For **Protocol**, choose `Email`.
   c. For **Endpoint**, enter an email address that you can use to receive the notifications.
   d. Choose **Create subscription**.
6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Subscriptions**.
3. Select the check box for the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Improve network performance with ENA Express on Windows 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 subnet.

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 subnet, 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.

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 subnet.
- The network path between the instances must not include middleware boxes. ENA Express doesn't currently support middleware boxes.

If any requirement is unmet, the instances use the standard TCP/UDP protocol but without SRD to communicate.

> **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 table contains instance types that support ENA Express.

Instance type	Architecture
General purpose	
`m6a.48xlarge`	x86_64
`m6a.metal`	x86_64
`m6i.8xlarge`	x86_64
`m6i.12xlarge`	x86_64
`m6i.16xlarge`	x86_64

Instance type	Architecture
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
m6id.32xlarge	x86_64
m6id.metal	x86_64
m7i.48xlarge	x86_64
m7i.metal-48xl	x86_64
Compute optimized	
c6a.48xlarge	x86_64
c6a.metal	x86_64
c6i.8xlarge	x86_64
c6i.12xlarge	x86_64
c6i.16xlarge	x86_64
c6i.24xlarge	x86_64
c6i.32xlarge	x86_64
c6i.metal	x86_64

Instance type	Architecture
c6id.8xlarge	x86_64
c6id.12xlarge	x86_64
c6id.16xlarge	x86_64
c6id.24xlarge	x86_64
c6id.32xlarge	x86_64
c6id.metal	x86_64
c7i.48xlarge	x86_64
c7i.metal-48xl	x86_64
Memory optimized	
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

Instance type	Architecture
r6id.24xlarge	x86_64
r6id.32xlarge	x86_64
r6id.metal	x86_64
r7i.48xlarge	x86_64
r7i.metal-48xl	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
x2iedn.32xlarge	x86_64
x2iedn.metal	x86_64
Storage optimized	
i4i.8xlarge	x86_64
i4i.12xlarge	x86_64
i4i.16xlarge	x86_64
i4i.24xlarge	x86_64
i4i.32xlarge	x86_64

Instance type	Architecture
i4i.metal	x86_64

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 on 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 network interfaces

For information on instance type settings, 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-0abcd123e456fabcd",
  "InstanceOwnerId": "111122223333",
  "Status": "attached",
  "EnaSrdSpecification": {
   "EnaSrdEnabled": true,
   "EnaSrdUdpSpecification": {
    "EnaSrdUdpEnabled": true
   }
  }
 },
 ...
 "NetworkInterfaceId": "eni-0d1234e5f6a78901b",
 "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 command 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.

Enable enhanced networking with the Intel 82599 VF interface on Windows instances

Amazon EC2 provides enhanced networking capabilities through the Intel 82599 VF interface, which uses the Intel `ixgbevf` driver.

Contents

- Requirements
- Test whether enhanced networking is enabled
- Enable enhanced networking on Windows

Requirements

To prepare for enhanced networking using the Intel 82599 VF interface, set up your instance as follows:

- Select from the following supported instance types: C3, C4, D2, I2, M4 (excluding `m4.16xlarge`), and R3.
- Launch the instance from a 64-bit HVM AMI. You can't enable enhanced networking on Windows Server 2008 and Windows Server 2003. Enhanced networking is already enabled for Windows Server 2012 R2 and Windows Server 2016 and later AMIs. Windows Server 2012 R2 includes Intel driver 1.0.15.3 and we recommend that you upgrade that driver to the latest version using the Pnputil.exe utility.
- Ensure that the instance has internet connectivity.
- Use AWS CloudShell from the AWS Management Console, or install and configure the AWS CLI or the AWS Tools for Windows PowerShell on any computer you choose, preferably your local desktop or laptop. For more information, see Access Amazon EC2 or the AWS CloudShell User Guide. Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as

enabling the `sriovNetSupport` attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Test whether enhanced networking is enabled

Enhanced networking with the Intel 82599 VF interface is enabled if the driver is installed on your instance and the `sriovNetSupport` attribute is set.

Driver

To verify that the driver is installed, connect to your instance and open Device Manager. You should see "Intel(R) 82599 Virtual Function" listed under **Network adapters**.

Instance attribute (sriovNetSupport)

To check whether an instance has the enhanced networking `sriovNetSupport` attribute set, use one of the following commands:

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.

Enable enhanced networking on Windows

If you launched your instance and it does not have enhanced networking enabled already, you must download and install the required network adapter driver on your instance, and then set the `sriovNetSupport` instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types. For more information, see Enhanced networking support.

> **Important**
>
> To view the latest version of the Intel driver in the Windows AMIs, see Details about AWS Windows AMI versions.

> **Warning**
>
> There is no way to disable the enhanced networking attribute after you've enabled it.

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.

2. [Windows Server 2016 and later] Run the following EC2 Launch PowerShell script to configure the instance after the driver is installed.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

> **Important**
>
> The administrator password will reset when you enable the initialize instance EC2 Launch script. You can modify the configuration file to disable the administrator password reset by specifying it in the settings for the initialization tasks. For steps on how to disable password reset, see Configure initialization tasks.

3. From the instance, download the Intel network adapter driver for your operating system:

   - **Windows Server 2022**

     Visit the download page and download `Wired_driver_`*`version`*`_x64.zip`.
   - **Windows Server 2019** including for Server version 1809 and later*

     Visit the download page and download `Wired_driver_`*`version`*`_x64.zip`.
   - **Windows Server 2016** including for Server version 1803 and earlier*

     Visit the download page and download `Wired_driver_`*`version`*`_x64.zip`.
   - **Windows Server 2012 R2**

     Visit the download page and download `Wired_driver_`*`version`*`_x64.zip`.
   - **Windows Server 2012**

     Visit the download page and download `Wired_driver_`*`version`*`_x64.zip`.
   - **Windows Server 2008 R2**

     Visit the download page and download `PROWinx64Legacy.exe`.

   *Server versions 1803 and earlier as well as 1809 and later are not specifically addressed on the Intel Drivers and Software pages.

4. Install the Intel network adapter driver for your operating system.

- **Windows Server 2008 R2**
  1. In the **Downloads** folder, locate the `PROWinx64Legacy.exe` file and rename it to `PROWinx64Legacy.zip`.
  2. Extract the contents of the `PROWinx64Legacy.zip` file.
  3. Open the command line, navigate to the extracted folder, and run the following command to use the `pnputil` utility to add and install the INF file in the driver store.

     ```
     C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
     ```

- **Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012**
  1. In the **Downloads** folder, extract the contents of the `Wired_driver_`*`version`*`_x64.zip` file.
  2. In the extracted folder, locate the `Wired_driver_`*`version`*`_x64.exe` file and rename it to `Wired_driver_`*`version`*`_x64.zip`.
  3. Extract the contents of the `Wired_driver_`*`version`*`_x64.zip` file.
  4. Open the command line, navigate to the extracted folder, and run one of the following commands to use the `pnputil` utility to add and install the INF file in the driver store.
     - Windows Server 2022

       ```
       C:\> pnputil -i -a PROXGB\Winx64\WS2022\vxs.inf
       ```

     - Windows Server 2019

       ```
       C:\> pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
       ```

     - Windows Server 2016

       ```
       C:\> pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
       ```

     - Windows Server 2012 R2

       ```
       C:\> pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
       ```

     - Windows Server 2012

```
C:\> pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
```

5. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: stop-instances (AWS CLI), Stop-EC2Instance (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.
6. From your local computer, enable the enhanced networking attribute using one of the following commands:

   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 a custom Windows 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.

Operating system optimizations

To achieve the maximum network performance on instances with enhanced networking, you might need to modify the default operating system configuration. We recommend the following configuration changes for applications that require high network performance. Other optimizations (such as turning on checksum offloading and enabling RSS, for example) are already in place on official Windows AMIs.

> **Note**
>
> TCP chimney offloading should be disabled in most use cases, and has been deprecated as of Windows Server 2016.

In addition to these operating system optimizations, you should also consider the maximum transmission unit (MTU) of your network traffic, and adjust according to your workload and network architecture. For more information, see Network maximum transmission unit (MTU) for your EC2 instance.

AWS regularly measures average round trip latencies between instances launched in a cluster placement group of 50us and tail latencies of 200us at the 99.9 percentile. If your applications require consistently low latencies, we recommend using the latest version of the ENA drivers on fixed performance instances built on the Nitro System.

Configure RSS CPU affinity

Receive side scaling (RSS) is used to distribute network traffic CPU load across multiple processors. By default, the official Amazon Windows AMIs are configured with RSS enabled. ENA ENIs provide up to eight RSS queues. By defining CPU affinity for RSS queues, as well as for other system processes, it is possible to spread the CPU load out over multi-core systems, enabling more network traffic to be processed. On instance types with more than 16 vCPUs, we recommend you use the `Set-NetAdapterRSS` PowerShell cmdlet (available from Windows Server 2012 and later), which manually excludes the boot processor (logical processor 0 and 1 when hyper-threading is enabled) from the RSS configuration for all ENIs, in order to prevent contention with various system components.

Windows is hyper-thread aware and will ensure the RSS queues of a single NIC are always placed on different physical cores. Therefore, unless hyper-threading is disabled, in order to completely prevent contention with other NICs, spread the RSS configuration of each NIC among a range of 16 logical processors. The `Set-NetAdapterRss` cmdlet allows you to define the per-NIC range of valid logical processors by defining the values of BaseProcessorGroup, BaseProcessorNumber, MaxProcessingGroup, MaxProcessorNumber, and NumaNode (optional). If there are not enough physical cores to completely eliminate inter-NIC contention, minimize the overlapping ranges or reduce the number of logical processors in the ENI ranges depending on the expected workload of the ENI (in other words, a low volume admin network ENI may not need as many RSS queues

assigned). Also, as previously noted, various components must run on CPU 0, and therefore we recommend excluding it from all RSS configurations when sufficient vCPUs are available.

For example, when there are three ENIs on a 72 vCPU instance with 2 NUMA nodes with hyper-threading enabled, the following commands spread the network load between the two CPUs without overlap and prevent the use of core 0 completely.

```
Set-NetAdapterRss -Name NIC1 -BaseProcessorGroup 0 -BaseProcessorNumber 2 -
MaxProcessorNumber 16
Set-NetAdapterRss -Name NIC2 -BaseProcessorGroup 1 -BaseProcessorNumber 0 -
MaxProcessorNumber 14
Set-NetAdapterRss -Name NIC3 -BaseProcessorGroup 1 -BaseProcessorNumber 16 -
MaxProcessorNumber 30
```

Note that these settings are persistent for each network adapter. If an instance is resized to one with a different number of vCPUs, you should reevaluate the RSS configuration for each enabled ENI. The complete Microsoft documentation for the `Set-NetAdapterRss` cmdlet can be found here: https://docs.microsoft.com/en-us/powershell/module/netadapter/set-netadapterrss.

Special note for SQL workloads: We also recommend that you review your I/O thread affinity settings along with your ENI RSS configuration to minimize I/O and network contention for the same CPUs. See affinity mask Server Configuration Option.

Monitor network performance for your EC2 instance

The Elastic Network Adapter (ENA) driver publishes network performance metrics from the instances where they are enabled. You can use these metrics to troubleshoot instance performance issues, choose the right instance size for a workload, plan scaling activities proactively, and benchmark applications to determine whether they maximize the performance available on an instance.

Amazon EC2 defines network maximums at the instance level to ensure a high-quality networking experience, including consistent network performance across instance sizes. AWS provides maximums for the following for each instance:

- **Bandwidth capability** – Each EC2 instance has a maximum bandwidth for aggregate inbound and outbound traffic, based on instance type and size. Some instances use a network I/O credit mechanism to allocate network bandwidth based on average bandwidth utilization. Amazon EC2 also has maximum bandwidth for traffic to AWS Direct Connect and the internet. 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 Windows instance

Requirements

- Install ENA driver version 2.2.2 or later. To verify the installed version, use Device Manager as follows.
  1. Open Device Manager by running `devmgmt.msc`.
  2. Expand **Network Adapters**.
  3. Choose **Amazon Elastic Network Adapter**, **Properties**.
  4. On the **Driver** tab, locate **Driver Version**.

  To upgrade your ENA driver, see Enhanced networking.
- To import these metrics to Amazon CloudWatch, install the CloudWatch agent. For more information, see Collect advanced network metrics in the *Amazon CloudWatch User Guide*.

Metrics for the ENA driver

The ENA driver delivers the following metrics to the instance in real time. They provide the cumulative number of packets queued or dropped on each network interface since the last driver reset.

Metric	Description	Supported on
`bw_in_allowance_exceeded`	The number of packets queued or dropped because the inbound aggregate bandwidth exceeded the maximum for the instance.	All instance types
`bw_out_allowance_exceeded`	The number of packets queued or dropped because the outbound aggregate bandwidth exceeded the maximum for the instance.	All instance types
`conntrack_allowance_exceeded`	The number of packets dropped because connection tracking exceeded the maximum for the instance and new connections could not be established. This can result in packet loss for traffic to or from the instance.	All instance types
`conntrack_allowance_available`	The number of tracked connections that can be established by the instance before hitting the Connections Tracked allowance of that instance type.	Nitro-based instance types 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	All instance types

Metric	Description	Supported on
	to the DNS service, the Instance Metadata Service, and the Amazon Time Sync Service.	
`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 Windows instance

You can view the metrics using any consumer of Windows performance counters. The data can be parsed according to the EnaPerfCounters manifest. This is an XML file that defines the performance counter provider and its countersets.

Manifest installation

If you launched the instance using an AMI that contains ENA driver 2.2.2 or later, or used the install script in the driver package for ENA driver 2.2.2, the manifest is already installed. To install the manifest manually, use the following steps:

1. Remove the existing manifest using the following command:

   ```
   unlodctr /m:EnaPerfCounters.man
   ```

2. Copy the manifest file `EnaPerfCounters.man` from the driver installation package to `%SystemRoot%\System32\drivers`.
3. Install the new manifest using the following command:

   ```
   lodctr /m:EnaPerfCounters.man
   ```

View metrics using Performance Monitor

1. Open Performance Monitor.
2. Press Ctrl+N to add new counters.

3. Choose **ENA Packets Shaping** from the list.
4. Select the instances to monitor and choose **Add**.
5. Choose **OK**.

Troubleshoot the Elastic Network Adapter (ENA) Windows driver

The Elastic Network Adapter (ENA) is designed to improve operating system health and to reduce unexpected hardware behavior or failures that can disrupt the operation of your Windows instance. The ENA architecture keeps device or driver failures as transparent to the operating system as possible.

This topic provides troubleshooting information for the ENA Windows driver.

Can't connect

If you are unable to connect to your instance, see Capture a screenshot of an unreachable instance.

> **Note**
>
> You can also connect to the instance through AWS Systems Manager Session Manager. However, to do so requires prior configuration. For more information, see Session Manager in the *AWS Systems Manager User Guide*.

Collect diagnostic information on the instance

The steps to open Windows operating system (OS) tools vary, depending on what version of the OS is installed on your instance. In the following sections, we use the **Run** dialog to open the tools, which works the same across all OS versions. However, you can access these tools using any method that you prefer.

Access the Run dialog

- Using the Windows logo key combination: `Windows + R`
- Using the search bar:
  - Enter `run` in the search bar.
  - Select the **Run** application from the search results.

Some steps require the context menu to access properties or context-sensitive actions. There are several ways to do this, depending on your OS version and hardware.

Access the context menu

- Using your mouse: right-click an item to bring up its context menu.
- Using your keyboard:
  - Depending on your OS version, use `Shift` + `F10`, or `Ctrl` + `Shift` + `F10`.
  - If you have the context key on your keyboard (three horizontal lines in a box), select the item you want and then press the context key.

If you can connect to your instance, use the following techniques to gather diagnostic information for troubleshooting.

Check ENA device status

To check the status of your ENA Windows driver using the Windows Device Manager, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.
6. Verify that the message in the **General** tab says "This device is working properly."

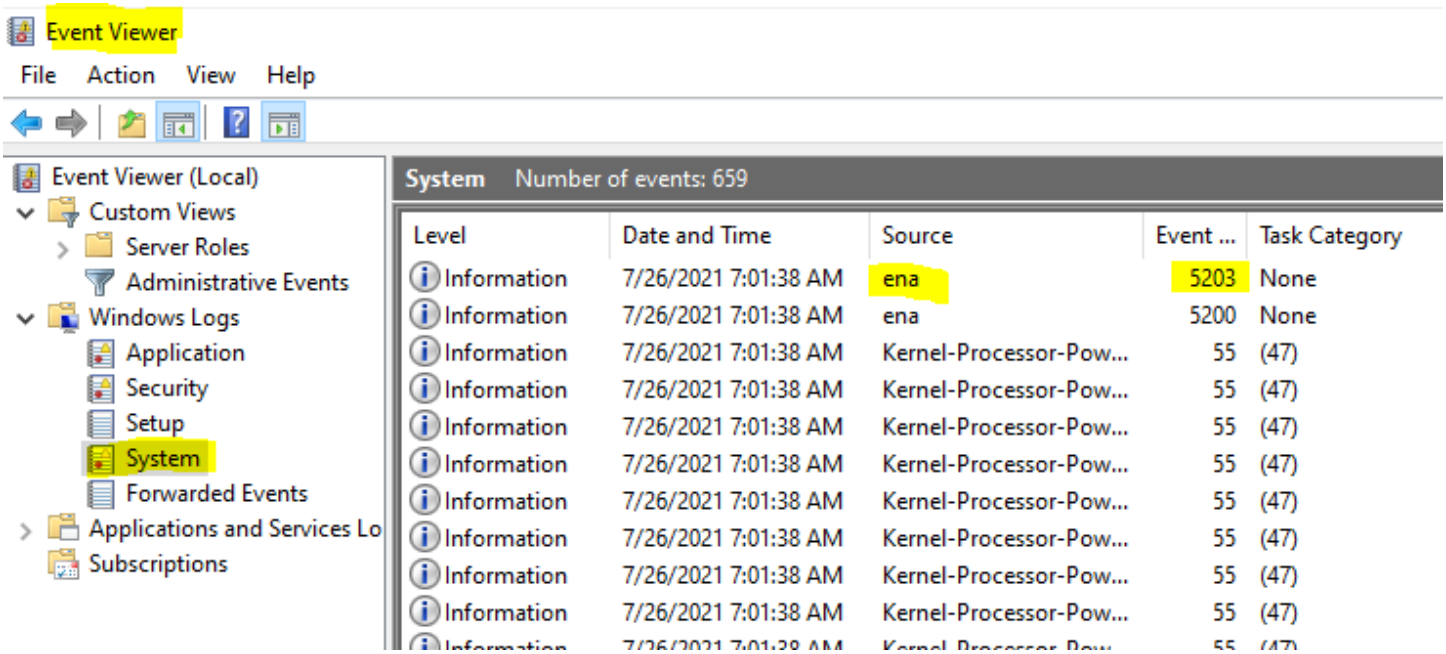
Investigate driver event messages

To review ENA Windows driver event logs using the Windows Event Viewer, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Event Viewer, enter `eventvwr.msc` in the **Run** box.
3. Choose **OK**. This opens the Event Viewer window.
4. Expand the **Windows Logs** menu, and then choose **System**.

5. Under **Actions**, in the top-right panel, choose **Filter Current Log**. This displays the filtering dialog.
6. In the **Event sources** box, enter `ena`. This limits results to events that were generated by the ENA Windows driver.
7. Choose **OK**. This shows filtered event log results in the detail sections of the window.
8. To drill down into the details, select an event message from the list.

The following example shows an ENA driver event in the Windows Event Viewer system events list:



Event message summary

The following table shows event messages that the ENA Windows driver generates.

Input

Event ID	ENA driver event description	Type
5001	Hardware is out of resources	Error
5002	Adapter has detected a hardware error	Error

Event ID	ENA driver event description	Type
5005	Adapter has timed out on NDIS operation that did not complete in a timely manner	Error
5032	Adapter has failed to reset the device	Error
5200	Adapter has been initialized	Informational
5201	Adapter has been halted	Informational
5202	Adapter has been paused	Informational
5203	Adapter has been restarted	Informational
5204	Adapter has been shut down	Informational
5205	Adapter has been reset	Error
5206	Adapter has been surprise removed	Error
5208	Adapter initialization routine has failed	Error
5210	Adapter has encountered and successfully recovered an internal issue	Error

Review performance metrics

The ENA Windows driver publishes network performance metrics from the instances where metrics are enabled. You can view and enable metrics on the instance using the native Performance Monitor application. For more information about the metrics that the ENA Windows driver produces, see Monitor network performance for your EC2 instance.

On instances where ENA metrics are enabled, and the Amazon CloudWatch agent is installed, CloudWatch collects the metrics that are associated with the counters in Windows Performance

Monitor, as well as some advanced metrics for ENA. These metrics are collected in addition to the metrics enabled by default on EC2 instances. For more information about the metrics, see Metrics collected by the CloudWatch agent in the *Amazon CloudWatch User Guide*.

> **Note**
>
> Performance metrics are available for ENA driver versions 2.4.0 and later (also for version 2.2.3). ENA driver version 2.2.4 was rolled back due to potential performance degradation on the sixth generation EC2 instances. We recommend that you upgrade to the current version of the driver to ensure that you have the latest updates.

Some of the ways that you can use performance metrics include:

- Troubleshoot instance performance issues.
- Choose the right instance size for a workload.
- Proactively plan scaling activities.
- Benchmark applications to determine if they maximize the performance available on an instance.

Refresh rate

By default, the driver refreshes metrics using a 1-second interval. However, the application that retrieves the metrics might use a different interval for polling. You can change the refresh interval in Device Manager, using the advanced properties for the driver.

To change the metrics refresh interval for the ENA Windows driver, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.
6. Open the **Advanced** tab in the pop-up window.
7. From the **Property** list, choose **Metrics Refresh Interval** to change the value.
8. When you are done, choose **OK**.

ENA adapter reset

The reset process starts when the ENA Windows driver detects an error on an adapter, and marks the adapter as unhealthy. The driver cannot reset itself, so it depends on the operating system to check the adapter health status, and call the reset handle for the ENA Windows driver. The reset process might result in a brief period of time where traffic loss occurs. However, TCP connections should be able to recover.

The ENA adapter might also indirectly request a device reset procedure, by failing to send a keep-alive notification. For example, if the ENA adapter reaches an unknown state after loading an irrecoverable configuration, it might stop sending keep-alive notifications.

Common causes for ENA adapter reset

- Keep-alive messages are missing

  The ENA adapter posts keep-alive events at a fixed rate (usually once every second). The ENA Windows driver implements a watchdog mechanism, which periodically checks for the presence of these keep-alive messages. If it detects one or more new messages since the last time it checked, it records a successful outcome. Otherwise, the driver concludes that the device experienced a failure, and initiates a reset sequence.
- Packets are stuck in transmit queues

  The ENA adapter verifies that packets are flowing through the transmit queues as expected. The ENA Windows driver detects if packets are getting stuck, and initiates a reset sequence if they are.
- Read timeout for Memory Mapped I/O (MMIO) registers

  To limit memory mapped I/O (MMIO) read operations, the ENA Windows driver accesses MMIO registers only during initialization and reset processes. If the driver detects a timeout, it takes one of the following actions, depending on what process was running:

  - If a timeout is detected during initialization, it fails the flow, which results in the driver displaying a yellow exclamation mark by the ENA adapter in Windows Device Manager.
  - If a timeout is detected during reset, it fails the flow. The OS then initiates a surprise removal of the ENA adapter, and recovers it by stopping and starting the adapter that was removed. For more information about surprise removal of a network interface card (NIC), see Handling the Surprise Removal of a NIC in the *Microsoft Windows Hardware Developer* documentation.

Troubleshooting scenarios

The following scenarios can help you troubleshoot issues that you might experience with the ENA Windows driver. We recommend that you start with upgrading your ENA driver, if you don't have the latest version. To find the latest driver for your Windows OS version, see Amazon ENA driver versions.

Unexpected ENA driver version installed

Description

After you go through the steps to install a specific version of the ENA driver, the Windows Device Manager shows that Windows installed a different version of the ENA driver.

Cause

When you run the install for a driver package, Windows ranks all of the driver packages that are valid for the given device in the local Driver Store before it begins. Then it selects the package with the lowest rank value as the best match. This can be different from the package that you intended to install. For more information about the device driver package selection process, see How Windows selects a driver package for a device on the *Microsoft documentation website*.

Solution

To ensure that Windows installs your chosen driver package version, you can remove lower ranked driver packages from the Driver Store with the PnPUtil command line tool.

Follow these steps to update the ENA driver:

1. Connect to your instance and log in as the local administrator.
2. Open the Device Manager properties window, as described in the Check ENA device status section. This opens the **General** tab of the **Amazon Elastic Network Adapter Properties** window.
3. Open the **Driver** tab.
4. Choose **Update Driver**. This opens the **Update Driver Software – Amazon Elastic Network Adapter** dialog box.

   a. On the **How do you want to search for driver software?** page, choose **Browse my computer for driver software**.

b. On the **Browse for driver software on your computer** page, choose **Let me pick from a list of device drivers on my computer**, located below the search bar.

c. On the **Select the device driver you want to install for this hardware** page, choose **Have Disk...**.

d. In the **Install from Disk** window, choose **Browse...**, next to the file location from the dropdown list.

e. Navigate to the location where you downloaded the target ENA driver package. Select the file named `ena.inf` and choose **Open**.

f. To start the install, choose **OK**, and then choose **Next**.

5. If the installer doesn't automatically reboot your instance, run the **Restart-Computer** PowerShell cmdlet.

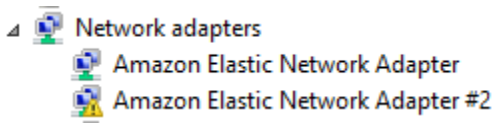
```
PS C:\> Restart-Computer
```

Device warning for ENA driver

Description

The ENA adapter icon in the Device Manager **Network adapters** section displays a warning sign (a yellow triangle with an exclamation mark inside).

The following example shows an ENA adapter with the warning icon in Windows Device Manager:



Cause

This device warning is commonly caused by environment issues, which might require more research, and often require a process of elimination to determine the underlying cause. For a full list of device errors, see Device Manager Error Messages in the *Microsoft Windows Hardware Developer* documentation.

Solution

The solution for this device warning depends on the root cause. The process of elimination described here includes a few basic steps to help identify and resolve the most common issues that

might have a simple solution. Additional root cause analysis is required when these steps do not resolve the issue.

Follow these steps to help identify and resolve common issues:

1. **Stop and start the device**

   Open the Device Manager properties window, as described in the Check ENA device status section. This opens the **General** tab of the **Amazon Elastic Network Adapter Properties** window, where the **Device status** displays the error code and a short message.

   a. Open the **Driver** tab.

   b. Choose **Disable Device**, and respond **Yes** to the warning message that displays.

   c. Choose **Enable Device**.

2. **Stop and start the EC2 instance**

   If the adapter still shows the warning icon in Device Manager, the next step is to stop and start the EC2 instance. This relaunches the instance on different hardware in most cases.

3. **Investigate possible instance resource issue**

   If you have stopped and started your EC2 instance, and the problem persists, this might indicate a resource issue on your instance, such as insufficient memory.

Connection timeout with adapter reset (error codes 5007, 5205)

Description

The Windows Event Viewer shows adapter timeout and reset events occurring in combination for ENA adapters. Messages resemble the following examples:

- **Event ID 5007**: Amazon Elastic Network Adapter : Timed out during an operation.
- **Event ID 5205**: Amazon Elastic Network Adapter : Adapter reset has been started.

Adapter resets cause minimal traffic disruption. Even when there are multiple resets, it would be unusual for them to cause any severe network disruption.

Cause

This sequence of events indicates that the ENA Windows driver initiated a reset for an ENA adapter that was unresponsive. However, the mechanism that the device driver uses to detect this issue is subject to false positives resulting from CPU 0 starvation.

Solution

If this combination of errors happens frequently, check your resource allocations to see where adjustments might be helpful.

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Resource Monitor, enter `resmon` in the **Run** box.
3. Choose **OK**. This opens the Resource Monitor window.
4. Open the **CPU** tab. Per-CPU usage graphs are shown along the right side of the Resource Monitor window.
5. Check the usage levels for CPU 0 to see if they are too high.

We recommend that you configure RSS to exclude CPU 0 for the ENA adapter on larger instance types (more than 16 vCPU). For smaller instance types, configuring RSS might improve the experience, but due to the lower number of available cores, testing is necessary to ensure that constraining CPU cores does not negatively impact performance.

Use the **Set-NetAdapterRss** command to configure RSS for your ENA adapter, as shown in the following example.

```
Set-NetAdapterRss -name (Get-NetAdapter | Where-Object {$_.InterfaceDescription -like
 "*Elastic*"}).Name -Baseprocessorgroup 0 -BaseProcessorNumber 1
```

Migrating to a sixth generation instance infrastructure impacts performance or attachment

Description

If you migrate to a sixth generation EC2 instance, you might experience reduced performance or ENA attachment failures if you haven't updated your ENA Windows driver version.

Cause

The sixth generation EC2 instance types require the following minimum version of the ENA Windows driver, based on the instance operating system (OS).

Minimum version

Windows Server version	ENA driver version
Windows Server 2008 R2	2.2.3 or 2.4.0
Windows Server 2012 and later	2.2.3 and later
Windows Workstation	2.2.3 and later

Solution

Before you upgrade to a sixth generation EC2 instance, make sure that the AMI you launch from has compatible drivers based on the instance OS as shown in the previous table. For more information, see What do I need to do before migrating my EC2 instance to a sixth generation instance to make sure that I get maximum network performance? in the *AWS re:Post Knowledge Center*.

Suboptimal performance for the elastic network interface

Description

The ENA interface is not performing as expected.

Cause

Root cause analysis for performance issues is a process of elimination. There are too many variables involved to name a common cause.

Solution

The first step in your root cause analysis is to review the diagnostic information for the instance that is not performing as expected, to determine if there are errors that might be causing the issue. For more information, see the Collect diagnostic information on the instance section.

You might need to modify the default operating system configuration to achieve maximum network performance on instances with enhanced networking. Some optimizations, such as turning on checksum offloading and enabling RSS, are configured by default in official Windows AMIs.

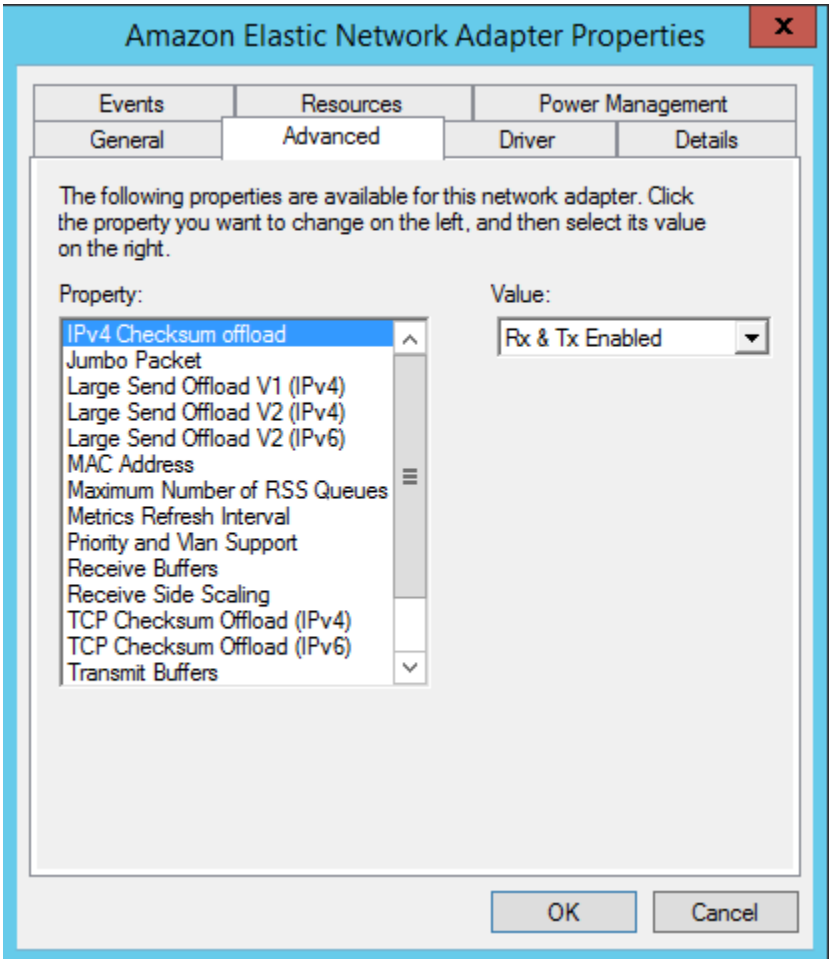
For other optimizations that you can apply to the ENA adapter, see the performance adjustments shown in ENA adapter performance adjustments.

We recommend that you proceed with caution, and limit device property adjustments to those that are listed in this section, or to specific changes that are recommended by the AWS support team.

To change ENA adapter properties, follow these steps:

1. Open the **Run** dialog using one of the methods described in the preceding section.
2. To open the Windows Device Manager, enter `devmgmt.msc` in the **Run** box.
3. Choose **OK**. This opens the Device Manager window.
4. Select the arrow to the left of **Network adapters** to expand the list.
5. Choose the name, or open the context menu for the **Amazon Elastic Network Adapter**, and then choose **Properties**. This opens the **Amazon Elastic Network Adapter Properties** dialog.
6. To make your changes, open the **Advanced** tab.
7. When you're done, choose **OK** to save your changes.

The following example shows an ENA adapter property in the Windows Device Manager:



ENA adapter performance adjustments

The following table includes properties that can be adjusted to improve performance for the ENA interface.

Input

Property	Description	Default value	Adjustment
Receive Buffers	Controls the number of entries in the software receive queues.	1024	Can be increased up to a maximum of 8192.
Receive Side Scaling (RSS)	Enables the efficient distribution of	Enabled	You can spread the load across multiple

Property	Description	Default value	Adjustment
	network receive processing acros s multiple CPUs in multiprocessor systems.		processors. To learn more, see Operating system optimizations.

Property	Description	Default value	Adjustment
Maximum Number of RSS Queues	Sets the maximum number of RSS queues allowed when RSS is enabled.	32	The number of RSS queues is determined during driver initialization, and includes the following limitations (among others): • RSS queue limit set by this property • Instance limits (vCPU count) • Hardware generation limits (up to 8 RSS queues in ENAv1, and up to 32 RSS queues in ENAv2) You can set the value from 1-32, depending on your instance and hardware generation limits. To learn more, see Operating system optimizations.

Property	Description	Default value	Adjustment
Jumbo packet	Enables the use of jumbo ethernet frames (more than 1500 bytes of payload).	Disabled (this limits payload to 1500 bytes or less)	Value can be set up to 9015, which translates to 9001 bytes of payload. This is the maximum payload for jumbo ethernet frames. See Considerations for using jumbo ethernet frames.

Considerations for using jumbo ethernet frames

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, which increases the percentage of the packet that is not packet overhead. Fewer packets are needed to send the same amount of usable data. However, traffic is limited to a maximum MTU of 1500 in the following cases:

- Traffic outside of a given AWS Region for EC2 Classic.
- Traffic outside of a single VPC.
- Traffic over an inter-Region VPC peering connection.
- Traffic over VPN connections.
- Traffic over an internet gateway.

> **Note**
>
> Packets over 1500 bytes are fragmented. If you have the `Don't Fragment` flag set in the IP header, these packets are dropped.
> Jumbo frames should be used with caution for internet-bound traffic, or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside of a VPC without impacting outbound traffic that's leaving the VPC, try one of the following options:

- Configure the MTU size by route.
- Use multiple network interfaces with different MTU sizes and different routes.

Recommended use cases for jumbo frames

Jumbo frames can be useful for traffic inside of and between VPCs. We recommend using jumbo frames for the following use cases:

- For instances that are collocated inside of a cluster placement group, jumbo frames help to achieve the maximum network throughput possible. For more information, see Placement groups.
- You can use jumbo frames for traffic between your VPCs and your on-premises networks over AWS Direct Connect. For more information about using AWS Direct Connect, and verifying jumbo frame capability, see Set network MTU for private virtual interfaces or transit virtual interfaces in the *AWS Direct Connect User Guide*.
- For more information about supported MTU sizes for transit gateways, see Quotas for your transit gateways in the *Amazon VPC Transit Gateways*.

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

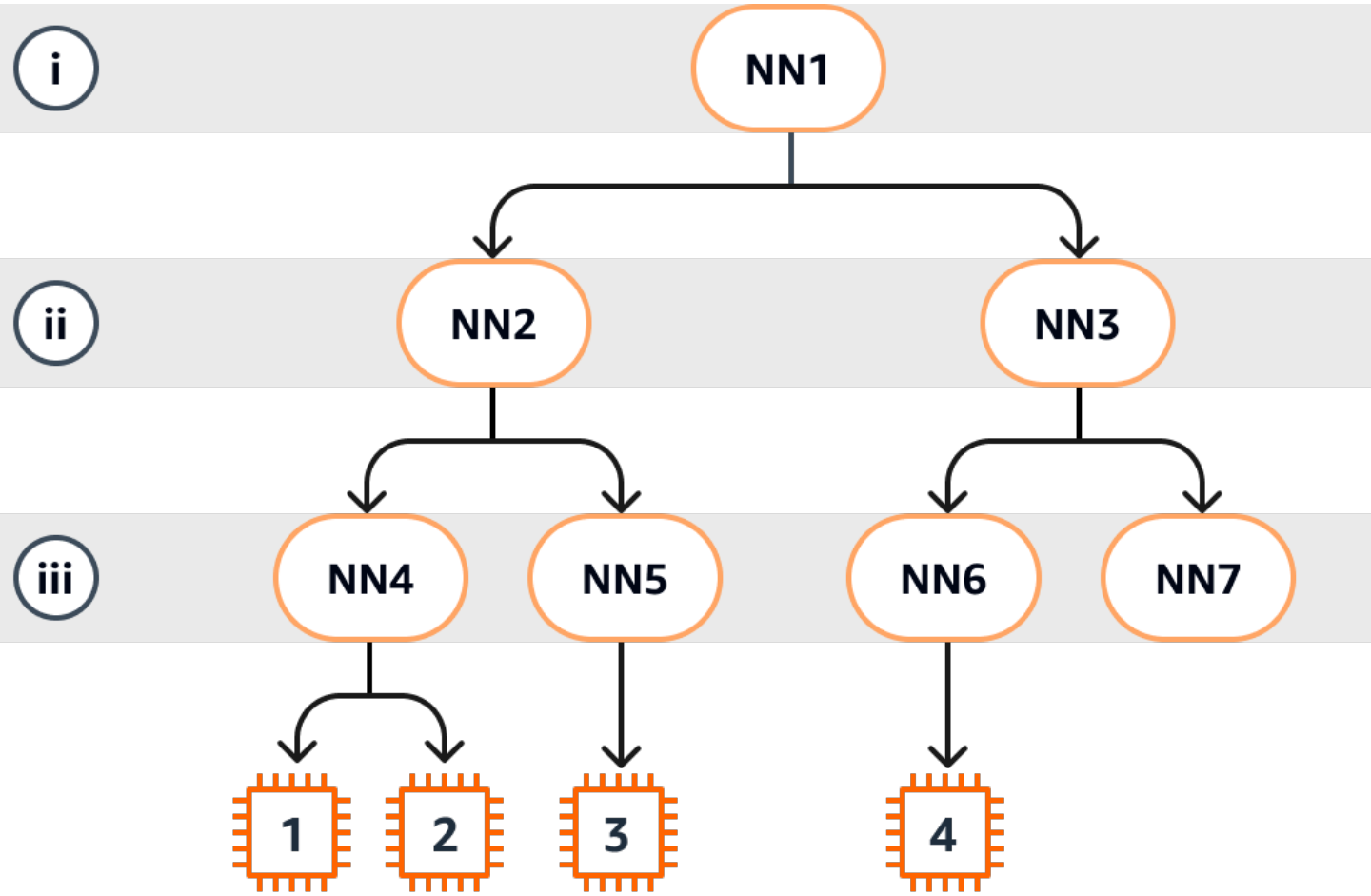
- How instance topology works
- Prerequisites for instance topology
- Amazon EC2 instance topology examples

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-444444444example",  //Corresponds to instance 4
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",          //Corresponds to NN1 - layer i
                "nn-3333333333example",          //Corresponds to NN3 - layer ii
                "nn-6666666666example"           //Corresponds to NN6 - layer iii -
connected to instance
```

```
            ],
            "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-444444444example",
            "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*"
```

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-444444444example",
            "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.

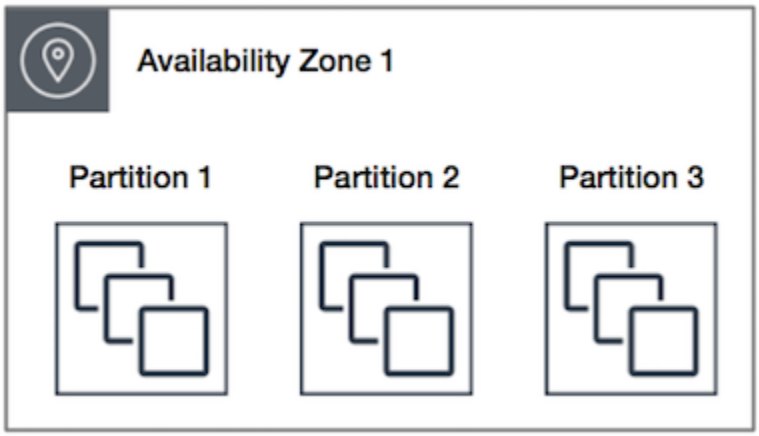
If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

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

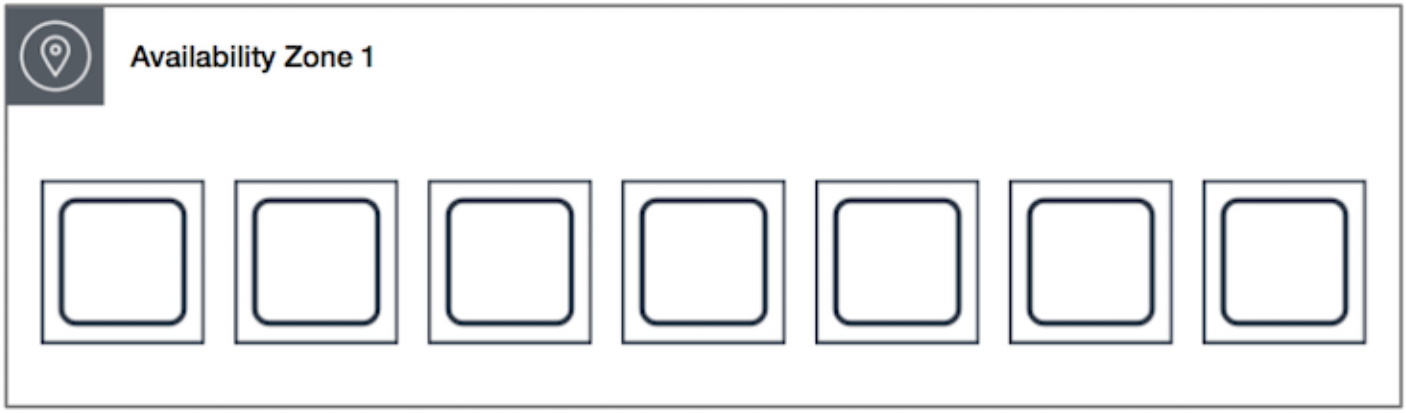
A spread placement group is a group of instances that are each placed on distinct hardware.

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) and M7i-flex instances.
  - The following previous generation instances: A1, C3, C4, G2, 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 **`Delete`** 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 Linux instances, switch to this page in the *Amazon EC2 User Guide for Linux Instances* guide: Network maximum transmission unit (MTU) for your EC2 instance.

Contents

- Jumbo frames (9001 MTU)
- Path MTU Discovery
- Check the path MTU between two hosts
- Check and set the MTU on your Windows 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, G2, 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 **mturoute.exe** command, which you can download and install from http://www.elifulkerson.com/projects/mturoute.php.

To check path MTU using mturoute.exe

1. Download **mturoute.exe** from http://www.elifulkerson.com/projects/mturoute.php.
2. Open a Command Prompt window and change to the directory where you downloaded **mturoute.exe**.
3. Use the following command to check the path MTU between your EC2 instance and another host. You can use a DNS name or an IP address as the destination. If the destination is another EC2 instance, verify that the security group allows inbound UDP traffic. This example checks the path MTU between an EC2 instance and `www.elifulkerson.com`.

```
.\mturoute.exe www.elifulkerson.com
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 10000 bytes. *
+ ICMP payload of 1472 bytes succeeded.
- ICMP payload of 1473 bytes is too big.
Path MTU: 1500 bytes.
```

   In this example, the path MTU is 1500.

Check and set the MTU on your Windows instance

Some drivers are configured to use jumbo frames, and others are configured to use standard frame sizes. You might want to use jumbo frames for network traffic within your VPC or standard frames for internet traffic. Whatever your use case, we recommend that you verify that your instances behave as expected.

If your instance runs in a Wavelength Zone, the maximum MTU value is 1300.

ENA Driver

For Driver Versions 1.5 and Earlier

You can change the MTU setting using Device Manager or the **Set-NetAdapterAdvancedProperty** command.

To get the current MTU setting using the **Get-NetAdapterAdvancedProperty** command, use the following command. Check the entry for the interface name `MTU`. A value of 9001 indicates that Jumbo frames are enabled. Jumbo frames are disabled by default.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Enable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -RegistryValue
 9001
```

Disable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -RegistryValue
 1500
```

For Driver Versions 2.1.0 and Later

You can change the MTU setting using Device Manager or the **Set-NetAdapterAdvancedProperty** command.

To get the current MTU setting using the **Get-NetAdapterAdvancedProperty** command, use the following command. Check the entry for the interface name `*JumboPacket`. A value of 9015 indicates that Jumbo frames are enabled. Jumbo frames are disabled by default.

Run **Get-NetAdapterAdvancedProperty** or use wildcard (asterisk) to detect all corresponding Ethernet names.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet*"
```

Run the following commands and include the Ethernet name you want to query.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Enable jumbo frames as follows.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9015
```

Disable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

Intel SRIOV 82599 driver

You can change the MTU setting using Device Manager or the **Set-NetAdapterAdvancedProperty** command.

To get the current MTU setting using the **Get-NetAdapterAdvancedProperty** command, use the following command. Check the entry for the interface name `*JumboPacket`. A value of 9014 indicates that Jumbo frames are enabled. (Note that the MTU size includes the header and the payload.) Jumbo frames are disabled by default.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Enable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9014
```

Disable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

AWS PV driver

You cannot change the MTU setting using Device Manager, but you can change it using the **netsh** command.

Get the current MTU setting using the following command. The name of the interface can vary. In the output, look for an entry with the name "Ethernet," "Ethernet 2," or "Local Area Connection". You'll need the interface name to enable or disable jumbo frames. A value of 9001 indicates that Jumbo frames are enabled.

```
netsh interface ipv4 show subinterface
```

Enable jumbo frames as follows:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=9001
```

Disable jumbo frames as follows:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500
```

Troubleshoot

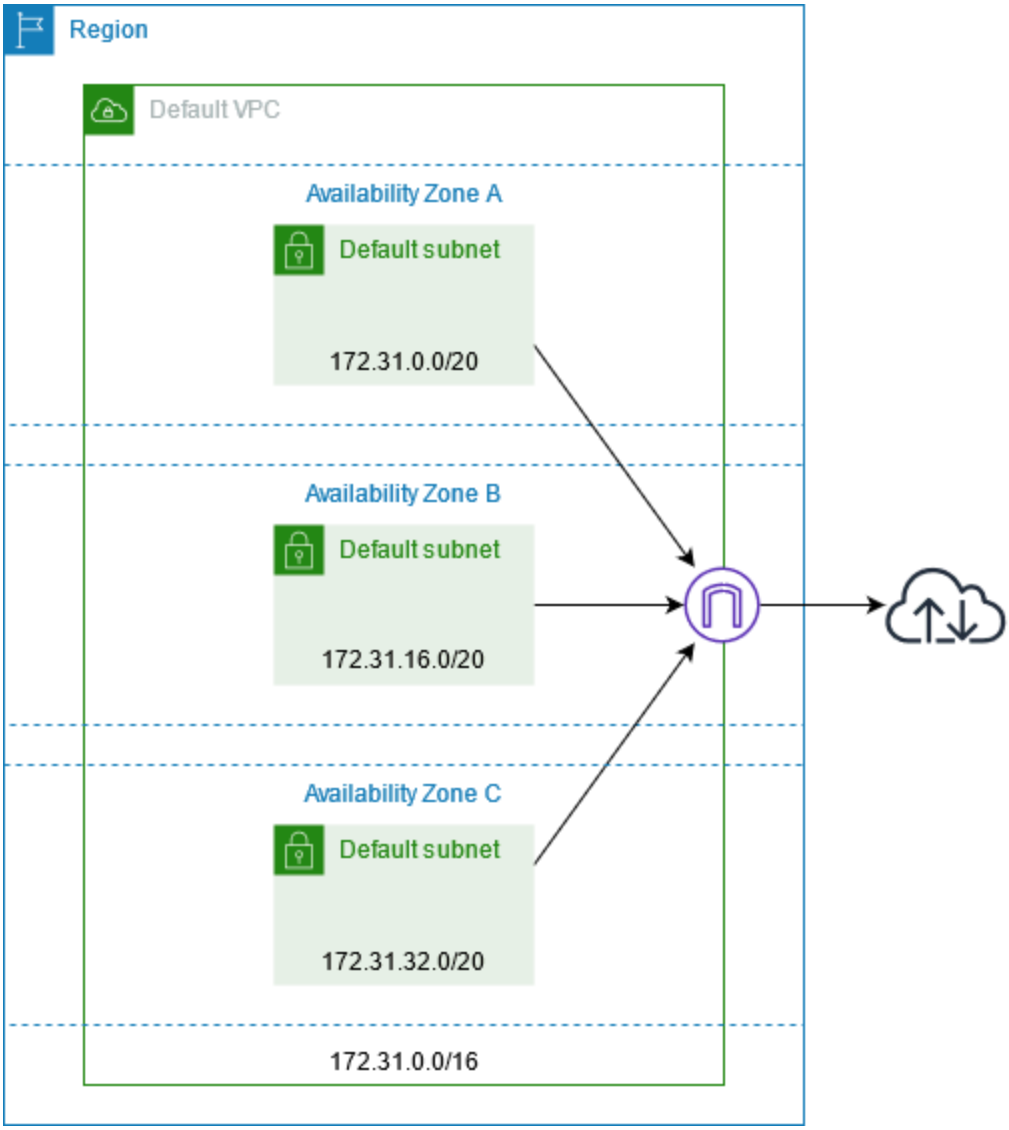
If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see Queries Appear to Hang in the *Amazon Redshift Management Guide*

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*.

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 Nitro System.

RDP access to your instances

To connect to an instance, you must authorize RDP 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 Prerequisites.

Ports and Protocols for Windows Amazon Machine Images (AMIs)

The following tables list the ports, protocols, and directions by workload for Windows Amazon Machine Images.

Contents

- AllJoyn Router
- Cast to Device
- Core Networking
- Delivery Optimization
- Diag Track
- DIAL Protocol Server
- Distributed File System (DFS) Management
- File and Printer Sharing
- File Server Remote Management
- ICMP v4 All
- Microsoft Edge
- Microsoft Media Foundation Network Source
- Multicast
- Remote Desktop
- Windows Device Management
- Windows Feature Experience Pack
- Windows Firewall Remote Management
- Windows Remote Management

AllJoyn Router

OS	Rule	Description	Port	Protocol	Direction
Windows Server 2016 Windows Server 2019 Windows Server 2022	AllJoyn Router (TCP-In)	Inbound rule for AllJoyn Router traffic [TCP]	Local: 9955 Remote: Any	TCP	In
	AllJoyn Router (TCP-Out)	Outbound rule for AllJoyn	Local: Any Remote: Any	TCP	Out

OS	Rule	Description	Port	Protocol	Direction
		Router traffic [TCP]			
	AllJoyn Router (UDP-In)	Inbound rule for AllJoyn Router traffic [UDP]	Local: Any Remote: Any	UDP	In
	AllJoyn Router (UDP-Out)	Outbound rule for AllJoyn Router traffic [UDP]	Local: Any Remote: Any	UDP	Out

Cast to Device

OS	Rule	Description	Port	Protocol	Direction
Windows Server 2016 Windows Server 2019 Windows Server 2022	Cast to Device functionality (qWave-TCP-In)	Inbound rule for the Cast to Device functiona lity to allow use of the Quality Windows Audio Video Experience Service. [TCP 2177]	Local: 2177 Remote: Any	TCP	In
	Cast to Device functionality	Outbound rule for the Cast to Device	Local: Any Remote: 2177	TCP	Out

OS	Rule	Description	Port	Protocol	Direction
	(qWave-TCP-Out)	functiona lity to allow use of the Quality Windows Audio Video Experience Service. [TCP 2177]			
	Cast to Device functionality (qWave-UDP-In)	Inbound rule for the Cast to Device functiona lity to allow use of the Quality Windows Audio Video Experience Service. [UDP 2177]	Local: 2177 Remote: Any	UDP	In

OS	Rule	Description	Port	Protocol	Direction
	Cast to Device functionality (qWave-UDP-Out)	Outbound rule for the Cast to Device functionality to allow use of the Quality Windows Audio Video Experience Service. [UDP 2177]	Local: Any Remote: 2177	UDP	Out
	Cast to Device SSDP Discovery (UDP-In)	Inbound rule to allow discovery of Cast to Device targets using SSDP	Local: Ply2Disc Remote: Any	UDP	In
	Cast to Device Streaming Server (HTTP-Streaming-In)	Inbound rule for the Cast to Device server to allow streaming using HTTP. [TCP 10246]	Local: 10246 Remote: Any	TCP	In

OS	Rule	Description	Port	Protocol	Direction
	Cast to Device Streaming Server (RTCP-Streaming-In)	Inbound rule for the Cast to Device server to allow streaming using RTSP and RTP. [UDP]	Local: Any Remote: Any	UDP	In
	Cast to Device Streaming Server (RTP-Streaming-Out)	Outbound rule for the Cast to Device server to allow streaming using RTSP and RTP. [UDP]	Local: Any Remote: Any	UDP	Out
	Cast to Device Streaming Server (RTSP-Streaming-In)	Inbound rule for the Cast to Device server to allow streaming using RTSP and RTP. [TCP 23554, 23555, 23556]	Local: 235, 542, 355, 523, 556 Remote: Any	TCP	In

OS	Rule	Description	Port	Protocol	Direction
	Cast to Device UPnP Events (TCP-In)	Inbound rule to allow receiving UPnP Events from Cast to Device targets	Local: 2869 Remote: Any	TCP	In

Core Networking

Windows Server 2016, 2019, and 2022

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2016 Windows Server 2019 Windows Server 2022	Destination Unreachable (ICMPv6-In)	Destination Unreachable error messages are sent from any node that a packet traverses which is unable to forward the packet for any reason except congestion.		ICMPv6	In
	Destination Unreachable Fragmenta	Destination Unreachable Fragmenta		ICMPv4	In

OS	Rule	Definition	Port	Protocol	Direction
	tion Needed (ICMPv4-In)	tion Needed error messages are sent from any node that a packet traverses which is unable to forward the packet because fragmenta tion was needed and the don't fragment bit was set.			

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - DNS (UDP-Out)	Outbound rule to allow DNS requests. DNS responses based on requests that match this rule are permitted regardless of source address. This behavior is classified as loose source mapping.	Local: Any Remote: 53	UDP	Out
	Dynamic Host Configuration Protocol (DHCP-In)	Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.	Local: 68 Remote: 67	UDP	In

OS	Rule	Definition	Port	Protocol	Direction
	Dynamic Host Configuration Protocol (DHCP-Out)	Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.	Local: 68 Remote: 67	UDP	Out
	Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)	Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.	Local: 546 Remote: 547	UDP	In

OS	Rule	Definition	Port	Protocol	Direction
	Dynamic Host Configuration Protocol for IPv6(DHCPV6-Out)	Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.	Local: 546 Remote: 547	UDP	Out
	Core Networking - Group Policy (LSASS-Out)	Outbound rule to allow remote LSASS traffic for Group Policy updates.	Local: Any Remote: Any	TCP	Out
	Core Networking - Group Policy (NP-Out)	Core Networking - Group Policy (NP-Out)	Local: Any Remote: 445	TCP	Out
	Core Networking - Group Policy (TCP-Out)	Outbound rule to allow remote RPC traffic for Group Policy updates.	Local: Any Remote: Any	TCP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Internet Group Management Protocol (IGMP-In)	IGMP messages are sent and received by nodes to create, join, and depart multicast groups.		2	In
	Core Networking - Internet Group Management Protocol (IGMP-Out)	IGMP messages are sent and received by nodes to create, join, and depart multicast groups.		2	Out
	Core Networking - IPHTTPS (TCP-In)	Inbound TCP rule to allow IPHTTPS tunneling technology to provide connectivity across HTTP proxies and firewalls.	Local: IPHTPS Remote: Any	TCP	In

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - IPHTTPS (TCP-Out)	Outbound TCP rule to allow IPHTTPS tunneling technology to provide connectivity across HTTP proxies and firewalls.	Local: Any Remote: IPHTPS	TCP	Out
	IPv6 (IPv6-In)	Inbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.		41	In

OS	Rule	Definition	Port	Protocol	Direction
	IPv6 (IPv6-Out)	Outbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressin g Protocol) and 6to4 tunneling services.		41	Out
	Multicast Listener Done (ICMPv6-In)	Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Done (ICMPv6-Out)	Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.		ICMPv6	Out
	Multicast Listener Query (ICMPv6-In)	An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Query (ICMPv6-Out)	An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership.		ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report (ICMPv6-In)	The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report (ICMPv6-Out)	The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.		ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report v2 (ICMPv6-In)	Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report v2 (ICMPv6-Out)	Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.		ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Advertise ment (ICMPv6-In)	Neighbor Discovery Advertise ment messages are sent by nodes to notify other nodes of link-laye r address changes or in response to a Neighbor Discovery Solicitation request.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Advertisement (ICMPv6-Out)	Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.		ICMPv6	Out
	Neighbor Discovery Solicitation (ICMPv6-In)	Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Solicitation (ICMPv6-Out)	Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.		ICMPv6	Out
	Packet Too Big (ICMPv6-In)	Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Packet Too Big (ICMPv6-Out)	Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.		ICMPv6	Out
	Parameter Problem (ICMPv6-In)	Parameter Problem error messages are sent by nodes when packets are incorrectly generated.		ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Parameter Problem (ICMPv6-Out)	Parameter Problem error messages are sent by nodes when packets are incorrectly generated.		ICMPv6	Out
	Router Advertisement (ICMPv6-In)	Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.		ICMPv6	In
	Router Advertisement (ICMPv6-Out)	Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.		ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Router Solicitation (ICMPv6-In)	Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-conf iguration.		ICMPv6	In
	Router Solicitation (ICMPv6-O ut)	Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-conf iguration.		ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - Teredo (UDP-In)	Inbound UDP rule to allow Teredo edge traversal. This technology provides address assignment and automatic tunneling for unicast IPv6 traffic when an IPv6/IPv4 host is located behind an IPv4 network address translator.	Local: Teredo Remote: Any	UDP	In

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - Teredo (UDP-Out)	Outbound UDP rule to allow Teredo edge traversal. This technology provides address assignment and automatic tunneling for unicast IPv6 traffic when an IPv6/IPv4 host is located behind an IPv4 network address translator.	Local: Any Remote: Any	UDP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Time Exceeded (ICMPv6-In)	Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.		ICMPv6	In
	Time Exceeded (ICMPv6-Out)	Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.		ICMPv6	Out

Windows Server 2012 and 2012 R2

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2012 Windows Server 2012 R2	Destination Unreachable (ICMPv6-In)	Destination Unreachable error messages are sent from any node that a packet traverses which is unable to forward the packet for any reason except congestion.	Local: 68 Remote: 67	ICMPv6	In
	Destination Unreachable Fragmentation Needed (ICMPv4-In)	Destination Unreachable Fragmentation Needed error messages are sent from any node that a packet traverses which is unable to forward the packet because	Local: 68 Remote: 67	ICMPv4	In

OS	Rule	Definition	Port	Protocol	Direction
		fragmentation was needed and the don't fragment bit was set.			
	Core Networking - DNS (UDP-Out)	Outbound rule to allow DNS requests. DNS responses based on requests that match this rule are permitted regardless of source address. This behavior is classified as loose source mapping.	Local: Any Remote: 53	UDP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Dynamic Host Configuration Protocol (DHCP-In)	Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.	Local: 68 Remote: 67	UDP	In
	Dynamic Host Configuration Protocol (DHCP-Out)	Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.	Local: 68 Remote: 67	UDP	Out
	Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)	Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.	Local: 546 Remote: 547	UDP	In

OS	Rule	Definition	Port	Protocol	Direction
	Dynamic Host Configuration Protocol for IPv6(DHCPV6-Out)	Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.	Local: 546 Remote: 547	UDP	Out
	Core Networking - Group Policy (LSASS-Out)	Outbound rule to allow remote LSASS traffic for Group Policy updates.	Local: Any Remote: Any	TCP	Out
	Core Networking - Group Policy (NP-Out)	Core Networking - Group Policy (NP-Out)	Local: Any Remote: 445	TCP	Out
	Core Networking - Group Policy (TCP-Out)	Outbound rule to allow remote RPC traffic for Group Policy updates.	Local: Any Remote: Any	TCP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Internet Group Management Protocol (IGMP-In)	IGMP messages are sent and received by nodes to create, join, and depart multicast groups.	Local: 68 Remote: 67	2	In
	Core Networking - Internet Group Management Protocol (IGMP-Out)	IGMP messages are sent and received by nodes to create, join, and depart multicast groups.	Local: 68 Remote: 67	2	Out
	Core Networking - IPHTTPS (TCP-In)	Inbound TCP rule to allow IPHTTPS tunneling technology to provide connectivity across HTTP proxies and firewalls.	Local: IPHTPS Remote: Any	TCP	In

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - IPHTTPS (TCP-Out)	Outbound TCP rule to allow IPHTTPS tunneling technology to provide connectivity across HTTP proxies and firewalls.	Local: Any Remote: IPHTPS	TCP	Out
	IPv6 (IPv6-In)	Inbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.	Local: Any Remote: 445	41	In

OS	Rule	Definition	Port	Protocol	Direction
	IPv6 (IPv6-Out)	Outbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressin g Protocol) and 6to4 tunneling services.	Local: Any Remote: 445	41	Out
	Multicast Listener Done (ICMPv6-In)	Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Done (ICMPv6-Out)	Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.	Local: 68 Remote: 67	ICMPv6	Out
	Multicast Listener Query (ICMPv6-In)	An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Query (ICMPv6-Out)	An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report (ICMPv6-In)	The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report (ICMPv6-Out)	The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report v2 (ICMPv6-In)	Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report v2 (ICMPv6-Out)	Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Advertisement (ICMPv6-In)	Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Advertisement (ICMPv6-Out)	Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.	Local: 68 Remote: 67	ICMPv6	Out
	Neighbor Discovery Solicitation (ICMPv6-In)	Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Solicitation (ICMPv6-Out)	Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.	Local: 68 Remote: 67	ICMPv6	Out
	Packet Too Big (ICMPv6-In)	Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Packet Too Big (ICMPv6-Out)	Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.	Local: 68 Remote: 67	ICMPv6	Out
	Parameter Problem (ICMPv6-In)	Parameter Problem error messages are sent by nodes when packets are incorrectly generated.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Parameter Problem (ICMPv6-Out)	Parameter Problem error messages are sent by nodes when packets are incorrectly generated.	Local: 68 Remote: 67	ICMPv6	Out
	Router Advertisement (ICMPv6-In)	Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.	Local: 68 Remote: 67	ICMPv6	In
	Router Advertisement (ICMPv6-Out)	Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Router Solicitation (ICMPv6-In)	Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-configuration.	Local: 68 Remote: 67	ICMPv6	In
	Router Solicitation (ICMPv6-Out)	Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-configuration.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - Teredo (UDP-In)	Inbound UDP rule to allow Teredo edge traversal. This technology provides address assignment and automatic tunneling for unicast IPv6 traffic when an IPv6/IPv4 host is located behind an IPv4 network address translator.	Local: Teredo Remote: Any	UDP	In

OS	Rule	Definition	Port	Protocol	Direction
	Core Networking - Teredo (UDP-Out)	Outbound UDP rule to allow Teredo edge traversal. This technology provides address assignment and automatic tunneling for unicast IPv6 traffic when an IPv6/IPv4 host is located behind an IPv4 network address translator.	Local: Any Remote: Any	UDP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Time Exceeded (ICMPv6-In)	Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.	Local: 68 Remote: 67	ICMPv6	In
	Time Exceeded (ICMPv6-Out)	Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.	Local: 68 Remote: 67	ICMPv6	Out

Windows Server 2008 R2 and SP2

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 R2 Windows Server 2008 SP2	Destination Unreachable (ICMPv6-In)	Destination Unreachable error messages are sent from any node that a packet traverses which is unable to forward the packet for any reason except congestion.	Local: 68 Remote: 67	ICMPv6	In
	Destination Unreachable Fragmentation Needed (ICMPv4-In)	Destination Unreachable Fragmentation Needed error messages are sent from any node that a packet traverses which is unable to forward the packet because	Local: 68 Remote: 67	ICMPv4	In

OS	Rule	Definition	Port	Protocol	Direction
		fragmentation was needed and the don't fragment bit was set.			
	Dynamic Host Configuration Protocol (DHCP-In)	Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.	Local: 68 Remote: 67	UDP	In
	Dynamic Host Configuration Protocol (DHCP-Out)	Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.	Local: 68 Remote: 67	UDP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)	Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.	Local: 546 Remote: 547	UDP	In
	Dynamic Host Configuration Protocol for IPv6(DHCPV6-Out)	Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.	Local: 546 Remote: 547	UDP	Out
	Internet Group Management Protocol (IGMP-In)	IGMP messages are sent and received by nodes to create, join, and depart multicast groups.	Local: 68 Remote: 67	2	In

OS	Rule	Definition	Port	Protocol	Direction
	IPv6 (IPv6-In)	Inbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.	Local: Any Remote: 445	41	In
	IPv6 (IPv6-Out)	Outbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.	Local: Any Remote: 445	41	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Done (ICMPv6-In)	Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.	Local: 68 Remote: 67	ICMPv6	In
	Multicast Listener Done (ICMPv6-Out)	Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Query (ICMPv6-In)	An IPv6 multicast -capable router uses the Multicast Listener Query message to query a link for multicast group membership.	Local: 68 Remote: 67	ICMPv6	In
	Multicast Listener Query (ICMPv6-Out)	An IPv6 multicast -capable router uses the Multicast Listener Query message to query a link for multicast group membership.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report (ICMPv6-In)	The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report (ICMPv6-Out)	The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report v2 (ICMPv6-In)	Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Multicast Listener Report v2 (ICMPv6-Out)	Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Advertise ment (ICMPv6-In)	Neighbor Discovery Advertise ment messages are sent by nodes to notify other nodes of link-laye r address changes or in response to a Neighbor Discovery Solicitation request.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Advertisement (ICMPv6-Out)	Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.	Local: 68 Remote: 67	ICMPv6	Out
	Neighbor Discovery Solicitation (ICMPv6-In)	Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Neighbor Discovery Solicitation (ICMPv6-Out)	Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.	Local: 68 Remote: 67	ICMPv6	Out
	Packet Too Big (ICMPv6-In)	Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Packet Too Big (ICMPv6-Out)	Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.	Local: 68 Remote: 67	ICMPv6	Out
	Parameter Problem (ICMPv6-In)	Parameter Problem error messages are sent by nodes when packets are incorrectly generated.	Local: 68 Remote: 67	ICMPv6	In

OS	Rule	Definition	Port	Protocol	Direction
	Parameter Problem (ICMPv6-Out)	Parameter Problem error messages are sent by nodes when packets are incorrectly generated.	Local: 68 Remote: 67	ICMPv6	Out
	Router Advertisement (ICMPv6-In)	Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.	Local: 68 Remote: 67	ICMPv6	In
	Router Advertisement (ICMPv6-Out)	Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Router Solicitation (ICMPv6-In)	Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-conf iguration.	Local: 68 Remote: 67	ICMPv6	In
	Router Solicitation (ICMPv6-O ut)	Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-conf iguration.	Local: 68 Remote: 67	ICMPv6	Out

OS	Rule	Definition	Port	Protocol	Direction
	Time Exceeded (ICMPv6-In)	Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.	Local: 68 Remote: 67	ICMPv6	In
	Time Exceeded (ICMPv6-Out)	Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.	Local: 68 Remote: 67	ICMPv6	Out

Delivery Optimization

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2019 Windows Server 2022	DeliveryOptimization-TCP-In	Inbound rule to allow Delivery Optimization to connect to remote endpoints.	Local: 7680 Remote: Any	TCP	In
	DeliveryOptimization-UDP-In	Inbound rule to allow Delivery Optimization to connect to remote endpoints.	Local: 7680 Remote: Any	UDP	In

Diag Track

Windows Server 2019 and 2022

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2019 Windows Server 2022	Connected User Experiences and Telemetry	Unified Telemetry Client Outbound Traffic.	Local: Any Remote: 443	TCP	Out

Windows Server 2016

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2016	Connected User Experiences and Telemetry	Unified Telemetry Client Outbound Traffic.	Local: Any Remote: Any	TCP	Out

DIAL Protocol Server

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2016 Windows Server 2019 Windows Server 2022	DIAL protocol server (HTTP-In)	Inbound rule for DIAL protocol server to allow remote control of Apps using HTTP.	Local: 10247 Remote: Any	TCP	In

Distributed File System (DFS) Management

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 R2	DFS Management (SMB-In)	Inbound rule to allow SMB traffic to manage the File Services role.	Local: 445 Remote: Any	TCP	In

OS	Rule	Definition	Port	Protocol	Direction
	DFS Management (WMI-In)	Inbound rule to allow WMI traffic to manage the File Services role.	Local: RPC Remote: Any	TCP	In
	DFS Management (DCOM-In)	Inbound rule to allow DCOM traffic to manage the File Services role.	Local: 135 Remote: Any	TCP	In
	DFS Management (TCP-In)	Inbound rule to allow TCP traffic to manage the File Services role.	Local: RPC Remote: Any	TCP	In

File and Printer Sharing

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 R2 Windows Server 2008 SP2 Windows Server 2012	File and Printer Sharing (Echo Request - ICMPv4-In)	Echo Request messages are sent as ping requests to other nodes.	Local: 5355 Remote: Any	ICMPv4	In
	File and Printer	Echo Request messages are	Local: 5355 Remote: Any	ICMPv4	Out

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2012 R2	Sharing (Echo Request - ICMPv4-Out)	sent as ping requests to other nodes.			
	File and Printer Sharing (Echo Request - ICMPv6-In)	Echo Request messages are sent as ping requests to other nodes.	Local: 5355 Remote: Any	ICMPv6	In
	File and Printer Sharing (Echo Request - ICMPv6-Out)	Echo Request messages are sent as ping requests to other nodes.	Local: 5355 Remote: Any	ICMPv6	Out
	File and Printer Sharing (LLMNR-UDP-In)	Inbound rule for File and Printer Sharing to allow Link Local Multicast Name Resolution.	Local: 5355 Remote: Any	UDP	In

OS	Rule	Definition	Port	Protocol	Direction
	File and Printer Sharing (LLMNR-UDP-Out)	Outbound rule for File and Printer Sharing to allow Link Local Multicast Name Resolution.	Local: Any Remote: 5355	UDP	Out
	File and Printer Sharing (NB-Datagram-In)	Inbound rule for File and Printer Sharing to allow NetBIOS Datagram transmission and reception.	Local: 138 Remote: Any	UDP	In
	File and Printer Sharing (NB-Datagram-Out)	Outbound rule for File and Printer Sharing to allow NetBIOS Datagram transmission and reception.	Local: Any Remote: 138	UDP	Out

OS	Rule	Definition	Port	Protocol	Direction
	File and Printer Sharing (NB-Name-In)	Inbound rule for File and Printer Sharing to allow NetBIOS Name Resolution.	Local: 137 Remote: Any	UDP	In
	File and Printer Sharing (NB-Name-Out)	Outbound rule for File and Printer Sharing to allow NetBIOS Name Resolution.	Local: Any Remote: 137	UDP	Out
	File and Printer Sharing (NB-Session-In)	Inbound rule for File and Printer Sharing to allow NetBIOS Session Service connections.	Local: 139 Remote: Any	TCP	In

OS	Rule	Definition	Port	Protocol	Direction
	File and Printer Sharing (NB-Session-Out)	Outbound rule for File and Printer Sharing to allow NetBIOS Session Service connections.	Local: Any Remote: 139	TCP	Out
	File and Printer Sharing (SMB-In)	Inbound rule for File and Printer Sharing to allow Server Message Block transmission and reception via Named Pipes.	Local: 445 Remote: Any	TCP	In
	File and Printer Sharing (SMB-Out)	Outbound rule for File and Printer Sharing to allow Server Message Block transmission and reception via Named Pipes.	Local: Any Remote: 445	TCP	Out

OS	Rule	Definition	Port	Protocol	Direction
	File and Printer Sharing (Spooler Service - RPC)	Inbound rule for File and Printer Sharing to allow the Print Spooler Service to communicate via TCP/RPC.	Local: RPC Remote: Any	TCP	In
	File and Printer Sharing (Spooler Service - RPC-EPMAP)	Inbound rule for the RPCSS service to allow RPC/TCP traffic for the Spooler Service.	Local: RPC-EPMap Remote: Any	TCP	In

File Server Remote Management

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 SP2 Windows Server 2012	File Server Remote Management (DCOM-In)	Inbound rule to allow DCOM traffic to manage the File Services role.	Local: 135 Remote: Any	TCP	In
Windows Server 2012 R2	File Server Remote	Inbound rule to allow SMB traffic to	Local: 445 Remote: Any	TCP	In

OS	Rule	Definition	Port	Protocol	Direction
	Management (SMB-In)	manage the File Services role.			
	WMI-In	Inbound rule to allow WMI traffic to manage the File Services role.	Local: RPC Remote: Any	TCP	In

ICMP v4 All

OS	Rule	Port	Protocol	Direction
Windows Server 2012 Windows Server 2012 R2	All ICMP v4	Local: 139 Remote: Any	ICMPv4	In

Microsoft Edge

OS	Rule	Port	Protocol	Direction
Windows Server 2022	Microsoft Edge (mDNS-In)	Local: 5353 Remote: Any	UDP	In

Microsoft Media Foundation Network Source

OS	Rule	Port	Protocol	Direction
Windows Server 2022	Microsoft Media Foundation Network Source IN [TCP 554]	Local: 554, 8554-8558 Remote: Any	TCP	In
	Microsoft Media Foundatio n Network Source IN [UDP 5004-5009]	Local: 5000-5020 Remote: Any	UDP	In
	Microsoft Media Foundation Network Source OUT [TCP ALL]	Local: Any Remote: 554, 8554-8558	TCP	In

Multicast

Windows Server 2019 and 2022

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2019 Windows Server 2022	mDNS (UDP-In)	Inbound rule for mDNS traffic.	Local: 5353 Remote: Any	UDP	In
	mDNS (UDP-Out)	Outbound rule for mDNS traffic.	Local: Any Remote: 5353	UDP	Out

Windows Server 2016

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2016	mDNS (UDP-In)	Inbound rule for mDNS traffic.	Local: mDNS Remote: Any	UDP	In
	mDNS (UDP-Out)	Outbound rule for mDNS traffic.	Local: 5353 Remote: Any	UDP	Out

Remote Desktop

Windows Server 2012 R2, 2016, 2019, and 2022

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2012 R2 Windows Server 2016 Windows Server 2019 Windows Server 2022	Remote Desktop - Shadow (TCP-In)	Inbound rule for the Remote Desktop service to allow shadowing of an existing Remote Desktop session.	Local: Any Remote: Any	TCP	In
	Remote Desktop - User Mode (TCP-In)	Inbound rule for the Remote Desktop service to	Local: 3389 Remote: Any	TCP	In

OS	Rule	Definition	Port	Protocol	Direction
		allow RDP traffic.			
	Remote Desktop - User Mode (UDP-In)	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	UDP	In

Windows Server 2012

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2012	Remote Desktop - User Mode (TCP-In)	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	TCP	In
	Remote Desktop - User Mode (UDP-In)	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	UDP	In

Windows Server 2008 SP2

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 SP2	Remote Desktop - Shadow (TCP-In)	Inbound rule for the Remote Desktop service to allow shadowing of an existing Remote Desktop session.	Local: Any Remote: Any	TCP	In
	Remote Desktop - User Mode (TCP-In)	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	TCP	In
	Remote Desktop - User Mode (UDP-In)	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	UDP	In

Windows Server 2008 R2

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 R2	RemoteFX (TCP-In)	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	TCP	In
	TCP-In	Inbound rule for the Remote Desktop service to allow RDP traffic.	Local: 3389 Remote: Any	TCP	In

Windows Device Management

Windows Server 2022

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2022	Windows Device Managemen t Certifica te Installer (TCP out)	Allow outbound TCP traffic from Windows Device Managemen t Certificate Installer.	Local: Any Remote: Any	TCP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Windows Device Managemen t Device Enroller (TCP out)	Allow outbound TCP traffic from Windows Device Managemen t Device Enroller.	Local: Any Remote: 80, 443	TCP	Out
	Windows Device Managemen t Enrollment Service (TCP out)	Allow outbound TCP traffic from Windows Device Managemen t Enrollment Service.	Local: Any Remote: Any	TCP	Out
	Windows Device Managemen t Sync Client (TCP out)	Allow outbound TCP traffic from Windows Device Managemen t Sync Client.	Local: Any Remote: Any	TCP	Out

Windows Server 2019

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2019	Windows Device	Allow outbound	Local: Any	TCP	Out

OS	Rule	Definition	Port	Protocol	Direction
	Management Certificate Installer (TCP out)	TCP traffic from Windows Device Management Certificate Installer.	Remote: Any		
	Windows Device Management Enrollment Service (TCP out)	Allow outbound TCP traffic from Windows Device Management Enrollment Service.	Local: Any Remote: Any	TCP	Out
	Windows Device Management Sync Client (TCP out)	Allow outbound TCP traffic from Windows Device Management Sync Client.	Local: Any Remote: Any	TCP	Out
	Windows Enrollment WinRT (TCP Out)	Allow outbound TCP traffic from Windows Enrollment WinRT.	Local: Any Remote: Any	TCP	Out

Windows Feature Experience Pack

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2022	Windows Feature Experience Pack	Windows Feature Experience Pack.		Any	Out

Windows Firewall Remote Management

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 SP2 Windows Server 2012 R2	Windows Firewall Remote Management (RPC)	Inbound rule for the Windows Firewall to be remotely managed via RPC/TCP.	Local: RPC Remote: Any	TCP	In
	Windows Firewall Remote Management (RPC-EPMAP)	Inbound rule for the RPCSS service to allow RPC/TCP traffic for the Windows Firewall.	Local: RPC-EPMap Remote: Any	TCP	In

Windows Remote Management

OS	Rule	Definition	Port	Protocol	Direction
Windows Server 2008 R2 Windows Server 2008 SP2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019 Windows Server 2022	Windows Remote Management (HTTP-In)	Inbound rule for Windows Remote Managemen t via WS-Manage ment.	Local: 5985 Remote: Any	TCP	In

For more information about Amazon EC2 security groups, see Amazon EC2 Security Groups for Windows Instances.

Security in Amazon EC2

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon EC2, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility includes the following areas:
  - Controlling network access to your instances, for example, through configuring your VPC and security groups. For more information, see Controlling network traffic.
  - Managing the credentials used to connect to your instances.
  - Managing the guest operating system and software deployed to the guest operating system, including updates and security patches. For more information, see Update management 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.

For security best practices for Amazon EC2 running Windows Server, see **Security and Network** under Best practices for Windows on Amazon EC2.

Contents

- Infrastructure security in Amazon EC2
- Resilience in Amazon EC2

- Data protection in Amazon EC2
- Windows virtualization-based security features
- Identity and access management for Amazon EC2
- Amazon EC2 key pairs and Amazon EC2 instances
- Amazon EC2 security groups for Windows instances
- Access Amazon EC2 using an interface VPC endpoint
- Configuration management in Amazon EC2
- Update management in Amazon EC2
- Change management in Amazon EC2
- Compliance validation for Amazon EC2
- Audit and accountability in 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. Configure Amazon EC2 instance security groups to permit the minimum required network traffic for the Amazon EC2instance and to allow access only from defined, expected, and approved locations. For example, if an Amazon EC2

instance is an IIS web server, configure its security groups to permit only inbound HTTP/HTTPS, Windows management traffic, and minimal outbound connections.

- Leverage security groups as the primary mechanism for controlling network access to Amazon EC2 instances. When necessary, use network ACLs sparingly to provide stateless, coarse-grain network control. Security groups are more versatile than network ACLs due to their ability to perform stateful packet filtering and create rules that reference other security groups. However, network ACLs can be effective as a secondary control for denying a specific subset of traffic or providing high-level subnet guard rails. Also, because network ACLs apply to an entire subnet, they can be used as defense-in-depth in case an instance is ever launched unintentionally without a correct security group.
- Centrally manage Windows Firewall settings with Group Policy Objects (GPO) to further enhance network controls. Customers often use the Windows Firewall for further visibility into network traffic and to complement security group filters, creating advanced rules to block specific applications from accessing the network or to filter traffic from a subset IP addresses. For example, the Windows Firewall can limit access to the EC2 metadata service IP address to specific users or applications. Alternatively, a public-facing service might use security groups to restrict traffic to specific ports and the Windows Firewall to maintain a list of explicitly blocked IP addresses.
- When managing Windows instances, limit access to a few well-defined centralized management servers or bastion hosts to reduce the environment's attack surface. Also, use secure administration protocols like RDP encapsulation over SSL/TLS. The Remote Desktop Gateway Quick Start provides best practices for deploying remote desktop gateway, including configuring RDP to use SSL/TLS.
- Use Active Directory or AWS Directory Service to tightly and centrally control and monitor interactive user and group access to Windows instances, and avoid local user permissions. Also avoid using Domain Administrators and instead create more granular, application-specific role-based accounts. Just Enough Administration (JEA) allows changes to Windows instances to be managed without interactive or administrator access. In addition, JEA enables organizations to lock down administrative access to the subset of Windows PowerShell commands required for instance administration. For additional information, see the section on "Managing OS-level Access to Amazon EC2" in the AWS Security Best Practices whitepaper.
- Systems Administrators should use Windows accounts with limited access to perform daily activities, and only elevate access when necessary to perform specific configuration changes. Additionally, only access Windows instances directly when absolutely necessary. Instead, leverage central configuration management systems such as EC2 Run Command, Systems Center

Configuration Manager (SCCM), Windows PowerShell DSC, or Amazon EC2 Systems Manager (SSM) to push changes to Windows servers.

- Configure Amazon VPC subnet route tables with the minimal required network routes. For example, place only Amazon EC2 instances that requite direct Internet access into subnets with routes to an Internet Gateway, and place only Amazon EC2 instances that need direct access to internal networks into subnets with routes to a virtual private gateway.
- Consider using additional security groups or ENIs to control and audit Amazon EC2 instance management traffic separately from regular application traffic. This approach allows customers to implement special IAM policies for change control, making it easier to audit changes to security group rules or automated rule-verification scripts. Multiple ENIs also provide additional options for controlling network traffic including the ability to create host-based routing policies or leverage different VPC subnet routing rules based on an ENI's assigned subnet.
- Use AWS Virtual Private Network or AWS Direct Connect to establish private connections from your remote networks to your VPCs. For more information, see Network-to-Amazon VPC Connectivity Options.
- Use VPC Flow Logs to monitor the traffic that reaches your instances.
- Use AWS Security Hub to check for unintended network accessibility from your instances.
- Use AWS Systems Manager Session Manager to access your instances remotely instead of opening inbound RDP ports.
- Use AWS Systems Manager Run Command to automate common administrative tasks instead of opening inbound RDP ports.
- Many of the Windows OS roles and Microsoft business applications also provide enhanced functionality such as IP Address Range restrictions within IIS, TCP/IP filtering policies in Microsoft SQL Server, and connection filter policies in Microsoft Exchange. Network restriction functionality within the application layer can provide additional layers of defense for critical business application servers.

In addition to restricting network access to each Amazon EC2 instance, Amazon VPC supports implementing additional network security controls like in-line gateways, proxy servers, and various network monitoring options.

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*.

You can also use Microsoft EFS and NTFS permissions for folder- and file-level encryption.

Instance store volumes

The data on NVMe instance store volumes is encrypted using an XTS-AES-256 cipher, implemented on a hardware module on the instance. The keys used to encrypt data that's written to locally-attached NVMe storage devices are per-customer, and per volume. The keys are generated by, and only reside within, the hardware module, which is inaccessible to AWS personnel. The encryption keys are destroyed when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

The data on HDD instance store volumes on H1, D3, and D3en instances is encrypted using XTS-AES-256 and one-time keys.

When you stop, hibernate, or terminate an instance, every block of storage in the instance store volume is reset. Therefore, your data cannot be accessed through the instance store of another instance.

Memory

Memory encryption is enabled on the following instances:

- Instances with AWS 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 and Transit Gateway cross-Region peering

All cross-Region traffic that uses Amazon VPC and Transit Gateway peering is automatically bulk-encrypted when it exits a Region. An additional layer of encryption is automatically provided at the physical layer for all cross-Region traffic, as previously noted in this section.

Encryption between instances

AWS provides secure and private connectivity between EC2 instances of all types. In addition, some instance types use the offload capabilities of the underlying Nitro System hardware to automatically encrypt in-transit traffic between instances. 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, M7i, M7i-flex

- **Compute optimized:** C5a, C5ad, C5n, C6a, C6i, C6id, C6in, C7a, C7i
- **Memory optimized:** R5dn, R5n, R6a, R6i, R6idn, R6in, R6id, R7a, R7i, R7iz, U-3tb1, U-6tb1, U-9tb1, U-12tb1, U-18tb1, U-24tb1, X2idn, X2iedn, X2iezn
- **Storage optimized:** D3, D3en, I3en, I4i
- **Accelerated computing:** G4ad, G4dn, G5, P3dn
- **High-performance computing:** Hpc6id, Hpc7a

- 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

RDP provides a secure communications channel for remote access to your Windows instances, whether directly or through EC2 Instance Connect. Remote access to your instances using AWS Systems Manager Session Manager or the Run Command is encrypted using TLS 1.2, and requests to create a connection are signed using SigV4 and authenticated and authorized by AWS Identity and Access Management.

It is your responsibility to use an encryption protocol, such as Transport Layer Security (TLS), to encrypt sensitive data in transit between clients and your Amazon EC2 instances.

Make sure to allow only encrypted connections between EC2 instances and the AWS API endpoints or other sensitive remote network services. You can enforce this through an outbound security group or Windows Firewall rules.

Windows virtualization-based security features

With the AWS Nitro System, you can enable certain Windows virtualization-based security (VBS) features. VBS is a suite of Windows security mechanisms that use hardware virtualization features to create an isolated compute environment. Currently, only Credential Guard is supported. For more information, see AWS Nitro System.

Topics

- Credential Guard

Credential Guard

The AWS Nitro System supports Credential Guard for Amazon Elastic Compute Cloud (Amazon EC2) Windows instances. Credential Guard is a Windows virtualization-based security (VBS) feature that enables the creation of isolated environments to protect security assets, such as Windows user credentials and code integrity enforcement, beyond Windows kernel protections. When you run EC2 Windows instances, Credential Guard uses the AWS Nitro System to protect Windows login credentials from being extracted from the OS memory.

Topics

- Prerequisites
- Launching a supported instance
- Disabling memory integrity
- Turning on Credential Guard
- Verifying Credential Guard is running

Prerequisites

Your Windows instance must meet the following prerequisites to utilize Credential Guard:

Amazon Machine Images (AMIs)

The AMI must be preconfigured to enable NitroTPM and UEFI Secure Boot. For more information on supported AMIs, see Prerequisites for launching a Windows instance with NitroTPM enabled.

Memory integrity

Memory integrity, also known as *hypervisor-protected code integrity (HVCI)* or *hypervisor enforced code integrity*, isn't supported. Before you turn on Credential Guard, you must ensure this feature is disabled. For more information, see Disabling memory integrity.

Instance types

The following instance types support Credential Guard across all sizes: `C5`, `C5d`, `C5n`, `C6i`, `C6id`, `C6in`, `M5`, `M5d`, `M5dn`, `M5n`, `M5zn`, `M6i`, `M6id`, `M6idn`, `M6in`, `R5`, `R5b`, `R5d`, `R5dn`, `R5n`, `R6i`, `R6id`, `R6idn`, `R6in`.

> **Note**
>
> Though NitroTPM has some required instance types in common, the instance type must be one of the above to support Credential Guard.

Launching a supported instance

You can use the Amazon EC2 console or AWS Command Line Interface (AWS CLI) to launch an instance which can support Credential Guard. You will need a compatible AMI ID for launching your instance which is unique for each AWS Region.

> **Tip**
>
> You can use the following link to discover and launch instances with compatible Amazon provided AMIs in the Amazon EC2 console: https://console.aws.amazon.com/ec2/v2/home?#Images:visibility=public-images;v=3;search=:TPM-Windows_Server;ownerAlias=amazon

Amazon EC2 console

To launch an instance using the Amazon EC2 console

Follow the steps to Launch an instance using the new launch instance wizard while specifying a supported instance type and preconfigured Windows AMI.

AWS CLI

To launch an instance using the AWS CLI

Use the run-instances command to launch an instance using a supported instance type and preconfigured Windows AMI.

```
aws ec2 run-instances \
    --image-id resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-English-Full-Base \
    --instance-type c6i.large \
    --region us-east-1 \
    --subnet-id subnet-id
    --key-name key-name
```

PowerShell

To launch an instance using the AWS Tools for PowerShell

Use the New-EC2Instance command to launch an instance using a supported instance type and preconfigured Windows AMI.

```
New-EC2Instance `
    -ImageId resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-English-Full-Base `
    -InstanceType c6i.large `
    -Region us-east-1 `
    -SubnetId subnet-id `
    -KeyName key-name
```

Disabling memory integrity

You can use the Local Group Policy Editor to disable memory integrity in supported scenarios. The following guidance can be applied for each configuration setting under **Virtualization Based Protection of Code Integrity**:

- **Enabled without lock** – Modify the setting to **Disabled** to disable memory integrity.

- **Enabled with UEFI lock** – Memory integrity has been enabled with UEFI lock. Memory integrity can't be disabled once it has been enabled with UEFI lock. We recommend creating a new instance with memory integrity disabled and terminating the unsupported instance if it's not in use.

To disable memory integrity with the Local Group Policy Editor

1. Connect to your instance as a user account with administrator privileges using the Remote Desktop Protocol (RDP). For more information, see Connect to your Windows instance using RDP.
2. Open the Start menu and search for **cmd** to start a command prompt.
3. Run the following command to open the Local Group Policy Editor: `gpedit.msc`
4. In the Local Group Policy Editor, choose **Computer Configuration**, **Administrative Templates**, **System**, **Device Guard**.
5. Select **Turn On Virtualization Based Security**, then select **Edit policy setting**.
6. Open the settings drop-down for **Virtualization Based Protection of Code Integrity**, choose **Disabled**, then choose **Apply**.
7. Reboot the instance to apply the changes.

Turning on Credential Guard

After you have launched a Windows instance with a supported instance type and compatible AMI, and confirmed that memory integrity is disabled, you can turn on Credential Guard.

> **Important**
>
> Administrator privileges are required to perform the following steps to turn on Credential Guard.

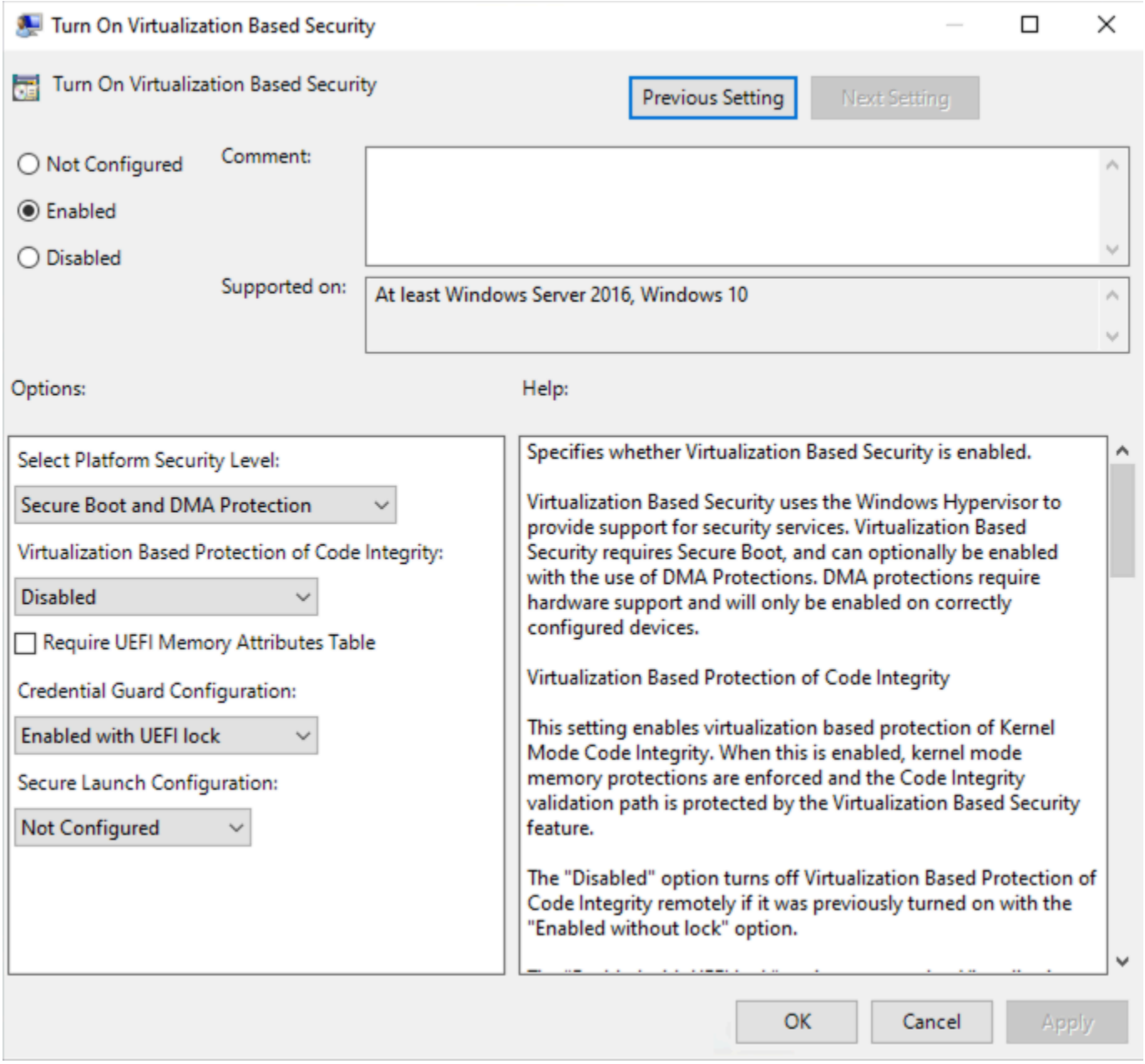
To turn on Credential Guard

1. Connect to your instance as a user account with administrator privileges using the Remote Desktop Protocol (RDP). For more information, see Connect to your Windows instance using RDP.
2. Open the Start menu and search for **cmd** to start a command prompt.

3. Run the following command to open the Local Group Policy Editor: `gpedit.msc`
4. In the Local Group Policy Editor, choose **Computer Configuration**, **Administrative Templates**, **System**, **Device Guard**.
5. Select **Turn On Virtualization Based Security**, then select **Edit policy setting**.
6. Choose **Enabled** within the **Turn On Virtualization Based Security** menu.
7. For **Select Platform Security Level**, choose **Secure Boot and DMA Protection**.
8. For **Credential Guard Configuration**, choose **Enabled with UEFI lock**.

> **Note**
>
> The remaining policy settings are not required to enable Credential Guard and can be left as **Not Configured**.

The following image displays the VBS settings configured as described previously:



9. Reboot the instance to apply the settings.

Verifying Credential Guard is running

You can use the Microsoft System Information (`Msinfo32.exe`) tool to confirm that Credential Guard is running.

> **Important**
>
> You must first reboot the instance to finish applying the policy settings required to enable Credential Guard.

To verify Credential Guard is running

1. Connect to your instance using the Remote Desktop Protocol (RDP). For more information, see Connect to your Windows instance using RDP.
2. Within the RDP session to your instance, open the Start menu and search for **`cmd`** to start a command prompt.
3. Open System Information by running the following command: `msinfo32.exe`
4. The Microsoft System Information tool lists the details for VBS configuration. Next to Virtualization-based security Services, confirm that **Credential Guard** appears as **Running**.

   The following image displays VBS is running as described previously:

Virtualization-based security	Running
Virtualization-based security Required Security Properties	Base Virtualization Support, Secure Boot, DMA Protection
Virtualization-based security Available Security Properties	Base Virtualization Support, Secure Boot, DMA Protection, UEFI Code Readonly, Mode Based Execution Control
Virtualization-based security Services Configured	Credential Guard
Virtualization-based security Services Running	Credential Guard

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
- Authorize inbound traffic for your Windows instances

Network access to your instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

For more information, see Authorize inbound traffic for your Windows instances.

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 per missions
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:`*`Resource`*`ID` condition keys with the specified API actions, the condition key value is used to specify the resulting resource that is created by the API action. `ec2:`*`Resource`*`ID` condition keys can't be used to specify a source resource that is specified in the API request. If you use one of the following `ec2:`*`Resource`*`ID` condition keys with a specified API, then you must always specify the wildcard (*). If you specify a different value, the condition always resolves to `*` during runtime. For example, to use the `ec2:ImageId` condition key with the **CopyImage** API, then you must specify the condition key as follows:

```
{
  "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`	• CreateDhcpOptions			
`ec2:ImageID`	• CopyImage • CreateImage • ImportImage • RegisterImage			
`ec2:InstanceID`	• RunInstances • ImportInstance			
`ec2:InternetGatewayID`	• CreateInternetGateway			
`ec2:NetworkAclID`	• CreateNetworkAcl			
`ec2:NetworkInterfaceID`	• CreateNetworkInterface			

Condition key	API action			
`ec2:PlacementGroupName`	• CreatePlacementGroup			
`ec2:RouteTableID`	• CreateRouteTable			
`ec2:SecurityGroupID`	• CreateSecurityGroup			
`ec2:SnapshotID`	• CopySnapshot • CreateSnapshot • CreateSnapshots • ImportSnapshots			
`ec2:SubnetID`	• CreateSubnet			
`ec2:VolumeID`	• CreateVolume • ImportVolume			

Condition key	API action			
ec2:VpcID	• CreateVpc			
ec2:VpcPeeringConnectionID	• CreateVpcPeeringConnection			

We recommend that you avoid using `ec2:Resource ID` 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=Webservers`. Users cannot override any of the launch template parameters in the `RunInstances` action.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": "arn:aws:ec2:region:account-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 allowe 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.

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.

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.

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.

AWS managed policy: EC2FastLaunchServiceRolePolicy

This policy is attached to the service-linked role named **AWSServiceRoleForEC2FastLaunch** to allow Amazon EC2 to create and manage a set of pre-provisioned snapshots that reduce the time it takes to launch instances from your Windows faster launching-enabled AMI. For more information, see the section called “Service-linked role”.

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
EC2FastLaunchServiceRolePolicy – New policy	Amazon EC2 added the Windows faster launching	November 26, 2021

Change	Description	Date
	feature to enable Windows AMIs to launch instances faster by creating a set of pre-provisioned snapshots.	
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*.

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

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri 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

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

   ```
   PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}
    -Method GET -Uri http://169.254.169.254/latest/meta-data/iam/security-
   credentials/role_name
   ```

   IMDSv1

   ```
   PS C:\> Invoke-RestMethod -uri 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

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

   ```
   PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}
    -Method GET -Uri http://169.254.169.254/latest/meta-data/iam/security-
   credentials/role_name
   ```

   IMDSv1

   ```
   PS C:\> Invoke-RestMethod -uri 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": "AIPAJLNLDX3AMYZNWYYAY",
            "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*.

Authorize inbound traffic for your Windows instances

Security groups enable you to control traffic to your instance, including the kind of traffic that can reach your instance. For example, you can allow computers from only your home network to access your instance using RDP. If your instance is a web server, you can allow all IP addresses to access your instance using HTTP or HTTPS, so that external users can browse the content on your web server.

Your default security groups and newly created security groups include default rules that do not enable you to access your instance from the internet. For more information, see Default security groups and Custom security groups. To enable network access to your instance, you must allow inbound traffic to your instance. To open a port for inbound traffic, add a rule to a security group that you associated with your instance when you launched it.

To connect to your instance, you must set up a rule to authorize RDP traffic from your computer's public IPv4 address. To allow RDP traffic from additional IP address ranges, add another rule for each range you need to authorize.

If you've enabled your VPC for IPv6 and launched your instance with an IPv6 address, you can connect to your instance using its IPv6 address instead of a public IPv4 address. Your local computer must have an IPv6 address and must be configured to use IPv6.

If you need to enable network access to a Linux instance, see Authorizing inbound traffic for your Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

Before you start

Decide who requires access to your instance; for example, a single host or a specific network that you trust such as your local computer's public IPv4 address. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address of your local computer for you. Alternatively, you can use the search phrase "what is my IP address" in an internet browser, or use the following service: Check IP. If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

> **Warning**
>
> If you use `0.0.0.0/0`, you enable all IPv4 addresses to access your instance using RDP. If you use `::/0`, you enable all IPv6 addresses to access your instance. You should authorize only a specific IP address or range of addresses to access your instance.

Windows Firewall may also block incoming traffic. If you're having trouble setting up access to your instance, you may have to disable Windows Firewall. For more information, see Remote Desktop can't connect to the remote computer.

Add a rule for inbound RDP traffic to a Windows instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group to enable you to connect to your Windows instance from your IP address using RDP.

To add a rule to a security group for inbound RDP traffic over IPv4 (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the top navigation bar, select a Region for the security group. Security groups are specific to a Region, so you should select the same Region in which you created your instance.
3. In the navigation pane, choose **Instances**.
4. Select your instance and, in bottom half of the screen, choose the **Security** tab. **Security groups** lists the security groups that are associated with the instance. **Inbound rules** displays a list of the inbound rules that are in effect for the instance.
5. For the security group to which you'll add the new rule, choose the security group ID link to open the security group.
6. On the **Inbound rules** tab, choose **Edit inbound rules**.
7. On the **Edit inbound rules** page, do the following:
   a. Choose **Add rule**.
   b. For **Type**, choose **RDP**.
   c. For **Source**, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer.

Alternatively, for **Source**, choose **Custom** and enter the public IPv4 address of your computer or network in CIDR notation. For example, if your IPv4 address is `203.0.113.25`, enter `203.0.113.25/32` to list this single IPv4 address in CIDR notation. If your company allocates addresses from a range, enter the entire range, such as `203.0.113.0/24`.

For information about finding your IP address, see Before you start.

d. Choose **Save rules**.

If you launched an instance with an IPv6 address and want to connect to your instance using its IPv6 address, you must add rules that allow inbound IPv6 traffic over RDP.

To add a rule to a security group for inbound RDP traffic over IPv6 (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the top navigation bar, select a Region for the security group. Security groups are specific to a Region, so you should select the same Region in which you created your instance.
3. In the navigation pane, choose **Instances**.
4. Select your instance and, in bottom half of the screen, choose the **Security** tab. **Security groups** lists the security groups that are associated with the instance. **Inbound rules** displays a list of the inbound rules that are in effect for the instance.
5. For the security group to which you'll add the new rule, choose the security group ID link to open the security group.
6. On the **Inbound rules** tab, choose **Edit inbound rules**.
7. On the **Edit inbound rules** page, do the following:

   a. Choose **Add rule**.

   b. For **Type**, choose **RDP**.

   c. For **Source**, choose **Custom** and enter the IPv6 address of your computer in CIDR notation. For example, if your IPv6 address is `2001:db8:1234:1a00:9691:9503:25ad:1761`, enter `2001:db8:1234:1a00:9691:9503:25ad:1761/128` to list the single IP address in CIDR notation. If your company allocates addresses from a range, enter the entire range, such as `2001:db8:1234:1a00::/64`.

   d. Choose **Save rules**.

> **Note**
>
> Be sure to run the following commands on your local system, not on the instance itself. For more information about these command line interfaces, see Access Amazon EC2.

To add a rule to a security group using the command line

1. Find the security group that is associated with your instance using one of the following commands:

   - describe-instance-attribute (AWS CLI)

     ```
     aws ec2 describe-instance-attribute --region region --instance-id instance_id --
     attribute groupSet
     ```

   - Get-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

     ```
     PS C:\> (Get-EC2InstanceAttribute -Region region -InstanceId instance_id -
     Attribute groupSet).Groups
     ```

   Both commands return a security group ID, which you use in the next step.

2. Add the rule to the security group using one of the following commands:

   - authorize-security-group-ingress (AWS CLI)

     ```
     aws ec2 authorize-security-group-ingress --region region --group-
     id security_group_id --protocol tcp --port 3389 --cidr cidr_ip_range
     ```

   - Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

     The `Grant-EC2SecurityGroupIngress` command needs an `IpPermission` parameter, which describes the protocol, port range, and IP address range to be used for the security group rule. The following command creates the `IpPermission` parameter:

     ```
     PS C:\> $ip1 = @{ IpProtocol="tcp"; FromPort="3389"; ToPort="3389";
      IpRanges="cidr_ip_range" }
     ```

```
PS C:\> Grant-EC2SecurityGroupIngress -Region region -GroupId security_group_id -
IpPermission @($ip1)
```

Assign a security group to an instance

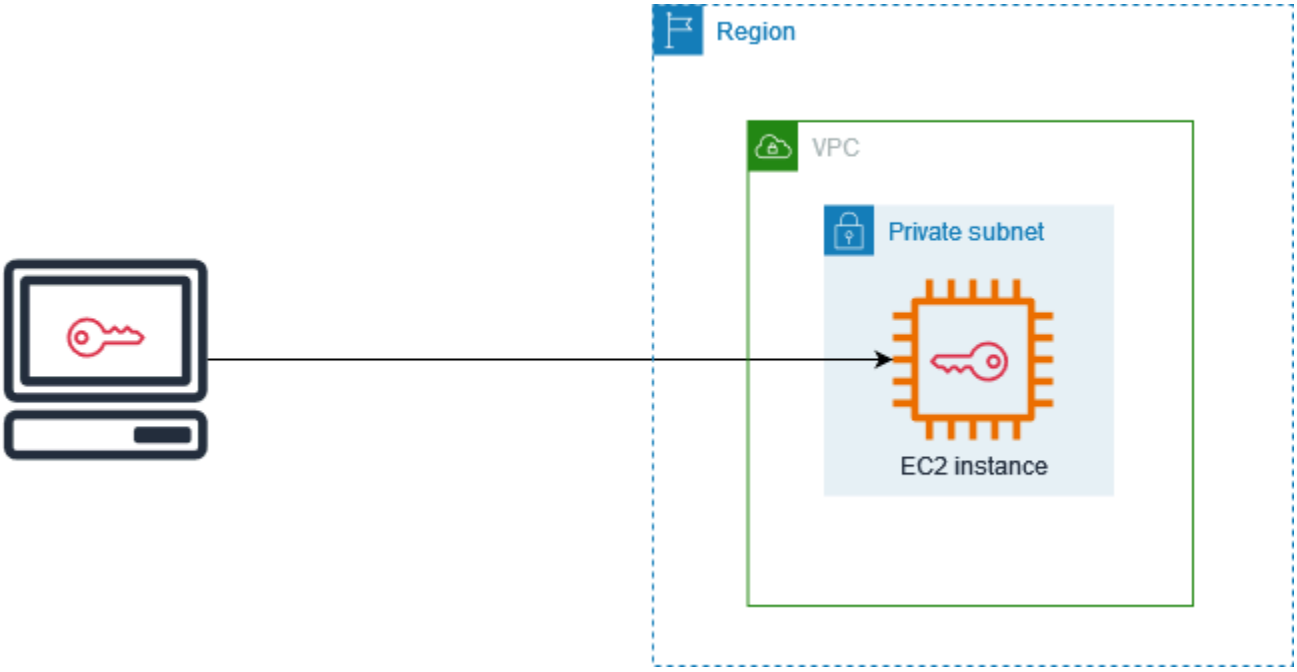
You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance, you can change its security groups. For more information, see the section called “Change an instance's security group”.

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 RDP, 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. For more information about connecting to your Windows instance, see Connect to your Windows 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 Windows 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
- 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 Windows instance using RDP after you have created a key pair, see Connect to your Windows 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 **RSA**. 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. To add a tag to the public key, choose **Add tag**, and enter the key and value for the tag. Repeat for each tag.
8. Choose **Create key pair**.
9. The private key file is automatically downloaded by your browser. The base file name is the name that you specified as the name of your key pair, and the file name extension is determined by the file format that you chose. Save the private key file in a safe place.

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

AWS CLI

To create a key pair using Amazon EC2

- 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
```

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 your key pair, you can create an RSA key pair by using a third-party tool, and then import the public key to Amazon EC2.

Requirements for key pairs

- Supported types: RSA. Amazon EC2 does not accept DSA keys.

> **Note**
>
> ED25519 keys are not supported for Windows instances.

- Supported formats:
  - OpenSSH public key format
  - 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.

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 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, `C:\keys\my-key-pair.pub`. The file name extension for this file is not important.
3. Save the private key to a local file that has the `.pem` or `.ppk` extension. For example, `C:\keys\my-key-pair.pem` or `C:\keys\my-key-pair.ppk`. The file name extension for this

file is important because only `.pem` files can be selected when connecting to your Windows instance from the EC2 console.

> **Important**
>
> Save the private key file in a safe place. You'll need to provide the name of your public key when you launch an instance, and the corresponding private key each time you connect to the instance.

After you have created the key pair, use one of the following methods to import your public key to Amazon EC2.

Console

To import the public key 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.

```
C:\> 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.

```
C:\> 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`.

```
C:\> 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

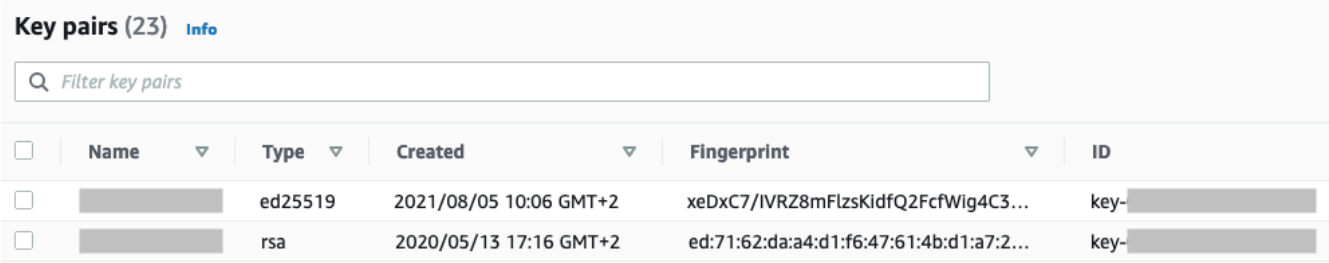
You can view the following information about your public keys that are stored in Amazon EC2: public key name, ID, key type, fingerprint, public key material, the date and time (in the UTC time zone) the key was created by Amazon EC2 (if the key was created by a third-party tool, then it's the date and 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.



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
 AAAAC3NzaC1lZDI1NTE5AAAAIIj7azlDjVHAsSxgcpCRZ3oWnTm0nAFM64y9jd22ioI/ 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 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

On your local Windows computer, you can use PuTTYgen to get the public key for your key pair.

Start PuTTYgen and choose **Load**. Select the `.ppk` or `.pem` private key file. PuTTYgen displays the public key under **Public key for pasting into OpenSSH authorized_keys file**. You can also view the public key by choosing **Save public key**, specifying a name for the file, saving the file, and then opening the file.

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

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

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri  http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
```

Example output

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE key-pair-name
```

For more information about instance metadata, see Retrieve instance metadata.

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
 AAAAC3NzaC1lZDI1NTE5AAAAIIj7azlDjVHAsSxgcpCRZ3oWnTm0nAFM64y9jd22ioI/ 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.

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.

How the fingerprints are calculated

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 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
Created by Amazon EC2	SHA-1
Imported to Amazon EC2	MD5[1]

[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.

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.

  ```
  openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt |
   openssl sha1 -c
  ```

- **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.

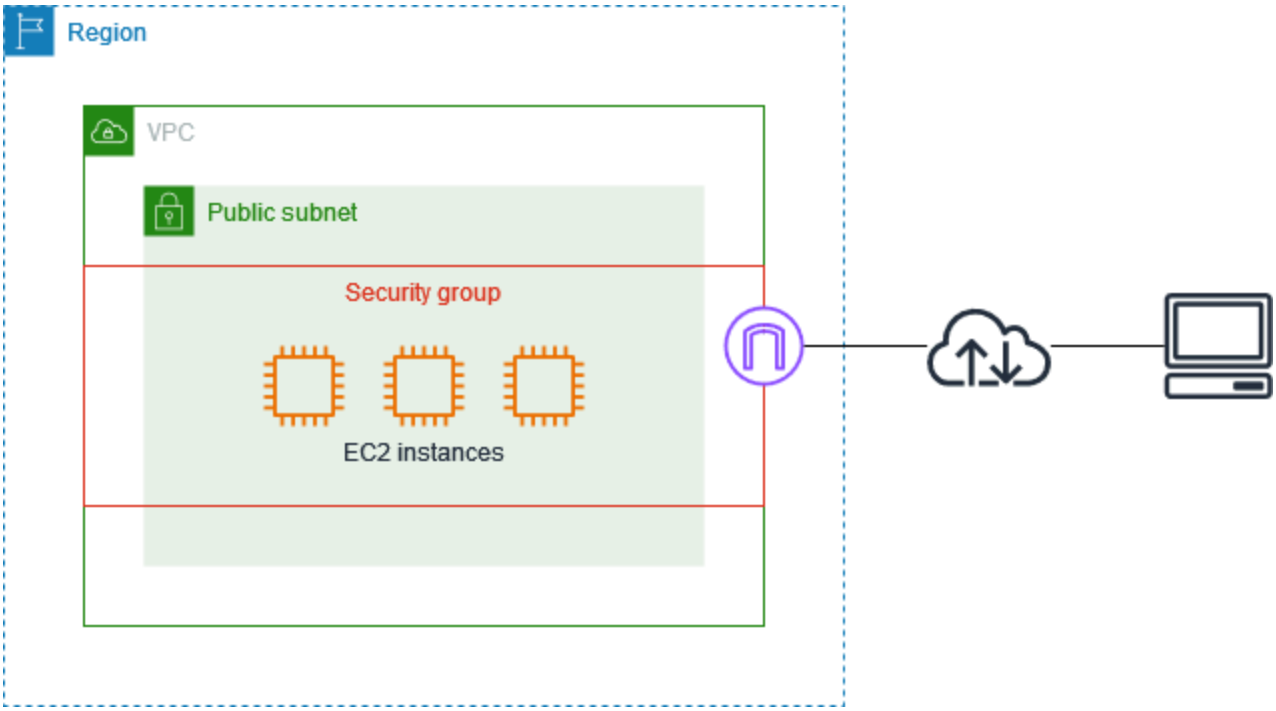
  ```
  ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER
   | openssl md5 -c
  ```

Amazon EC2 security groups for Windows instances

A *security group* acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the

outgoing traffic from your instance. When you launch an instance, you can specify one or more security groups. If you don't specify a security group, Amazon EC2 uses the default security group 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 RDP traffic from your network, then you can connect to your instance from your computer using RDP. 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 Linux instance, see Amazon EC2 security groups for Linux instances in the *Amazon EC2 User Guide for Linux 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, `pl-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 3389 (RDP) from IP address 203.0.113.1, and another rule that allows access to TCP port 3389 from everyone, everyone has access to TCP port 3389.

When you add, update, or remove rules, the changes are automatically applied to all instances associated with the security group.

Security group connection tracking

Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied. With this approach, security groups are stateful. This means that responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa.

As an example, suppose that you initiate 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 `conntrack_allowance_available` network performance metric to determine the number of tracked connections still available for that instance type.

To determine whether packets were dropped because the network traffic for your instance exceeded the maximum number of connections that can be tracked, use the `conntrack_allowance_exceeded` network performance metric. For more information, see Monitor network performance for 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 Nitro-based EC2 instances.

There are three configurable timeouts:

- **TCP established timeout**: Timeout (in seconds) for idle TCP connections in an established state. Min: 60 seconds. Max: 432000 seconds (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 *conntrack_allowance_exceeded* and *conntrack_allowance_available* metrics enable you to monitor dropped packets and tracked connection utilization to proactively manage EC2 instance capacity with scale up or out actions to help meet network connections demand before dropping packets. If you are observing *conntrack_allowance_exceeded* drops on your EC2 instances, you may benefit from setting a lower TCP established timeout to account for stale TCP/UDP sessions resulting from improper clients or network middle boxes.
- Typically, load balancers or firewalls have TCP Established idle timeout in the range of 60 to 90 minutes. If you are running workloads that are expected to handle a very high number of connections (greater than 100k) from appliances like network firewalls, configuring a similar timeout on an EC2 network interface is advised.
- If you are running 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
sg-1234567890abcdef0	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
0.0.0.0/0	All	All	Allows all outbound IPv4 traffic.

Destination	Protocol	Port range	Description
::/0	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 new launch template using parameters you define.

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

Protocol type	Protocol number	Port	Source IP	Notes
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.

Protocol type	Protocol number	Port	Notes
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 has an IPv6 address, you can enter an IPv6 address or range.

Rules to connect to instances from an instance with the same security group

To allow instances that are associated with the same security group to communicate with each other, you must explicitly add rules for this.

> **Note**
>
> If you configure routes to forward the traffic between two instances in different subnets through a middlebox appliance, you must ensure that the security groups for both instances allow traffic to flow between the instances. The security group for each instance must reference the private IP address of the other instance, or the CIDR range of the subnet that contains the other instance, as the source. If you reference the security group of the other instance as the source, this does not allow traffic to flow between the instances.

The following table describes the inbound rule for a security group that enables associated instances to communicate with each other. The rule allows all types of traffic.

Protocol type	Protocol number	Ports	Source IP
-1 (All)	-1 (All)	-1 (All)	The ID of the security group, or the CIDR range of the subnet that contains the other instance (see note).

Rules for ping/ICMP

The **ping** command is a type of ICMP traffic. To ping your instance, you must add one of the following inbound ICMP rules.

Type	Protocol	Source		
Custom ICMP - IPv4	Echo request	The public IPv4 address of your computer, a specific IPv4 address, or an IPv4 or IPv6 address from anywhere.		
All ICMP - IPv4	IPv4 ICMP (1)	The public IPv4 address of your computer, a specific IPv4 address, or an IPv4 or IPv6 address from anywhere.		

To use the **ping6** command to ping the IPv6 address for your instance, you must add the following inbound ICMPv6 rule.

Type	Protocol	Source		
All ICMP - IPv6	IPv6 ICMP (58)	The IPv6 address of your computer, a specific IPv4 address, or an IPv4 or IPv6 address from anywhere.		

DNS server rules

If you've set up your EC2 instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

For the source IP, specify one of the following:

- An IP address or range of IP addresses (in CIDR block notation) in a network
- The ID of a security group for the set of instances in your network that require access to the DNS server

Protocol type	Protocol number	Port
TCP	6	53
UDP	17	53

Amazon EFS rules

If you're using an Amazon EFS file system with your Amazon EC2 instances, the security group that you associate with your Amazon EFS mount targets must allow traffic over the NFS protocol.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	2049 (NFS)	The ID of the security group	Allows inbound NFS access from resources (including the mount tar get) associated with this security group

To mount an Amazon EFS file system on your Amazon EC2 instance, you must connect to your instance. Therefore, the security group associated with your instance must have rules that allow inbound SSH from your local computer or local network.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	22 (SSH)	The IP address range of your local computer, or the range of IP addresses (in CIDR block notation) for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing rules

If you'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"
            }
        }
    }]
}
```

Configuration management in Amazon EC2

Amazon Machine Images (AMIs) provide an initial configuration for an Amazon EC2 instance, which includes the Windows OS and optional customer-specific customizations, such as applications and security controls. Create an AMI catalog containing customized security configuration baselines to ensure all Windows instances are launched with standard security controls. Security baselines can be baked into an AMI, bootstrapped dynamically when an EC2 instance is launched, or packaged as a product for uniform distribution through AWS Service Catalog portfolios. For more information on securing an AMI, see Best Practices for Building an AMI.

Each Amazon EC2 instance should adhere to organizational security standards. Do not install any Windows roles and features that are not required, and do install software to protect against malicious code (antivirus, antimalware, exploit mitigation), monitor host-integrity, and perform intrusion detection. Configure security software to monitor and maintain OS security settings, protect the integrity of critical OS files, and alert on deviations from the security baseline. Consider implementing recommended security configuration benchmarks published by Microsoft, the Center for Internet Security (CIS), or the National Institute of Standards and Technology (NIST). Consider using other Microsoft tools for particular application servers, such as the Best Practice Analyzer for SQL Server.

AWS customers can also run Amazon Inspector assessments to improve the security and compliance of applications deployed on Amazon EC2 instances. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices and includes a knowledge base of hundreds of rules mapped to common security compliance standards (for example, PCI DSS) and vulnerability definitions. Examples of built-in rules include checking if remote root login is enabled, or if vulnerable software versions are installed. These rules are regularly updated by AWS security researchers.

Update management in Amazon EC2

We recommend that you regularly patch, update, and secure the operating system and applications on your EC2 instances. You can use AWS Systems Manager Patch Manager to automate the process of installing security-related updates for both the operating system and applications.

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.

You should configure Windows Update on your Amazon EC2 instances running Windows Server. By default, you will not receive Windows updates on AMIs provided by AWS. For more information, see Best practices for Windows on Amazon EC2.

For a list of the latest Amazon EC2 AMIs running Windows Server, see Details About AWS Windows AMI Versions.

Change management in Amazon EC2

After initial security baselines are applied to Amazon EC2 instances at launch, control ongoing Amazon EC2 changes to maintain the security of your virtual machines. Establish a change management process to authorize and incorporate changes to AWS resources (such as security groups, route tables, and network ACLs) as well as to OS and application configurations (such as Windows or application patching, software upgrades, or configuration file updates).

AWS provides several tools to help manage changes to AWS resources, including AWS CloudTrail, AWS Config, AWS CloudFormation, and AWS Elastic Beanstalk, AWS OpsWorks, and management packs for Systems Center Operations Manager and System Center Virtual Machine Manager. Note that Microsoft releases Windows patches every Tuesday (sometimes even daily) and AWS updates all Windows AMIs managed by AWS within five days after Microsoft releases a patch. Therefore it is important to continually patch all baseline AMIs, update AWS CloudFormation templates and Auto Scaling group configurations with the latest AMI IDs, and implement tools to automate running instance patch management.

Microsoft provides several options for managing Windows OS and application changes. SCCM, for example, provides full lifecycle coverage of environment modifications. Select tools that address business requirements and control how changes will affect application SLAs, capacity,

security, and disaster recovery procedures. Avoid manual changes and instead leverage automated configuration management software or command line tools such as the EC2 Run Command or Windows PowerShell to implement scripted, repeatable change processes. To assist with this requirement, use bastion hosts with enhanced logging for all interactions with your Windows instances to ensure that all events and tasks are automatically recorded.

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.

Amazon EC2 provides Amazon Machine Images (AMI) for Windows Server to help you meet the compliance standards of the Security Technical Implementation Guide (STIG). These AMIs are pre-configured with a number of STIG standards to help you get started with your deployments while meeting STIG compliance standards. For more information, see STIG Hardened Amazon EC2 Windows Server AMIs.

Audit and accountability in Amazon EC2

AWS CloudTrail, AWS Config, and AWS Config Rules provide audit and change tracking features for auditing AWS resource changes. Configure Windows event logs to send local log files to a centralized log management system to preserve log data for security and operational behavior analysis. Microsoft System Center Operations Manager (SCOM) aggregates information about Microsoft applications deployed to Windows instances and applies preconfigured and custom rulesets based on application roles and services. System Center Management Packs build on SCOM to provide application-specific monitoring and configuration guidance. These Management Packs support Windows Server Active Directory, SharePoint Server 2013, Exchange Server 2013, Lync Server 2013, SQL Server 2014, and many more servers and technologies. The AWS Management Pack for Microsoft System Center Operations Manager (SCOM) and the Systems Manager for Microsoft System Center Virtual Machine Manager (SCVMM) integrate with Microsoft Systems Center to help you monitor and manage your on-premises and AWS environments together.

In addition to Microsoft systems management tools, customers can use Amazon CloudWatch to monitor instance CPU utilization, disk performance, network I/O, and perform host and instance status checks. The EC2Config and EC2Launch services provide access to additional, advanced features for Windows instances. For example, they can export Windows system, security, application, and Internet Information Services (IIS) logs to CloudWatch Logs which can then be

integrated with Amazon CloudWatch metrics and alarms. Customers can also create scripts that export Windows performance counters to Amazon CloudWatch custom metrics.

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 a Nitro-based instance. 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 Windows instances
- 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 Windows instances

To launch a Windows instance with NitroTPM enabled, the following prerequisites must be in place. For the prerequisites for launching a Linux instance, see Prerequisites for launching Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

AMI

Requires an AMI with NitroTPM enabled.

The following Windows AMIs are preconfigured to enable NitroTPM and 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 NitroTPM by using the import-image command.

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 TPM-Windows_Server-2022-English-Full-Base, 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
- **High-performance computing**: 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.

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 Windows, use the TPM management console, tpm.msc.

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
```

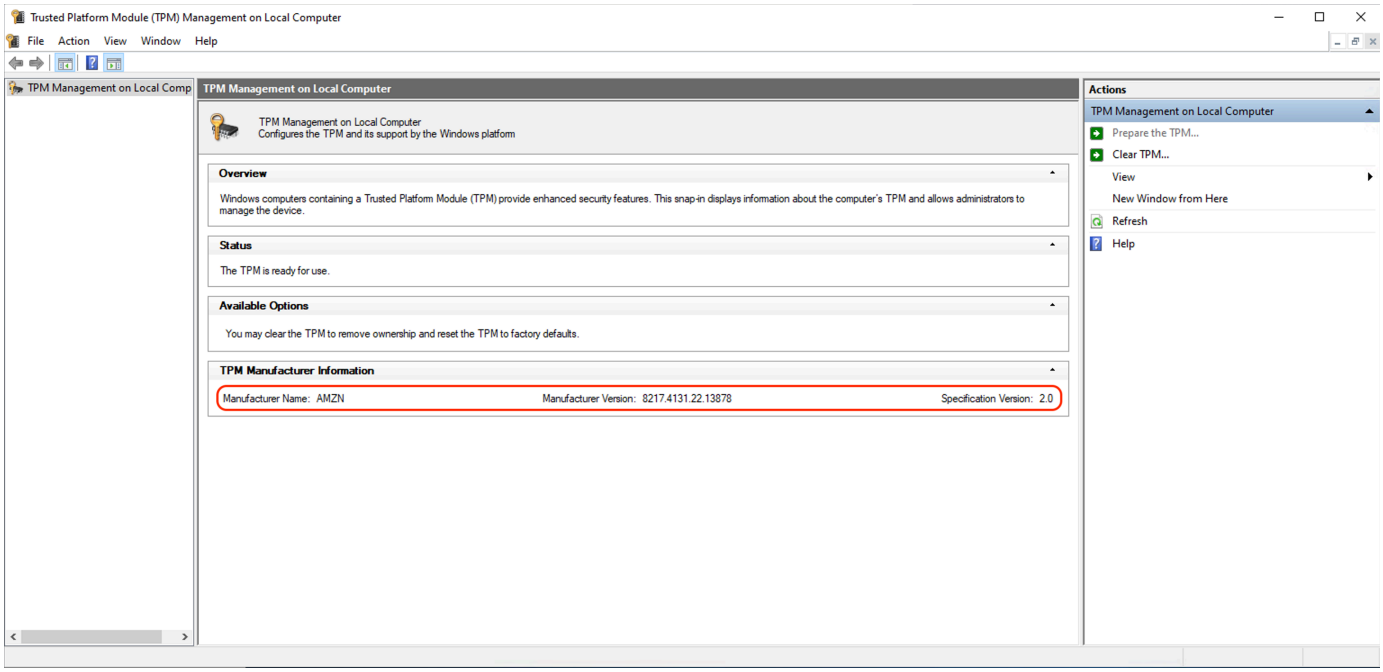
If NitroTPM support is enabled on the instance, `"TpmSupport": "v2.0"` appears in the output.

```
"Instances": {
            "InstanceId":"0123456789example",
            "InstanceType":"c5.large",
            ...
            "BootMode": "uefi",
            "TpmSupport": "v2.0"
            ...
            }
```

To verify whether NitroTPM is accessible inside an Amazon EC2 Windows instance

1. Connect to your EC2 Windows instance.
2. On the instance, run the tpm.msc program.

   The **TPM Management on Local Computer** window opens.
3. Check the **TPM Manufacturer Information** field. It contains the manufacturer's name and the version of the NitroTPM on the instance.



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.

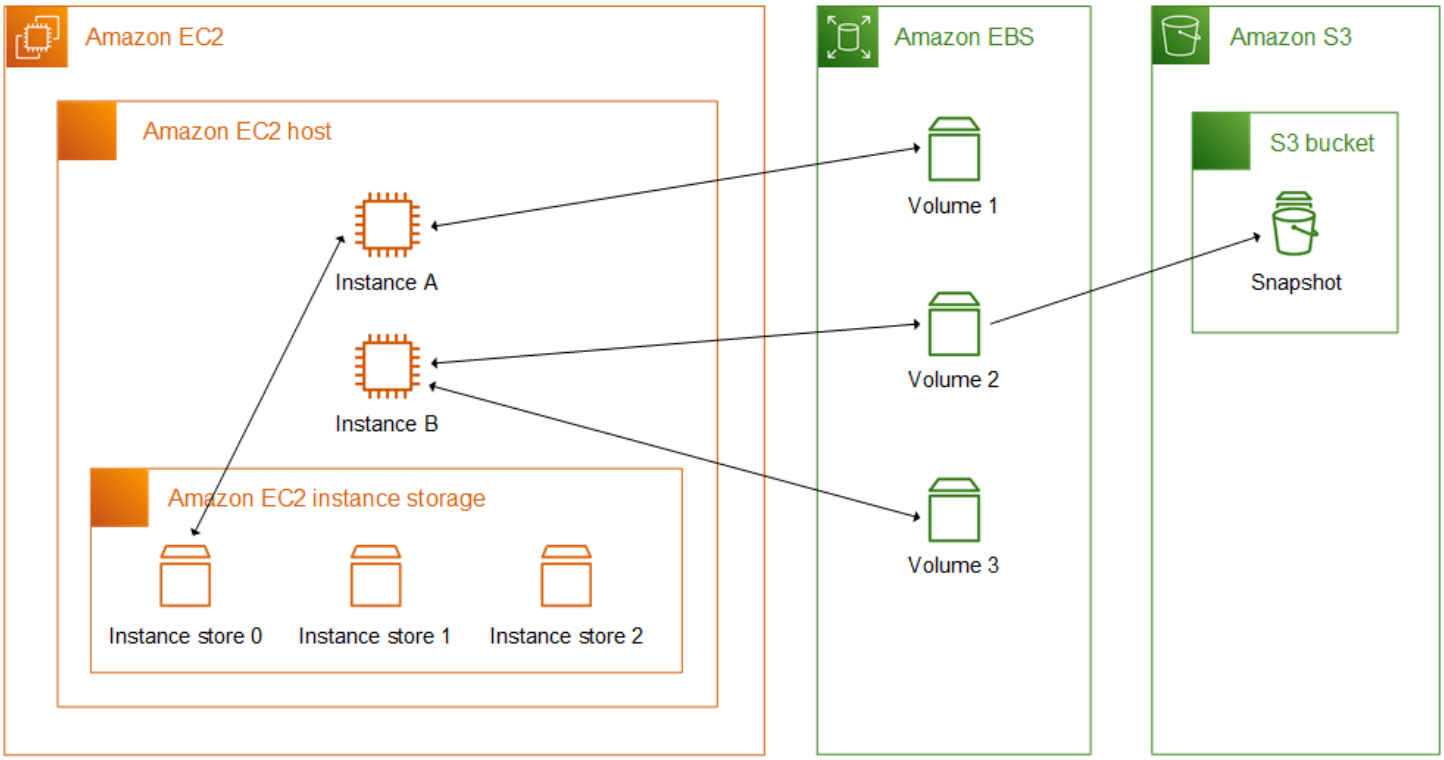
Amazon S3

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.

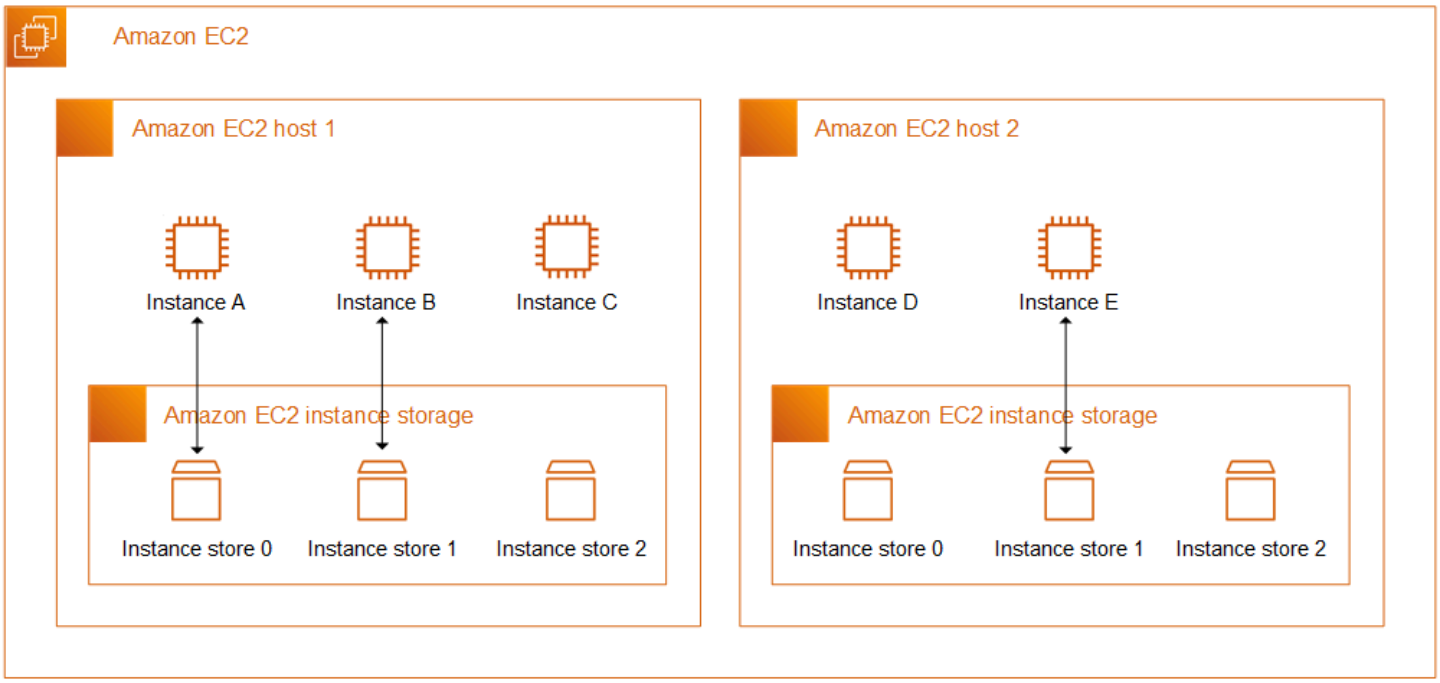
For more information about working with volumes and snapshots, see the Amazon EBS User Guide.

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 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	

Event	What happens to your data?
The instance is rebooted	The data persists
The instance is stopped	The data does not persist
The instance is hibernated	The data does not persist
The instance is terminated	The data does not persist
The instance type is changed	The data does not persist *
A Windows AMI is created from the instance	The data does not persist in the created AMI **
An EBS-backed AMI is created from the instance	The data does not persist in the created AMI **
An instance store-backed AMI is created from the instance	The data persists in the AMI bundle uploaded to Amazon S3 ***
User-initiated OS events	
A shutdown is initiated	The data does not persist †
A restart is initiated	The data persists
AWS scheduled events	
Instance stop	The data does not persist
Instance reboot	The data persists
System reboot	The data persists
Instance retirement	The data does not persist
Unplanned events	
Simplified automatic recovery	The data does not persist
CloudWatch action based recovery	The data does not persist

Event	What happens to your data?
The underlying disk fails	The data on the failed disk does not persist
Power failure	The data persists upon reboot

* If the new instance type supports instance store, the instance gets the number of instance store volumes supported by the new instance type, but the data does not transfer to the new instance. If the new instance type does not support instance store, the instance does not get the instance store volumes.

** The data is not included in the EBS-backed AMI, and it is not included on instance store volumes attached to instances launched from that AMI.

*** The data is included in the AMI bundle that is uploaded to Amazon S3. When you launch an instance from that AMI, the instance gets the instance store volumes bundled in the AMI with the data they contained at the time the AMI was created.

† Termination protection and stop protection do not protect instances against instance stops or terminations as a result of shutdowns initiated through the operating system on the instance. Data stored on instance store volumes does not persist in both instance stop and termination events.

Instance store 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, G2, 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 following table provides the quantity, size, type, and performance optimizations of instance store volumes available on each supported instance type.

Topics

- General purpose
- Compute optimized
- Memory optimized
- Storage optimized
- Accelerated computing
- High-performance computing

General purpose

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
`m1.small`	1 x 160 GB	HDD	✓	
`m1.medium`	1 x 410 GB	HDD	✓	
`m1.large`	2 x 420 GB	HDD	✓	
`m1.xlarge`	4 x 420 GB	HDD	✓	
`m2.xlarge`	1 x 420 GB	HDD	✓	
`m2.2xlarge`	1 x 850 GB	HDD	✓	

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
m2.4xlarge	2 x 840 GB	HDD	✓	
m3.medium	1 x 4 GB	SSD	✓	
m3.large	1 x 32 GB	SSD	✓	
m3.xlarge	2 x 40 GB	SSD	✓	
m3.2xlarge	2 x 80 GB	SSD	✓	
m5ad.large	1 x 75 GB	NVMe SSD		✓
m5ad.xlarge	1 x 150 GB	NVMe SSD		✓
m5ad.2xlarge	1 x 300 GB	NVMe SSD		✓
m5ad.4xlarge	2 x 300 GB	NVMe SSD		✓
m5ad.8xlarge	2 x 600 GB	NVMe SSD		✓
m5ad.12xlarge	2 x 900 GB	NVMe SSD		✓
m5ad.16xlarge	4 x 600 GB	NVMe SSD		✓
m5ad.24xlarge	4 x 900 GB	NVMe SSD		✓
m5d.large	1 x 75 GB	NVMe SSD		✓
m5d.xlarge	1 x 150 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
m5d.2xlarge	1 x 300 GB	NVMe SSD		✓
m5d.4xlarge	2 x 300 GB	NVMe SSD		✓
m5d.8xlarge	2 x 600 GB	NVMe SSD		✓
m5d.12xlarge	2 x 900 GB	NVMe SSD		✓
m5d.16xlarge	4 x 600 GB	NVMe SSD		✓
m5d.24xlarge	4 x 900 GB	NVMe SSD		✓
m5d.metal	4 x 900 GB	NVMe SSD		✓
m5dn.large	1 x 75 GB	NVMe SSD		✓
m5dn.xlarge	1 x 150 GB	NVMe SSD		✓
m5dn.2xlarge	1 x 300 GB	NVMe SSD		✓
m5dn.4xlarge	2 x 300 GB	NVMe SSD		✓
m5dn.8xlarge	2 x 600 GB	NVMe SSD		✓
m5dn.12xlarge	2 x 900 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
m5dn.16xlarge	4 x 600 GB	NVMe SSD		✓
m5dn.24xlarge	4 x 900 GB	NVMe SSD		✓
m5dn.metal	4 x 900 GB	NVMe SSD		✓
m6id.large	1 x 118 GB	NVMe SSD		✓
m6id.xlarge	1 x 237 GB	NVMe SSD		✓
m6id.2xlarge	1 x 474 GB	NVMe SSD		✓
m6id.4xlarge	1 x 950 GB	NVMe SSD		✓
m6id.8xlarge	1 x 1900 GB	NVMe SSD		✓
m6id.12xlarge	2 x 1425 GB	NVMe SSD		✓
m6id.16xlarge	2 x 1900 GB	NVMe SSD		✓
m6id.24xlarge	4 x 1425 GB	NVMe SSD		✓
m6id.32xlarge	4 x 1900 GB	NVMe SSD		✓
m6id.metal	4 x 1900 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
m6idn.large	1 x 118 GB	NVMe SSD		✓
m6idn.xlarge	1 x 237 GB	NVMe SSD		✓
m6idn.2xlarge	1 x 474 GB	NVMe SSD		✓
m6idn.4xlarge	1 x 950 GB	NVMe SSD		✓
m6idn.8xlarge	1 x 1900 GB	NVMe SSD		✓
m6idn.12xlarge	2 x 1425 GB	NVMe SSD		✓
m6idn.16xlarge	2 x 1900 GB	NVMe SSD		✓
m6idn.24xlarge	4 x 1425 GB	NVMe SSD		✓
m6idn.32xlarge	4 x 1900 GB	NVMe SSD		✓
m6idn.metal	4 x 1900 GB	NVMe SSD		✓

Compute optimized

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
c1.medium	1 x 350 GB	HDD	✓	
c1.xlarge	4 x 420 GB	HDD	✓	
c3.large	2 x 16 GB	SSD	✓	
c3.xlarge	2 x 40 GB	SSD	✓	
c3.2xlarge	2 x 80 GB	SSD	✓	
c3.4xlarge	2 x 160 GB	SSD	✓	
c3.8xlarge	2 x 320 GB	SSD	✓	
c5ad.large	1 x 75 GB	NVMe SSD		✓
c5ad.xlarge	1 x 150 GB	NVMe SSD		✓
c5ad.2xlarge	1 x 300 GB	NVMe SSD		✓
c5ad.4xlarge	2 x 300 GB	NVMe SSD		✓
c5ad.8xlarge	2 x 600 GB	NVMe SSD		✓
c5ad.12xlarge	2 x 900 GB	NVMe SSD		✓
c5ad.16xlarge	2 x 1200 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
c5ad.24xlarge	2 x 1900 GB	NVMe SSD		✓
c5d.large	1 x 50 GB	NVMe SSD		✓
c5d.xlarge	1 x 100 GB	NVMe SSD		✓
c5d.2xlarge	1 x 200 GB	NVMe SSD		✓
c5d.4xlarge	1 x 400 GB	NVMe SSD		✓
c5d.9xlarge	1 x 900 GB	NVMe SSD		✓
c5d.12xlarge	2 x 900 GB	NVMe SSD		✓
c5d.18xlarge	2 x 900 GB	NVMe SSD		✓
c5d.24xlarge	4 x 900 GB	NVMe SSD		✓
c5d.metal	4 x 900 GB	NVMe SSD		✓
c6id.large	1 x 118 GB	NVMe SSD		✓
c6id.xlarge	1 x 237 GB	NVMe SSD		✓
c6id.2xlarge	1 x 474 GB	NVMe SSD		✓
c6id.4xlarge	1 x 950 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
c6id.8xlarge	1 x 1900 GB	NVMe SSD		✓
c6id.12xlarge	2 x 1425 GB	NVMe SSD		✓
c6id.16xlarge	2 x 1900 GB	NVMe SSD		✓
c6id.24xlarge	4 x 1425 GB	NVMe SSD		✓
c6id.32xlarge	4 x 1900 GB	NVMe SSD		✓
c6id.metal	4 x 1900 GB	NVMe SSD		✓

Memory optimized

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
r3.large	1 x 32 GB	SSD	✓	
r3.xlarge	1 x 80 GB	SSD	✓	
r3.2xlarge	1 x 160 GB	SSD	✓	
r3.4xlarge	1 x 320 GB	SSD	✓	
r3.8xlarge	2 x 320 GB	SSD	✓	
r5ad.large	1 x 75 GB	NVMe SSD		✓
r5ad.xlarge	1 x 150 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
r5ad.2xlarge	1 x 300 GB	NVMe SSD		✓
r5ad.4xlarge	2 x 300 GB	NVMe SSD		✓
r5ad.8xlarge	2 x 600 GB	NVMe SSD		✓
r5ad.12xlarge	2 x 900 GB	NVMe SSD		✓
r5ad.16xlarge	4 x 600 GB	NVMe SSD		✓
r5ad.24xlarge	4 x 900 GB	NVMe SSD		✓
r5d.large	1 x 75 GB	NVMe SSD		✓
r5d.xlarge	1 x 150 GB	NVMe SSD		✓
r5d.2xlarge	1 x 300 GB	NVMe SSD		✓
r5d.4xlarge	2 x 300 GB	NVMe SSD		✓
r5d.8xlarge	2 x 600 GB	NVMe SSD		✓
r5d.12xlarge	2 x 900 GB	NVMe SSD		✓
r5d.16xlarge	4 x 600 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
r5d.24xlarge	4 x 900 GB	NVMe SSD		✓
r5d.metal	4 x 900 GB	NVMe SSD		✓
r5dn.large	1 x 75 GB	NVMe SSD		✓
r5dn.xlarge	1 x 150 GB	NVMe SSD		✓
r5dn.2xlarge	1 x 300 GB	NVMe SSD		✓
r5dn.4xlarge	2 x 300 GB	NVMe SSD		✓
r5dn.8xlarge	2 x 600 GB	NVMe SSD		✓
r5dn.12xlarge	2 x 900 GB	NVMe SSD		✓
r5dn.16xlarge	4 x 600 GB	NVMe SSD		✓
r5dn.24xlarge	4 x 900 GB	NVMe SSD		✓
r5dn.metal	4 x 900 GB	NVMe SSD		✓
r6idn.large	1 x 118 GB	NVMe SSD		✓
r6idn.xlarge	1 x 237 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
r6idn.2xlarge	1 x 474 GB	NVMe SSD		✓
r6idn.4xlarge	1 x 950 GB	NVMe SSD		✓
r6idn.8xlarge	1 x 1900 GB	NVMe SSD		✓
r6idn.12xlarge	2 x 1425 GB	NVMe SSD		✓
r6idn.16xlarge	2 x 1900 GB	NVMe SSD		✓
r6idn.24xlarge	4 x 1425 GB	NVMe SSD		✓
r6idn.32xlarge	4 x 1900 GB	NVMe SSD		✓
r6idn.metal	4 x 1900 GB	NVMe SSD		✓
r6id.large	1 x 118 GB	NVMe SSD		✓
r6id.xlarge	1 x 237 GB	NVMe SSD		✓
r6id.2xlarge	1 x 474 GB	NVMe SSD		✓
r6id.4xlarge	1 x 950 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
r6id.8xlarge	1 x 1900 GB	NVMe SSD		✓
r6id.12xlarge	2 x 1425 GB	NVMe SSD		✓
r6id.16xlarge	2 x 1900 GB	NVMe SSD		✓
r6id.24xlarge	4 x 1425 GB	NVMe SSD		✓
r6id.32xlarge	4 x 1900 GB	NVMe SSD		✓
r6id.metal	4 x 1900 GB	NVMe SSD		✓
x1.16xlarge	1 x 1920 GB	SSD	✓	
x1.32xlarge	2 x 1920 GB	SSD	✓	
x2idn.16xlarge	1 x 1900 GB	NVMe SSD		✓
x2idn.24xlarge	2 x 1425 GB	NVMe SSD		✓
x2idn.32xlarge	2 x 1900 GB	NVMe SSD		✓
x2idn.metal	2 x 1900 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
x2iedn.xlarge	1 x 118 GB	NVMe SSD		✓
x2iedn.2xlarge	1 x 237 GB	NVMe SSD		✓
x2iedn.4xlarge	1 x 475 GB	NVMe SSD		✓
x2iedn.8xlarge	1 x 950 GB	NVMe SSD		✓
x2iedn.16xlarge	1 x 1900 GB	NVMe SSD		✓
x2iedn.24xlarge	2 x 1425 GB	NVMe SSD		✓
x2iedn.32xlarge	2 x 1900 GB	NVMe SSD		✓
x2iedn.metal	2 x 1900 GB	NVMe SSD		✓
x1e.xlarge	1 x 120 GB	SSD	✓	
x1e.2xlarge	1 x 240 GB	SSD	✓	
x1e.4xlarge	1 x 480 GB	SSD	✓	
x1e.8xlarge	1 x 960 GB	SSD	✓	

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
x1e.16xlarge	1 x 1920 GB	SSD	✓	
x1e.32xlarge	2 x 1920 GB	SSD	✓	
z1d.large	1 x 75 GB	NVMe SSD		✓
z1d.xlarge	1 x 150 GB	NVMe SSD		✓
z1d.2xlarge	1 x 300 GB	NVMe SSD		✓
z1d.3xlarge	1 x 450 GB	NVMe SSD		✓
z1d.6xlarge	1 x 900 GB	NVMe SSD		✓
z1d.12xlarge	2 x 900 GB	NVMe SSD		✓
z1d.metal	2 x 900 GB	NVMe SSD		✓

Storage optimized

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
d2.xlarge	3 x 2048 GB	HDD	✓	
d2.2xlarge	6 x 2048 GB	HDD	✓	
d2.4xlarge	12 x 2048 GB	HDD	✓	
d2.8xlarge	24 x 2048 GB	HDD	✓	

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
d3.xlarge	3 x 1980 GB	NVMe HDD		✓
d3.2xlarge	6 x 1980 GB	NVMe HDD		✓
d3.4xlarge	12 x 1980 GB	NVMe HDD		✓
d3.8xlarge	24 x 1980 GB	NVMe HDD		✓
d3en.xlarge	2 x 13980 GB	NVMe HDD		✓
d3en.2xlarge	4 x 13980 GB	NVMe HDD		✓
d3en.4xlarge	8 x 13980 GB	NVMe HDD		✓
d3en.6xlarge	12 x 13980 GB	NVMe HDD		✓
d3en.8xlarge	16 x 13980 GB	NVMe HDD		✓
d3en.12xlarge	24 x 13980 GB	NVMe HDD		✓
h1.2xlarge	1 x 2000 GB	HDD	✓	
h1.4xlarge	2 x 2000 GB	HDD	✓	
h1.8xlarge	4 x 2000 GB	HDD	✓	
h1.16xlarge	8 x 2000 GB	HDD	✓	
i2.xlarge	1 x 800 GB	SSD	✓	

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
i2.2xlarge	2 x 800 GB	SSD	✓	
i2.4xlarge	4 x 800 GB	SSD	✓	
i2.8xlarge	8 x 800 GB	SSD	✓	
i3.large	1 x 475 GB	NVMe SSD		✓
i3.xlarge	1 x 950 GB	NVMe SSD		✓
i3.2xlarge	1 x 1900 GB	NVMe SSD		✓
i3.4xlarge	2 x 1900 GB	NVMe SSD		✓
i3.8xlarge	4 x 1900 GB	NVMe SSD		✓
i3.16xlarge	8 x 1900 GB	NVMe SSD		✓
i3.metal	8 x 1900 GB	NVMe SSD		✓
i3en.large	1 x 1250 GB	NVMe SSD		✓
i3en.xlarge	1 x 2500 GB	NVMe SSD		✓
i3en.2xlarge	2 x 2500 GB	NVMe SSD		✓
i3en.3xlarge	1 x 7500 GB	NVMe SSD		✓
i3en.6xlarge	2 x 7500 GB	NVMe SSD		✓
i3en.12xlarge	4 x 7500 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
i3en.24xlarge	8 x 7500 GB	NVMe SSD		✓
i3en.metal	8 x 7500 GB	NVMe SSD		✓
i4i.large	1 x 468 GB	NVMe SSD		✓
i4i.xlarge	1 x 937 GB	NVMe SSD		✓
i4i.2xlarge	1 x 1875 GB	NVMe SSD		✓
i4i.4xlarge	1 x 3750 GB	NVMe SSD		✓
i4i.8xlarge	2 x 3750 GB	NVMe SSD		✓
i4i.12xlarge	3 x 3750 GB	NVMe SSD		✓
i4i.16xlarge	4 x 3750 GB	NVMe SSD		✓
i4i.24xlarge	6 x 3750 GB	NVMe SSD		✓
i4i.32xlarge	8 x 3750 GB	NVMe SSD		✓
i4i.metal	8 x 3750 GB	NVMe SSD		✓

Accelerated computing

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
g2.2xlarge	1 x 60 GB	SSD	✓	
g2.8xlarge	2 x 120 GB	SSD	✓	
g4ad.xlarge	1 x 150 GB	NVMe SSD		✓
g4ad.2xlarge	1 x 300 GB	NVMe SSD		✓
g4ad.4xlarge	1 x 600 GB	NVMe SSD		✓
g4ad.8xlarge	1 x 1200 GB	NVMe SSD		✓
g4ad.16xlarge	2 x 1200 GB	NVMe SSD		✓
g4dn.xlarge	1 x 125 GB	NVMe SSD		✓
g4dn.2xlarge	1 x 225 GB	NVMe SSD		✓
g4dn.4xlarge	1 x 225 GB	NVMe SSD		✓
g4dn.8xlarge	1 x 900 GB	NVMe SSD		✓
g4dn.12xlarge	1 x 900 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
g4dn.16xlarge	1 x 900 GB	NVMe SSD		✓
g4dn.metal	2 x 900 GB	NVMe SSD		✓
g5.xlarge	1 x 250 GB	NVMe SSD		✓
g5.2xlarge	1 x 450 GB	NVMe SSD		✓
g5.4xlarge	1 x 600 GB	NVMe SSD		✓
g5.8xlarge	1 x 900 GB	NVMe SSD		✓
g5.12xlarge	1 x 3800 GB	NVMe SSD		✓
g5.16xlarge	1 x 1900 GB	NVMe SSD		✓
g5.24xlarge	1 x 3800 GB	NVMe SSD		✓
g5.48xlarge	2 x 3800 GB	NVMe SSD		✓
p3dn.24xlarge	2 x 900 GB	NVMe SSD		✓

High-performance computing

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
hpc6id.32xlarge	4 x 3800 GB	NVMe SSD		✓

* Volumes attached to certain instances suffer a first-write penalty unless initialized.

** For more information, see Instance store volume TRIM support.

Instance store volume performance

The following documentation describes the I/O performance of the instance store volumes.

- General purpose instances
- Compute optimized instances
- Memory optimized instances
- Storage optimized instances
- Accelerated computing instances

To query instance store volume information using the AWS CLI

You can use the describe-instance-types AWS CLI command to display information about an instance type, such as its instance store volumes. The following example displays the total size of instance storage for all R5 instances with instance store volumes.

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r5*" "Name=instance-storage-
supported,Values=true" \
    --query "InstanceTypes[].[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, G2, 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 mounted automatically. For Windows instances, the root volume must be an Amazon EBS volume; instance store is not supported for the root volume.

Contents

- Add instance store volumes to an 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 Windows 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 Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see List disks using Disk Management.

To manually mount an instance store volume

1. Choose **Start**, enter **Computer Management**, and then press **Enter**.
2. In left-hand panel, choose **Disk Management**.
3. If you are prompted to initialize the volume, choose the volume to initialize, select the required partition type depending on your use case, and then choose **OK**.
4. In the list of volumes, right-click the volume to mount, and then choose **New Simple Volume**.
5. On the wizard, choose **Next**.
6. On the Specify Volume Size screen, choose **Next** to use the maximum volume size. Alternatively, choose a volume size that is between the minimum and maximum disk space.
7. On the Assign a Drive Letter or Path screen, do one of the following, and choose **Next**.
   - To mount the volume with a drive letter, choose **Assign the following drive letter** and then choose the drive letter to use.
   - To mount the volume as a folder, choose **Mount in the following empty NTFS folder** and then choose **Browse** to create or select the folder to use.
   - To mount the volume without a drive letter or path, choose **Do not assign a drive letter or drive path**.

8. On the Format Partition screen, specify whether or not to format the volume. If you choose to format the volume, choose the required file system and unit size, and specify a volume label.
9. Choose **Next**, **Finish**.

SSD instance store volumes

Like other instance store volumes, you must map the SSD instance store volumes for your instance when you launch it. The data on an SSD instance volume persists only for the life of its associated instance. For more information, see Add instance store volumes to your EC2 instance.

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.

The latest AWS Windows AMIs for the following operating systems contain the AWS NVMe drivers used to interact with SSD instance store volumes that are exposed as NVMe block devices for better performance:

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

After you connect to your instance, you can verify that you see the NVMe volumes in Disk Manager. On the taskbar, open the context (right-click) menu for the Windows logo and choose **Disk Management**. On Windows Server 2008 R2, choose **Start**, **Administrative Tools**, **Computer Management**, **Disk Management**.

The AWS Windows AMIs provided by Amazon include the AWS NVMe driver. If you are not using the latest AWS Windows AMIs, you can install the current AWS NVMe driver.

The data on NVMe instance storage is encrypted using an XTS-AES-256 block cipher implemented in a hardware module on the instance. The encryption keys are generated using the hardware module and are unique to each NVMe instance storage device. All encryption keys are destroyed

when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

Non-NVMe SSD volumes

The following instances support instance store volumes that use non-NVMe SSDs to deliver high random I/O performance: C3, G2, I2, M3, R3, and X1. For more information about the instance store volumes supported by each instance type, see Instance store volumes.

Instance store volume TRIM support

Some instance types support SSD volumes with TRIM. For more information, see Instance store volumes.

Instances running Windows Server 2012 R2 support TRIM as of AWS PV Driver version 7.3.0. Instances running earlier versions of Windows Server do not support TRIM.

Instance store volumes that support TRIM are fully trimmed before they are allocated to your instance. These volumes are not formatted with a file system when an instance launches, so you must format them before they can be mounted and used. For faster access to these volumes, you should skip the TRIM operation when you format them. On Windows, to temporarily disable TRIM support during initial formatting, use the `fsutil behavior set DisableDeleteNotify 1` command. After formatting is complete, re-enable TRIM support by using `fsutil behavior set DisableDeleteNotify 0`.

With instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller when you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. On Windows, use the `fsutil behavior set DisableDeleteNotify 0` command to ensure TRIM support is enabled during normal operation.

File storage

Cloud file storage is a method for storing data in the cloud that provides servers and applications access to data through shared file systems. This compatibility makes cloud file storage ideal for workloads that rely on shared file systems and provides simple integration without code changes.

There are many file storage solutions that exist, ranging from a single node file server on a compute instance using block storage as the underpinnings with no scalability or few redundancies

to protect the data, to a do-it-yourself clustered solution, to a fully-managed solution. The following content introduces some of the storage services provided by AWS for use with Windows.

Contents

- Use Amazon S3 with Amazon EC2
- 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.

AWS Tools for Windows PowerShell

Windows instances have the benefit of a graphical browser that you can use to access the Amazon S3 console directly; however, for scripting purposes, Windows users can also use the AWS Tools for Windows PowerShell to move objects to and from Amazon S3.

Use the following command to copy an Amazon S3 object to your Windows instance.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key path-to-file -
LocalFile my_copied_file.ext
```

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. The AWS CLI enables users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the AWS Command Line Interface detail page.

The **aws s3 cp** command is similar to the Unix **cp** command. You can copy files from Amazon S3 to your instance, copy files from your instance to Amazon S3, and copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
aws s3 sync s3://remote_S3_bucket local_directory
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the Amazon Simple Storage Service 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. To use Amazon EFS with a Linux instance, see Amazon Elastic File System (Amazon EFS) in the *Amazon EC2 User Guide for Linux 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.

> **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.

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	*file system security group*
UDP	20001 - 20003	*file system security group*
UDP	4049	*file system security group*
UDP	2049	*file system security group*
UDP	635	*file system security group*
UDP	4045 - 4046	*file system security group*
TCP	4049	*file system security group*
TCP	635	*file system security group*
TCP	2049	*file system security group*
TCP	111	*file system security group*
TCP	4045 - 4046	*file system security group*
TCP	20001 - 20003	*file system security group*

Protocol type	Port number	Destination
All	All	*file system security group*

The security group that is created and attached to the file system is tagged with `Name=fsx-sg-1`. The value in the tag is automatically incremented each time the launch instance wizard creates a security group for Amazon FSx file systems.

The security group includes the following rules.

Inbound rules

Protocol type	Port number	Source
UDP	2049	*instance security group*
UDP	20001 - 20003	*instance security group*
UDP	4049	*instance security group*
UDP	111	*instance security group*
UDP	635	*instance security group*
UDP	4045 - 4046	*instance security group*
TCP	4045 - 4046	*instance security group*
TCP	635	*instance security group*
TCP	2049	*instance security group*
TCP	4049	*instance security group*
TCP	20001 - 20003	*instance security group*
TCP	111	*instance security group*

Outbound rules

Protocol type	Port number	Destination
All	All	0.0.0.0/0

User data script

If you choose to automatically attach user data scripts, the launch instance wizard adds the following user data to the instance. This script installs the necessary packages, mounts the file system, and updates your instance settings so that the file system will automatically re-mount whenever the instance restarts.

```
#cloud-config
package_update: true
package_upgrade: true
runcmd:
- yum install -y nfs-utils
- apt-get -y install nfs-common
- svm_id_1=svm_id
- file_system_id_1=file_system_id
- vol_path_1=/vol1
- fsx_mount_point_1=/mnt/fsx/fs1
- mkdir -p "${fsx_mount_point_1}"
- if [ -z "$svm_id_1" ]; then printf "\n${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
 0 0\n" >> /etc/fstab; else printf "\n${svm_id_1}.${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
 0\n" >> /etc/fstab; fi
- retryCnt=15; waitTime=30; while true; do mount -a -t nfs4 defaults; if [ $? = 0 ] ||
 [ $retryCnt -lt 1 ]; then echo File system mounted successfully; break; fi; echo File
 system not available, retrying to mount.; ((retryCnt--)); sleep $waitTime; done;
```

Mount an Amazon FSx file system at launch

To mount a new or existing Amazon FSx file system at launch

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

2. In the navigation pane, choose **Instances** and then choose **Launch instance** to open the launch instance wizard.
3. In the **Application and OS Images** section, select the AMI to use.
4. In the **Instance type** section, select the instance type.
5. In the **Key pair** section, select an existing key pair or create a new one.
6. In the **Network settings** section, do the following:

   a. Choose **Edit**.

   b. If you want to **mount an existing file system**, for **Subnet**, choose the file system's preferred subnet. We recommend that you launch the instance into the same Availability Zone as the file system's preferred subnet to optimize performance.

      If you want to **create a new file system** to mount to an instance, for **Subnet**, choose the subnet into which to launch the instance.

      > **Important**
      >
      > You must select a subnet to enable the Amazon FSx functionality in the new launch instance wizard. If you do not select a subnet, you will not be able to mount an existing file system or create a new one.

7. In the **Storage** section, do the following:

   a. Configure the volumes as needed.

   b. Expand the **File systems** section and select **FSx**.

   c. Choose **Add shared file system**.

   d. For **File system**, select the file system to mount.

      > **Note**
      >
      > The list displays all Amazon FSx for NetApp ONTAP and Amazon FSx for OpenZFS file systems in your account in the selected Region.

   e. To automatically create and attach the security groups needed to enable access to the file system, select **Automatically create and attach security groups**. If you prefer to create the security groups manually, clear the 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

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

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

Instance size	Volume limit
metal-16xl |metal-24xl	39
metal-32xl |metal-48xl	79

Shared Amazon EBS volume limit

All other Nitro instance types 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 apply:

- 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.
- `mac1.metal` instances support a maximum of 16 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

The following table shows the volume limits for Xen-based Windows instances based on the driver used. That these numbers include the root volume, plus any attached instance store volumes and Amazon EBS volumes.

> **Important**
>
> Attaching more than the following number of volumes to a Xen-based Windows instance is supported on a best effort basis only and is not guaranteed.

Driver	Volume Limit
AWS PV	26
Citrix PV	26
Red Hat PV	17

We recommend that you do not that attach more than 26 volumes to a Xen-based Windows instance with AWS PV or Citrix PV drivers, as it is likely to cause performance issues.

To determine which PV drivers your instance is using, or to upgrade your Windows instance from Red Hat to Citrix PV drivers, see Upgrade PV drivers on Windows instances.

For more information about how device names are related to volumes, see Map disks to volumes on your Windows instance.

Amazon EC2 instance root device volume

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

Topics

- Configure the root volume to persist
- Confirm that a root volume is configured to persist
- Change the initial size of the root volume

Configure the root volume to persist

By default, the root volume is deleted when the instance terminates (the `DeleteOnTermination` attribute is `true`). Using the console, you can change `DeleteOnTermination` when you launch an instance. To change this attribute for an existing instance, you must use the command line.

Topics

- Configure the root volume to persist during instance launch
- Configure the root volume to persist for an existing instance

Configure the root volume to persist during instance launch

You can configure the root volume to persist when you launch an instance using the Amazon EC2 console or the command line tools.

Console

To configure the root volume to persist when you launch an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then choose **Launch instances**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect **Delete On Termination** for the root volume.
6. Complete the remaining wizard pages, and then choose **Launch**.

AWS CLI

To configure the root volume to persist when you launch an instance using the AWS CLI

Use the run-instances command and include a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
C:\> aws ec2 run-instances --block-device-mappings file://mapping.json ...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`.

```
C:\> aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

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

Tools for Windows PowerShell

To confirm that a root volume is configured to persist using the AWS Tools for Windows PowerShell

Use the Get-EC2Instance and verify that the `DeleteOnTermination` attribute in the `BlockDeviceMappings` response element is set to `false`.

```
C:\> (Get-EC2Instance -InstanceId i-
i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

Change the initial size of the root volume

By default, the size of the root volume is determined by the size of the snapshot. You can increase the initial size of the root volume using the block device mapping of the instance as follows.

1. Determine the device name of the root volume specified in the AMI, as described in View the EBS volumes in an AMI block device mapping.
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.

Device names on Windows instances

When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

The number of volumes that your instance can support is determined by the operating system. For more information, see Instance volume limits.

Contents

- Available device names
- Device name considerations

For information about device names on Linux instances, see Device naming on Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

Available device names

Windows AMIs use one of the following sets of drivers to permit access to virtualized hardware: AWS PV, Citrix PV, and RedHat PV. For more information, see Paravirtual drivers for Windows instances.

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume.

Driver type	Available	Reserved for root volume	Recommended for EBS volumes	Instance store volumes
AWS PV, Citrix PV	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z] *	xvdc[a-x] xvd[a-e] **
Red Hat PV	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-p]	xvdc[a-x] xvd[a-e]

* For Citrix PV and Red Hat PV, if you map an EBS volume with the name `xvda`, Windows does not recognize the volume (the volume is visible for AWS PV or AWS NVMe).

** NVMe instance store volumes are automatically enumerated and assigned a Windows drive letter.

For more information about instance store volumes, see Amazon EC2 instance store. 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 a Windows drive letter.
- AWS Windows AMIs come with additional software that prepares an instance when it first boots up. This is either the EC2Config service (Windows AMIs prior to Windows Server 2016) or EC2Launch (Windows Server 2016 and later). After the devices have been mapped to drives, they are initialized and mounted. The root drive is initialized and mounted as `C:\`. By default, when an EBS volume is attached to a Windows instance, it can show up as any drive letter on the instance. You can change the settings to set the drive letters of the volumes per your specifications. For instance store volumes, the default depends on the driver. AWS PV drivers and Citrix PV drivers assign instance store volumes drive letters going from Z: to A:. Red Hat drivers assign instance store volumes drive letters going from D: to Z:. For more information, see Configure a Windows instance using the EC2Config service (legacy), Configure a Windows instance using EC2Launch, and Map disks to volumes on your Windows instance.

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 Amazon EC2 instance root device volume.

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 Windows instances.

For Instance store volumes, you also specify the following information:

- The virtual device: `ephemeral[0-23]`. Note that the number and size of available instance store volumes for your instance varies by instance type.

For NVMe instance store volumes, the following information also applies:

- These volumes are automatically enumerated and assigned a device name; including them in your block device mapping has no effect.

For EBS volumes, you also specify the following information:

- The ID of the snapshot to use to create the block device (snap-*xxxxxxxx*). This value is optional as long as you specify a volume size. You can't specify the ID of an archived snapshot.
- The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- Whether to delete the volume on instance termination (`true` or `false`). The default value is `true` for the root device volume and `false` for attached volumes. When you create an AMI, its block device mapping inherits this setting from the instance. When you launch an instance, it inherits this setting from the AMI.
- The volume type, which can be gp2 and gp3 for General Purpose SSD, `io1` and `io2` for Provisioned IOPS SSD, `st1` for Throughput Optimized HDD, `sc1` for Cold HDD, or `standard` for Magnetic.
- The number of input/output operations per second (IOPS) that the volume supports. (Used only with `io1` and `io2` volumes.)

Block device mapping instance store caveats

There are several caveats to consider when launching instances with AMIs that have instance store volumes in their block device mappings.

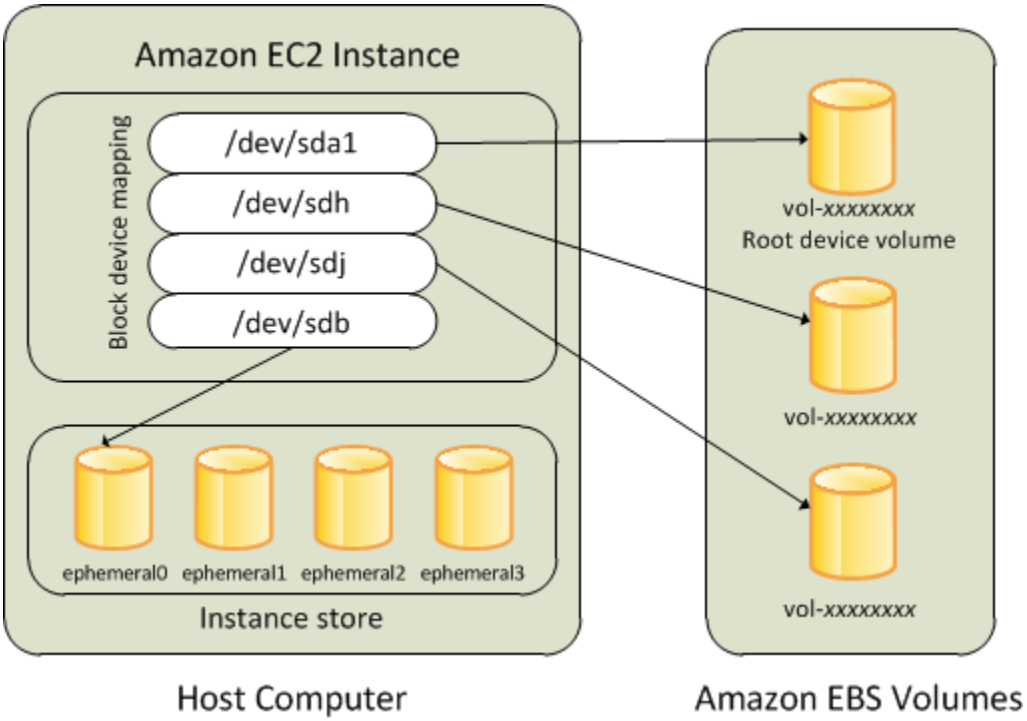
- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store volume.
- Instance store volumes can only be mapped at launch time. You cannot stop an instance without instance store volumes (such as the `t2.micro`), change the instance to a type that supports instance store volumes, and then restart the instance with instance store volumes. However, you can create an AMI from the instance and launch it on an instance type that supports instance store volumes, and map those instance store volumes to the instance.
- If you launch an instance with instance store volumes mapped, and then stop the instance and change it to an instance type with fewer instance store volumes and restart it, the instance store volume mappings from the initial launch still show up in the instance metadata. However, only the maximum number of supported instance store volumes for that instance type are available to the instance.

  > **Note**
  >
  > When an instance is stopped, all data on the instance store volumes is lost.

- Depending on instance store capacity at launch time, M3 instances may ignore AMI instance store block device mappings at launch unless they are specified at launch. You should specify instance store block device mappings at launch time, even if the AMI you are launching has the instance store volumes mapped in the AMI, to ensure that the instance store volumes are available when the instance launches.

Example block device mapping

This figure shows an example block device mapping for an EBS-backed instance. It maps `/dev/sdb` to `ephemeral0` and maps two EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the EBS volume that is the root device volume, `/dev/sda1`.



Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings in Specify a block device mapping for an AMI and Update the block device mapping when launching an instance.

How devices are made available in the operating system

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a Windows instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots, and then the Ec2Config service initializes and mounts the drives. The root device volume is mounted as `C:\`. The instance store volumes are mounted as `Z:\`, `Y:\`, and so on. When an EBS volume is mounted, it can be mounted using any available drive letter. However, you can configure how the Ec2Config Service assigns drive letters to EBS volumes; for more information, see Configure a Windows instance using the EC2Config service (legacy).

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": "xvdb",
    "VirtualName": "ephemeral0"
}
```

To add an empty 100 GiB gp2 volume, use the following mapping.

```
{
    "DeviceName": "xvdg",
    "Ebs": {
      "VolumeSize": 100
    }
}
```

To add an EBS volume based on a snapshot, use the following mapping.

```
{
    "DeviceName": "xvdh",
    "Ebs": {
      "SnapshotId": "snap-xxxxxxxx"
```

```
    }
}
```

To omit a mapping for a device, use the following mapping.

```
{
    "DeviceName": "xvdj",
    "NoDevice": ""
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the following commands (AWS Tools for Windows PowerShell):

- New-EC2Image
- Register-EC2Image

View the EBS volumes in an AMI block device mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. Choose **EBS images** from the **Filter** list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:

   - **Root Device Type** (`ebs`)
   - **Root Device Name** (for example, `/dev/sda1`)
   - **Block Devices** (for example, `/dev/sda1=snap-1234567890abcdef0:8:true`)

   If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (This screen doesn't display instance store volumes.)

To view the EBS volumes for an AMI using the command line

Use the describe-images (AWS CLI) command or Get-EC2Image (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an AMI.

Instance block device mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI.

Limitations

- For the root volume, you can only modify the following: volume size, volume type, and the **Delete on Termination** flag.
- When you modify an EBS volume, you can't decrease its size. Therefore, you must specify a snapshot whose size is equal to or greater than the size of the snapshot specified in the block device mapping of the AMI.

Contents

- Update the block device mapping when launching an instance
- 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:

- xvdb=ephemeral0
- xvdh=snap-1234567890abcdef0
- xvdj=:100

To prevent `xvdj` from attaching to an instance launched from this AMI, use the following mapping.

```
{
    "DeviceName": "xvdj",
    "NoDevice": ""
}
```

To increase the size of `xvdh` to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for `xvdh`, because specifying the device name is enough to identify the volume.

```
{
    "DeviceName": "xvdh",
    "Ebs": {
      "VolumeSize": 300
    }
}
```

To 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, `xvdc`, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect. If the instance supports NVMe instance store volumes, they are automatically enumerated and assigned an NVMe device name.

```
{
    "DeviceName": "xvdc",
    "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`, `xvdf`, and `xvdj`)

   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 Disk Management or PowerShell to list both EBS and instance store NVMe volumes. For more information, see the section called “List NVMe volumes”.

HDD or SSD instance store volumes

You can use instance metadata to query the HDD or SSD instance store volumes in the block device mapping. NVMe instance store volumes are not included.

The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`. For more information, see Instance metadata and user data.

First, connect to your running instance. From the instance, use this query to get its block device mapping.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-
mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store–backed `m1.small` instance looks like this.

```
ami
ephemeral0
root
swap
```

The `ami` device is the root device as seen by the instance. The instance store volumes are named `ephemeral[0-23]`. The `swap` device is for the page file. If you've also mapped EBS volumes, they appear as `ebs1`, `ebs2`, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-
mapping/ephemeral0
```

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, open Windows Disk Management. To learn how many instance store volumes are supported by each instance type, see Instance store volumes.

Map disks to volumes on your Windows instance

Your Windows instance comes with an EBS volume that serves as the root volume. If your Windows instance uses AWS PV or Citrix PV drivers, you can optionally add up to 25 volumes, making a total of 26 volumes. For more information, see Instance volume limits.

Depending on the instance type of your instance, you'll have from 0 to 24 possible instance store volumes available to the instance. To use any of the instance store volumes that are available to your instance, you must specify them when you create your AMI or launch your instance. You can also add EBS volumes when you create your AMI or launch your instance, or attach them while your instance is running.

When you add a volume to your instance, you specify the device name that Amazon EC2 uses. For more information, see Device names on Windows instances. AWS Windows Amazon Machine Images (AMIs) contain a set of drivers that are used by Amazon EC2 to map instance store and EBS volumes to Windows disks and drive letters. If you launch an instance from a Windows AMI that uses AWS PV or Citrix PV drivers, you can use the relationships described on this page to map your Windows disks to your instance store and EBS volumes. If your Windows AMI uses Red Hat PV drivers, you can update your instance to use the Citrix drivers. For more information, see Upgrade PV drivers on Windows instances.

Contents

- List NVMe volumes
  - List NVMe disks using Disk Management
  - List NVMe disks using PowerShell
  - Map NVMe EBS volumes
- List volumes
  - List disks using Disk Management
  - Map disk devices to device names
    - Instance store volumes
    - EBS volumes
  - List disks using PowerShell

List NVMe volumes

You can find the disks on your Windows instance using Disk Management or Powershell.

List NVMe disks using Disk Management

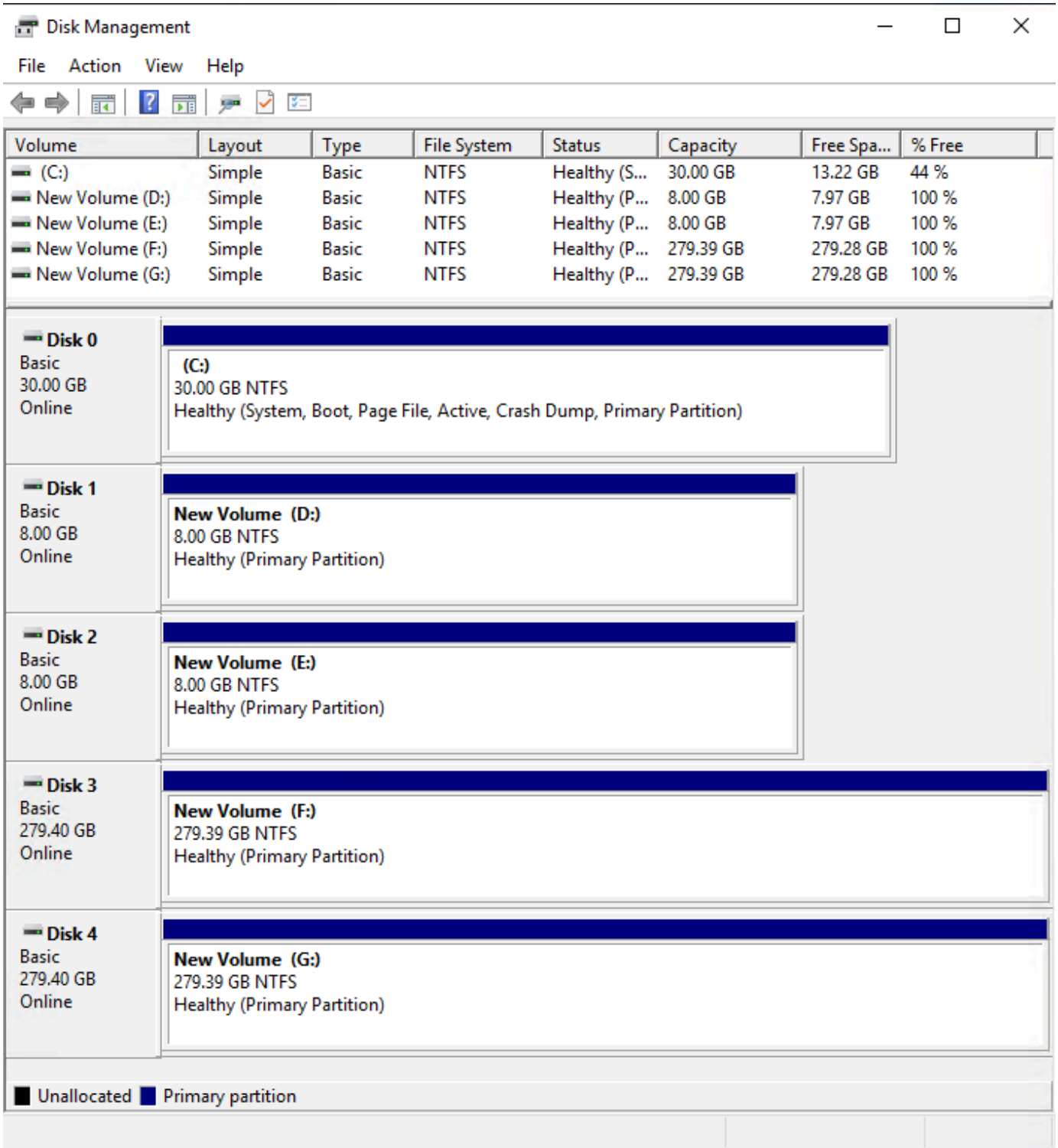
You can find the disks on your Windows instance using Disk Management.

To find the disks on your Windows instance

1. Log in to your Windows instance using Remote Desktop. For more information, see Connect to your Windows instance.
2. Start the Disk Management utility.

3. Review the disks. The root volume is an EBS volume mounted as `C:\`. If there are no other disks shown, then you didn't specify additional volumes when you created the AMI or launched the instance.

   The following is an example that shows the disks that are available if you launch an `r5d.4xlarge` instance with two additional EBS volumes.


Disk Management
File Action View Help
Volume Layout Type File System Status Capacity Free Spa... % Free
(C:) Simple Basic NTFS Healthy (S... 30.00 GB 13.22 GB 44 %
New Volume (D:) Simple Basic NTFS Healthy (P... 8.00 GB 7.97 GB 100 %
New Volume (E:) Simple Basic NTFS Healthy (P... 8.00 GB 7.97 GB 100 %
New Volume (F:) Simple Basic NTFS Healthy (P... 279.39 GB 279.28 GB 100 %
New Volume (G:) Simple Basic NTFS Healthy (P... 279.39 GB 279.28 GB 100 %
Disk 0
Basic
30.00 GB
Online
(C:)
30.00 GB NTFS
Healthy (System, Boot, Page File, Active, Crash Dump, Primary Partition)
Disk 1
Basic
8.00 GB
Online
New Volume (D:)
8.00 GB NTFS
Healthy (Primary Partition)
Disk 2
Basic
8.00 GB
Online
New Volume (E:)
8.00 GB NTFS
Healthy (Primary Partition)
Disk 3
Basic
279.40 GB
Online
New Volume (F:)
279.39 GB NTFS
Healthy (Primary Partition)
Disk 4
Basic
279.40 GB
Online
New Volume (G:)
279.39 GB NTFS
Healthy (Primary Partition)
Unallocated
Primary partition

List NVMe disks using PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume. It is intended for use with instances build on the Nitro System, which use NVMe EBS and instance store volumes.

Connect to your Windows instance and run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the following script and save it as `mapping.ps1` on your Windows instance.

```
# List the disks for NVMe volumes

function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function GetEBSVolumeId {
    param($Path)
    $SerialNumber = (Get-Disk -Path $Path).SerialNumber
    if($SerialNumber -clike 'vol*'){
        $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("vol","vol-")
    }
    else {
       $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("AWS","AWS-")
    }
    return $EbsVolumeId
}

function GetDeviceName{
    param($EbsVolumeId)
    if($EbsVolumeId -clike 'vol*'){

        $Device  = ((Get-EC2Volume -VolumeId $EbsVolumeId ).Attachment).Device
        $VolumeName = ""
    }
     else {
        $Device = "Ephemeral"
        $VolumeName = "Temporary Storage"
    }
```

```
    Return $Device,$VolumeName
}

function GetDriveLetter{
    param($Path)
    $DiskNumber =  (Get-Disk -Path $Path).Number
    if($DiskNumber -eq 0){
        $VirtualDevice = "root"
        $DriveLetter = "C"
        $PartitionNumber = (Get-Partition -DriveLetter C).PartitionNumber
    }
    else
    {
        $VirtualDevice = "N/A"
        $DriveLetter = (Get-Partition -DiskNumber $DiskNumber).DriveLetter
        if(!$DriveLetter)
        {
            $DriveLetter = ((Get-Partition -DiskId $Path).AccessPaths).Split(",")[0]
        }
        $PartitionNumber = (Get-Partition -DiskId $Path).PartitionNumber
    }

    return $DriveLetter,$VirtualDevice,$PartitionNumber

}

$Report = @()
foreach($Path in (Get-Disk).Path)
{
    $Disk_ID = ( Get-Partition -DiskId $Path).DiskId
    $Disk = ( Get-Disk -Path $Path).Number
    $EbsVolumeId  = GetEBSVolumeId($Path)
    $Size =(Get-Disk -Path $Path).Size
    $DriveLetter,$VirtualDevice, $Partition = (GetDriveLetter($Path))
    $Device,$VolumeName = GetDeviceName($EbsVolumeId)
    $Disk = New-Object PSObject -Property @{
      Disk          = $Disk
      Partitions    = $Partition
      DriveLetter   = $DriveLetter
      EbsVolumeId   = $EbsVolumeId
      Device        = $Device
      VirtualDevice = $VirtualDevice
      VolumeName= $VolumeName
    }
```

```
 $Report += $Disk
}

$Report | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, VolumeName
```

Run the script as follows:

```
PS C:\> .\mapping.ps1
```

The following is example output for an instance with a root volume, two EBS volumes, and two instance store volumes.

```
Disk Partitions DriveLetter EbsVolumeId           Device    VirtualDevice VolumeName
---- ---------- ----------- -----------           ------    ------------- ----------
   0          1 C           vol-03683f1d861744bc7 /dev/sda1 root
   1          1 D           vol-082b07051043174b9 xvdb      N/A
   2          1 E           vol-0a4064b39e5f534a2 xvdc      N/A
   3          1 F           AWS-6AAD8C2AEEE1193F0 Ephemeral N/A           Temporary
 Storage
   4          1 G           AWS-13E7299C2BD031A28 Ephemeral N/A           Temporary
 Storage
```

If you did not configure your credentials for Tools for Windows PowerShell on the Windows instance, the script cannot get the EBS volume ID and uses N/A in the `EbsVolumeId` column.

Map NVMe EBS volumes

With instances built on the Nitro System, EBS volumes are exposed as NVMe devices. You can use the Get-Disk command to map Windows disk numbers to EBS volume IDs.

```
PS C:\> Get-Disk
Number Friendly Name Serial Number                    HealthStatus
 OperationalStatus      Total Size Partition

                      Style
------ ------------- -------------                    ------------
 -----------------      ---------- ----------
3      NVMe Amazo... AWS6AAD8C2AEEE1193F0_00000001.   Healthy              Online
             279.4 GB MBR
4      NVMe Amazo... AWS13E7299C2BD031A28_00000001.   Healthy              Online
             279.4 GB MBR
```

```
2      NVMe Amazo... vol0a4064b39e5f534a2_00000001.   Healthy              Online
                 8 GB MBR
0      NVMe Amazo... vol03683f1d861744bc7_00000001.   Healthy              Online
                30 GB MBR
1      NVMe Amazo... vol082b07051043174b9_00000001.   Healthy              Online
                 8 GB MBR
```

You can also run the **ebsnvme-id** command to map NVMe disk numbers to EBS volume IDs and device names.

```
PS C:\> C:\PROGRAMDATA\Amazon\Tools\ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-03683f1d861744bc7
Device Name: sda1

Disk Number: 1
Volume ID: vol-082b07051043174b9
Device Name: xvdb

Disk Number: 2
Volume ID: vol-0a4064b39e5f534a2
Device Name: xvdc
```

List volumes

You can find the disks on your Windows instance using Disk Management or Powershell.

List disks using Disk Management

You can find the disks on your Windows instance using Disk Management.

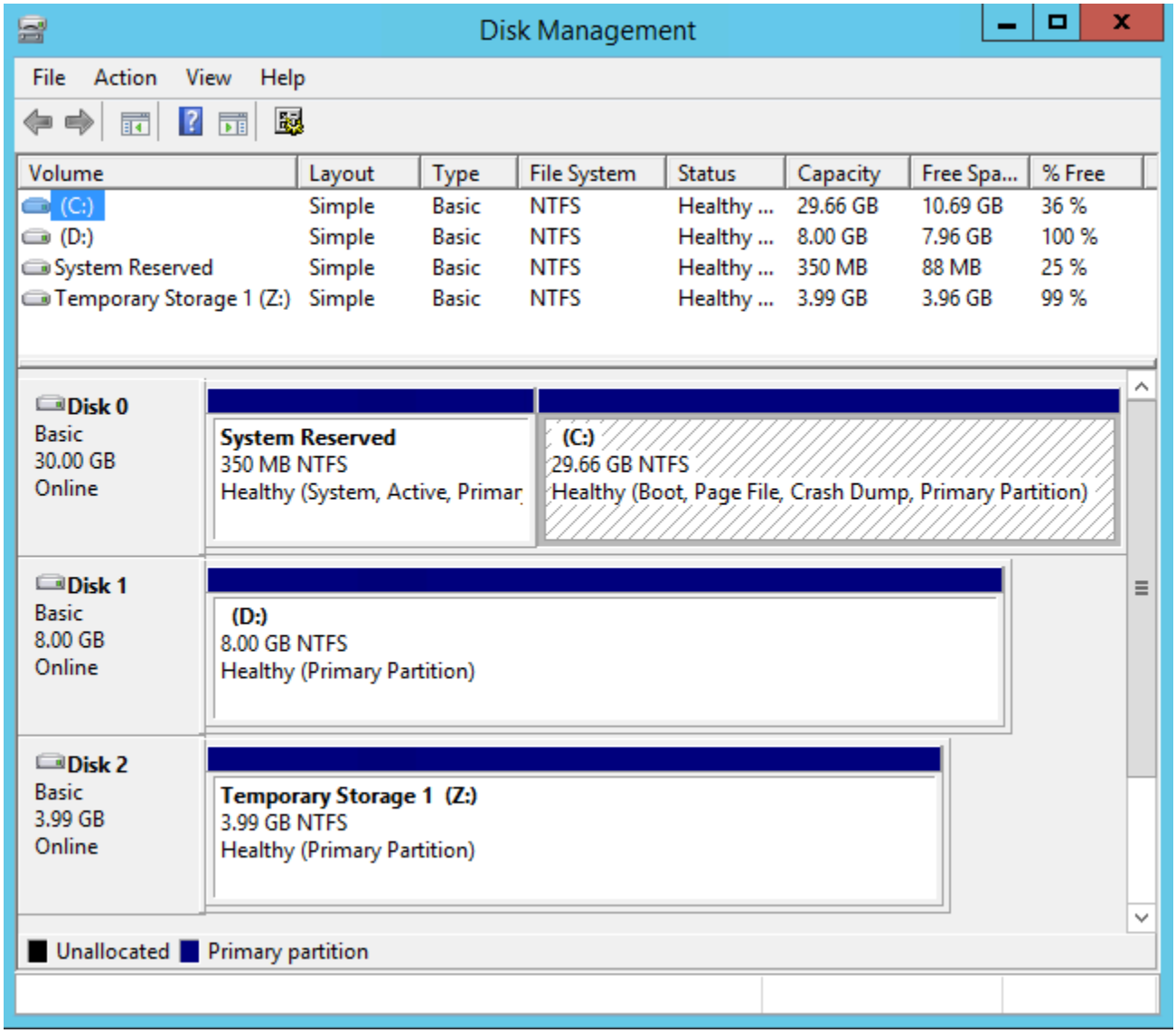
To find the disks on your Windows instance

1. Log in to your Windows instance using Remote Desktop. For more information, see Connect to your Windows instance.
2. Start the Disk Management utility.

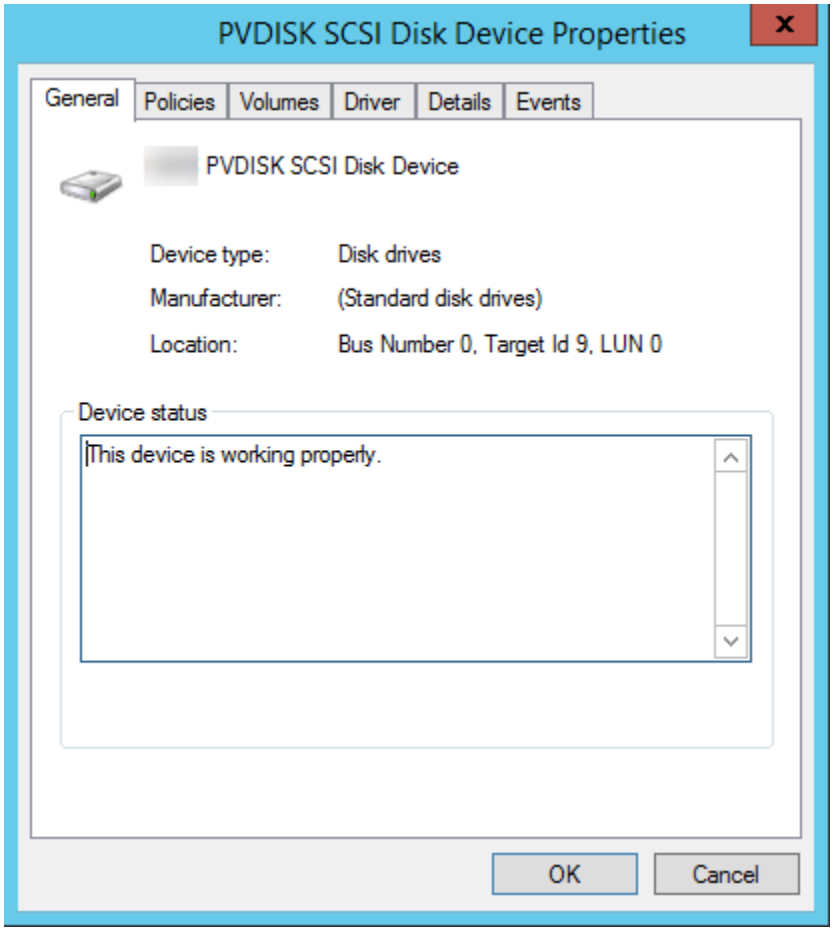
   On Windows Server 2012 and later, on the taskbar, right-click the Windows logo, and then choose **Disk Management**. On Windows Server 2008, choose **Start**, **Administrative Tools**, **Computer Management**, **Disk Management**.

3. Review the disks. The root volume is an EBS volume mounted as `C:\`. If there are no other disks shown, then you didn't specify additional volumes when you created the AMI or launched the instance.

   The following is an example that shows the disks that are available if you launch an `m3.medium` instance with an instance store volume (Disk 2) and an additional EBS volume (Disk 1).



4. Right-click the gray pane labeled Disk 1, and then select **Properties**. Note the value of **Location** and look it up in the tables in Map disk devices to device names. For example, the following disk has the location Bus Number 0, Target Id 9, LUN 0. According to the table for EBS volumes, the device name for this location is `xvdj`.



Map disk devices to device names

The block device driver for the instance assigns the actual volume names when mounting volumes.

Mappings

- Instance store volumes
- EBS volumes

Instance store volumes

The following table describes how the Citrix PV and AWS PV drivers map non-NVMe instance store volumes to Windows volumes. The number of available instance store volumes is determined by the instance type. For more information, see Instance store volumes.

Location	Device name
Bus Number 0, Target ID 78, LUN 0	xvdca
Bus Number 0, Target ID 79, LUN 0	xvdcb
Bus Number 0, Target ID 80, LUN 0	xvdcc
Bus Number 0, Target ID 81, LUN 0	xvdcd
Bus Number 0, Target ID 82, LUN 0	xvdce
Bus Number 0, Target ID 83, LUN 0	xvdcf
Bus Number 0, Target ID 84, LUN 0	xvdcg
Bus Number 0, Target ID 85, LUN 0	xvdch
Bus Number 0, Target ID 86, LUN 0	xvdci
Bus Number 0, Target ID 87, LUN 0	xvdcj
Bus Number 0, Target ID 88, LUN 0	xvdck
Bus Number 0, Target ID 89, LUN 0	xvdcl

EBS volumes

The following table describes how the Citrix PV and AWS PV drivers map non-NVME EBS volumes to Windows volumes.

Location	Device name
Bus Number 0, Target ID 0, LUN 0	/dev/sda1
Bus Number 0, Target ID 1, LUN 0	xvdb
Bus Number 0, Target ID 2, LUN 0	xvdc
Bus Number 0, Target ID 3, LUN 0	xvdd

Location	Device name
Bus Number 0, Target ID 4, LUN 0	xvde
Bus Number 0, Target ID 5, LUN 0	xvdf
Bus Number 0, Target ID 6, LUN 0	xvdg
Bus Number 0, Target ID 7, LUN 0	xvdh
Bus Number 0, Target ID 8, LUN 0	xvdi
Bus Number 0, Target ID 9, LUN 0	xvdj
Bus Number 0, Target ID 10, LUN 0	xvdk
Bus Number 0, Target ID 11, LUN 0	xvdl
Bus Number 0, Target ID 12, LUN 0	xvdm
Bus Number 0, Target ID 13, LUN 0	xvdn
Bus Number 0, Target ID 14, LUN 0	xvdo
Bus Number 0, Target ID 15, LUN 0	xvdp
Bus Number 0, Target ID 16, LUN 0	xvdq
Bus Number 0, Target ID 17, LUN 0	xvdr
Bus Number 0, Target ID 18, LUN 0	xvds
Bus Number 0, Target ID 19, LUN 0	xvdt
Bus Number 0, Target ID 20, LUN 0	xvdu
Bus Number 0, Target ID 21, LUN 0	xvdv
Bus Number 0, Target ID 22, LUN 0	xvdw
Bus Number 0, Target ID 23, LUN 0	xvdx

Location	Device name
Bus Number 0, Target ID 24, LUN 0	xvdy
Bus Number 0, Target ID 25, LUN 0	xvdz

List disks using PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume.

Requirements and limitations

- Requires Windows Server 2012 or later.
- Requires credentials to get the EBS volume ID. You can configure a profile using the Tools for PowerShell, or attach an IAM role to the instance.
- Does not support NVMe volumes.
- Does not support dynamic disks.

Connect to your Windows instance and run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the following script and save it as `mapping.ps1` on your Windows instance.

```
# List the disks
function Convert-SCSITargetIdToDeviceName {
  param([int]$SCSITargetId)
  If ($SCSITargetId -eq 0) {
    return "sda1"
  }
  $deviceName = "xvd"
  If ($SCSITargetId -gt 25) {
    $deviceName += [char](0x60 + [int]($SCSITargetId / 26))
  }
  $deviceName += [char](0x61 + $SCSITargetId % 26)
  return $deviceName
}
```

```
[string[]]$array1 = @()
[string[]]$array2 = @()
[string[]]$array3 = @()
[string[]]$array4 = @()

Get-WmiObject Win32_Volume | Select-Object Name, DeviceID | ForEach-Object {
  $array1 += $_.Name
  $array2 += $_.DeviceID
}

$i = 0
While ($i -ne ($array2.Count)) {
  $array3 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-Disk).SerialNumber) -
replace "_[^ ]*$" -replace "vol", "vol-"
  $array4 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-Disk).FriendlyName)
  $i ++
}

[array[]]$array = $array1, $array2, $array3, $array4

Try {
  $InstanceId = Get-EC2InstanceMetadata -Category "InstanceId"
  $Region = Get-EC2InstanceMetadata -Category "Region" | Select-Object -ExpandProperty
 SystemName
}
Catch {
  Write-Host "Could not access the instance Metadata using AWS Get-EC2InstanceMetadata
 CMDLet.
Verify you have AWSPowershell SDK version '3.1.73.0' or greater installed and Metadata
 is enabled for this instance." -ForegroundColor Yellow
}
Try {
  $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
 $InstanceId).Instances.BlockDeviceMappings
  $VirtualDeviceMap = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping").GetEnumerator() | Where-Object { $_.Key -ne "ami" }
}
Catch {
  Write-Host "Could not access the AWS API, therefore, VolumeId is not available.
Verify that you provided your access keys or assigned an IAM role with adequate
 permissions." -ForegroundColor Yellow
}

Get-disk | ForEach-Object {
```

```
  $DriveLetter = $null
  $VolumeName = $null
  $VirtualDevice = $null
  $DeviceName = $_.FriendlyName

  $DiskDrive = $_
  $Disk = $_.Number
  $Partitions = $_.NumberOfPartitions
  $EbsVolumeID = $_.SerialNumber -replace "_[^ ]*$" -replace "vol", "vol-"
  if ($Partitions -ge 1) {
    $PartitionsData = Get-Partition -DiskId $_.Path
    $DriveLetter = $PartitionsData.DriveLetter | Where-object { $_ -notin @("",
$null) }
    $VolumeName = (Get-PSDrive | Where-Object { $_.Name -in
@($DriveLetter) }).Description | Where-object { $_ -notin @("", $null) }
  }
  If ($DiskDrive.path -like "*PROD_PVDISK*") {
    $BlockDeviceName = Convert-SCSITargetIdToDeviceName((Get-WmiObject -Class
Win32_Diskdrive | Where-Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" +
$DiskDrive.Number) }).SCSITargetId)
    $BlockDeviceName = "/dev/" + $BlockDeviceName
    $BlockDevice = $BlockDeviceMappings | Where-Object { $BlockDeviceName -like "*" +
$_.DeviceName + "*" }
    $EbsVolumeID = $BlockDevice.Ebs.VolumeId
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -eq
$BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON_EC2_NVME*") {
    $BlockDeviceName = (Get-EC2InstanceMetadata -Category
"BlockDeviceMapping").ephemeral((Get-WmiObject -Class Win32_Diskdrive | Where-Object
{ $_.DeviceID -eq ("\\.\PHYSICALDRIVE" + $DiskDrive.Number) }).SCSIPort - 2)
    $BlockDevice = $null
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -eq
$BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
```

```
      }
    }
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  ElseIf ($DiskDrive.path -like "*NETAPP*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $EbsVolumeID = "FSxN Volume"
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  Else {
    $BlockDeviceName = $null
    $BlockDevice = $null
  }
  New-Object PSObject -Property @{
    Disk          = $Disk;
    Partitions    = $Partitions;
    DriveLetter   = If ($DriveLetter -eq $null) { "N/A" } Else { $DriveLetter };
    EbsVolumeId   = If ($EbsVolumeID -eq $null) { "N/A" } Else { $EbsVolumeID };
    Device        = If ($BlockDeviceName -eq $null) { "N/A" } Else
 { $BlockDeviceName };
    VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
    VolumeName    = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
    DeviceName    = If ($DeviceName -eq $null) { "N/A" } Else { $DeviceName };
  }
} | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions, DriveLetter,
 EbsVolumeId, Device, VirtualDevice, DeviceName, VolumeName
```

Run the script as follows:

```
PS C:\> .\mapping.ps1
```

The following is example output.

```
Disk  Partitions  DriveLetter   EbsVolumeId             Device      VirtualDevice
 DeviceName              VolumeName
----  ----------  -----------   -----------             ------      -------------
 ----------              ----------
   0           1            C   vol-0561f1783298efedd   /dev/sda1   N/A
 NVMe Amazon Elastic B   N/A
   1           1            D   vol-002a9488504c5e35a   xvdb        N/A
 NVMe Amazon Elastic B   N/A
   2           1            E   vol-0de9d46fcc907925d   xvdc        N/A
 NVMe Amazon Elastic B   N/A
```

If you did not provide your credentials on the Windows instance, the script cannot get the EBS volume ID and uses N/A in the `EbsVolumeId` column.

Application-consistent snapshots based on Windows VSS

You can take application-consistent snapshots of all Amazon EBS volumes attached to your Windows on Amazon EC2 instances using AWS Systems Manager Run Command. The snapshot process uses the Windows Volume Shadow Copy Service (VSS) to take EBS volume level backups of VSS-aware applications. The snapshots include data from pending transactions between these applications and the disk. You don't have to shut down your instances or disconnect them when you need to back up all attached volumes.

There is no additional cost to use VSS-enabled EBS snapshots. You only pay for EBS snapshots created by the backup process. For more information, see How is my EBS snapshot bill calculated?

Contents

- What is AWS VSS?
- Prerequisites
- Create VSS-enabled EBS snapshots
- Troubleshooting
- Restore EBS volumes from VSS-enabled EBS snapshots
- AWS VSS solution version history

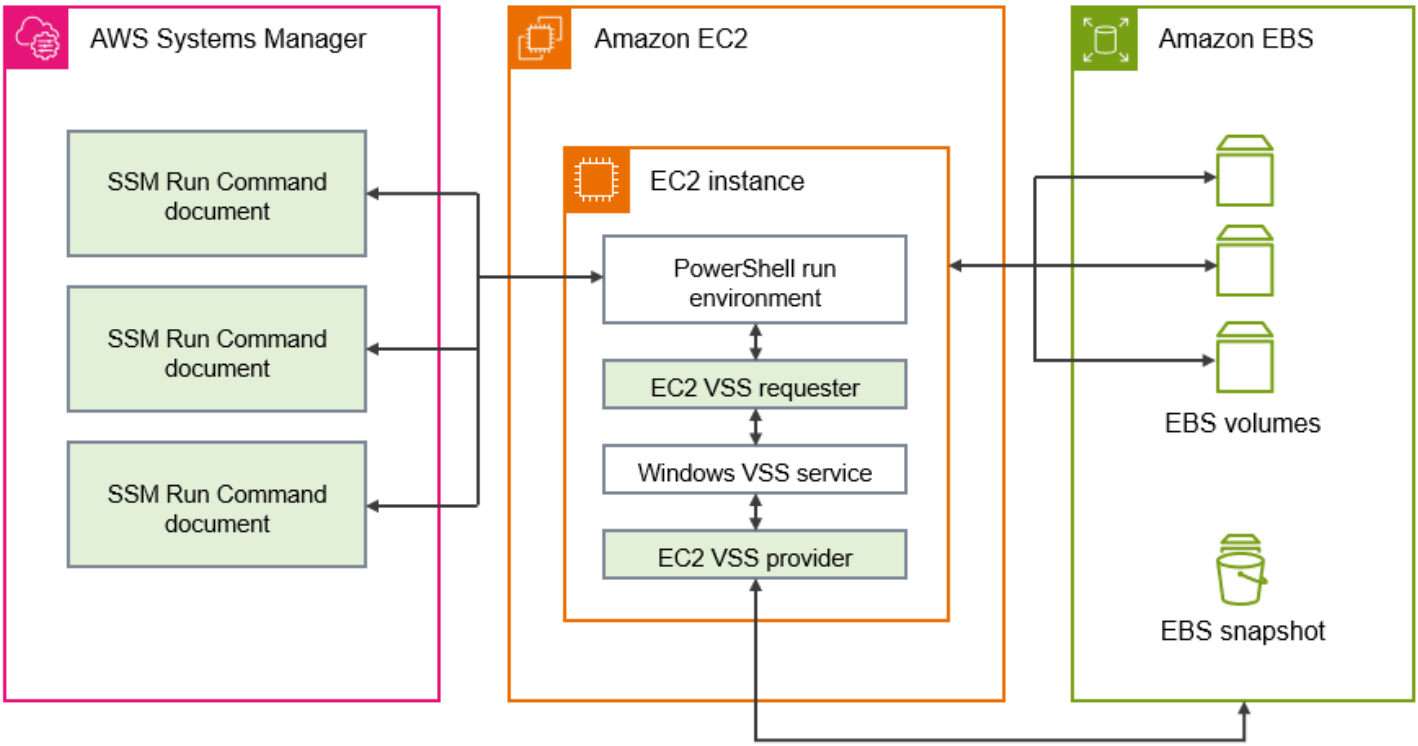
What is AWS VSS?

Volume Snapshot Copy Service (VSS) is a backup and recovery technology included in Microsoft Windows. It can create backup copies, or snapshots, of computer files or volumes while they are in use. For more information, see Volume Shadow Copy Service.

To create an application-consistent snapshot, the following software components are involved.

- *VSS service* — Part of the Windows operating system
- *VSS requester* — The software that requests the creation of shadow copies
- *VSS writer* — Typically provided as part of an application, such as SQL Server, to ensure a consistent data set to back up
- *VSS provider* — The component that creates the shadow copies of the underlying volumes

The AWS VSS solution comprises of multiple Systems Manager (SSM) Run Command documents that facilitate backup creation, and a Systems Manager Distributor package, called `AwsVssComponents`, that includes an *EC2 VSS requester* and an *EC2 VSS provider*. The `AwsVssComponents` package must be installed on EC2 Windows instances to take application-consistent snapshots of EBS volumes. The following diagram illustrates the relationship between these software components.



How the AWS VSS solution works

The process for taking application-consistent, VSS-enabled EBS snapshots consists of the following steps.

1. Complete the Prerequisites.
2. Enter parameters for the `AWSEC2-VssInstallAndSnapshot` SSM document and run this document by using Run Command. For more information, see Run the AWSEC2-VssInstallAndSnapshot command document (recommended).
3. The Windows VSS service on your instance coordinates all ongoing I/O operations for running applications.
4. The system flushes all I/O buffers and temporarily pauses all I/O operations. The pause lasts, at most, ten seconds.
5. During the pause, the system creates snapshots of all volumes attached to the instance.
6. The pause is lifted and I/O resumes operation.
7. The system adds all newly-created snapshots to the list of EBS snapshots. The system tags all VSS-enabled EBS snapshots successfully created by this process with **AppConsistent:true**.
8. If you need to restore from a snapshot, you can use the standard EBS process of creating a volume from a snapshot, or you can restore all volumes to an instance by using a sample script, as described in Restore EBS volumes from VSS-enabled EBS snapshots.

Prerequisites

You can create VSS snapshots with Systems Manager Run Command, AWS Backup, or Amazon Data Lifecycle Manager. The following prerequisites apply for all solutions.

Prerequisites

- System requirements
- IAM permissions
- VSS components

System requirements

Install the SSM Agent

VSS is orchestrated by the AWS Systems Manager (Systems Manager) using PowerShell. Ensure that you have installed SSM Agent version `3.0.502.0` or later on your EC2 instance. If you are already using an older version of the SSM Agent, update it using Run Command. For more information, see Setting up Systems Manager for Amazon EC2 instances and Working with SSM Agent on Amazon EC2 instances for Windows Server in the *AWS Systems Manager User Guide*.

Amazon EC2 Windows instance requirements

VSS-enabled EBS snapshots are supported for instances running Windows Server 2012 and later. For older versions of Windows, see Windows version support table in AWS VSS solution version history.

.NET Framework version

The `AwsVssComponents` package requires .NET Framework version `4.6` or later. If you are using Windows 2012 or 2012 R2, the default .NET Framework version is earlier than `4.6` and you must install version `4.6` or later using Windows Update.

AWS Tools for Windows PowerShell version

Ensure that your instance is running AWS Tools for Windows PowerShell version `3.3.48.0` or later. To check your version, run the following command in the PowerShell terminal on the instance.

```
C:\> Get-AWSPowerShellVersion
```

If you need to update AWS Tools for Windows PowerShell on your instance, see Installing the AWS Tools for Windows PowerShell in the *AWS Tools for Windows PowerShell User Guide*.

Windows PowerShell version

Ensure that your instance is running Windows PowerShell major version 3, 4, or 5. To check your version, run the following command in a PowerShell terminal on the instance.

```
C:\> $PSVersionTable.PSVersion
```

PowerShell language mode

Ensure that your instance has the PowerShell language mode set to `FullLanguage`. For more information, see about_Language_Modes in the Microsoft documentation.

IAM permissions

VSS requires the IAM role attached to your Amazon EC2 Windows instance to have the necessary IAM policies attached. The following sections describe how to create the required IAM policy and instance profile. The policy enables Systems Manager to create snapshots, tag snapshots, and attach metadata, such as device ID, to the default snapshot tags that the system creates.

To create an IAM policy for VSS-enabled snapshots

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**, and then choose **Create policy**.
3. On the **Create policy** page, choose the **JSON** tab, and then replace the default content with the following JSON policy.

> **Note**
>
> The following policy contains all recommended IAM permissions. You can scope IAM permissions if needed. For more information, see Scope down permissions for VSS (advanced).

> **Note**
>
> `ec2:DescribeInstanceAttribute` is supported on `AwsVssComponents` version 2.2.1 and later. If you have an older version of `AwsVssComponents` installed, specify `ec2:DescribeInstances` instead.

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

```
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*::image/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstanceAttribute",
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:CreateImage",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Choose **Next**.
5. For **Policy name**, enter **`VssSnapshotRolePolicy`**.
6. Choose **Create policy**.

The following procedure creates an instance profile using the IAM policy created above and attaches it to an instance.

To create and attach the instance profile

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, and then choose **Create role**.
3. For **Trusted entity type**, choose **AWS Service**.
4. For **Use case**, choose **EC2**, and then choose **Next**.
5. Search for and select the following policies, and then choose **Next**.
   - **AmazonSSMManagedInstanceCore**
   - **VssSnapshotRolePolicy** (This is the policy you created earlier)

6. For **Role name**, enter a name for the role, such as **`VssSnapshotRole`**. And for **Description**, replace the default text with a description of the role's purpose.
7. Choose **Create role**.
8. Attach this role to the instances for which you want to create VSS-enabled EBS snapshots. For more information, see Attach an IAM role to an instance.

Scope down permissions for VSS (advanced)

To scope down the IAM policy needed for VSS (advanced)

- If you only use the **CreateAmi** option or create VSS snapshots through the AWS Backup service, then you can omit `ec2:CreateSnapshot` and `ec2:CreateSnapshots` from the second policy statement.
- If you do not use the **CreateAmi** option, then you can omit `arn:aws:ec2:*::image/*` from the first policy statement and you can omit `ec2:CreateImage` and `ec2:DescribeImages` from the second policy statement.

VSS components

To create application-consistent snapshots on Windows operating systems, the `AwsVssComponents` package must be installed on the instance. The package contains an on-instance EC2 VSS Agent that functions as the VSS requester, and an EC2 VSS provider for EBS volumes.

There are several ways that you can install the component onto an existing instance:

- (Recommended) Run the AWSEC2-VssInstallAndSnapshot command document (recommended). This automatically installs or updates if needed every time it runs.
- Manually install VSS components on an instance.
- Update VSS components on your instances on a schedule.

You can also create an AMI with EC2 Image Builder that uses the `aws-vss-components-windows` managed component to install the `AwsVssComponents` package for the image. The managed component uses AWS Systems Manager Distributor to install the package. After Image Builder creates the image, every instance that you launch from the associated AMI will have the VSS package installed on it. For more information about how you can create an AMI with the VSS

package installed, see Distributor package managed components for Windows in the *EC2 Image Builder User Guide*.

Contents

- Manually install VSS components on an instance
- Update VSS components on your instances on a schedule

Manually install VSS components on an instance

Your EC2 Windows instance must have VSS components installed before you can create application-consistent snapshots with Systems Manager. If you don't run the `AWSEC2-VssInstallAndSnapshot` command document to automatically install or update the package every time you create application-consistent snapshots, you must manually install the package.

You must also install manually if you plan to use one of the following methods to create application-consistent snapshots from your EC2 instance.

- Create VSS snapshots using AWS Backup
- Create VSS snapshots using Amazon Data Lifecycle Manager

If you need to perform a manual install, we recommend that you use the latest AWS VSS component package to improve the reliability and performance of application-consistent snapshots on your EC2 Windows instances.

> **Note**
>
> To automatically install or update the `AwsVssComponents` package whenever you create application-consistent snapshots, we recommend that you use Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` document. For more information, see Run the AWSEC2-VssInstallAndSnapshot command document (recommended).

To install the VSS components on an Amazon EC2 Windows instance, follow the steps for your preferred environment.

Console

To install the VSS components using SSM Distributor

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose **Run Command**.
3. Choose **Run command**.
4. For **Command document**, choose the button next to **AWS-ConfigureAWSPackage**.
5. For **Command parameters**, do the following:
   a. Verify that **Action** is set to **Install**.
   b. For **Name**, enter `AwsVssComponents`.
   c. For **Version**, enter a version or leave the field empty so that Systems Manager installs the latest version.
6. For **Targets**, identify the instances on which you want to run this operation by specifying tags or selecting instances manually.

   > **Note**
   > If you choose to select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? in the *AWS Systems Manager User Guide* for troubleshooting tips.

7. For **Other parameters**:
   - (Optional) For **Comment**, type information about this command.
   - For **Timeout (seconds)**, specify the number of seconds for the system to wait before failing the overall command execution.
8. (Optional) For **Rate control**:
   - For **Concurrency**, specify either a number or a percentage of instances on which to run the command at the same time.

> **Note**
>
> If you selected targets by choosing Amazon EC2 tags, and you are not certain how many instances use the selected tags, then limit the number of instances that can run the document at the same time by specifying a percentage.

- For **Error threshold**, specify when to stop running the command on other instances after it fails on either a number or a percentage of instances. For example, if you specify three errors, then Systems Manager stops sending the command when the fourth error is received. Instances still processing the command might also send errors.

9. (Optional) For **Output options** section, if you want to save the command output to a file, select the box next to **Enable writing to an S3 bucket**. Specify the bucket and (optional) prefix (folder) names.

> **Note**
>
> The S3 permissions that grant the ability to write the data to an S3 bucket are those of the instance profile assigned to the instance, not those of the user performing this task. For more information, see Create an IAM Instance Profile for Systems Manager in the *AWS Systems Manager User Guide*.

10. (Optional) Specify options for **SNS notifications**.

    For information about configuring Amazon SNS notifications for Run Command, see Configuring Amazon SNS Notifications for AWS Systems Manager.

11. Choose **Run**.

AWS CLI

Use the following procedure to download and install the `AwsVssComponents` package on your instances by using Run Command from the AWS CLI. The package installs two components: a VSS requester and a VSS provider. The system copies these components to a directory on the instance, and then registers the provider DLL as a VSS provider.

To install the VSS package by using the AWS CLI

- Run the following command to download and install the required VSS components for Systems Manager.

```
aws ssm send-command \
  --document-name "AWS-ConfigureAWSPackage" \
  --instance-ids "i-01234567890abcdef" \
  --parameters '{"action":["Install"],"name":["AwsVssComponents"]}'
```

PowerShell

Use the following procedure to download and install the `AwsVssComponents` package on your instances by using Run Command from the Tools for Windows PowerShell. The package installs two components: a VSS requester and a VSS provider. The system copies these components to a directory on the instance, and then registers the provider DLL as a VSS provider.

To install the VSS package using the AWS Tools for Windows PowerShell

- Run the following command to download and install the required VSS components for Systems Manager.

```
Send-SSMCommand -DocumentName AWS-ConfigureAWSPackage -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'action'='Install';'name'='AwsVssComponents'}
```

Verify the signature on AWS VSS components

Use the following procedure to verify the signature on the `AwsVssComponents` package.

1. Connect to your Windows instance. For more information, see Connect to your Windows instance.
2. Navigate to C:\Program Files\Amazon\AwsVssComponents.
3. Open the context (right-click) menu for `ec2-vss-agent.exe`, and then choose **Properties**.
4. Navigate to the **Digital Signatures** tab and verify that the name of the signer is Amazon Web Services Inc.

5. Use the preceding steps to verify the signature on `Ec2VssInstaller` and `Ec2VssProvider.dll`.

Update VSS components on your instances on a schedule

We recommend that you keep the VSS components updated with the latest recommended version. There are several different ways that you can update components when a new version of the `AwsVssComponents` package is released.

Update methods

- You can repeat the steps described in Manually install VSS components on an instance when a new version of the AWS VSS components is released.
- You can configure a Systems Manager State Manager association to automatically download and install new or updated VSS components when the `AwsVssComponents` package becomes available.
- You can automatically install or update the `AwsVssComponents` package whenever you create application-consistent snapshots, when you use Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` document.

> **Note**
>
> We recommend that you use Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` command document, which automatically installs or updates the `AwsVssComponents` package before it creates the application-consistent snapshots. For more information, see Run the AWSEC2-VssInstallAndSnapshot command document (recommended).

To create a Systems Manager State Manager association, follow the steps for your preferred environment.

Console

To create a State Manager association using the console

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.

2. In the navigation pane, choose **State Manager**.

   Or, if the Systems Manager homepage opens first, open the navigation pane and then choose **State Manager**.

3. Choose **Create association**.
4. In the **Name** field, enter a descriptive name.
5. In the **Document** list, choose **AWS-ConfigureAWSPackage**.
6. In the **Parameters** section, choose **Install** from the **Action** list.
7. For **Installation type**, choose **Uninstall and reinstall**.
8. In the **Name** field, enter `AwsVssComponents`. You can keep the **Version** and **Additional Arguments** fields empty.
9. In the **Targets** section, choose an option.

   > **Note**
   >
   > If you choose to target instances by using tags, and you specify tags that map to Linux instances, the association succeeds on the Windows instance but fails on the Linux instances. The overall status of the association shows **Failed**.

10. In the **Specify schedule** section, choose an option.
11. In the **Advanced options** section, for **Compliance severity**, choose a severity level for the association. For more information, see About State Manager association compliance. For **Change Calendars**, select a preconfigured change calendar. For more information, see about AWS Systems Manager Change Calendar.
12. For **Rate control**, do the following:

    - For **Concurrency**, specify either a number or a percentage of managed nodes on which to run the command at the same time.
    - For **Error threshold**, specify when to stop running the command on other managed nodes after it fails on either a number or a percentage of nodes.

13. (Optional) For **Output options**, to save the command output to a file, select **Enable writing output to S3**. Enter the bucket and prefix (folder) names in the boxes.
14. Choose **Create association**, and then choose **Close**. The system attempts to create the association on the instances and immediately apply the state.

> **Note**
>
> If EC2 instances for Windows Server show a status of **Failed**, verify that the SSM Agent is running on the instance, and verify that the instance is configured with an AWS Identity and Access Management (IAM) role for Systems Manager. For more information, see Setting up AWS Systems Manager.

AWS CLI

You can run create-association AWS CLI command to update a Distributor package on a schedule without taking the associated application offline. Only new or updated files in the package are replaced.

To create a State Manager association using the AWS CLI

1. Install and configure the AWS CLI, if you haven't already. For information, see Install or update the latest version of the AWS CLI.
2. Run the following command to create an association. The value of `--name`, the document name, is always `AWS-ConfigureAWSPackage`. The following command uses the key `InstanceIds` to specify target instances.

```
aws ssm create-association \
 --name "AWS-ConfigureAWSPackage" \
 --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["AwsVssComponents"]}' \
 --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"i-01234567890abcdef\",
\"i-000011112222abcde\"]}]
```

For information about other options you can use with the `create-association` command, see create-association in the AWS Systems Manager section of the AWS CLI Command Reference.

Create VSS-enabled EBS snapshots

This section includes the steps to create VSS-enabled EBS snapshots.

You can create VSS-enabled EBS snapshots of EBS volumes attached to your EC2 instances. Before you attempt to create a VSS-enabled snapshot, ensure that the Prerequisites are met.

Topics

- Create VSS snapshots with AWS Systems Manager command documents
- Create VSS snapshots using AWS Backup
- Create VSS snapshots using Amazon Data Lifecycle Manager

Create VSS snapshots with AWS Systems Manager command documents

You can use AWS Systems Manager command documents to create VSS-enabled snapshots. The following content introduces the command documents that are available, and the runtime parameters that the documents use to create your snapshots.

Before you use any of the Systems Manager command documents, ensure that you've met all Prerequisites.

Topics

- Parameters for Systems Manager VSS snapshot documents
- Run Systems Manager VSS snapshot command documents

Parameters for Systems Manager VSS snapshot documents

The Systems Manager documents that create VSS snapshots all use the following parameters, except where noted:

ExcludeBootVolume (string, optional)

This setting excludes boot volumes from the backup process if you create snapshots. To exclude boot volumes from your snapshots, set **ExcludeBootVolume** to `True`, and **CreateAmi** to `False`.

If you create an AMI for your backup, this parameter should be set to `False`. The default value for this parameter is `False`.

NoWriters (string, optional)

To exclude application VSS writers from the snapshot process, set this parameter to `True`. Excluding application VSS writers can help you resolve conflicts with third-party VSS backup components. The default value for this parameter is `False`.

CopyOnly (string, optional)

If you use the native SQL Server backup in addition to AWS VSS, performing a Copy-only backup prevents AWS VSS from breaking the native differential backup chain. To perform a Copy-only backup operation, set this parameter to `True`.

The default value for this parameter is `False`, which causes AWS VSS to perform a full backup operation.

CreateAmi (string, optional)

To create a VSS-enabled Amazon Machine Image (AMI) to back up your instance, set this parameter to `True`. The default value for this parameter is `False`, which backs up your instance with an EBS snapshot instead.

For more information about creating an AMI from an instance, see Create a Windows AMI from a running instance.

AmiName (string, optional)

If the **CreateAmi** option is set to `True`, specify the name of the AMI that the backup creates.

description (string, optional)

Specify a description for the snapshots or image that this process creates.

tags (string, optional)

We recommend that you tag your snapshots and images to help you locate and manage your resources, for example, to restore volumes from a list of snapshots. The system adds the `Name` key, with a blank value where you can specify the name that you want to apply to your output snapshots or images.

If you want to specify additional tags, separate tags with a semicolon in between. For example, `Key=Environment,Value=Test;Key=User,Value=TestUser1`.

By default, the system adds the following reserved tags for VSS-enabled snapshots and images.

- **Device** – For VSS-enabled snapshots, this is the device name of the EBS volume that the snapshot captures.
- **AppConsistent** – This tag indicates the successful creation of a VSS-enabled snapshot or AMI.
- **AwsVssConfig** – This identifies snapshots and AMIs that are created with VSS enabled. The tag includes meta information such as the `AwsVssComponents` version.

> **Warning**
>
> Specifying any of these reserved tags in your parameter list will cause an error.

executionTimeout (string, optional)

Specify the maximum time in seconds to run the snapshot creation process on the instance, or to create an AMI from the instance. Increasing this timeout allows the command to wait longer for VSS to start its freeze and complete tagging of the resources it creates. This timeout only applies to the snapshot or AMI creation steps. The initial step to install or update the `AwsVssComponents` package is not included in the timeout.

CollectDiagnosticLogs (string, optional)

To collect more information during snapshot and AMI creation steps, set this parameter to `True`. The default value for this parameter is `False`.

VssVersion (string, optional)

For the `AWSEC2-VssInstallAndSnapshot` document only, you can specify the `VssVersion` parameter to install a specific version of `AwsVssComponents` package on your instance. Leave this parameter blank to install the recommended default version.

If the specified version of the `AwsVssComponents` package is already installed, the script skips the install step and moves on to the backup step. For a list of `AwsVssComponents` package versions and operating support, see AWS VSS solution version history.

Run Systems Manager VSS snapshot command documents

You can create VSS-enabled EBS snapshots with AWS Systems Manager command documents as follows.

Run the AWSEC2-VssInstallAndSnapshot command document (recommended)

When you use AWS Systems Manager to run the `AWSEC2-VssInstallAndSnapshot` document, the script runs the following steps.

1. The script first installs or updates the `AwsVssComponents` package on your instance, depending on whether it's already installed.
2. The script creates the application-consistent snapshots after the first step completes.

To run the `AWSEC2-VssInstallAndSnapshot` document, follow the steps for your preferred environment.

Console

Create VSS-enabled EBS snapshots from the console

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. Select **Run Command** from the navigation pane. This shows a list of commands that are currently running in your account, if applicable.
3. Choose **Run command**. This opens a list of command documents that you have access to.
4. Select `AWSEC2-VssInstallAndSnapshot` from the list of command documents. To streamline results, you can enter all or part of the document name. You can also filter by the owner, by platform types, or by tags.

   When you select a command document, details populate below the list.
5. Select `Default version at runtime` from the **Document version** list.
6. Configure the **Command parameters** to define how `AWSEC2-VssInstallAndSnapshot` will install the `AwsVssComponents` package and back up with VSS snapshots or an AMI. For parameter details, see Parameters for Systems Manager VSS snapshot documents.
7. For **Target selection**, specify tags or select instances manually to identify the instances on which to run this operation.

   > **Note**
   >
   > If you select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? for troubleshooting tips.
8. For additional parameters to define Systems Manager Run Command behavior such as **Rate control**, enter values as described in Running commands from the console.
9. Choose **Run**.

   If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the Systems Manager command output for details about why the execution failed.

If the command successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS volumes.

AWS CLI

You can run the following commands in the AWS CLI to create VSS-enabled EBS snapshots and get the status of your snapshot creation.

Create VSS-enabled EBS snapshots

Run the following command to create VSS-enabled EBS snapshots. To create the snapshots, you must identify the instances with the `--instance-ids` parameter. For more information about other parameters that you can use, see Parameters for Systems Manager VSS snapshot documents.

```
aws ssm send-command \
 --document-name "AWSEC2-VssInstallAndSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"],"VssVersion":[""]}'
```

If successful, the command document populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the command output for details about why the execution failed.

Get command status

To get the current status of the snapshots, run the following command using the command ID returned from **send-command**.

```
aws ssm get-command-invocation
 --instance-ids "i-01234567890abcdef" \
 --command-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
 --plugin-name "CreateVssSnapshot"
```

PowerShell

Run the following commands with AWS Tools for Windows PowerShell to create VSS-enabled EBS snapshots and get the current runtime status for the creation of your output. Specify parameters described in the prior list to modify the behavior of the snapshot process.

Create VSS-enabled EBS snapshots with Tools for Windows PowerShell

Run the following command to create VSS-enabled EBS snapshots or AMIs.

```
Send-SSMCommand -DocumentName "AWSEC2-VssInstallAndSnapshot" -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value';'VssVersion'=''}
```

Get command status

To get the current status of the snapshots, run the following command using the command ID returned from **Send-SSMCommand**.

```
Get-SSMCommandInvocationDetail -InstanceId "i-01234567890abcdef" -CommandId
 "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" -PluginName "CreateVssSnapshot"
```

If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the command output for details about why the execution failed.

Run the AWSEC2-CreateVssSnapshot command document

To run the `AWSEC2-CreateVssSnapshot` document, follow the steps for your preferred environment.

Console

Create VSS-enabled EBS snapshots from the console

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. Select **Run Command** from the navigation pane. This shows a list of commands that are currently running in your account, if applicable.
3. Choose **Run command**. This opens a list of command documents that you have access to.
4. Select `AWSEC2-CreateVssSnapshot` from the list of command documents. To streamline results, you can enter all or part of the document name. You can also filter by the owner, by platform types, or by tags.

When you select a command document, details populate below the list.

5. Select `Default version at runtime` from the **Document version** list.
6. Configure the **Command parameters** to define how `AWSEC2-CreateVssSnapshot` will back up with VSS snapshots or an AMI. For parameter details, see Parameters for Systems Manager VSS snapshot documents.
7. For **Target selection**, specify tags or select instances manually to identify the instances on which to run this operation.

   > **Note**
   >
   > If you select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? for troubleshooting tips.

8. For additional parameters to define Systems Manager Run Command behavior such as **Rate control**, enter values as described in Running commands from the console.
9. Choose **Run**.

   If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the Systems Manager command output for details about why the execution failed. If the command successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS volumes.

AWS CLI

You can run the following command in the AWS CLI to create VSS-enabled EBS snapshots.

Create VSS-enabled EBS snapshots

Run the following command to create VSS-enabled EBS snapshots. To create the snapshots, you must identify the instances with the `--instance-ids` parameter. For more information about other parameters that you can use, see Parameters for Systems Manager VSS snapshot documents.

```
aws ssm send-command \
 --document-name "AWSEC2-CreateVssSnapshot" \
```

```
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"]}'
```

If successful, the command document populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the command output for details about why the execution failed.

PowerShell

Run the following command with AWS Tools for Windows PowerShell to create VSS-enabled EBS snapshots.

Create VSS-enabled EBS snapshots with Tools for Windows PowerShell

Run the following command to create VSS-enabled EBS snapshots. To create the snapshots, you must identify the instances with the `InstanceId` parameter. You can specify more than one instance to create snapshots for. For more information about other parameters that you can use, see Parameters for Systems Manager VSS snapshot documents.

```
Send-SSMCommand -DocumentName AWSEC2-CreateVssSnapshot -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value'}
```

If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the command output for details about why the execution failed. If the command successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS snapshots.

Run command documents for a Windows Failover Cluster with shared EBS storage

You can use any of the command line procedures described in the previous section to create a VSS-enabled snapshot. The command document (`AWSEC2-VssInstallAndSnapshot` or `AWSEC2-CreateVssSnapshot`) must run on the primary node in your cluster. The document will fail on the secondary nodes as they don't have access to the shared disks. If your primary and secondary change dynamically, you can run the AWS Systems Manager Run Command document on multiple

nodes with the expectation that the command will succeed on the primary node and fail on secondary nodes.

Run the AWSEC2-ManageVssIO SSM command document

You can use the following script and the pre-defined `AWSEC2-ManageVssIO` SSM document to temporarily pause I/O, create VSS-enabled EBS snapshots, and restart I/O. This process runs in the context of the user who runs the command. If the user has sufficient permission to create and tag snapshots, then AWS Systems Manager can create and tag VSS-enabled EBS snapshots without the need for the additional IAM snapshot role on the instance.

In contrast, the command document (`AWSEC2-VssInstallAndSnapshot` or `AWSEC2-CreateVssSnapshot`) requires that you assign the IAM snapshot role to each instance for which you want to create EBS snapshots. If you don't want to provide additional IAM permissions to your instances for policy or compliance reasons, then you can use the following script.

Before you begin

Note the following important details about this process:

- This process uses a PowerShell script (`CreateVssSnapshotAdvancedScript.ps1`) to take snapshots of all volumes on the instances you specify, except root volumes. If you need to take snapshots of root volumes, then you must use the `AWSEC2-CreateVssSnapshot` SSM document.
- The script calls the `AWSEC2-ManageVssIO` document twice. The first time with the `Action` parameter set to `Freeze`, which pauses all I/O on the instances. The second time, the `Action` parameter is set to `Thaw`, which forces I/O to resume.
- Don't attempt to use the `AWSEC2-ManageVssIO` document without using the CreateVssSnapshotAdvancedScript.ps1 script. Microsoft's VSS framework requires that the `Freeze` and `Thaw` actions be called no more than ten seconds apart, and manually calling these actions without the script could result in errors.

To create VSS-enabled EBS snapshots by using the `AWSEC2-ManageVssIO` SSM document

1. Download the CreateVssSnapshotAdvancedScript.zip file and extract the file contents.
2. Open `CreateVssSnapshotAdvancedScript.ps1` in a text editor, edit the sample call at the bottom of the script with a valid EC2 instance ID, snapshot description, and desired tag values, and then run the script from PowerShell.

If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the command output for details about why the execution failed. If the command was successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS volumes.

> **Note**
>
> To automate backups, you can create an AWS Systems Manager maintenance window task that uses the `AWSEC2-VssInstallAndSnapshot` document. For more information, see Working with Maintenance Windows (Console) in the *AWS Systems Manager User Guide*.

Create VSS snapshots using AWS Backup

You can create a VSS backup when using AWS Backup by enabling VSS in the console or CLI. Ensure that you have met the prerequisites before you create the VSS-enabled backup plan. For more information, see Creating Windows VSS backups in the *AWS Backup Developer Guide*.

> **Note**
>
> AWS Backup doesn't automatically install the `AwsVssComponents` package on your instance. You must perform a manual install on your instance. For more information, see Manually install VSS components on an instance.

Create VSS snapshots using Amazon Data Lifecycle Manager

You can create VSS snapshots using Amazon Data Lifecycle Manager by enabling pre and post scripts in your snapshot lifecycle policies. For more information, see Automating application-consistent snapshots with pre and post scripts.

> **Note**
>
> Amazon Data Lifecycle Manager doesn't automatically install the `AwsVssComponents` package on your instance. You must perform a manual install on your instance. For more information, see Manually install VSS components on an instance.

Troubleshooting

Before you try any other troubleshooting steps, we recommend that you verify the following details.

- Ensure that you've met all Prerequisites.
- Verify that you're using the latest Windows OS version support of the `AwsVssComponents` package for your operating system. The issue that you've observed might have been addressed in newer versions.

Topics

- General: Check the log files
- General: Using VSS on instances with proxy configured
- Error: Thaw pipe connection timed out, error on thaw, timeout waiting for VSS Freeze, or other timeout errors
- Error: Cannot invoke method. Method invocation is supported only on core types in this language mode

General: Check the log files

If you experience problems or receive error messages when creating VSS-enabled EBS snapshots, you can view the command output in the Systems Manager console. You can also view the following logs:

- `%ProgramData%\Amazon\SSM\InstanceData\`*`InstanceID`*`\document\orchestration\`*`SSMCommandID`*`\awsrunPowerShellScript\runPowerShellScript\stdout`
- `%ProgramData%\Amazon\SSM\InstanceData\`*`InstanceID`*`\document\orchestration\`*`SSMCommandID`*`\awsrunPowerShellScript\runPowerShellScript\stderr`

You can also open the Event Viewer Windows application and choose **Windows Logs**, **Application** to view additional logs. To see events specifically from the EC2 Windows VSS Provider and the Volume Shadow Copy Service, filter by **Source** on the terms **`Ec2VssSoftwareProvider`** and **`VSS`**.

If you're using Systems Manager with VPC endpoints, and the Systems Manager Run Command failed, verify that you configured the following endpoint correctly: **com.amazonaws.*`region`*.ec2**.

Without the EC2 endpoint defined, the call to enumerate attached EBS volumes fails, which causes the Systems Manager command to fail. For more information about setting up VPC endpoints with Systems Manager, see Create a Virtual Private Cloud Endpoint in the *AWS Systems Manager User Guide*.

General: Using VSS on instances with proxy configured

If you experience problems when creating VSS-enabled EBS snapshots on instances that use a proxy to reach EC2 endpoints, make sure of the following:

- The proxy is configured so that the EC2 service endpoints in the instance's Region and IMDS are reachable by AWS Tools for Windows PowerShell running as SYSTEM.
- `AwsVssComponents` version 2.0.1 or later is installed. Starting with `AwsVssComponents` version 2.0.1, the EC2 VSS provider supports using the system's configured WinHTTP proxy. For more information about configuring WinHTTP proxy, see Netsh Commands for Windows Hypertext Transfer Protocol (WINHTTP) on the Microsoft website.

Error: Thaw pipe connection timed out, error on thaw, timeout waiting for VSS Freeze, or other timeout errors

The EC2 Windows VSS Provider might time out due to activity or services on the instance preventing VSS-enabled snapshots from proceeding in a timely manner. The Windows VSS Framework provides a non-configurable 10-second window during which communication to the file system is paused. During this time, `AWSEC2-CreateVssSnapshot` snapshots your volumes.

The following issues can cause the EC2 Windows VSS Provider to run into time limits during a snapshot:

- Excessive I/O to a volume
- Slow responsiveness of the EC2 API on the instance
- Fragmented volumes
- Incompatibility with some antivirus software
- Issues with a VSS Application writer
- When Module Logging is enabled for a large number of PowerShell modules, that can cause PowerShell scripts to run slowly

Most timeout issues that occur when you run the `AWSEC2-CreateVssSnapshot` command document are related to the workload on the instance being too high at the time of the backup. The following actions can help you take a successful snapshot:

- Retry the `AWSEC2-CreateVssSnapshot` command to see if the snapshot attempt is successful. If retrying succeeds in some cases, reducing the instance load might make snapshots more successful.
- Wait a while for the workload on the instance to decrease, and retry the `AWSEC2-CreateVssSnapshot` command. Alternatively, you can attempt snapshots when the instance is known to be under low stress.
- Attempt VSS snapshots when the antivirus software on the system is turned off. If this resolves the issue, refer to the antivirus software instructions and configure it to allow VSS snapshots.
- If there is a high volume of Amazon EC2 API calls in your account within the same Region where you're running a snapshot, API throttling might delay snapshot operations. To reduce throttling impact, use the latest `AwsVssComponents` package (version 2.1.0 and above, with the prerequisite permissions). This package utilizes the EC2 `CreateSnapshots` API action to reduce the number of mutating actions like per-volume snapshot creation and tagging.
- If you have multiple `AWSEC2-CreateVssSnapshot` command scripts running at the same time, you can take the following steps to reduce concurrency issues.
  - Consider scheduling snapshots during periods of lower API activity.
  - If you use **Run Command** in the Systems Manager console (or **SendCommand** in the API) to run the command script, you can use Systems Manager rate controls to reduce concurrency.

    You can also use Systems Manager rate controls to reduce concurrency for services like AWS Backup that use Systems Manager to run the command script.
- Run the command `vssadmin list writers` in a shell and see if it reports any errors in the **Last error** field for any writers on the system. If any writers report a **time out** error, consider retrying snapshots when the instance is under less load.
- When you use smaller instance types like *`t2 | t3 | t3a`*.nano or *`t2 | t3 | t3a`*.micro, timeouts due to memory and CPU constraints can occur. The following actions might help reduce timeout issues.
  - Try closing memory or CPU intensive applications before taking snapshots.
  - Try taking snapshots during periods of lower instance activity.

Error: Cannot invoke method. Method invocation is supported only on core types in this language mode

You will encounter this error when the PowerShell language mode is not set to `FullLanguage`. The `AWSEC2-CreateVssSnapshot` and `AWSEC2-ManageVssIo` SSM documents require PowerShell to be configured to `FullLanguage` mode.

To verify the language mode, run the following command on the instance in a PowerShell console:

```
$ExecutionContext.SessionState.LanguageMode
```

For more information about language modes, see about_Language_Modes in the Microsoft documentation.

Restore EBS volumes from VSS-enabled EBS snapshots

You can use the `RestoreVssSnapshotSampleScript.ps1` script to restore volumes on an instance from VSS-enabled EBS snapshots. This script performs the following tasks:

- Stops an instance
- Removes all existing drives from the instance (except the boot volume, if it was excluded)
- Creates new volumes from the snapshots
- Attaches the volumes to the instance by using the device ID tag on the snapshot
- Restarts the instance

> **Important**
>
> The following script detaches all volumes attached to an instance, and then creates new volumes from a snapshot. Make sure that you have properly backed-up the instance. The old volumes are not deleted. If you want, you can edit the script to delete the old volumes.

To restore volumes from VSS-enabled EBS snapshots

1. Download the RestoreVssSnapshotSampleScript.zip file and extract the file contents.

2. Open `RestoreVssSnapshotSampleScript.ps1` in a text editor and edit the sample call at the bottom of the script with a valid EC2 instance ID and EBS snapshot ID, and then run the script from PowerShell.

AWS VSS solution version history

Topics

- AwsVssComponents package versions
- Windows OS version support

AwsVssComponents package versions

The following table describes the released versions of the AWS VSS component package.

Version	Details	Release date
2.3.1	Added new default tag `AwsVssConfig` to identify snapshots and AMIs created by AWS VSS.	March 7, 2024
2.2.1	• Added support for using the `DescribeInstanceAttribute` API. • Bug fixes and reliability improvements. • Deprecated support for Windows Server 2012 and 2012 R2. AWS VSS components version 2.2.1 installation on Windows Server 2012 and 2012 R2 will fail. AWS VSS components version 2.1.0 is the last version to support Windows Server 2012 and 2012 R2.	January 18, 2024
2.1.0	Added support for using the `CreateSnapshots` API.	November 6, 2023
2.0.1	Added support for using the WinHTTP proxy settings.	October 26, 2023

Version	Details	Release date
2.0.0	Added capability to the AWS VSS component to create snapshots and AMIs, which enables compatibility with PowerShell module logging, script block logging, and transcription features.	April 28, 2023
1.3.2.0	Fixed a case where installation failure is not reported correctly.	May 10, 2022
1.3.1.0	• Fixed snapshots failing on domain controllers in relation to an NTDS VSS writer logging error. • Fixed VSS agent error when uninstalling version 1.0 VSS provider.	February 6, 2020
1.3.00	• Improved logging by reducing unwanted verbosity. • Fixed regionalization issues during installation. • Fixed return codes for some provider registration error conditions. • Fixed various installation issues.	March 19, 2019
1.2.00	• Added command line parameters `-nw` (no-writers) and `-copy` (copy-only) to agent. • Fixed EventLog errors caused by improper memory allocation calls.	November 15, 2018
1.1	Fixed AWS VSS components that were being used incorrectly as the default Windows Backup and Restore provider.	December 12, 2017

Version	Details	Release date
1.0	Initial release.	November 20, 2017

Windows OS version support

The following table shows which AWS VSS solution versions you should run on each version of Windows Server on Amazon EC2.

Windows Server version	AwsVssComponents version	AWSEC2-VssInstallAndSnapshot version name	AWSEC2-CreateVssSnapshot version name	AWSEC2-ManageVssIO version name
Windows Server 2022	default	default	default	default
Windows Server 2019	default	default	default	default
Windows Server 2016	default	default	default	default
Windows Server 2012 R2	2.1.0	not supported	2012R2	2012R2
Windows Server 2012	2.1.0	not supported	2012R2	2012R2

Windows Server version	AwsVssComponents version	AWSEC2-VssInstallAndSnapshot version name	AWSEC2-CreateVssSnapshot version name	AWSEC2-ManageVssIO version name
Windows Server 2008 R2	1.3.1.0	not supported	2008R2	2008R2

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-generate 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.

```
C:\> 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.

```
C:\> 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.

```
C:\> 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.

```
C:\> 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.

```
C:\> aws rbin list-rules --resource-type EBS_SNAPSHOT|EC2_IMAGE
```

Example

The following example command provides lists all retention rules that retain snapshots.

```
C:\> aws rbin list-rules --resource-type EBS_SNAPSHOT
```

To view information for a specific retention rule

Use the get-rule AWS CLI command.

```
C:\> aws rbin get-rule --identifier rule_ID
```

Example

The following example command provides information about retention rule `pwxIkFcvge4`.

```
C:\> 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.

```
C:\> 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.

```
C:\> 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.

```
C:\> aws rbin lock-rule \
--identifier rule_ID \
--lock-configuration
 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=number_of_days}'
```

Example

The following example command locks retention rule `6lsJ2Fa9nh9` and sets the unlock delay period to 15 days.

```
C:\> aws rbin lock-rule \
--identifier 6lsJ2Fa9nh9 \
--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.

```
C:\> aws rbin unlock-rule \
--identifier rule_ID
```

Example

The following example command unlocks retention rule `6lsJ2Fa9nh9`

```
C:\> aws rbin unlock-rule \
--identifier 6lsJ2Fa9nh9
```

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.

```
C:\> 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/nOoSBBtItF3` with tag `purpose=production`.

```
C:\> aws rbin tag-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 \
--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.

```
C:\> 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/nOoSBBtItF3`.

```
C:\> aws rbin list-tags-for-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3
```

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.

```
C:\> 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/nOoSBBtItF3`.

```
C:\> aws rbin untag-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 \
--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.

```
C:\> aws rbin delete-rule --identifier rule_ID
```

Example

The following example command deletes retention rule `6lsJ2Fa9nh9`.

```
C:\> aws rbin delete-rule --identifier 6lsJ2Fa9nh9
```

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.

```
C:\> 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.

```
C:\> 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 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.

```
C:\> aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snapshot_id
```

For example, the following command restores snapshot `snap-01234567890abcdef` from the Recycle Bin.

```
C:\> 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*.

```
C:\> 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.

```
C:\> 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.

```
C:\> aws ec2 restore-image-from-recycle-bin --image-id ami_id
```

For example, the following command restores AMI `ami-01234567890abcdef` from the Recycle Bin.

```
C:\> 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 botocore/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 botocore/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 botocore/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 botocore/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 botocore/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 botocore/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 informati on, 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	The key pairs that you create using Amazon EC2 are tied to the Region where you created them. You can create your own RSA key pair and upload it to the Region in which you want to use it; therefore, you can make your key pair globally available by uploading it to each Region. For more information, see Amazon EC2 key pairs and Amazon EC2 instances.
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

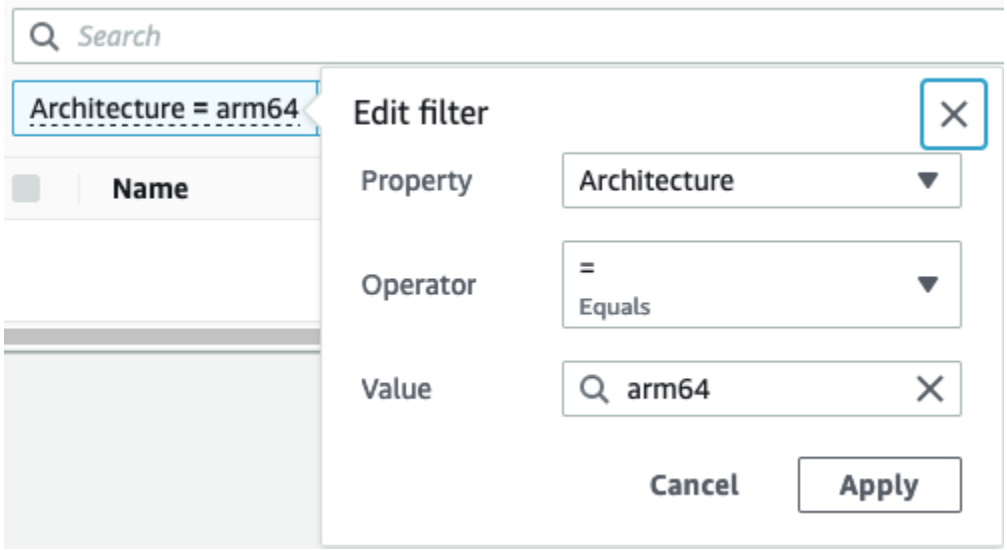
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose the option that corresponds to the resource type. For example, to list your instances, choose **Instances**.

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.



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

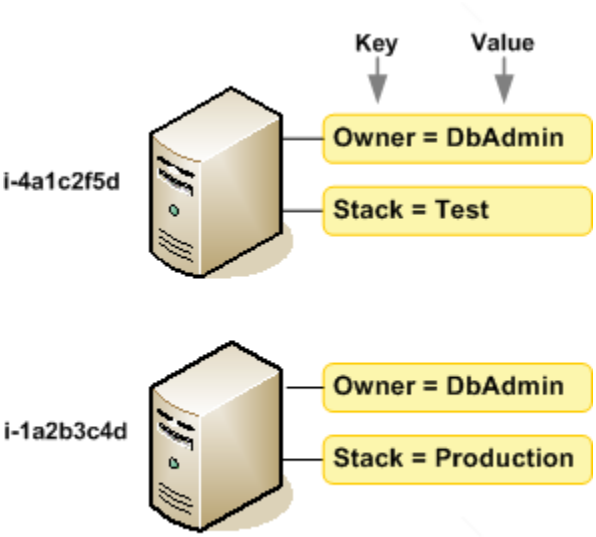
- Tags and access management
- Tag your resources for billing
- Work with tags using the console
- Work with tags using the command line
- Work with instance tags in instance metadata
- Add tags to a resource using CloudFormation

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 of 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 Windows 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 \
```

```
    --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.

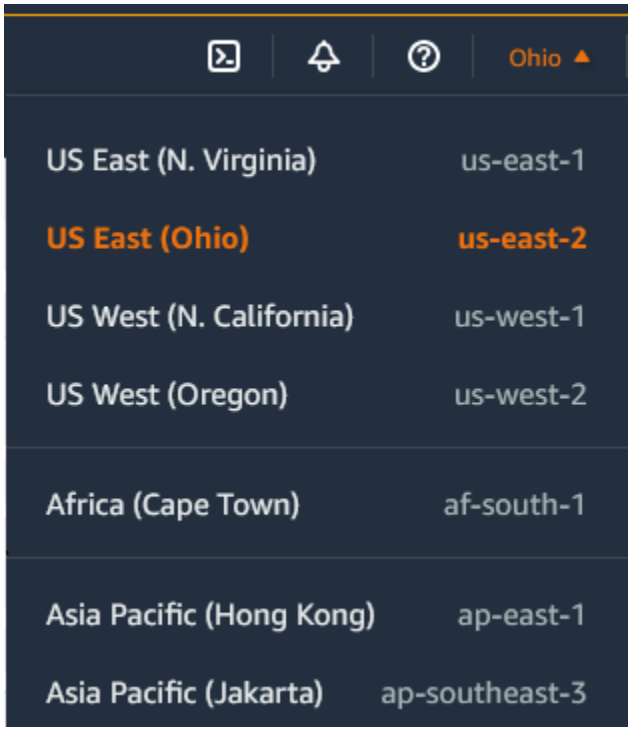
For more information, see Amazon EC2 endpoints and quotas and Amazon EBS endpoints and quotas in the *Amazon Web Services General Reference*.

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.



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*.

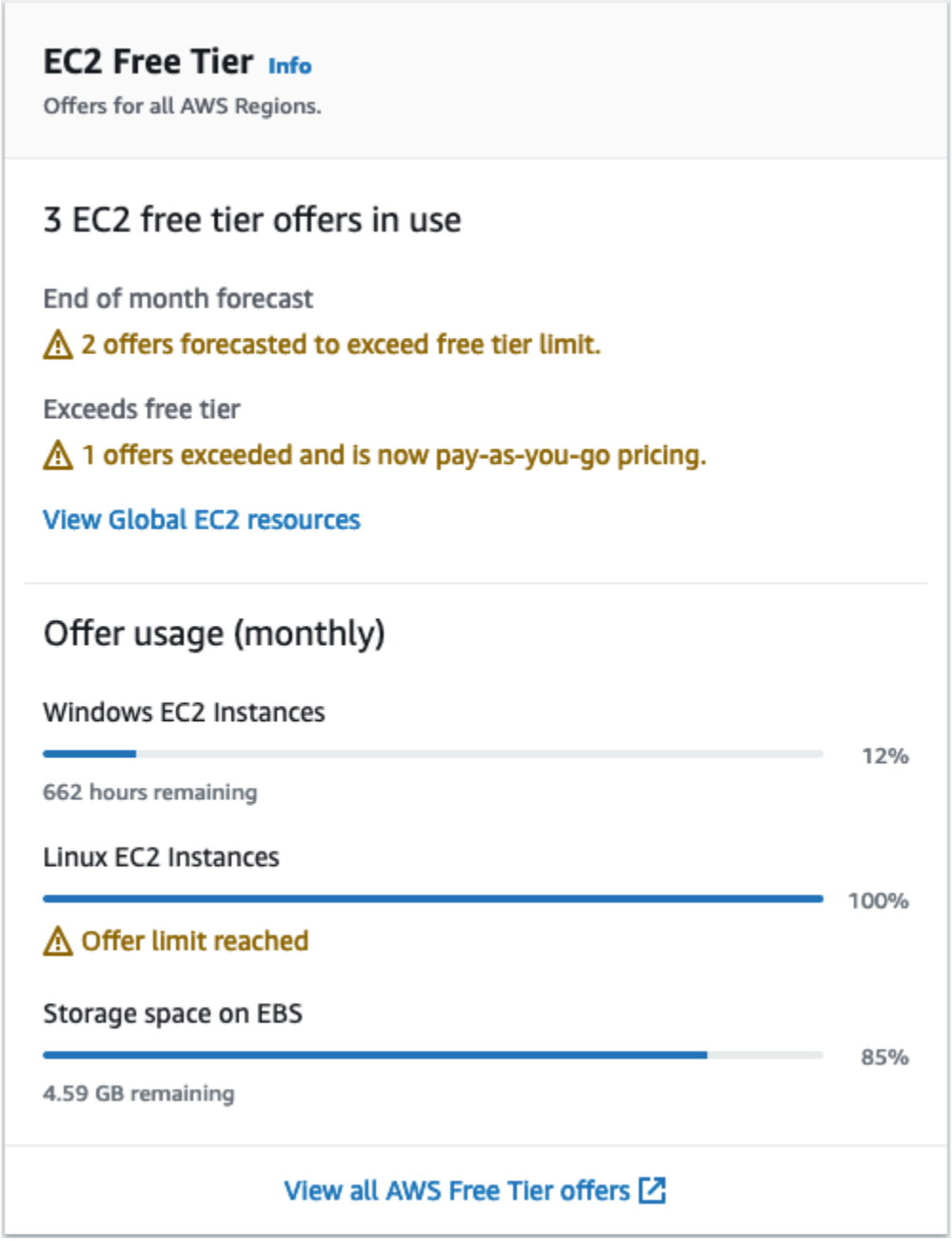
Track your Free Tier usage

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).



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 Windows instances

The following procedures and tips can help you troubleshoot problems with your Amazon EC2 Windows instances.

Contents

- Common issues with Windows instances
- Common messages troubleshooting Windows instances
- Troubleshoot instance launch issues
- Troubleshoot connecting to your Windows instance
- Troubleshoot an unreachable instance
- Reset a lost or expired Windows administrator password
- Troubleshoot stopping your instance
- Troubleshoot instance termination (shutting down)
- Troubleshoot Sysprep
- Use EC2Rescue for Windows Server
- EC2 Serial Console for Windows instances
- Send a diagnostic interrupt (for advanced users)

To get additional information for troubleshooting problems with your instance, use Use EC2Rescue for Windows Server. For information about troubleshooting issues with PV drivers, see Troubleshoot PV drivers.

Common issues with Windows instances

The following are troubleshooting tips to help you solve common issues with EC2 Windows Server instances.

Issues

- EBS volumes don't initialize on Windows Server 2016 and 2019
- Boot an EC2 Windows instance into Directory Services Restore Mode (DSRM)
- Instance loses network connectivity or scheduled tasks don't run when expected

- Unable to get console output
- Windows Server 2012 R2 not available on the network
- Disk signature collision

EBS volumes don't initialize on Windows Server 2016 and 2019

Instances created from Amazon Machine Images (AMIs) for Windows Server 2016 and 2019 use the EC2Launch v1 agent for a variety of startup tasks, including initializing EBS volumes. By default, EC2Launch v1 doesn't initialize secondary volumes. However, you can configure EC2Launch v1 to initialize these disks automatically, as follows.

Map drive letters to volumes

1. Connect to the instance to configure and open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` file in a text editor.
2. Specify the volume settings as follows:

```
{
"driveLetterMapping": [
 {
   "volumeName": "sample volume",
   "driveLetter": "H"
 }]
}
```

3. Save your changes and close the file.
4. Open Windows PowerShell and use the following command to run the EC2Launch v1 script that initializes the disks:

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

To initialize the disks each time the instance boots, add the `-Schedule` flag as follows:

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -
Schedule
```

The EC2Launch v1 agent can run instance initialization scripts such as `initializeDisks.ps1` in parallel with the `InitializeInstance.ps1` script. If the

`InitializeInstance.ps1` script reboots the instance, it might interrupt other scheduled tasks that run at instance startup. To avoid any potential conflicts, we recommend that you add logic to your `initializeDisks.ps1` script to ensure that instance initialization has finished first.

> **Note**
>
> If the EC2Launch script does not initialize the volumes, ensure that the volumes are online. If the volumes are offline, run the following command to bring all disks online.
>
> ```
> PS C:\> Get-Disk | Where-Object IsOffline -Eq $True | Set-Disk -IsOffline
> $False
> ```

Boot an EC2 Windows instance into Directory Services Restore Mode (DSRM)

If an instance running Microsoft Active Directory experiences a system failure or other critical issues you can troubleshoot the instance by booting into a special version of Safe Mode called *Directory Services Restore Mode* (DSRM). In DSRM you can repair or recover Active Directory.

Driver support for DSRM

How you enable DSRM and boot into the instance depends on the drivers the instance is running. In the EC2 console you can view driver version details for an instance from the System Log. The following table shows which drivers are supported for DSRM.

Driver Versions	DSRM Supported?	Next Steps
Citrix PV 5.9	No	Restore the instance from a backup. You cannot enable DSRM.
AWS PV 7.2.0	No	Though DSRM is not supported for this driver, you can still detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability

Driver Versions	DSRM Supported?	Next Steps
		Zone as a secondary volume. You can then enable DSRM (as described in this section).
AWS PV 7.2.2 and later	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
Enhanced Networking	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).

For information about how to enable Enhanced Networking, see Enabling Enhanced Networking on Windows Instances in a VPC. For more information about upgrading AWS PV drivers, see Upgrade PV drivers on Windows instances.

Configure an instance to boot into DSRM

EC2 Windows instances do not have network connectivity before the operating system is running. For this reason, you cannot press the F8 button on your keyboard to select a boot option. You must use one of the following procedures to boot an EC2 Windows Server instance into DSRM.

If you suspect that Active Directory has been corrupted and the instance is still running, you can configure the instance to boot into DSRM using either the System Configuration dialog box or the command prompt.

To boot an online instance into DSRM using the System Configuration dialog box

1. In the **Run** dialog box, type `msconfig` and press Enter.
2. Choose the **Boot** tab.
3. Under **Boot options** choose **Safe boot**.
4. Choose **Active Directory repair** and then choose **OK**. The system prompts you to reboot the server.

To boot an online instance into DSRM using the command line

From a Command Prompt window, run the following command:

```
bcdedit /set safeboot dsrepair
```

If an instance is offline and unreachable, you must detach the root volume and attach it to another instance to enable DSRM mode.

To boot an offline instance into DSRM

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate and select the affected instance. Choose **Instance state**, **Stop instance**.
4. Choose **Launch instances** and create a temporary instance in the same Availability Zone as the affected instance. Choose an instance type that uses a different version of Windows. For example, if your instance is Windows Server 2008, then choose a Windows Server 2008 R2 instance.

   > **Important**
   >
   > If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. Detach the volume and attach it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to make the volume available for use.
8. Open a command prompt and run the following command. Replace *D* with the actual drive letter of the secondary volume you just attached:

   ```
   bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
   ```

9. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
10. In the EC2 console, detach the affected volume from the temporary instance and reattach it to your original instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.

11. Start the instance.
12. After the instance passes the health checks in the EC2 console, connect to the instance using Remote Desktop and verify that it boots into DSRM mode.
13. (Optional) Delete or stop the temporary instance you created in this procedure.

Instance loses network connectivity or scheduled tasks don't run when expected

If you restart your instance and it loses network connectivity, it's possible that the instance has the wrong time.

By default, Windows instances use Coordinated Universal Time (UTC). If you set the time for your instance to a different time zone and then restart it, the time becomes offset and the instance temporarily loses its IP address. The instance regains network connectivity eventually, but this can take several hours. The amount of time that it takes for the instance to regain network connectivity depends on the difference between UTC and the other time zone.

This same time issue can also result in scheduled tasks not running when you expect them to. In this case, the scheduled tasks do not run when expected because the instance has the incorrect time.

To use a time zone other than UTC persistently, you must set the **RealTimeIsUniversal** registry key. Without this key, an instance uses UTC after you restart it.

To resolve time issues that cause a loss of network connectivity

1. Ensure that you are running the recommended PV drivers. For more information, see Upgrade PV drivers on Windows instances.
2. Verify that the following registry key exists and is set to 1: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**

Unable to get console output

For Windows instances, the instance console displays the output from tasks performed during the Windows boot process. If Windows boots successfully, the last message logged is `Windows is Ready to use`. Note that you can also display event log messages in the console, but this feature is not enabled by default. For more information, see EC2 service properties.

To get the console output for your instance using the Amazon EC2 console, select the instance, and then choose **Actions**, **Monitor and troubleshoot**, **Get system log**. To get the console output using the command line, use one of the following commands: get-console-output (AWS CLI) or Get-EC2ConsoleOutput (AWS Tools for Windows PowerShell).

For instances running Windows Server 2012 R2 and earlier, if the console output is empty, it could indicate an issue with the EC2Config service, such as a misconfigured configuration file, or that Windows failed to boot properly. To fix the issue, download and install the latest version of EC2Config. For more information, see Install the latest version of EC2Config.

Windows Server 2012 R2 not available on the network

For information about troubleshooting a Windows Server 2012 R2 instance that is not available on the network, see Windows Server 2012 R2 loses network and storage connectivity after an instance reboot.

Disk signature collision

You can check for and resolve disk signature collisions using EC2Rescue for Windows Server. Or, you can manually resolve disk signature issues by performing the following steps.

> **Warning**
>
> The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see Configure the Registry.

1. Open a command prompt, type **regedit.exe**, and press Enter.
2. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
3. Type **Windows Boot Manager** and then choose **Find Next**.
4. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.
5. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).

6. Locate the four-byte disk signature at offset 0x38 in the data. This is the Boot Configuration Database signature (BCD). Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is E9EB3AA5:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

7. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

8. Run the `select disk` DiskPart command and specify the disk number for the volume with the disk signature collision.

> **Tip**
>
> To check the disk number for the volume with the disk signature collision, use the **Disk Management** utility. Open a command prompt, type `compmgmt.msc` and press **Enter**. In the left-hand navigation panel, double-click **Disk Management**. In the **Disk Management** utility, check the disk number for the offline volume with the disk signature collision.

```
DISKPART> select disk 1
Disk 1 is now the selected disk.
```

9. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk
Disk ID: 0C764FA8
```

10. If the disk signature shown in the previous step doesn't match the disk signature that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

Common messages troubleshooting Windows instances

This section includes tips to help you troubleshoot issues based on common messages.

Topics

- "Password is not available"
- "Password not available yet"
- "Cannot retrieve Windows password"
- "Waiting for the metadata service"
- "Unable to activate Windows"
- "Windows is not genuine (0x80070005)"
- "No Terminal Server License Servers available to provide a license"
- "Some settings are managed by your organization"

"Password is not available"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

You can generate a password for the Administrator account for instances launched using a custom Windows AMI. To generate the password, you will need to configure some settings in the operating system before the AMI is created. For more information, see Create a custom Windows AMI.

If your Windows instance isn't configured to generate a random password, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password is not available.
The instance was launched from a custom AMI, or the default password has changed. A
password cannot be retrieved for this instance. If you have forgotten your password,
 you can
reset it using the Amazon EC2 configuration service. For more information, see
 Passwords for a
Windows Server instance.
```

Check the console output for the instance to see whether the AMI that you used to launch it was created with password generation disabled. If password generation is disabled, the console output contains the following:

```
Ec2SetPassword: Disabled
```

If password generation is disabled and you don't remember the password for the original instance, you can reset the password for this instance. For more information, see Reset a lost or expired Windows administrator password.

"Password not available yet"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

Your password should be available within a few minutes. If the password isn't available, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password not available yet.
Please wait at least 4 minutes after launching an instance before trying to retrieve
 the
auto-generated password.
```

If it's been longer than four minutes and you still can't get the password, it's possible that the launch agent for your instance is not configured to generate a password. Verify by checking whether the console output is empty. For more information, see Unable to get console output.

Also verify that the AWS Identity and Access Management (IAM) account being used to access the Management Portal has the `ec2:GetPasswordData` action allowed. For more information about IAM permissions, see What is IAM?.

"Cannot retrieve Windows password"

To retrieve the auto-generated password for the Administrator account, you must use the private key for the key pair that you specified when you launched the instance. If you didn't specify a key pair when you launched the instance, you'll receive the following message.

```
Cannot retrieve Windows password
```

You can terminate this instance and launch a new instance using the same AMI, making sure to specify a key pair.

"Waiting for the metadata service"

A Windows instance must obtain information from its instance metadata before it can activate itself. By default, the `WaitForMetaDataAvailable` setting ensures that the EC2Config service waits for the instance metadata to be accessible before continuing with the boot process. For more information, see Instance metadata and user data.

If the instance is failing the instance reachability test, try the following to resolve this issue.

- Check the CIDR block for your VPC. A Windows instance cannot boot correctly if it's launched into a VPC that has an IP address range from `224.0.0.0` to `255.255.255.255` (Class D and Class E IP address ranges). These IP address ranges are reserved, and should not be assigned to host devices. We recommend that you create a VPC with a CIDR block from the private (non-publicly routable) IP address ranges as specified in RFC 1918.
- It's possible that the system has been configured with a static IP address. Try creating a network interface and attaching it to the instance.
- **To enable DHCP on a Windows instance that you can't connect to**

  1. Stop the affected instance and detach its root volume.
  2. Launch a temporary instance in the same Availability Zone as the affected instance.

     > **Warning**
     >
     > If your temporary instance is based on the same AMI that the original instance is based on, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012.

  3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.

4. From the temporary instance, open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key that you just loaded and navigate to `ControlSet001\Services\Tcpip\Parameters\Interfaces`. Each network interface is listed by a GUID. Select the correct network interface. If DHCP is disabled and a static IP address assigned, `EnableDHCP` is set to 0. To enable DHCP, set `EnableDHCP` to 1, and delete the following keys if they exist: `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. Select the key again, and from the **File** menu, choose **Unload Hive**.

   > **Note**
   >
   > If you have multiple network interfaces, you'll need to identify the correct interface to enable DHCP. To identify the correct network interface, review the following key values `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. These values display the static configuration of the previous instance.

6. (Optional) If DHCP is already enabled, it's possible that you don't have a route to the metadata service. Updating EC2Config can resolve this issue.

   a. Download and install the latest version of the EC2Config service. For more information about installing this service, see Install the latest version of EC2Config.

   b. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.

   c. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).

   d. Select the key that you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe -q` as the data.

   e. Select the key again, and from the **File** menu, choose **Unload Hive**.

7. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

> **Warning**
>
> The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see Configure the Registry.

a. Open a command prompt, type **regedit.exe**, and press Enter.

b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.

c. Type **Windows Boot Manager** and then choose **Find Next**.

d. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.

e. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).

f. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. Using the **Disk Management** utility, bring the drive offline.

> **Note**
>
> The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

9. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
10. Restore the root volume of the affected instance by attaching the volume as `/dev/sda1`.
11. Start the affected instance.

If you are connected to the instance, open an Internet browser from the instance and enter the following URL for the metadata server:

```
http://169.254.169.254/latest/meta-data/
```

If you can't contact the metadata server, try the following to resolve the issue:

- Download and install the latest version of the EC2Config service. For more information about installing this service, see Install the latest version of EC2Config.
- Check whether the Windows instance is running RedHat PV drivers. If so, update to Citrix PV drivers. For more information, see Upgrade PV drivers on Windows instances.
- Verify that the firewall, IPSec, and proxy settings do not block outgoing traffic to the metadata service (`169.254.169.254`) or the AWS KMS servers (the addresses are specified

in `TargetKMSServer` elements in `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`).

- Verify that you have a route to the metadata service (`169.254.169.254`) using the following command.

```
route print
```

- Check for network issues that might affect the Availability Zone for your instance. Go to http://status.aws.amazon.com/.

"Unable to activate Windows"

Windows instances use Windows AWS KMS activation. You can receive this message: `A problem occurred when Windows tried to activate. Error Code 0xC004F074`, if your instance can't reach the AWS KMS server. Windows must be activated every 180 days. EC2Config attempts to contact the AWS KMS server before the activation period expires to ensure that Windows remains activated.

If you encounter a Windows activation issue, use the following procedure to resolve the issue.

For EC2Config (Windows Server 2012 R2 AMIs and earlier)

1. Download and install the latest version of the EC2Config service. For more information about installing this service, see Install the latest version of EC2Config.
2. Log onto the instance and open the following file: `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`.
3. Locate the **Ec2WindowsActivate** plugin in the `config.xml` file. Change the state to **Enabled** and save your changes.
4. In the Windows Services snap-in, restart the EC2Config service or reboot the instance.

If this does not resolve the activation issue, follow these additional steps.

1. Set the AWS KMS target: **C:\> slmgr.vbs /skms 169.254.169.250:1688**
2. Activate Windows: **C:\> slmgr.vbs /ato**

For EC2Launch (Windows Server 2016 AMIs and later)

1. From a PowerShell prompt with administrative rights, import the EC2Launch module:

```
PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1"
```

2. Call the Add-Routes function to see the list of new routes:

```
PS C:\> Add-Routes
```

3. Call the Set-ActivationSettings function:

```
PS C:\> Set-Activationsettings
```

4. Then, run the following script to activate Windows:

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

For both EC2Config and EC2Launch, if you are still receiving an activation error, verify the following information.

- Verify that you have routes to the AWS KMS servers. Open `C:\Program Files \Amazon\Ec2ConfigService\Settings\ActivationSettings.xml` and locate the `TargetKMSServer` elements. Run the following command and check whether the addresses for these AWS KMS servers are listed.

```
route print
```

- Verify that the AWS KMS client key is set. Run the following command and check the output.

```
C:\Windows\System32\slmgr.vbs /dlv
```

  If the output contains Error: product key not found, the AWS KMS client key isn't set. If the AWS KMS client key isn't set, look up the client key as described in this Microsoft article: AWS KMS Client Setup Keys, and then run the following command to set the AWS KMS client key.

```
C:\Windows\System32\slmgr.vbs /ipk client_key
```

- Verify that the system has the correct time and time zone. If you are using Windows Server 2008 or later and a time zone other than UTC, add the following registry key and set it to 1 to ensure that the time is correct: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal`.
- If Windows Firewall is enabled, temporarily disable it using the following command.

```
netsh advfirewall set allprofiles state off
```

"Windows is not genuine (0x80070005)"

Windows instances use Windows AWS KMS activation. If an instance is unable to complete the activation process, it reports that the copy of Windows is not genuine.

Try the suggestions for "Unable to activate Windows".

"No Terminal Server License Servers available to provide a license"

By default, Windows Server is licensed for two simultaneous users through Remote Desktop. If you need to provide more than two users with simultaneous access to your Windows instance through Remote Desktop, you can purchase a Remote Desktop Services client access license (CAL) and install the Remote Desktop Session Host and Remote Desktop Licensing Server roles.

Check for the following issues:

- You've exceeded the maximum number of concurrent RDP sessions.
- You've installed the Windows Remote Desktop Services role.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance as a user. You can try the following:
  - Connect to the instance from the command line using an `/admin` parameter, for example:

    ```
    mstsc /v:instance /admin
    ```

    For more information, see the following Microsoft article: Access Remote Desktop Via Command Line.
  - Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

"Some settings are managed by your organization"

Instances launched from the latest Windows Server AMIs might show a Windows Update dialog message stating "Some settings are managed by your organization." This message appears as a result of changes in Windows Server and does not impact the behavior of Windows Update or your ability to manage update settings.

To remove the warning

1. Open `gpedit.msc` and navigate to **Computer Configuration**, **Administrative Templates**, **Windows Components**, **Windows updates**. Edit **Configure Automatic Update**, and set it to **enabled**.
2. In a command prompt, update group policy using **gpupdate /force**.
3. Close and reopen the Windows Update Settings. You will see the above message about your settings being managed by your organization, followed by "We'll automatically download updates, except on metered connections (where charges may apply). In that case, we'll automatically download those updates required to keep Windows running smoothly."
4. Return to `gpedit.msc` and set the group policy back to **not configured**. Run **gpupdate /force** again.
5. Close the command prompt and wait a few minutes.
6. Reopen the Windows Update Settings. You should not see the message "Some settings are managed by your organization."

Troubleshoot 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
- High CPU usage shortly after Windows starts

- 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.

```
C:\> 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*.

High CPU usage shortly after Windows starts

If Windows Update is set to **Check for updates but let me choose whether to download and install them** (the default instance setting) this check can consume anywhere from 50 - 99% of the CPU on the instance. If this CPU consumption causes problems for your applications, you can manually change Windows Update settings in **Control Panel** or you can use the following script in the Amazon EC2 user data field:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v
 AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

When you run this script, specify a value for /d. The default value is 3. Possible values include the following:

1. Never check for updates
2. Check for updates but let me choose whether to download and install them
3. Download updates but let me choose whether to install them
4. Install updates automatically

After you modify the user data for your instance, you can run it. For more information, see View and update the instance user data and User data execution.

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 Windows instance

The following are possible problems you might have and error messages you might see while trying to connect to your Windows instance.

Contents

- Remote Desktop can't connect to the remote computer

- Error using the macOS RDP client
- RDP displays a black screen instead of the desktop
- Unable to remotely log on to an instance with a user that is not an administrator
- Troubleshooting Remote Desktop issues using AWS Systems Manager
- Enable Remote Desktop on an EC2 instance with remote registry
- I've lost my private key. How can I connect to my Windows instance?

Remote Desktop can't connect to the remote computer

Try the following to resolve issues related to connecting to your instance:

- Verify that you're using the correct public DNS hostname. (In the Amazon EC2 console, select the instance and check **Public DNS (IPv4)** in the details pane.) If your instance is in a VPC and you do not see a public DNS name, you must enable DNS hostnames. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*.
- Verify that your instance has a public IPv4 address. If not, you can associate an Elastic IP address with your instance. For more information, see Elastic IP addresses.
- To connect to your instance using an IPv6 address, check that your local computer has an IPv6 address and is configured to use IPv6. If you launched an instance from a Windows Server 2008 SP2 AMI or earlier, your instance is not automatically configured to recognize an IPv6 address assigned to the instance. For more information, see Configure IPv6 on your instances in the *Amazon VPC User Guide*.
- Verify that your security group has a rule that allows RDP access. For more information, see Create a security group.
- If you copied the password but get the error `Your credentials did not work`, try typing them manually when prompted. It's possible that you missed a character or got an extra white space character when you copied the password.
- Verify that the instance has passed status checks. For more information, see Status checks for your instances and Troubleshoot instances with failed status checks (*Amazon EC2 User Guide for Linux Instances*).
- Verify that the route table for the subnet has a route that sends all traffic destined outside the VPC to the internet gateway for the VPC. For more information, see Creating a custom route table (Internet Gateways) in the *Amazon VPC User Guide*.

- Verify that Windows Firewall, or other firewall software, is not blocking RDP traffic to the instance. We recommend that you disable Windows Firewall and control access to your instance using security group rules. You can use AWSSupport-TroubleshootRDP to disable the Windows Firewall profiles using SSM Agent . To disable Windows Firewall on a Windows instance that is not configured for AWS Systems Manager, use AWSSupport-ExecuteEC2Rescue, or use the following manual steps:

Manual steps

1. Stop the affected instance and detach its root volume.
2. Launch a temporary instance in the same Availability Zone as the affected instance.

> **Warning**
>
> If your temporary instance is based on the same AMI that the original instance is based on, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012.

3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key you just loaded and navigate to `ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy`. For each key with a name of the form *xxxx*Profile, select the key and change `EnableFirewall` from 1 to 0. Select the key again, and from the **File** menu, choose **Unload Hive**.
6. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

> **Warning**
>
> The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see Configure the Registry.

a. Open a command prompt, type **regedit.exe**, and press Enter.

b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.

c. Type **Windows Boot Manager** and then choose **Find Next**.

d. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.

e. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).

f. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

7. Using the **Disk Management** utility, bring the drive offline.

> **Note**
>
> The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

8. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
9. Restore the root volume of the affected instance by attaching it as `/dev/sda1`.
10. Start the instance.

- Verify that Network Level Authentication is disabled on instances that are not part of an Active Directory domain (use AWSSupport-TroubleshootRDP to disable NLA).
- Verify that the Remote Desktop Service (TermService) Startup Type is Automatic and the service is started (use AWSSupport-TroubleshootRDP to enable and start the RDP service).
- Verify that you are connecting to the correct Remote Desktop Protocol port, which by default is 3389 (use AWSSupport-TroubleshootRDP to read the current RDP port and change it back to 3389).
- Verify that Remote Desktop connections are allowed on your instance (use AWSSupport-TroubleshootRDP to enable Remote Desktop connections).
- Verify that the password has not expired. If the password has expired, you can reset it. For more information, see Reset a lost or expired Windows administrator password.

- If you attempt to connect using a user that you created on the instance and receive the error The `user cannot connect to the server due to insufficient access privileges`, verify that you granted the user the right to log on locally. For more information, see Grant a Member the Right to Logon Locally.
- If you attempt more than the maximum allowed concurrent RDP sessions, your session is terminated with the message `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.` By default, you are allowed two concurrent RDP sessions to your instance.

Error using the macOS RDP client

If you are connecting to a Windows Server 2012 R2 instance using the Remote Desktop Connection client from the Microsoft website, you may get the following error:

```
Remote Desktop Connection cannot verify the identity of the computer that you want to
 connect to.
```

Download the Microsoft Remote Desktop app from the Mac App Store and use the app to connect to your instance.

RDP displays a black screen instead of the desktop

Try the following to resolve this issue:

- Check the console output for additional information. To get the console output for your instance using the Amazon EC2 console, select the instance, and then choose **Actions**, **Monitor and troubleshoot**, **Get system log**.
- Verify that you are running the latest version of your RDP client.
- Try the default settings for the RDP client. For more information, see Remote Session Environment.
- If you are using Remote Desktop Connection, try starting it with the `/admin` option as follows.

  ```
  mstsc /v:instance /admin
  ```

- If the server is running a full-screen application, it might have stopped responding. Use Ctrl +Shift+Esc to start Windows Task Manager, and then close the application.

- If the server is over-utilized, it might have stopped responding. To monitor the instance using the Amazon EC2 console, select the instance and then select the **Monitoring** tab. If you need to change the instance type to a larger size, see Change the instance type.

Unable to remotely log on to an instance with a user that is not an administrator

If you are not able to remotely log on to a Windows instance with a user that is not an administrator account, ensure that you have granted the user the right to log on locally. See Grant a user or group the right to log on locally to the domain controllers in the domain.

Troubleshooting Remote Desktop issues using AWS Systems Manager

You can use AWS Systems Manager to troubleshoot issues connecting to your Windows instance using RDP.

AWSSupport-TroubleshootRDP

The AWSSupport-TroubleshootRDP automation document allows the user to check or modify common settings on the target instance that can impact Remote Desktop Protocol (RDP) connections, such as the **RDP Port**, **Network Layer Authentication (NLA)**, and **Windows Firewall** profiles. By default, the document reads and outputs the values of these settings.

The AWSSupport-TroubleshootRDP automation document can be used with EC2 instances, on-premises instances, and virtual machines (VMs) that are enabled for use with AWS Systems Manager (managed instances). In addition, it can also be used with EC2 instances for Windows Server that are *not* enabled for use with Systems Manager. For information about enabling instances for use with AWS Systems Manager, see Managed nodes in the *AWS Systems Manager User Guide*.

To troubleshoot using the AWSSupport-TroubleshootRDP document

1. Log in to the Systems Manager Console.
2. Verify that you are in the same Region as the impaired instance.
3. Choose **Documents** from the left navigation pane.
4. On the **Owned by Amazon** tab, enter `AWSSupport-TroubleshootRDP` in the search field. When the `AWSSupport-TroubleshootRDP` document appears, select it.

5. Choose **Execute automation**.
6. For **Execution Mode**, choose **Simple execution**.
7. For **Input parameters**, **InstanceId**, enable **Show interactive instance picker**.
8. Choose your Amazon EC2 instance.
9. Review the examples, then choose **Execute**.
10. To monitor the execution progress, for **Execution status**, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose an item from **Step ID.**

AWSSupport-TroubleshootRDP examples

The following examples show you how to accomplish common troubleshooting tasks using AWSSupport-TroubleshootRDP. You can use either the example AWS CLI start-automation-execution command or the provided link to the AWS Management Console.

Example Example: Check the current RDP status

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=instance_id, Action=Custom" --region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region#documentVersion=$LATEST
```

Example Example: Disable the Windows Firewall

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=instance_id, Action=Custom, Firewall=Disable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&Firewall=Disable
```

Example Example: Disable Network Level Authentication

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=instance_id, Action=Custom, NLASettingAction=Disable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion
```

Example Example: Set RDP Service Startup Type to Automatic and start the RDP service

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=instance_id, Action=Custom, RDPServiceStartupType=Auto,
 RDPServiceAction=Start" --region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/
AWSSupport-TroubleshootRDP?region=region_code#documentVersion=
$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example Example: Restore the default RDP Port (3389)

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=instance_id, Action=Custom, RDPPortAction=Modify" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RDPPortAction=Modify
```

Example Example: Allow remote connections

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=instance_id, Action=Custom, RemoteConnections=Enable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RemoteConnections=Enable
```

AWSSupport-ExecuteEC2Rescue

The AWSSupport-ExecuteEC2Rescue automation document uses Use EC2Rescue for Windows Server to automatically troubleshoot and restore EC2 instance connectivity and RDP issues. For more information, see Run the EC2Rescue tool on unreachable instances.

The AWSSupport-ExecuteEC2Rescue automation document requires a stop and restart of the instance. Systems Manager Automation stops the instance and creates an Amazon Machine Image (AMI). Data stored in instance store volumes is lost. The public IP address changes if you are not using an Elastic IP address. For more information, see Run the EC2Rescue tool on unreachable instances in the *AWS Systems Manager User Guide*.

To troubleshoot using the AWSSupport-ExecuteEC2Rescue document

1. Open the Systems Manager console.
2. Verify that you are in the same Region as the impaired Amazon EC2 instance.
3. In the navigation panel, choose **Documents**.
4. Search for and select the `AWSSupport-ExecuteEC2Rescue` document, and then choose **Execute automation**.
5. In **Execution Mode**, choose **Simple execution**.
6. In the **Input parameters** section, for **UnreachableInstanceId**, enter the Amazon EC2 instance ID of the unreachable instance.

7. (Optional) For **LogDestination**, enter the Amazon Simple Storage Service (Amazon S3) bucket name if you want to collect operating system logs for troubleshooting your Amazon EC2 instance. Logs are automatically uploaded to the specified bucket.
8. Choose **Execute**.
9. To monitor the execution progress, in **Execution status**, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose the **Step ID**.

Enable Remote Desktop on an EC2 instance with remote registry

If your unreachable instance is not managed by AWS Systems Manager Session Manager, then you can use remote registry to enable Remote Desktop.

1. From the EC2 console, stop the unreachable instance.
2. Detach the root volume of the unreachable instance and attach it to a reachable instance in the same Availability Zone as a storage volume. If you don't have a reachable instance in the same Availability Zone, launch one. Note the device name of the root volume on the unreachable instance.
3. On the reachable instance, open Disk Management. You can do so by running the following command in the Command Prompt window.

   ```
   diskmgmt.msc
   ```

4. Right click the newly attached volume that came from the unreachable instance, and then choose **Online**.
5. Open the Windows Registry Editor. You can do so by running the following command in the Command Prompt window.

   ```
   regedit
   ```

6. In Registry Editor, choose **HKEY_LOCAL_MACHINE**, then select **File**, **Load Hive**.
7. Select the drive of the attached volume, navigate to `\Windows\System32\config\`, select `SYSTEM`, and then choose **Open**.
8. For **Key Name**, enter a unique name for the hive and choose **OK**.
9. Back up the registry hive before making any changes to the registry.

a. In the Registry Editor console tree, select the hive that you loaded: **HKEY_LOCAL_MACHINE***`your-key-name`*.
b. Choose **File**, **Export**.
c. In the Export Registry File dialog box, choose the location to which you want to save the backup copy, and then type a name for the backup file in the **File name** field.
d. Choose **Save**.

10. In Registry Editor, navigate to `HKEY_LOCAL_MACHINE\`*`your key name`*`\ControlSet001\Control\Terminal Server`, and then, in the details pane, double-click **fDenyTSConnections**.
11. In the **Edit DWORD** value box, enter `0` in the **Value data** field.
12. Choose **OK**.

> **Note**
>
> If the value in the **Value data** field is 1, then the instance will deny remote desktop connections. A value of `0` allows remote desktop connections.

13. In Registry Editor, choose **HKEY_LOCAL_MACHINE***`your-key-name`*, then select **File**, **Unload Hive**.
14. Close Registry Editor and Disk Management.
15. From the EC2 console, detach the volume from the reachable instance and then reattach it to the unreachable instance. When attaching the volume to the unreachable instance, enter the device name that you saved earlier in the **device** field.
16. Restart the unreachable instance.

I've lost my private key. How can I connect to my Windows instance?

When you connect to a newly-launched Windows instance, you decrypt the password for the Administrator account using the private key for the key pair that you specified when you launched the instance.

If you lose the Administrator password and you no longer have the private key, you must reset the password or create a new instance. For more information, see Reset a lost or expired Windows administrator password. For steps to reset the password using an Systems Manager document, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

Troubleshoot an unreachable instance

You can use the following methods to troubleshoot an unreachable Windows instance. For information about troubleshooting an unreachable Linux instance, see Troubleshoot an unreachable instance in the *EC2 User Guide for Linux*.

Contents

- Instance reboot
- Instance console output
- Capture a screenshot of an unreachable instance
- Common screenshots
- 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 Windows instances, the instance console output includes the last three system event log errors.

You can optionally retrieve the latest serial console output at any time during the instance lifecycle. This option is only supported on Instances built on the 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)

Capture a screenshot of an unreachable instance

If you are unable to reach your instance via RDP, 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 also use EC2 Rescue on instances running Windows Server 2008 or later to gather and analyze data from offline instances.

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

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)

Common screenshots

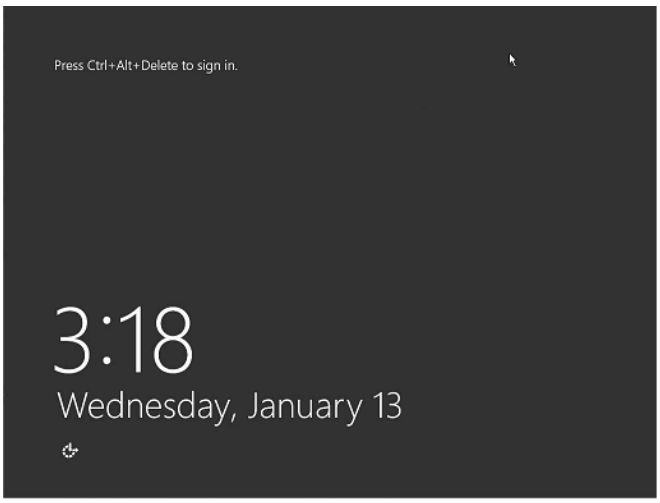
You can use the following information to help you troubleshoot an unreachable Windows instance based on screenshots returned by the service.

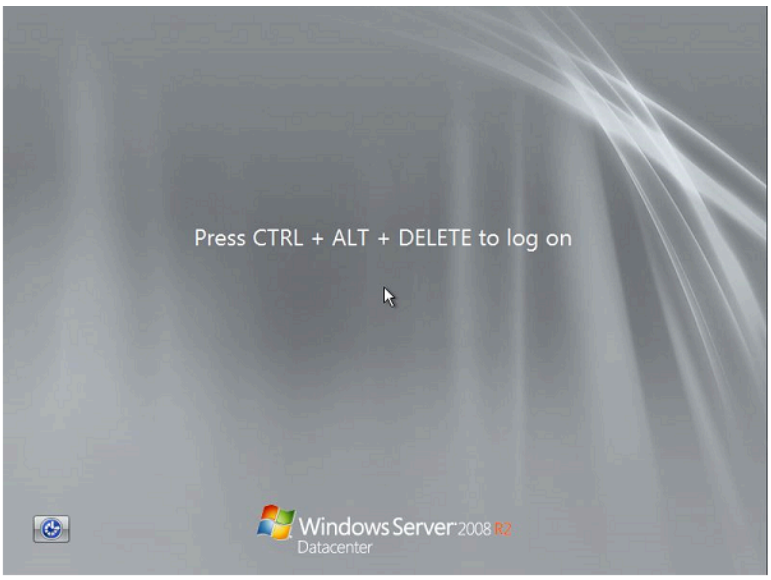
- Log on screen (Ctrl+Alt+Delete)
- Recovery console screen
- Windows boot manager screen
- Sysprep screen

- Getting ready screen
- Windows Update screen
- Chkdsk

Log on screen (Ctrl+Alt+Delete)

Console Screenshot Service returned the following.





If an instance becomes unreachable during logon, there could be a problem with your network configuration or Windows Remote Desktop Services. An instance can also be unresponsive if a process is using large amounts of CPU.

Network configuration

Use the following information to verify that your AWS, Microsoft Windows, and local (or on-premises) network configurations aren't blocking access to the instance.

AWS network configuration

Configuration	Verify
Security group configuration	Verify that port 3389 is open for your security group. Verify you are connecting to the right public IP address. If the instance was not associated with an Elastic IP, the public IP changes after the instance stops/starts. For

Configuration	Verify
	more information, see Remote Desktop can't connect to the remote computer.
VPC configuration (Network ACLs)	Verify that the access control list (ACL) for your Amazon VPC is not blocking access. For information, see Network ACLs in the *Amazon VPC User Guide*.
VPN configuration	If you are connecting to your VPC using a virtual private network (VPN), verify VPN tunnel connectivity. For more information, see How do I troubleshoot VPN tunnel connectivity to an Amazon VPC?

Windows network configuration

Configuration	Verify
Windows Firewall	Verify that Windows Firewall isn't blocking connections to your instance. Disable Windows Firewall as described in bullet 7 of the remote desktop troubleshooting section, Remote Desktop can't connect to the remote computer.
Advanced TCP/IP configuration (Use of static IP)	The instance may be unresponsive because you configured a static IP address. For a VPC, create a network interface and attach it to the instance.

Local or on-premises network configuration

Verify that a local network configuration isn't blocking access. Try to connect to another instance in the same VPC as your unreachable instance. If you can't access another instance, work with your local network administrator to determine whether a local policy is restricting access.

Remote Desktop Services issues

If the instance can't be reached during logon, there could a problem with Remote Desktop Services (RDS) on the instance.

> **Tip**
>
> You can use the `AWSSupport-TroubleshootRDP` runbook to check and modify various settings that might affect Remote Desktop Protocol (RDP) connections. For more information, see `AWSSupport-TroubleshootRDP` in the *AWS Systems Manager Automation runbook reference*.

Remote Desktop Services configuration

Configuration	Verify
RDS is running	Verify that RDS is running on the instance. Connect to the instance using the Microsoft Management Console (MMC) Services snap-in (`services.msc`). In the list of services, verify that **Remote Desktop Services** is **Running**. If it isn't, start it and then set the startup type to **Automatic**. If you can't connect to the instance by using the Services snap-in, detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume, and modify the Start registry key. When you are finished, reattach the root volume to the original instance.
RDS is enabled	Even if the service is started, it might be disabled. Detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume, and enable the service by modifying the **Terminal Server** registry key as described in Enable Remote Desktop on an EC2 instance with remote registry.

Configuration	Verify
	When you are finished, reattach the root volume to the original instance.

High CPU usage

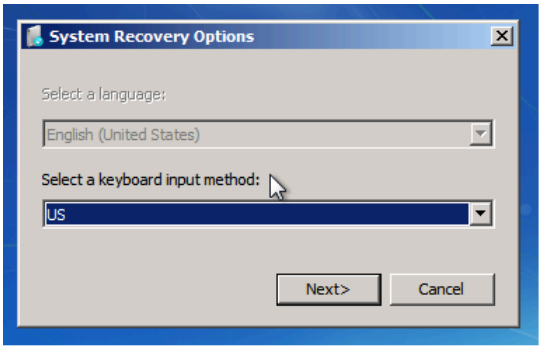
Check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch. If **CPUUtilization (Maximum)** is a high number, wait for the CPU to go down and try connecting again. High CPU usage can be caused by:

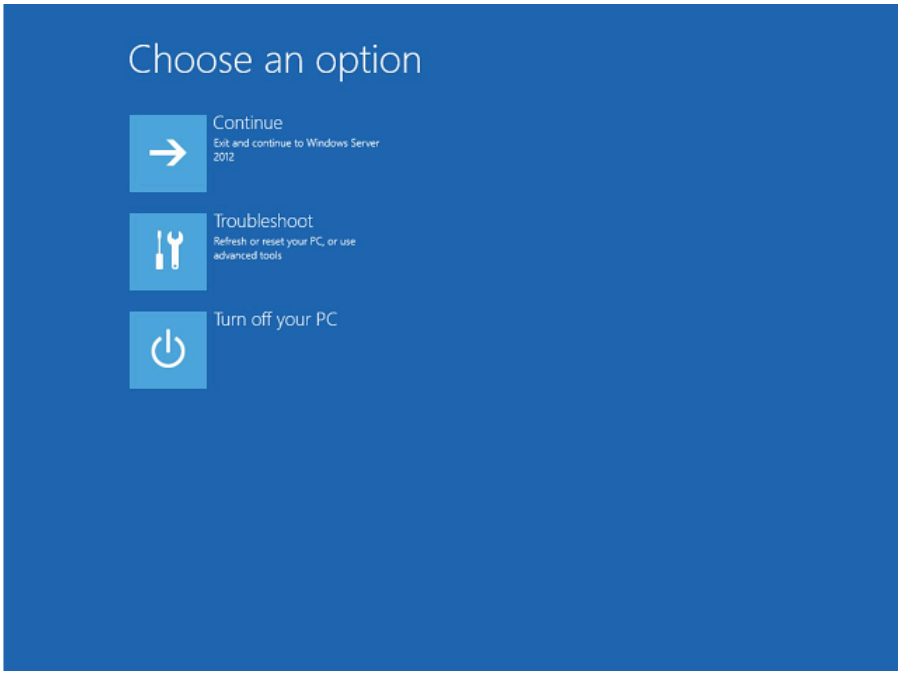
- Windows Update
- Security Software Scan
- Custom Startup Script
- Task Scheduler

For more information, see Get Statistics for a Specific Resource in the *Amazon CloudWatch User Guide*. For additional troubleshooting tips, see High CPU usage shortly after Windows starts.

Recovery console screen

Console Screenshot Service returned the following.





The operating system might boot into the Recovery console and get stuck in this state if the `bootstatuspolicy` is not set to `ignoreallfailures`. Use the following procedure to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

By default, the policy configuration for public Windows AMIs provided by AWS is set to `ignoreallfailures`.

1. Stop the unreachable instance.
2. Create a snapshot of the root volume. The root volume is attached to the instance as `/dev/sda1`.

   Detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability Zone as a secondary volume.

   > **Warning**
   >
   > If your temporary instance and the original instance were launched using the same AMI, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. If you must create a temporary instance using the same AMI, to avoid a disk signature collision, complete the steps in Disk signature collision.
   > Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses an AMI for Windows Server 2008 R2, launch the temporary instance using an AMI for Windows Server 2012.

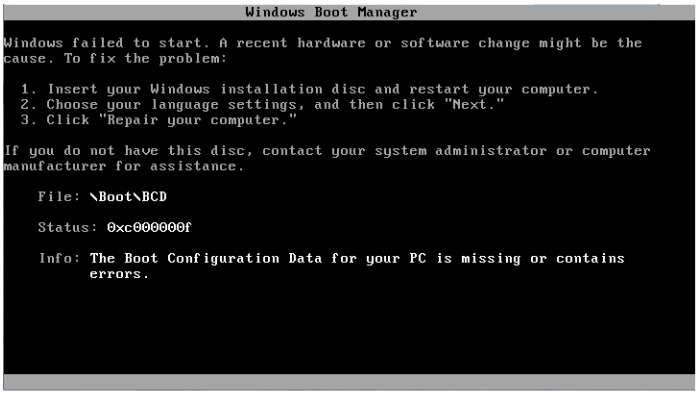
3. Log in to the instance and run the following command from a command prompt to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

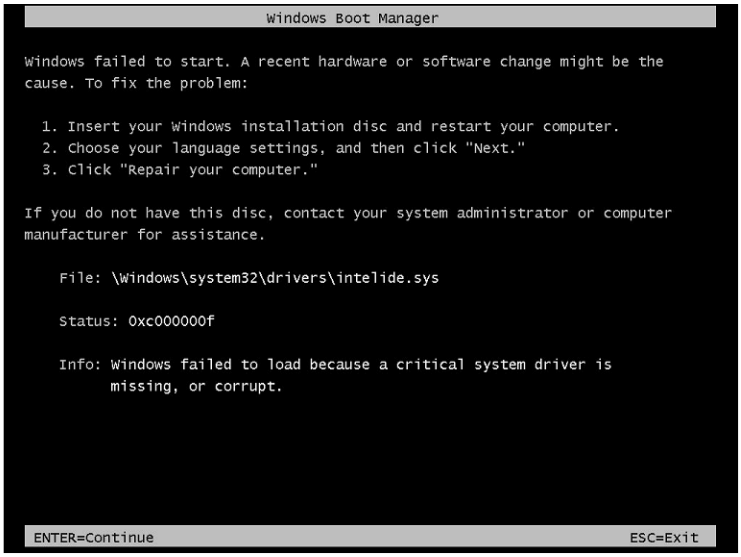
   ```
   bcdedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy
    ignoreallfailures
   ```

4. Reattach the volume to the unreachable instance and start the instance again.

Windows boot manager screen

Console Screenshot Service returned the following.

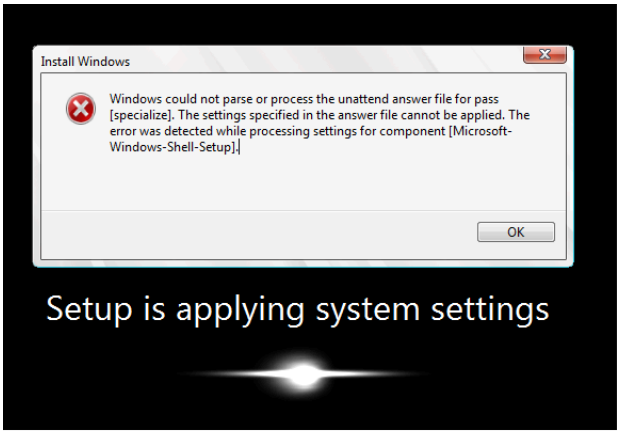


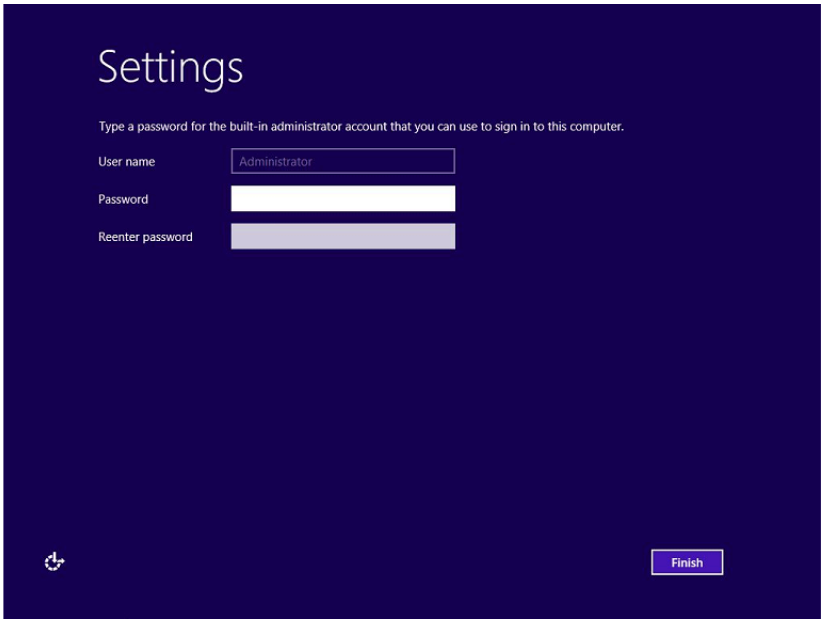


The operating system experienced a fatal corruption in the system file and/or the registry. When the instance is stuck in this state, you should recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach any root volumes from the unreachable instance, take a snapshot of those volume or create an AMI from them, and attach them to another instance in the same Availability Zone as a secondary volume.

Sysprep screen

Console Screenshot Service returned the following.

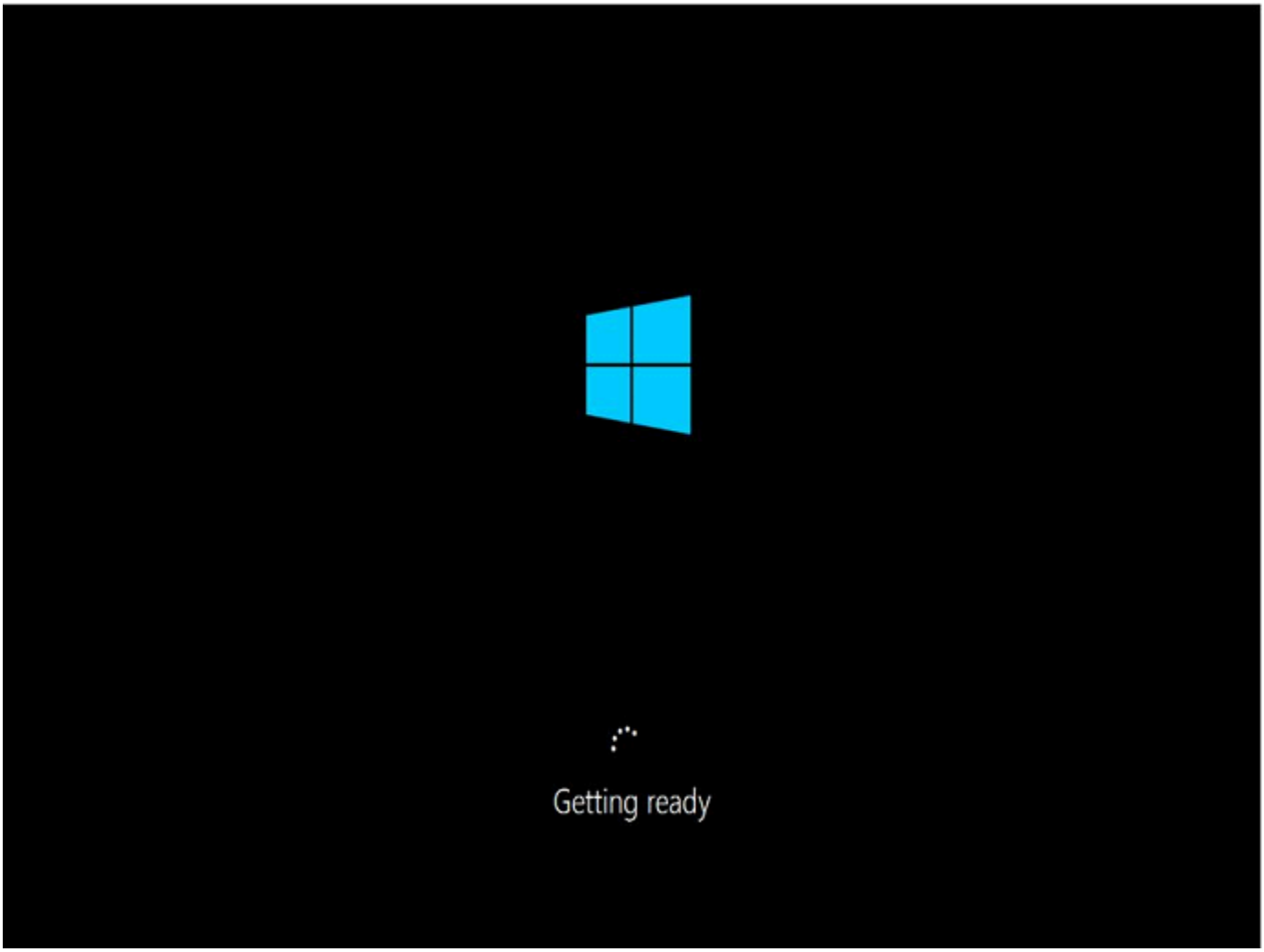




You may see this screen if you did not use the EC2Config Service to call Sysprep or if the operating system failed while running Sysprep. You can reset the password using EC2Rescue. Otherwise, Create a standardized Amazon Machine Image (AMI) using Sysprep.

Getting ready screen

Console Screenshot Service returned the following.

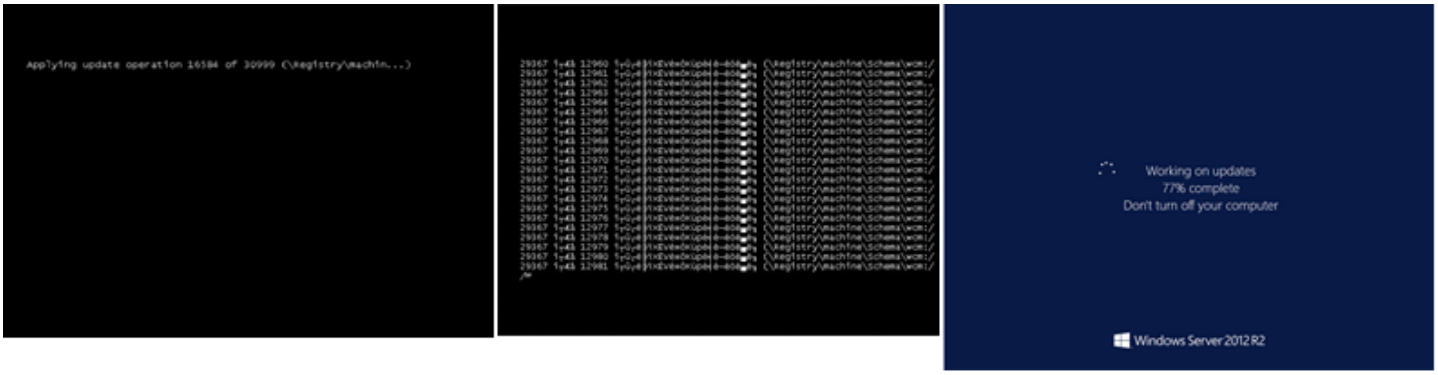


Refresh the Instance Console Screenshot Service repeatedly to verify that the progress ring is spinning. If the ring is spinning, wait for the operating system to start up. You can also check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch to see if the operating system is active. If the progress ring is not spinning, the instance may be stuck at the boot process. Reboot the instance. If rebooting does not solve the problem, recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach the root volume from the unreachable instance, take a snapshot of the volume

or create an AMI from it. Then attach it to another instance in the same Availability Zone as a secondary volume.

Windows Update screen

Console Screenshot Service returned the following.

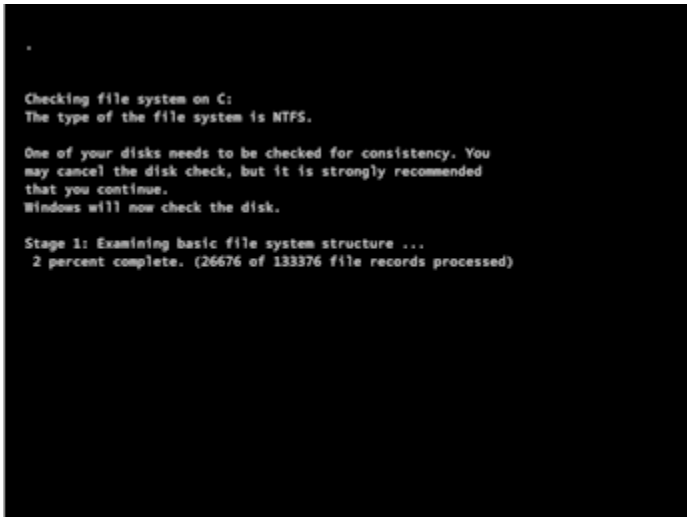


The Windows Update process is updating the registry. Wait for the update to finish. Do not reboot or stop the instance as this may cause data corruption during the update.

> **Note**
>
> The Windows Update process can consume resources on the server during the update. If you experience this problem often, consider using faster instance types and faster EBS volumes.

Chkdsk

Console Screenshot Service returned the following.



Windows is running the chkdsk system tool on the drive to verify file system integrity and fix logical file system errors. Wait for process to complete.

Instance recovery when a host computer fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You are notified of such an event ahead of time by email.

To recover an Amazon EBS-backed instance running on a host computer that failed

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.
4. Restore any important data.

For more information, see Stop and start your instance.

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.

Reset a lost or expired Windows administrator password

If you are no longer able to access your Windows Amazon EC2 instance because the Windows administrator password is lost or expired, you can reset the password.

> **Note**
>
> There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

The manual methods to reset the administrator password use EC2Launch v2, EC2Config, or EC2Launch.

- For all supported Windows AMIs that include the EC2Launch v2 agent, use EC2Launch v2.
- For Windows AMIs before Windows Server 2016, use the EC2Config service.
- For Windows Server 2016 and later AMIs, use the EC2Launch service.

These procedures also describe how to connect to an instance if you lost the key pair that was used to create the instance. Amazon EC2 uses a public key to encrypt a piece of data, such as a password, and a private key to decrypt the data. The public and private keys are known as a *key pair*. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

> **Note**
>
> If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Using EC2Rescue for Windows Server with Systems Manager Run Command.

Contents

- Reset the Windows administrator password using EC2Launch v2

- Reset the Windows administrator password using EC2Config
- Reset the Windows administrator password using EC2Launch

Reset the Windows administrator password using EC2Launch v2

If you have lost your Windows administrator password and are using a supported Windows AMI that includes the EC2Launch v2 agent, you can use EC2Launch v2 to generate a new password.

If you are using a Windows Server 2016 or later AMI that does not include the EC2Launch v2 agent, see Reset the Windows administrator password using EC2Launch.

If you are using a Windows Server AMI earlier than Windows Server 2016 that does not include the EC2Launch v2 agent, see Reset the Windows administrator password using EC2Config.

> **Note**
>
> If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Using EC2Rescue for Windows Server with Systems Manager Run Command.

> **Note**
>
> There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Launch v2, you need to do the following:

- Step 1: Verify that the EC2Launch v2 agent is running
- Step 2: Detach the root volume from the instance
- Step 3: Attach the volume to a temporary instance
- Step 4: Delete the `.run-once` file
- Step 5: Restart the original instance

Step 1: Verify that the EC2Launch v2 agent is running

Before you attempt to reset the administrator password, verify that the EC2Launch v2 agent is installed and running. You use the EC2Launch v2 agent to reset the administrator password later in this section.

To verify that the EC2Launch v2 agent is running

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then select the instance that requires a password reset. This instance is referred to as the *original* instance in this procedure.
3. Choose **Actions**, **Monitor and troubleshoot**, **Get system log**.
4. Locate the EC2 Launch entry, for example, **Launch: EC2Launch v2 service v2.0.124**. If you see this entry, the EC2Launch v2 service is running.

   If the system log output is empty, or if the EC2Launch v2 agent is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see Capture a screenshot of an unreachable instance.

Step 2: Detach the root volume from the instance

You can't use EC2Launch v2 to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires a password reset and choose **Actions**, **Instance state**, **Stop instance**. After the status of the instance changes to **Stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.

   a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.

b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.

c. Choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**. In the navigation pane, choose **AMIs**. After the image status changes to **available**, continue to the next step.

d. Select the image and choose **Actions**, and then **Launch**.

e. Complete the wizard, selecting the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch**.

f. When prompted, choose the key pair that you created for the new instance, select the acknowledgement check box, and then choose **Launch Instances**.

g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.

5. Detach the root volume from the original instance as follows:

   a. In the **Description** pane of the original instance, note the ID of the EBS volume listed as the **Root device**.

   b. In the navigation pane, choose **Volumes**.

   c. In the list of volumes, select the volume noted in the previous step, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

Step 3: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to modify the configuration file.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:

   a. In the navigation pane, choose **Instances**, choose **Launch instances**, and then select an AMI.

> **Important**
>
> To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2019, launch the temporary instance using the base AMI for Windows Server 2016.

b. Leave the default instance type and choose **Next: Configure Instance Details**.

c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

> **Important**
>
> The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.

d. On the **Review Instance Launch** page, choose **Launch**.

e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.

   c. For **Device**, type **`xvdf`** (if it isn't already there), and choose **Attach**.

Step 4: Delete the `.run-once` file

You must now delete the `.run-once` file from the offline volume attached to the instance. This directs EC2Launch v2 to run all tasks with a frequency of `once`, which includes setting the administrator password. The file path in the secondary volume that you attached will be similar to `D:\ProgramData\Amazon\EC2Launch\state\.run-once`.

To delete the .run-once file

1. Open the **Disk Management** utility, and bring the drive online using these instructions: Making an Amazon EBS volume available for use.
2. Locate the `.run-once` file in the disk you brought online.
3. Delete the .run-once file.

> **Important**
>
> Any scripts set to run once will be triggered by this action.

Step 5: Restart the original instance

After you have deleted the `.run-once` file, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

1. Reattach the volume to the original instance as follows:
   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions**, **Attach Volume**.
   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.
   c. For **Device**, type **`/dev/sda1`**.
   d. Choose **Attach**. After the volume status changes to `in-use`, continue to the next step.
2. In the navigation pane, choose **Instances**. Select the original instance and choose **Instance state**, **Start instance**. After the instance state changes to `Running`, continue to the next step.
3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see Connect to your Windows instance.

   > **Important**
   >
   > The instance gets a new public IP address after you stop and start it. Make sure to connect to the instance using its current public DNS name. For more information, see Instance lifecycle.

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

Reset the Windows administrator password using EC2Config

If you have lost your Windows administrator password and are using a Windows AMI before Windows Server 2016, you can use the EC2Config agent to generate a new password.

If you are using a Windows Server 2016 or later AMI, see Reset the Windows administrator password using EC2Launch or, you can use the EC2Rescue tool, which uses the EC2Launch service to generate a new password.

> **Note**
>
> If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Using EC2Rescue for Windows Server with Systems Manager Run Command.

> **Note**
>
> There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Config, you need to do the following:

- Step 1: Verify that the EC2Config service is running
- Step 2: Detach the root volume from the instance
- Step 3: Attach the volume to a temporary instance
- Step 4: Modify the configuration file
- Step 5: Restart the original instance

Step 1: Verify that the EC2Config service is running

Before you attempt to reset the administrator password, verify that the EC2Config service is installed and running. You use the EC2Config service to reset the administrator password later in this section.

To verify that the EC2Config service is running

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then select the instance that requires a password reset. This instance is referred to as the *original* instance in this procedure.
3. (New console) Choose **Actions**, **Monitor and troubleshoot**, **Get system log**.

   (Old console) Choose **Actions**, **System Settings**, **Get System Log**.
4. Locate the EC2 Agent entry, for example, **EC2 Agent: Ec2Config service v3.18.1118**. If you see this entry, the EC2Config service is running.

   If the system log output is empty, or if the EC2Config service is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see Capture a screenshot of an unreachable instance.

Step 2: Detach the root volume from the instance

You can't use EC2Config to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires a password reset and choose **Actions**, **Instance state**, **Stop instance**. After the status of the instance changes to **Stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.

a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.

b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.

c. Choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**. In the navigation pane, choose **AMIs**. After the image status changes to **available**, continue to the next step.

d. Select the image and choose **Actions**, and then **Launch**.

e. Complete the wizard, selecting the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch**.

f. When prompted, choose the key pair that you created for the new instance, select the acknowledgement check box, and then choose **Launch Instances**.

g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.

5. Detach the root volume from the original instance as follows:

   a. In the **Description** pane of the original instance, note the ID of the EBS volume listed as the **Root device**.

   b. In the navigation pane, choose **Volumes**.

   c. In the list of volumes, select the volume noted in the previous step, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

Step 3: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to modify the configuration file.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:

   a. In the navigation pane, choose **Instances**, choose **Launch instances**, and then select an AMI.

   > **Important**
   >
   > To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2019, launch the temporary instance using the base AMI for Windows Server 2016.

   b. Leave the default instance type and choose **Next: Configure Instance Details**.

   c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

   > **Important**
   >
   > The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.

   d. On the **Review Instance Launch** page, choose **Launch**.

   e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.

   c. For **Device**, type **`xvdf`** (if it isn't already there), and choose **Attach**.

Step 4: Modify the configuration file

After you have attached the volume to the temporary instance as a secondary volume, modify the `Ec2SetPassword` plugin in the configuration file.

To modify the configuration file

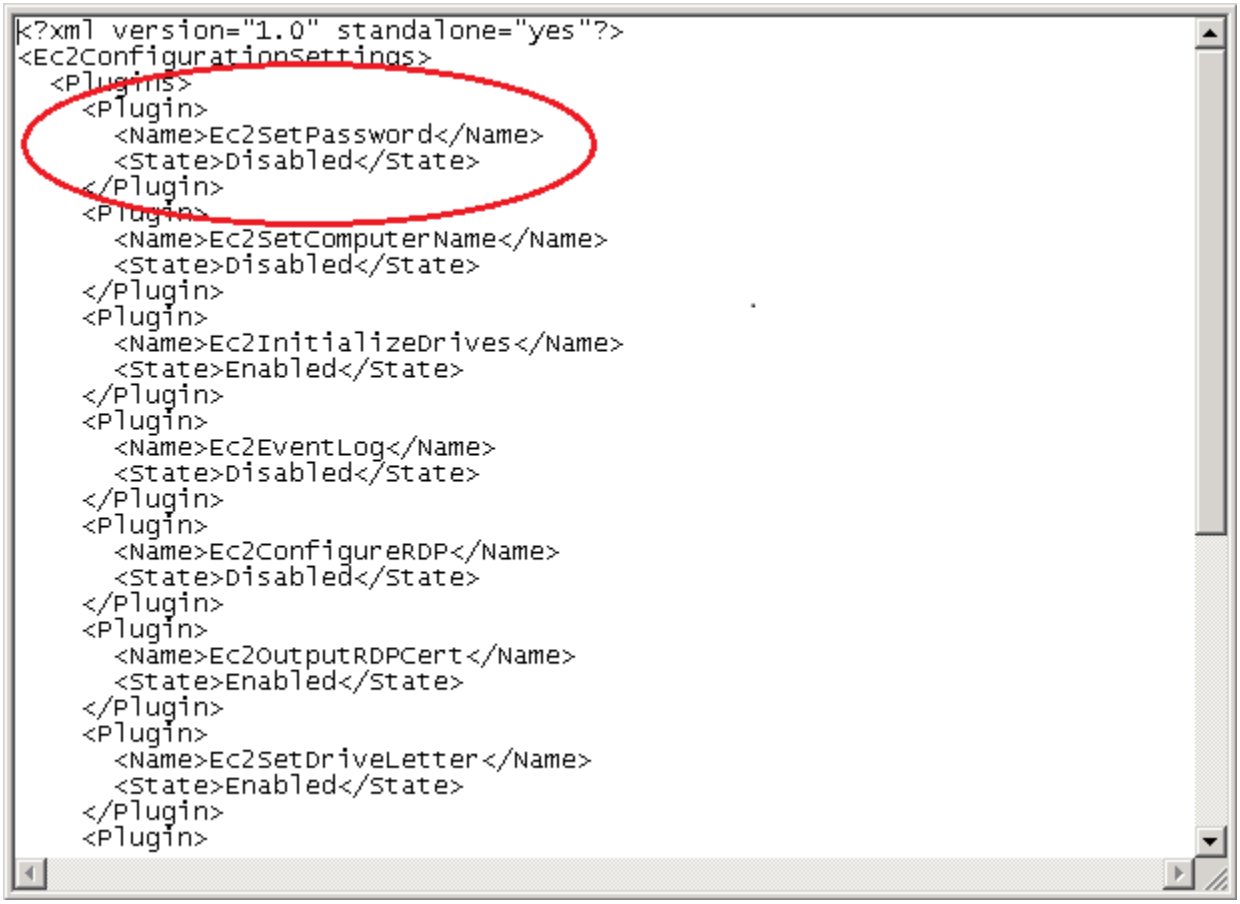
1. From the temporary instance, modify the configuration file on the secondary volume as follows:

   a. Launch and connect to the temporary instance.

   b. Open the **Disk Management** utility, and bring the drive online using these instructions: Making an Amazon EBS volume available for use.

   c. Navigate to the secondary volume, and open `\Program Files\Amazon\Ec2ConfigService\Settings\config.xml` using a text editor, such as Notepad.

   d. At the top of the file, find the plugin with the name `Ec2SetPassword`, as shown in the screenshot. Change the state from `Disabled` to `Enabled` and save the file.



2. After you have modified the configuration file, detach the secondary volume from the temporary instance as follows:

   a. Using the **Disk Management** utility, bring the volume offline.

   b. Disconnect from the temporary instance and return to the Amazon EC2 console.

   c. In the navigation pane, choose **Volumes**, select the volume, and then choose **Actions**, **Detach Volume**. After the volume's status changes to **available**, continue with the next step.

Step 5: Restart the original instance

After you have modified the configuration file, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

1. Reattach the volume to the original instance as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.

   c. For **Device**, type **`/dev/sda1`**.

   d. Choose **Attach**. After the volume status changes to `in-use`, continue to the next step.

2. In the navigation pane, choose **Instances**. Select the original instance and choose **Instance state**, **Start instance**. After the instance state changes to `Running`, continue to the next step.

3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see Connect to your Windows instance.

   > **Important**
   >
   > The instance gets a new public IP address after you stop and start it. Make sure to connect to the instance using its current public DNS name. For more information, see Instance lifecycle.

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

Reset the Windows administrator password using EC2Launch

If you have lost your Windows administrator password and are using a Windows Server 2016 or later AMI, you can use the EC2Rescue tool, which uses the EC2Launch service to generate a new password.

If you are using a Windows Server 2016 or later AMI that does not include the EC2Launch v2 agent, you can use EC2Launch v2 to generate a new password.

If you are using a Windows Server AMI earlier than Windows Server 2016, see Reset the Windows administrator password using EC2Config.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

> **Note**
>
> If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Using EC2Rescue for Windows Server with Systems Manager Run Command.

> **Note**
>
> There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Launch, you need to do the following:

- Step 1: Detach the root volume from the instance
- Step 2: Attach the volume to a temporary instance

- Step 3: Reset the administrator password
- Step 4: Restart the original instance

Step 1: Detach the root volume from the instance

You can't use EC2Launch to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires a password reset and choose **Actions**, **Instance state**, **Stop instance**. After the status of the instance changes to **Stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.

   a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.

   b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.

   c. Choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**. In the navigation pane, choose **AMIs**. After the image status changes to **available**, continue to the next step.

   d. Select the image and choose **Actions**, and then **Launch**.

   e. Complete the wizard, selecting the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch**.

   f. When prompted, choose the key pair that you created for the new instance, select the acknowledgement check box, and then choose **Launch Instances**.

   g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.

5. Detach the root volume from the original instance as follows:

   a. In the **Description** pane of the original instance, note the ID of the EBS volume listed as the **Root device**.

   b. In the navigation pane, choose **Volumes**.

   c. In the list of volumes, select the volume noted in the previous step, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

Step 2: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to run EC2Launch.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:

   a. In the navigation pane, choose **Instances**, choose **Launch instances**, and then select an AMI.

   > **Important**
   >
   > To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2019, launch the temporary instance using the base AMI for Windows Server 2016.

   b. Leave the default instance type and choose **Next: Configure Instance Details**.

   c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

> **Important**
>
> The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.

d. On the **Review Instance Launch** page, choose **Launch**.

e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.

   c. For **Device**, type **`xvdf`** (if it isn't already there), and choose **Attach**.

Step 3: Reset the administrator password

Next, connect to the temporary instance and use EC2Launch to reset the administrator password.

To reset the administrator password

1. Connect to the temporary instance and use the EC2Rescue for Windows Server tool on the instance to reset the administrator password as follows:

   a. Download the EC2Rescue for Windows Server zip file, extract the contents, and run **EC2Rescue.exe**.

   b. On the **License Agreement** screen, read the license agreement, and, if you accept the terms, choose **I Agree**.

   c. On the **Welcome to EC2Rescue for Windows Server** screen, choose **Next**.

   d. On the **Select mode** screen, choose **Offline instance**.

   e. On the **Select a disk** screen, select the **xvdf** device and choose **Next**.

   f. Confirm the disk selection and choose **Yes**.

   g. After the volume has loaded, choose **OK**.

h. On the **Select Offline Instance Option** screen, choose **Diagnose and Rescue**.

i. On the **Summary** screen, review the information and choose **Next**.

j. On the **Detected possible issues** screen, select **Reset Administrator Password** and choose **Next**.

k. On the **Confirm** screen, choose **Rescue**, **OK**.

l. On the **Done** screen, choose **Finish**.

m. Close the EC2Rescue for Windows Server tool, disconnect from the temporary instance, and then return to the Amazon EC2 console.

2. Detach the secondary (`xvdf`) volume from the temporary instance as follows:

   a. In the navigation pane, choose **Instances** and select the temporary instance.

   b. On the **Storage** tab for the temporary instance, note the ID of the EBS volume listed as **xvdf**.

   c. In the navigation pane, choose **Volumes**.

   d. In the list of volumes, select the volume noted in the previous step, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

Step 4: Restart the original instance

After you have reset the administrator password using EC2Launch, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

To restart the original instance

1. Reattach the volume to the original instance as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.

   c. For **Device**, type **`/dev/sda1`**.

   d. Choose **Attach**. After the volume status changes to `in-use`, continue to the next step.

2. In the navigation pane, choose **Instances**. Select the original instance and choose **Instance state**, **Start instance**. After the instance state changes to `Running`, continue to the next step.
3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see Connect to your Windows instance.
4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

Troubleshoot stopping your instance

If you have stopped your Amazon EBS-backed instance and it appears stuck in the `stopping` state, there may be an issue with the underlying host computer.

There is no cost for instance usage while an instance is in the `stopping` state or in any other state except `running`. You are only charged for instance usage when an instance is in the `running` state.

Force stop the instance

Force the instance to stop using either the console or the AWS CLI.

> **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 Windows AMI from a running 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.

```
C:\> 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 Sysprep

If you experience problems or receive error messages during image preparations, review the following logs. Log location varies depending on whether you are running EC2Config, EC2Launch v1, or EC2Launch v2 with Sysprep.

- `%WINDIR%\Panther\Unattendgc` (EC2Config, EC2Launch v1, and EC2Launch v2)
- `%WINDIR%\System32\Sysprep\Panther` (EC2Config, EC2Launch v1, and EC2Launch v2)
- `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt` (EC2Config only)
- `C:\ProgramData\Amazon\Ec2Config\Logs` (EC2Config only)
- `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\EC2Launch.log` (EC2Launch v1 only)
- `%ProgramData%\Amazon\EC2Launch\log\agent.log` (EC2Launch v2 only)

If you receive an error message during image preparation with Sysprep, the OS might not be reachable. To review the log files, you must stop the instance, attach its root volume to another healthy instance as a secondary volume, and then review the logs mentioned earlier on the secondary volume. For more information about the purpose of the log files by name, see Windows Setup-Related Log Files in the Microsoft documentation.

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

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

There are two options for resolving this issue:

Option 1

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

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

Option 2

1. Edit the relevant file, as follows:
   - Windows Server 2012 R2 and earlier – Edit the EC2Config answer file (`C:\Program Files \Amazon\Ec2ConfigService\sysprep2008.xml`).
   - Windows Server 2016 and 2019 – Edit the unattend.xml answer file (`C:\ProgramData \Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml`).
   - Windows Server 2022 – Edit the unattend.xml answer file (`C:\ProgramData\Amazon \EC2Launch\sysprep\unattend.xml`).
2. Change `<CopyProfile>true</CopyProfile>` to `<CopyProfile>false</ CopyProfile>`.
3. Run Sysprep again. Note that this configuration change will delete the built-in administrator user profile after Sysprep completes.

Use EC2Rescue for Windows Server

EC2Rescue for Windows Server is an easy-to-use tool that you run on an Amazon EC2 Windows Server instance to diagnose and troubleshoot possible problems. It is valuable for collecting log files and troubleshooting issues and also proactively searching for possible areas of concern. It can even examine Amazon EBS root volumes from other instances and collect relevant logs for troubleshooting Windows Server instances using that volume.

EC2Rescue for Windows Server has two different modules: a data collector module that collects data from all different sources, and an analyzer module that parses the data collected against a series of predefined rules to identify issues and provide suggestions.

The EC2Rescue for Windows Server tool only runs on Amazon EC2 instances running Windows Server 2008 R2 and later. When the tool starts, it checks whether it is running on an Amazon EC2 instance.

The `AWSSupport-ExecuteEC2Rescue` runbook uses the EC2Rescue tool to troubleshoot and, where possible, fix common connectivity issues with the specified EC2 instance. For more information, and to run this automation, see AWSSupport-ExecuteEC2Rescue.

> **Note**
>
> If you are using a Linux instance, see EC2Rescue for Linux.

Contents

- Use EC2Rescue for Windows Server GUI
- Use EC2Rescue for Windows Server with the command line
- Use EC2Rescue for Windows Server with Systems Manager Run Command

Use EC2Rescue for Windows Server GUI

EC2Rescue for Windows Server can perform the following analysis on **offline instances**:

Option	Description

Diagnose and Rescue	EC2Rescue for Windows Server can detect and address issues with the following service settings:
	• System Time
	  • **RealTimeisUniversal** - Detects whether the `RealTimeisUniversal` registry key is enabled. If disabled, Windows system time drifts when the timezone is set to a value other than UTC.
	• Windows Firewall
	  • **Domain networks** - Detects whether this Windows Firewall profile is enabled or disabled.
	  • **Private networks** - Detects whether this Windows Firewall profile is enabled or disabled.

Option	Description

	• **Guest or public networks** - Detects whether this Windows Firewall profile is enabled or disabled.
	
	• Remote Desktop
	  • **Service Start** - Detects whether the Remote Desktop service is enabled.
	  • **Remote Desktop Connections** - Detects whether this is enabled.
	  • **TCP Port** - Detects which port the Remote Desktop service is listening on.
	
	• EC2Config (Windows Server 2012 R2 and earlier)
	  • **Installation** - Detects which EC2Config version is installed.
	  • **Service Start** - Detects whether the EC2Config service is enabled.
	  • **Ec2SetPassword** - Generates a new administrator password.
	  • **Ec2HandleUserData** - Allows you to run a user data script on the next boot of the instance.
	
	• EC2Launch (Windows Server 2016 and later)
	  • **Installation** - Detects which EC2Launch version is installed.
	  • **Ec2SetPassword** - Generates a new administrator password.
	
	• Network Interface

Option	Description
	• **DHCP Service Startup** - Detects whether the DHCP service is enabled. • **Ethernet detail** - Displays informati on about the network driver version, if detected. • **DHCP on Ethernet** - Detects whether DHCP is enabled. • Disk signature status • **Signature on disk** and **Signature on Boot Configuration Database (BCD)** - Detects whether the disk signature and the BCD signature are the same. If the values are different, EC2Rescue attempts to overwrite the disk signature with the signature on BCD.
Restore	Perform one of the following actions: • **Last Known Good Configuration** - Attempts to boot the instance into the last known bootable state. • **Restore registry from backup** - Restores the registry from `\Windows\System32\config\RegBack` .
Capture Logs	Allows you to capture logs on the instance for analysis.

EC2Rescue for Windows Server can collect the following data from **active and offline instances**:

Item	Description
Event Log	Collects application, system, and EC2Config event logs.

Item	Description
Registry	Collects `SYSTEM` and `SOFTWARE` hives.
Windows Update Log	Collects log files generated by Windows Update. **Note** In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.
Sysprep Log	Collects log files generated by the Windows System Preparation tool.
Driver Setup Log	Collects Windows SetupAPI logs (`setupapi.dev.log` and `setupapi.setup.log`).
Boot Configuration	Collects `HKEY_LOCAL_MACHINE\BCD00000000` hive.
Memory Dump	Collects any memory dump files that exist on the instance.
EC2Config File	Collects log files generated by the EC2Config service.
EC2Launch File	Collects log files generated by the EC2Launch scripts.
SSM Agent File	Collects log files generated by SSM Agent and Patch Manager logs.
EC2 ElasticGPUs File	Collects event logs related to elastic GPUs.
ECS	Collects logs related to Amazon ECS.

Item	Description
CloudEndure	Collects log files related to CloudEndure Agent.

EC2Rescue for Windows Server can collect the following additional data from **active instances**:

Item	Description
System Information	Collects MSInfo32.
Group Policy Result	Collects a Group Policy report.

Analyze an offline instance

The **Offline Instance** option is useful for debugging boot issues with Windows instances.

To perform an action on an offline instance

1. From a working Windows Server instance, download the EC2Rescue for Windows Server tool and extract the files.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   This command will download the EC2Rescue .zip file to the desktop of the currently logged in user.
2. Stop the faulty instance, if it is not stopped already.
3. Detach the EBS root volume from the faulty instance and attach the volume to a working Windows instance that has EC2Rescue for Windows Server installed.
4. Run the EC2Rescue for Windows Server tool on the working instance and choose **Offline Instance**.

5. Select the disk of the newly mounted volume and choose **Next**.
6. Confirm the disk selection and choose **Yes**.
7. Choose the offline instance option to perform and choose **Next**.

The EC2Rescue for Windows Server tool scans the volume and collects troubleshooting information based on the selected log files.

Collect data from an active instance

You can collect logs and other data from an active instance.

To collect data from an active instance

1. Connect to your Windows instance.
2. Download the EC2Rescue for Windows Server tool to your Windows instance and extract the files.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   This command will download the EC2Rescue .zip file to the desktop of the currently logged in user.
3. Open the EC2Rescue for Windows Server application and accept the license agreement.
4. Choose **Next**, **Current instance**, **Capture logs**.
5. Select the data items to collect and choose **Collect...**. Read the warning and choose **Yes** to continue.
6. Choose a file name and location for the ZIP file and choose **Save**.
7. After EC2Rescue for Windows Server completes, choose **Open Containing Folder** to view the ZIP file.
8. Choose **Finish**.

Use EC2Rescue for Windows Server with the command line

The EC2Rescue for Windows Server command line interface (CLI) allows you to run an EC2Rescue for Windows Server plugin (referred as an "action") programmatically.

The EC2Rescue for Windows Server tool has two execution modes:

- **/online**—This allows you to take action on the instance that EC2Rescue for Windows Server is installed on, such as collect log files.
- **/offline:<device_id>**—This allows you to take action on the offline root volume that is attached to a separate Amazon EC2 Windows instance, on which you have installed EC2Rescue for Windows Server.

Download the EC2Rescue for Windows Server tool to your Windows instance and extract the files. You can view the help file using the following command:

```
EC2RescueCmd.exe /help
```

EC2Rescue for Windows Server can perform the following actions on an Amazon EC2 Windows instance:

- Collect action
- Rescue action
- Restore action

Collect action

> **Note**
>
> You can collect all logs, an entire log group, or an individual log within a group.

EC2Rescue for Windows Server can collect the following data from **active and offline instances**.

Log group	Available logs	Description
`all`		Collects all available logs.

Log group	Available logs	Description
`eventlog`	• `'Application'` • `'System'` • `'EC2ConfigService'`	Collects application, system, and EC2Config event logs.
`memory-dump`	• `'Memory Dump File'` • `'Mini Dump Files'`	Collects any memory dump files that exist on the instance.
`ec2config`	• `'Log Files'` • `'Configuration Files'`	Collects log files generated by the EC2Config service.
`ec2launch`	• `'Logs'` • `'Config'`	Collects log files generated by the EC2Launch scripts.
`ssm-agent`	• `'Log Files'` • `'Patch Baseline Logs'` • `'InstanceData'`	Collects log files generated by SSM Agent and Patch Manager logs.
`sysprep`	`'Log Files'`	Collects log files generated by the Windows System Preparation tool.
`driver-setup`	• `'SetupAPI Log Files'` • `'DPInst Log File'` • `'AWS PV Setup Log File'`	Collects Windows SetupAPI logs (`setupapi.dev.log` and `setupapi.setup.log`).
`registry`	• `'SYSTEM'` • `'SOFTWARE'` • `'BCD'`	Collects `SYSTEM` and `SOFTWARE` hives.
`egpu`	• `'Event Log'` • `'System Files'`	Collects event logs related to elastic GPUs.

Log group	Available logs	Description
`boot-config`	`'BCDEDIT Output'`	Collects `HKEY_LOCAL_MACHINE\BCD00000000` hive.
`windows-update`	`'Log Files'`	Collects log files generated by Windows Update. **Note** In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.
`cloudendure`	• `'Migrate Script Logs'` • `'Driver Logs'` • `'CloudEndure File List'`	Collects log files related to CloudEndure Agent.

EC2Rescue for Windows Server can collect the following additional data from **active instances**.

Log group	Available logs	Description
`system-info`	`'MSInfo32 Output'`	Collects `MSInfo32`.
`gpresult`	`'GPResult Output'`	Collects a Group Policy report.

The following are the available options:

- **/output:<outputFilePath>** - Required destination file path location to save collected log files in zip format.

- **/no-offline** - Optional attribute used in offline mode. Does not set the volume offline after completing the action.
- **/no-fix-signature** - Optional attribute used in offline mode. Does not fix a possible disk signature collision after completing the action.

Examples

The following are examples using the EC2Rescue for Windows Server CLI.

Online mode examples

Collect all available logs:

```
EC2RescueCmd /accepteula /online /collect:all /output:<outputFilePath>
```

Collect only a specific log group:

```
EC2RescueCmd /accepteula /online /collect:ec2config /output:<outputFilePath>
```

Collect individual logs within a log group:

```
EC2RescueCmd /accepteula /online /collect:'ec2config.Log Files,driver-setup.SetupAPI
 Log Files' /output:<outputFilePath>
```

Offline mode examples

Collect all available logs from an EBS volume. The volume is specified by the device_id value.

```
EC2RescueCmd /accepteula /offline:xvdf /collect:all /output:<outputFilePath>
```

Collect only a specific log group:

```
EC2RescueCmd /accepteula /offline:xvdf /collect:ec2config /output:<outputFilePath>
```

Rescue action

EC2Rescue for Windows Server can detect and address issues with the following service settings:

Service group	Available actions	Description
`all`		
`system-time`	`'RealTimeIsUniversal'`	System Time • **RealTimeisUniversal** - Detects whether the `RealTimeisUniversal` registry key is enabled. If disabled, Windows system time drifts when the timezone is set to a value other than UTC.
`firewall`	• `'Domain networks'` • `'Private networks'` • `'Guest or public networks'`	Windows Firewall • **Domain networks** - Detects whether this Windows Firewall profile is enabled or disabled. • **Private networks** - Detects whether this Windows Firewall profile is enabled or disabled. • **Guest or public networks** - Detects whether this Windows Firewall profile is enabled or disabled.
rdp	• `'Service Start'` • `'Remote Desktop Connections'` • `'TCP Port'`	Remote Desktop • **Service Start** - Detects whether the Remote Desktop service is enabled.

Service group	Available actions	Description
		• **Remote Desktop Connections** - Detects whether this is enabled. • **TCP Port** - Detects which port the Remote Desktop service is listening on.
`ec2config`	• `'Service Start'` • `'Ec2SetPassword'` • `'Ec2HandleUserData'`	EC2Config • **Service Start** - Detects whether the EC2Config service is enabled. • **Ec2SetPassword** - Generates a new administrator password. • **Ec2HandleUserData** - Allows you to run a user data script on the next boot of the instance.
`ec2launch`	`'Reset Administrator Password'`	Generates a new Windows administrator password.
`network`	`'DHCP Service Startup'`	Network Interface • **DHCP Service Startup** - Detects whether the DHCP service is enabled.

The following are the available options:

- **/level:<level>** - Optional attribute for the check level that the action should trigger. Allowed values are: `information`, `warning`, `error`, `all`. By default, it is set to `error`.
- **/check-only** - Optional attribute that generates a report but makes no modifications to the offline volume.

- **/no-offline** - Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature** - Optional attribute that does not fix a possible disk signature collision after completing the action.

Rescue examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Attempt to fix all identified issues on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:all
```

Attempt to fix all issues within a service group on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:firewall
```

Attempt to fix a specific item within a service group on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:rdp.'Service Start'
```

Specify multiple issues to attempt to fix on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:'system-
time.RealTimeIsUniversal,ec2config.Service Start'
```

Restore action

EC2Rescue for Windows Server can detect and address issues with the following service settings:

Service Group	Available Actions	Description
Restore Last Known Good Configuration	`lkgc`	**Last Known Good Configuration** - Attempts to boot the instance into the last known bootable state.

Service Group	Available Actions	Description
Restore Windows registry from latest backup	`regback`	**Restore registry from backup** - Restores the registry from `\Windows\System32\config\RegBack` .

The following are the available options:

- **/no-offline**—Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature**—Optional attribute that does not fix a possible disk signature collision after completing the action.

Restore examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Restore last known good configuration on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

Restore the last Windows registry backup on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

Use EC2Rescue for Windows Server with Systems Manager Run Command

AWS Support provides you with a Systems Manager Run Command document to interface with your Systems Manager-enabled instance to run EC2Rescue for Windows Server. The Run Command document is called `AWSSupport-RunEC2RescueForWindowsTool`.

This Systems Manager Run Command document performs the following tasks:

- Downloads and verifies EC2Rescue for Windows Server.
- Imports a PowerShell module to ease your interaction with the tool.
- Runs EC2RescueCmd with the provided command and parameters.

The Systems Manager Run Command document accepts three parameters:

- **Command**—The EC2Rescue for Windows Server action. The current allowed values are:
  - **ResetAccess**—Resets the local Administrator password. The local Administrator password of the current instance will be reset and the randomly generated password will be securely stored in Parameter Store as `/EC2Rescue/Password/<INSTANCE_ID>`. If you select this action and provide no parameters, passwords are encrypted automatically with the default KMS key. Optionally, you can specify a KMS key ID in Parameters to encrypt the password with your own key.
  - **CollectLogs**—Runs EC2Rescue for Windows Server with the `/collect:all` action. If you select this action, `Parameters` must include an Amazon S3 bucket name to upload the logs to.
  - **FixAll**—Runs EC2Rescue for Windows Server with the `/rescue:all` action. If you select this action, `Parameters` must include the block device name to rescue.
- **Parameters**—The PowerShell parameters to pass for the specified command.

> **Note**
>
> In order for the **ResetAccess** action to work, your Amazon EC2 instance needs to have the following policy attached in order to write the encrypted password to Parameter Store. Please wait a few minutes before attempting to reset the password of an instance after you have attached this policy to the related IAM role.
> Using the default KMS key:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
```

```
      "Resource": [
        "arn:aws:ssm:region:account_id:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    }
  ]
}
```

Using a custom KMS key:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:region:account_id:parameter/EC2Rescue/
Passwords/<instanceid>"
        ]
      },
      {
      "Effect": "Allow",
      "Action": [
      "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:region:account_id:key/<kmskeyid>"
      ]
    }
  ]
}
```

The following procedure describes how to view the JSON for this document in the Amazon EC2 console.

To view the JSON for the Systems Manager Run Command document

1. Open the Systems Manager console at https://console.aws.amazon.com/systems-manager/home.
2. In the navigation pane, expand **Shared Services** and choose **Documents**.
3. In the search bar, set **Owner** as **Owned by Me or Amazon** and set the **Document name prefix** to `AWSSupport-RunEC2RescueForWindowsTool`.
4. Select the `AWSSupport-RunEC2RescueForWindowsTool` document, choose **Contents**, and then view the JSON.

Examples

Here are some examples on how to use the Systems Manager Run Command document to run EC2Rescue for Windows Server, using the AWS CLI. For more information about sending commands with the AWS CLI, see the AWS CLI Command Reference.

Attempt to fix all identified issues on an offline root volume

Attempt to fix all identified issues on an offline root volume attached to an Amazon EC2 Windows instance:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --comment "EC2Rescue offline volume xvdf" --parameters
 "Command=FixAll, Parameters='xvdf'" --output text
```

Collect logs from the current Amazon EC2 Windows instance

Collect all logs from the current online Amazon EC2 Windows instance and upload them to an Amazon S3 bucket:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --comment "EC2Rescue online log collection to S3" --
parameters "Command=CollectLogs, Parameters='YOURS3BUCKETNAME'" --output text
```

Collect logs from an offline Amazon EC2 Windows instance volume

Collect all logs from an offline volume attached to an Amazon EC2 Windows instance and upload them to Amazon S3 with a presigned URL:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --comment "EC2Rescue offline log collection to S3"
 --parameters "Command=CollectLogs, Parameters=\"-Offline -BlockDeviceName xvdf -
S3PreSignedUrl 'YOURS3PRESIGNEDURL'\"" --output text
```

Reset the local Administrator password

The following examples show methods you can use to reset the local Administrator password. The output provides a link to Parameter Store, where you can find the randomly generated secure password you can then use to RDP to your Amazon EC2 Windows instance as the local Administrator.

Reset the local Administrator password of an online instance using the default AWS KMS key alias/aws/ssm:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --comment "EC2Rescue online password reset" --parameters
 "Command=ResetAccess" --output text
```

Reset the local Administrator password of an online instance using a KMS key:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --comment "EC2Rescue online password reset" --parameters
 "Command=ResetAccess, Parameters=a133dc3c-a2g4-4fc6-a873-6c0720104bf0" --output text
```

> **Note**
>
> In this example, the KMS key is `a133dc3c-a2g4-4fc6-a873-6c0720104bf0`.

EC2 Serial Console for Windows instances

With the EC2 serial console, you have access to your Amazon EC2 instance's serial port, which you can use to troubleshoot boot, network configuration, and other issues. The serial console does not require your instance to have any networking capabilities. With the serial console, you can enter commands to an instance as if your keyboard and monitor are directly attached to the instance's serial port. The serial console session lasts during instance reboot and stop. During reboot, you can view all of the boot messages from the start.

Access to the serial console is not available by default. Your organization must grant account access to the serial console and configure IAM policies to grant your users access to the serial console. Serial console access can be controlled at a granular level by using instance IDs, resource tags, and other IAM levers. For more information, see Configure access to the EC2 Serial Console.

The serial console can be accessed by using the EC2 console or the AWS CLI.

The serial console is available at no additional cost.

If you are using a Linux instance, see EC2 Serial Console for Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

Topics

- Prerequisites
- Configure access to the EC2 Serial Console
- Connect to the EC2 Serial Console
- Disconnect from the EC2 Serial Console
- Troubleshoot your Windows 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 for all virtualized instances built on the Nitro System.

Not supported on bare metal instances.

Grant access

You must complete the configuration tasks to grant access to the EC2 Serial Console. For more information, see Configure access to the EC2 Serial Console.

Support for browser-based client

To connect to the serial console using the browser-based client, your browser must support WebSocket. If your browser does not support WebSocket, connect to the serial console using your own key and an SSH client.

Instance state

Must be `running`.

You can't connect to the serial console if the instance is in the `pending`, `stopping`, `stopped`, `shutting-down`, or `terminated` state.

For more information about the instance states, see 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 Windows instance via the serial console, you can use the Special Admin Console (SAC). Before you can use SAC, you must first enable SAC and the boot menu on every instance on which you'll use it.

For the instructions to configure your chosen troubleshooting tool on Linux, see Configure your chosen troubleshooting tool in the *Amazon EC2 User Guide for Linux Instances*.

Enable SAC and the boot menu

> **Note**
>
> If you enable SAC on an instance, the EC2 services that rely on password retrieval will not work from the Amazon EC2 console. Windows on Amazon EC2 launch agents (EC2Config, EC2Launch v1, and EC2Launch v2) rely on the serial console to execute various tasks. These tasks do not perform successfully when you enable SAC on an instance. For more information about Windows on Amazon EC2 launch agents, see Configure your Windows instance. If you enable SAC, you can disable it later. For more information, see Disable SAC and the boot menu.

Use one of the following methods to enable SAC and the boot menu on an instance.

PowerShell

To enable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from an elevated PowerShell command line.
2. Enable SAC.

```
bcdedit /ems '{current}' on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. Enable the boot menu.

```
bcdedit /set '{bootmgr}' displaybootmenu yes
bcdedit /set '{bootmgr}' timeout 15
bcdedit /set '{bootmgr}' bootems yes
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Command prompt

To enable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from the command prompt.
2. Enable SAC.

```
bcdedit /ems {current} on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. Enable the boot menu.

```
bcdedit /set {bootmgr} displaybootmenu yes
bcdedit /set {bootmgr} timeout 15
bcdedit /set {bootmgr} bootems yes
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Configure access to the EC2 Serial Console

To configure access to the serial console, you must grant serial console access at the account level and then configure IAM policies to grant access to your users.

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

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.

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*.

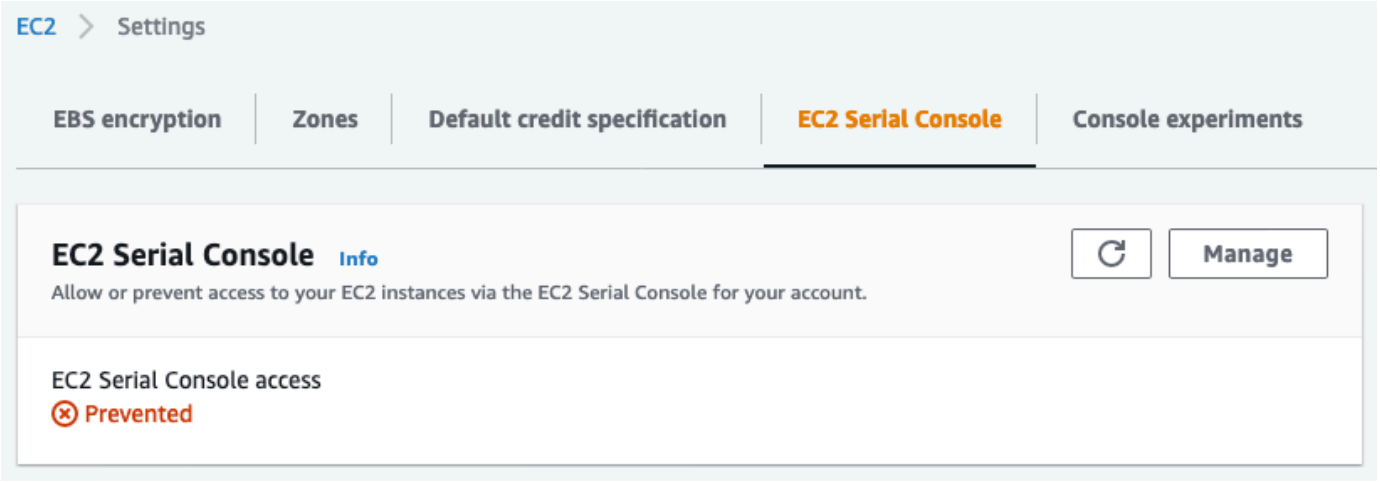
View account access status to the serial console

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}"
                }
```

```
            }
        }
    ]
}
```

Connect to the EC2 Serial Console

You can connect to the serial console of your EC2 instance by using the Amazon EC2 console or via SSH. After connecting to the serial console, you can use it for troubleshooting boot, network configuration, and other issues. For more information about troubleshooting, see Troubleshoot your Windows instance using the EC2 Serial Console.

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 Windows: COM1
- 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.
   - **Reboot your instance.** You can reboot your instance by using the EC2 console or the AWS CLI. For more information, see Reboot your instance.

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.

```
C:\> 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.

```
C:\> 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.
   - **Reboot your instance.** You can reboot your instance by using the EC2 console or the AWS CLI. For more information, see Reboot your instance.

EC2 Serial Console endpoints and fingerprints

The following are the service endpoints and fingerprints for EC2 Serial Console. To connect programmatically to an instance's serial console, you use an EC2 Serial Console endpoint. The EC2 Serial Console endpoints and fingerprints are unique for each AWS Region.

Region Name	Region	Endpoint	Fingerprint
US East (Ohio)	us-east-2	serial-console.ec2-instance-connect.us-east-2.aws	SHA256:EhwPkTzRtTY7TRSzz26XbB0/HvV9jRM7mCZN0xw/d/0
US East (N. Virginia)	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
US West (N. California)	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OHldlcMET8u7QLSX3jmRTRAPFHVtqbyoLZBMUCqiH3Y
US West (Oregon)	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EMCIe23TqKaBI6yGHain

Region Name	Region	Endpoint	Fingerprint
			qZcMwqNkD hhAVHa1O2JxVUc
Africa (Cape Town)	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RM WWZ2fVePe JUqzjO5jL2KIgXsczo Hlz21Ed00biiWI
Asia Pacific (Hong Kong)	ap-east-1	ec2-serial-console.ap-east-1.api.aws	SHA256:T0 Q1lpiXxChoZHplnAkj bP7tkm2xXViC9bJFsj Ynifk
Asia Pacific (Hyderabad)	ap-south-2	ec2-serial-console.ap-south-2.api.aws	SHA256:WJ gPBSwV4/shN +OPITValoewAuYj1 5DVW845JEhDKRs
Asia Pacific (Jakarta)	ap-southeast-3	ec2-serial-console.ap-southeast-3.api.aws	SHA256:5ZwgrCh+lfn s32XITqL/4O0zIfbx4 bZgsYFqy3o8mIk
Asia Pacific (Melbourne)	ap-southeast-4	ec2-serial-console.ap-southeast-4.api.aws	SHA256:Av aq27hFgLvjn5gTSShZ 0oV7h90p0 GG46wfOeT6ZJvM
Asia Pacific (Mumbai)	ap-south-1	serial-console.ec2-instance-connect.ap-south-1.aws	SHA256:oB LXcYmklqHHEbliARxE gH8IsO51rezTPiSM35 BsU40
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+FrcZM2dSrqrI
Asia Pacific (Singapore)	ap-southeast-1	serial-console.ec2-instance-connect.ap-southeast-1.aws	SHA256:PLFNn7WnCQDHx3qmwLu1Gy/O8TUX7LQgZuaC6L45CoY
Asia Pacific (Sydney)	ap-southeast-2	serial-console.ec2-instance-connect.ap-southeast-2.aws	SHA256:yFvMwUK9lEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4
Asia Pacific (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.api.amazonwebservices.com.cn	SHA256:TdgrNZkiQOdVfYEBUhO4SzUA09VWI5rYOZGTogpwmiM
Europe (Frankfurt)	eu-central-1	serial-console.ec2-instance-connect.eu-central-1.aws	SHA256:aCMFS/yIcOdOlkXvOl8AmZ1Toe+bBnrJJ3Fy0k0De2c
Europe (Ireland)	eu-west-1	serial-console.ec2-instance-connect.eu-west-1.aws	SHA256:h2AaGAWO4Hathhtm6ezs3Bj7udgUxi2qTrHjZAwCW6E
Europe (London)	eu-west-2	serial-console.ec2-instance-connect.eu-west-2.aws	SHA256:a69rd5CE/AEG4Amm53I6lkD1ZPvS/BCV3tTPW2RnJg8
Europe (Milan)	eu-south-1	ec2-serial-console.eu-south-1.api.aws	SHA256:lC0kOVJnpgFyBVrxn0A7n99ecLbXSX95cuuS7X7QK30
Europe (Paris)	eu-west-3	serial-console.ec2-instance-connect.eu-west-3.aws	SHA256:q8ldnAf9pymeNe8BnFVngY3RPAr/kxswJUzfrlxeEWs

Region Name	Region	Endpoint	Fingerprint
Europe (Spain)	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:GoCW2DFRlu669QNxqFxEcsR6fZUz/4F4n7T45ZcwoEc
Europe (Stockholm)	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tkGFFUVUDvocDiGSS3Cu8Gdl6w2uI32EPNpKFKLwX84
Europe (Zurich)	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8Ppx2mBMf6WdCw0NUlzKfwM4/IfRz4OaXFutQXWp6mk
Israel (Tel Aviv)	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR6q8v6kNNPi8+QSFQ4dj5dimNmZPTgwgsM1SNvtYyU
Middle East (Bahrain)	me-south-1	ec2-serial-console.me-south-1.api.aws	SHA256:nPjLLKHu2QnLdUq2kVArsoK5xvPJOMRJKCBzCDqC3k8
Middle East (UAE)	me-central-1	ec2-serial-console.me-central-1.api.aws	SHA256:zpb5duKiBZ+l0dFwPeyykB4MPBYhI/XzXNeFSDKBvLE

Region Name	Region	Endpoint	Fingerprint
South America (São Paulo)	sa-east-1	ec2-serial-console.sa-east-1.api.aws	SHA256:rd2+/32Ognjew1yVIemENaQzC+Botbih62OqAPDq1dI
AWS GovCloud (US-East)	us-gov-east-1	serial-console.ec2-instance-connect.us-gov-east-1.amazonaws.com	SHA256:tIwe19GWsoyLClrtvu38YEEh+DHIkqnDcZnmtebvF28
AWS GovCloud (US-West)	us-gov-west-1	serial-console.ec2-instance-connect.us-gov-west-1.amazonaws.com	SHA256:kfOFRWLaOZfB+utbd3bRf8OlPf8nGO2YZLqXZiIw5DQ

Disconnect from the EC2 Serial Console

If you no longer need to be connected to your instance's EC2 Serial Console, you can disconnect from it. When you disconnect from the serial console, any shell session running on the instance will continue to run. If you want to end the shell session, you'll need to end it before disconnecting from the serial console.

Considerations

- The serial console connection typically lasts for 1 hour unless you disconnect from it. However, during system maintenance, Amazon EC2 will disconnect the serial console session.
- It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.

The way to disconnect from the serial console depends on the client.

Browser-based client

To disconnect from the serial console, close the serial console in-browser terminal window.

Standard OpenSSH client

To disconnect from the serial console, use the following command to close the SSH connection. This command must be run immediately following a new line.

```
C:\> ~.
```

The command that you use for closing an SSH connection might be different depending on the SSH client that you're using.

Troubleshoot your Windows instance using the EC2 Serial Console

By using EC2 Serial Console, you can troubleshoot boot, network configuration, and other issues by connecting to your instance's serial port.

Before commencing, be sure to check the prerequisites.

Topics

- Use SAC to troubleshoot your Windows instance

For information about troubleshooting your Linux instance, see Troubleshoot your Linux instance using the EC2 Serial Console in the *Amazon EC2 User Guide for Linux Instances*.

Use SAC to troubleshoot your Windows instance

The Special Admin Console (SAC) capability of Windows provides a way to troubleshoot a Windows instance. By connecting to the instance's serial console and using SAC, you can interrupt the boot process and boot Windows in safe mode.

Before you can use SAC, make sure you have completed the prerequisites, which include granting access to the serial console and enabling SAC and the boot menu.

> **Note**
>
> If you enable SAC on an instance, the EC2 services that rely on password retrieval will not work from the Amazon EC2 console. Windows on Amazon EC2 launch agents (EC2Config, EC2Launch v1, and EC2Launch v2) rely on the serial console to execute various tasks. These tasks do not perform successfully when you enable SAC on an instance. For more information about Windows on Amazon EC2 launch agents, see Configure your Windows instance. If you enable SAC, you can disable it later. For more information, see Disable SAC and the boot menu.

Topics

- Use SAC
- Use the boot menu
- Disable SAC and the boot menu

Use SAC

To use SAC

1. Connect to the serial console.

   If SAC is enabled on the instance, the serial console displays the SAC> prompt.

```
Computer is booting, SAC started and initialized.

Use the "ch -?" command for information about using channels.
Use the "?" command for general help.


SAC>?
EVENT: The CMD command is now available.
SAC
```

2. To display the SAC commands, enter **?**, and then press **Enter**.

   Expected output

```
SAC>?
ch                    Channel management commands.  Use ch -? for more help.
cmd                   Create a Command Prompt channel.
d                     Dump the current kernel log.
f                     Toggle detailed or abbreviated tlist info.
? or help             Display this list.
i                     List all IP network numbers and their IP addresses.
i <#> <ip> <subnet> <gateway> Set IPv4 addr., subnet and gateway.
id                    Display the computer identification information.
k <pid>               Kill the given process.
l <pid>               Lower the priority of a process to the lowest possible.
lock                  Lock access to Command Prompt channels.
m <pid> <MB-allow>    Limit the memory usage of a process to <MB-allow>.
p                     Toggle paging the display.
r <pid>               Raise the priority of a process by one.
s                     Display the current time and date (24 hour clock used).
s mm/dd/yyyy hh:mm    Set the current time and date (24 hour clock used).
t                     Tlist.
restart               Restart the system immediately.
shutdown              Shutdown the system immediately.
crashdump             Crash the system. You must have crash dump enabled.
```

3. To create a command prompt channel (such as `cmd0001` or `cmd0002`), enter **`cmd`**, and then press **Enter**.
4. To view the command prompt channel, press **ESC**, and then press **TAB**.

   Expected output

```
Name:                 Cmd0001
Description:          Command
Type:                 VT-UTF8
Channel GUID:         ef9f20a0-1287-11eb-82b0-0e4ba51872e5
Application Type GUID: 63d02271-8aa4-11d5-bccf-00b0d014a2d0

Press <esc><tab> for next channel.
Press <esc><tab>0 to return to the SAC channel.
Use any other key to view this channel.
```

5. To switch channels, press **ESC+TAB+channel number** together. For example, to switch to the `cmd0002` channel (if it has been created), press **ESC+TAB+2**.
6. Enter the credentials required by the command prompt channel.

```
Please enter login credentials.
Username: Administrator
Domain  : .
Password: *******
```

   The command prompt is the same full-featured command shell that you get on a desktop, but with the exception that it does not allow the reading of characters that were already output.

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>diskpart

Microsoft DiskPart version 10.0.17763.1

Copyright (C) Microsoft Corporation.
On computer: EC2AMAZ-ASR4SAI

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online           46 GB    46 GB

DISKPART>
```

PowerShell can also be used from the command prompt.

Note that you might need to set the progress preference to silent mode.

```
PS C:\Windows\system32> $ProgressPreference="SilentlyContinue"
PS C:\Windows\system32> $computerInfo = Get-ComputerInfo
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Name
Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Description
Intel64 Family 6 Model 85 Stepping 4
PS C:\Windows\system32> _
```

Use the boot menu

If the instance has the boot menu enabled and is restarted after connecting via SSH, you should see the boot menu, as follows.

```
                         Windows Boot Manager

Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)

    Windows Server [EMS Enabled]                                    >




To specify an advanced option for this choice, press F8.



Tools:

     Windows Memory Diagnostic



 ENTER=Choose                    TAB=Menu                    ESC=Cancel
```

Boot menu commands

ENTER

Starts the selected entry of the operating system.

TAB

Switches to the Tools menu.

ESC

Cancels and restarts the instance.

ESC followed by 8

Equivalent to pressing **F8**. Shows advanced options for the selected item.

ESC key + left arrow

Goes back to the initial boot menu.

> **Note**
>
> The ESC key alone does not take you back to the main menu because Windows is waiting to see if an escape sequence is in progress.

```
                         Advanced Boot Options

Choose Advanced Options for: Windows Server
(Use the arrow keys to highlight your choice.)

   Repair Your Computer

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video
    Last Known Good Configuration (advanced)
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement
    Disable Early Launch Anti-Malware Driver

    Start Windows Normally

Description: View a list of system recovery tools you can use to repair
             startup problems, run diagnostics, or restore your system.

 ENTER=Choose                                                ESC=Cancel
```

Disable SAC and the boot menu

If you enable SAC and the boot menu, you can disable these features later.

Use one of the following methods to disable SAC and the boot menu on an instance.

PowerShell

To disable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from an elevated PowerShell command line.
2. First disable the boot menu by changing the value to no.

```
bcdedit /set '{bootmgr}' displaybootmenu no
```

3. Then disable SAC by changing the value to `off`.

```
bcdedit /ems '{current}' off
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Command prompt

To disable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from the command prompt.
2. First disable the boot menu by changing the value to no.

```
bcdedit /set {bootmgr} displaybootmenu no
```

3. Then disable SAC by changing the value to `off`.

```
bcdedit /ems {current} off
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Send a diagnostic interrupt (for advanced users)

> **Warning**
>
> Diagnostic interrupts are intended for use by advanced users. Incorrect usage could negatively impact your instance. Sending a diagnostic interrupt to an instance could trigger an instance to crash and reboot, which could lead to the loss of data.

You can send a diagnostic interrupt to an unreachable or unresponsive Windows instance to manually trigger a *stop error*. Stop errors are commonly referred to as *blue screen errors*.

In general, Windows operating systems crash and reboot when a stop error occurs, but the specific behavior depends on its configuration. A stop error can also cause the operating system to write debugging information, such as a kernel memory dump, to a file. You can then use this information to conduct root cause analysis to debug the instance.

The memory dump data is generated locally by the operating system on the instance itself.

Before sending a diagnostic interrupt to your instance, we recommend that you consult the documentation for your operating system and then make the necessary configuration changes.

Contents

- Supported instance types
- 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 Nitro System and AWS Graviton.

Prerequisites

Before using a diagnostic interrupt, you should configure your instance's operating system to perform the actions you need when a stop error occurs.

To configure Windows to generate a memory dump when a stop error occurs

1. Connect to your instance.
2. Open the **Control Panel** and choose **System**, **Advanced system settings**.
3. In the **System Properties** dialog box, choose the **Advanced** tab.
4. In the **Startup and Recovery** section, choose **Settings...**.
5. In the **System failure** section, configure the settings as needed, and then choose **OK**.

For more information about configuring Windows stop errors, see Overview of memory dump file options for Windows.

Send a diagnostic interrupt

After you have completed the necessary configuration changes, you can send a diagnostic interrupt to your instance using the AWS CLI or Amazon EC2 API.

To send a diagnostic interrupt to your instance (AWS CLI)

Use the send-diagnostic-interrupt command and specify the instance ID.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

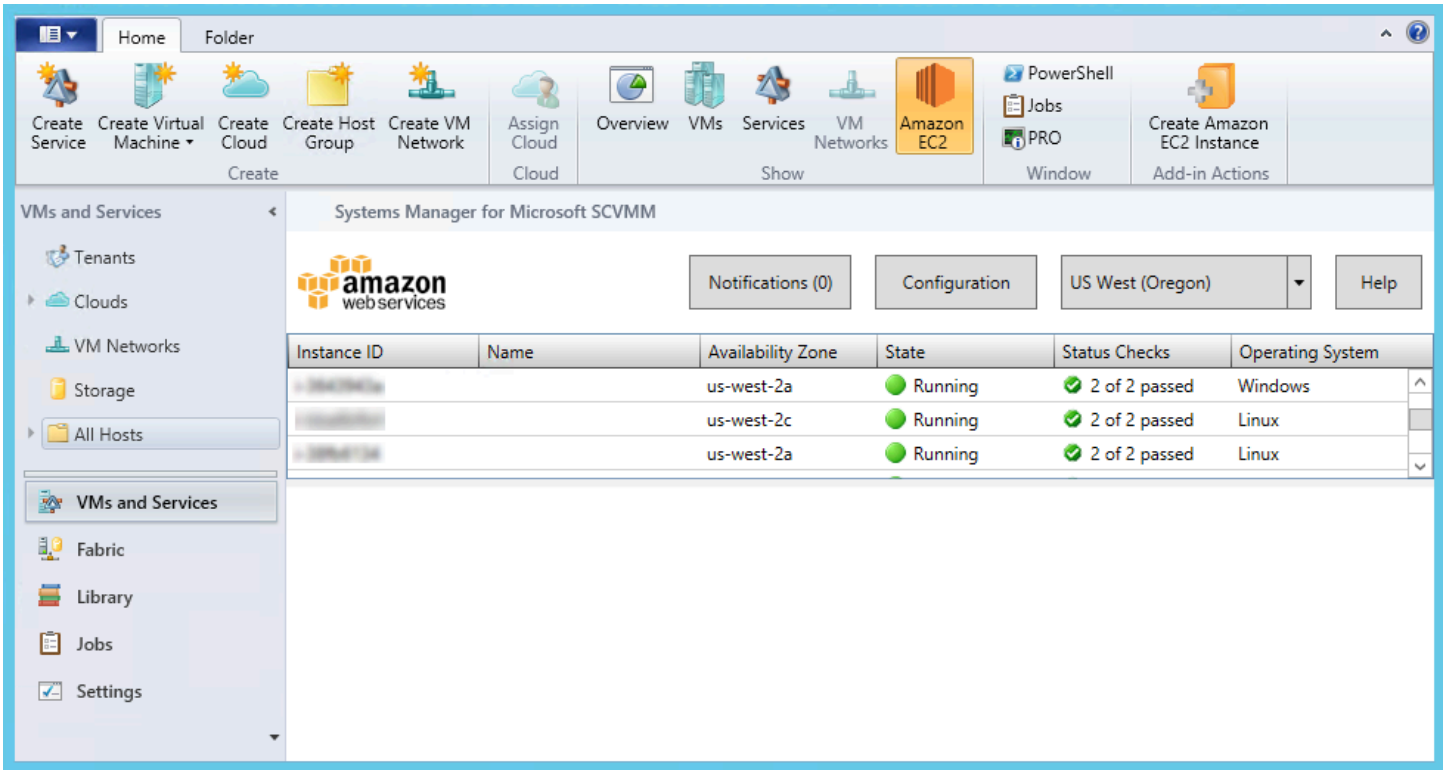
To send a diagnostic interrupt to your instance (AWS Tools for Windows PowerShell)

Use the Send-EC2DiagnosticInterrupt cmdlt and specify the instance ID.

```
PS C:\> Send-EC2DiagnosticInterrupt -InstanceId i-1234567890abcdef0
```

AWS Systems Manager for Microsoft System Center VMM

AWS Systems Manager for Microsoft System Center Virtual Machine Manager (SCVMM) provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. It is implemented as an add-in for the VMM console. For more information, see AWS Add-ins for Microsoft System Center.



Features

- Administrators can grant permissions to users so that they can manage EC2 instances from SCVMM.
- Users can launch, view, reboot, stop, start, and terminate instances, if they have the required permissions.
- Users can get the passwords for their Windows instances and connect to them using RDP.
- Users can get the public DNS names for their Linux instances and connect to them using SSH.
- Users can import their Hyper-V Windows virtual machines from SCVMM to Amazon EC2.

Limitations

- Users must have an account that they can use to log in to SCVMM.
- You can't import Linux virtual machines from SCVMM to Amazon EC2.
- This is not a comprehensive tool for creating and managing AWS resources. The add-in enables SCVMM users to get started quickly with the basic tasks for managing their EC2 instances. Future releases might support managing additional AWS resources.

Requirements

- An AWS account
- Microsoft System Center VMM 2012 R2 or System Center VMM 2012 SP1 with the latest update roll-up

Get started

To get started, see the following documentation:

- Setting Up
- Managing EC2 Instances
- Troubleshooting

Set up AWS Systems Manager for Microsoft SCVMM

When you set up AWS Systems Manager, users in your organization can access your AWS resources. The process involves creating accounts, deploying the add-in, and providing your credentials.

Tasks

- Sign up for AWS
- Set up access for users
- Deploy the add-in
- Provide your AWS credentials

Sign up for AWS

When you sign up for Amazon Web Services, your AWS account is automatically signed up for all services in AWS. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, see Sign up for an AWS account for instructions on how to create one.

Set up access for users

The first time that you use Systems Manager, you must provide AWS credentials. To enable multiple users to access the same AWS account using unique credentials and permissions, create a user for each user. You can create one or more groups with policies that grant permissions to perform limited tasks. Then you can create one or more users, and add each user to the appropriate group.

To create an Administrators group

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. In the **Group Name** box, specify **`Administrators`** and then choose **Next Step**.
4. On the **Attach Policy** page, select the **AdministratorAccess** AWS managed policy.
5. Choose **Next Step** and then choose **Create Group**.

To create a group with limited access to Amazon EC2

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. In the **Group Name** box, specify a meaningful name for the group and then choose **Next Step**.
4. On the **Attach Policy** page, do not select an AWS managed policy — choose **Next Step**, and then choose **Create Group**.
5. Choose the name of the group you've just created. On the **Permissions** tab, choose **Inline Policies**, and then **click here**.
6. Select the **Custom Policy** radio button and then choose **Select**.

7. Enter a name for the policy and a policy document that grants limited access to Amazon EC2, and then choose **Apply Policy**. For example, you can specify one of the following custom policies.

Grant users in this group permission to view information about EC2 instances only

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "iam:ListInstanceProfiles"
      ],
      "Resource": "*"
    }
  ]
}
```

Grant users in this group permission to perform all operations on EC2 instances that are supported by the add-in

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListInstanceProfiles",
        "ec2:Describe*", "ec2:CreateKeyPair",
        "ec2:CreateTags", "ec2:DeleteTags",
        "ec2:RunInstances", "ec2:GetPasswordData",
        "ec2:RebootInstances", "ec2:StartInstances",
        "ec2:StopInstances", "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

Grant users in this group permission to import a VM to Amazon EC2

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets", "s3:CreateBucket",
        "s3:DeleteBucket", "s3:DeleteObject",
        "s3:GetBucketLocation", "s3:GetObject",
        "s3:ListBucket", "s3:PutObject",
        "ec2:DescribeTags", "ec2:CancelConversionTask",
        "ec2:DescribeConversionTasks", "ec2:DescribeInstanceAttribute",
        "ec2:CreateImage", "ec2:AttachVolume",
        "ec2:ImportInstance", "ec2:ImportVolume",
        "dynamodb:DescribeTable", "dynamodb:CreateTable",
        "dynamodb:Scan", "dynamodb:PutItem", "dynamodb:UpdateItem"
      ],
      "Resource": "*"
    }
  ]
}
```

To create a user, get the user's AWS credentials, and grant the user permissions

Users need programmatic access if they want to interact with AWS outside of the AWS Management Console. The way to grant programmatic access depends on the type of user that's accessing AWS.

To grant users programmatic access, choose one of the following options.

Which user needs programmatic access?	To	By
Workforce identity (Users managed in IAM Identity Center)	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Configuring the AWS CLI to use AWS IAM

Which user needs programmatic access?	To	By
		Identity Center in the *AWS Command Line Interface User Guide*. • For AWS SDKs, tools, and AWS APIs, see IAM Identity Center authentication in the *AWS SDKs and Tools Reference Guide*.
IAM	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions in Using temporary credentials with AWS resources in the *IAM User Guide*.
IAM	(Not recommended) Use long-term credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Authenticating using IAM user credentials in the *AWS Command Line Interface User Guide*. • For AWS SDKs and tools, see Authenticate using long-term credentials in the *AWS SDKs and Tools Reference Guide*. • For AWS APIs, see Managing access keys for IAM users in the *IAM User Guide*.

Deploy the add-in

Add-ins for System Center VMM are distributed as `.zip` files. To deploy the add-in, use the following procedure.

To deploy the add-in

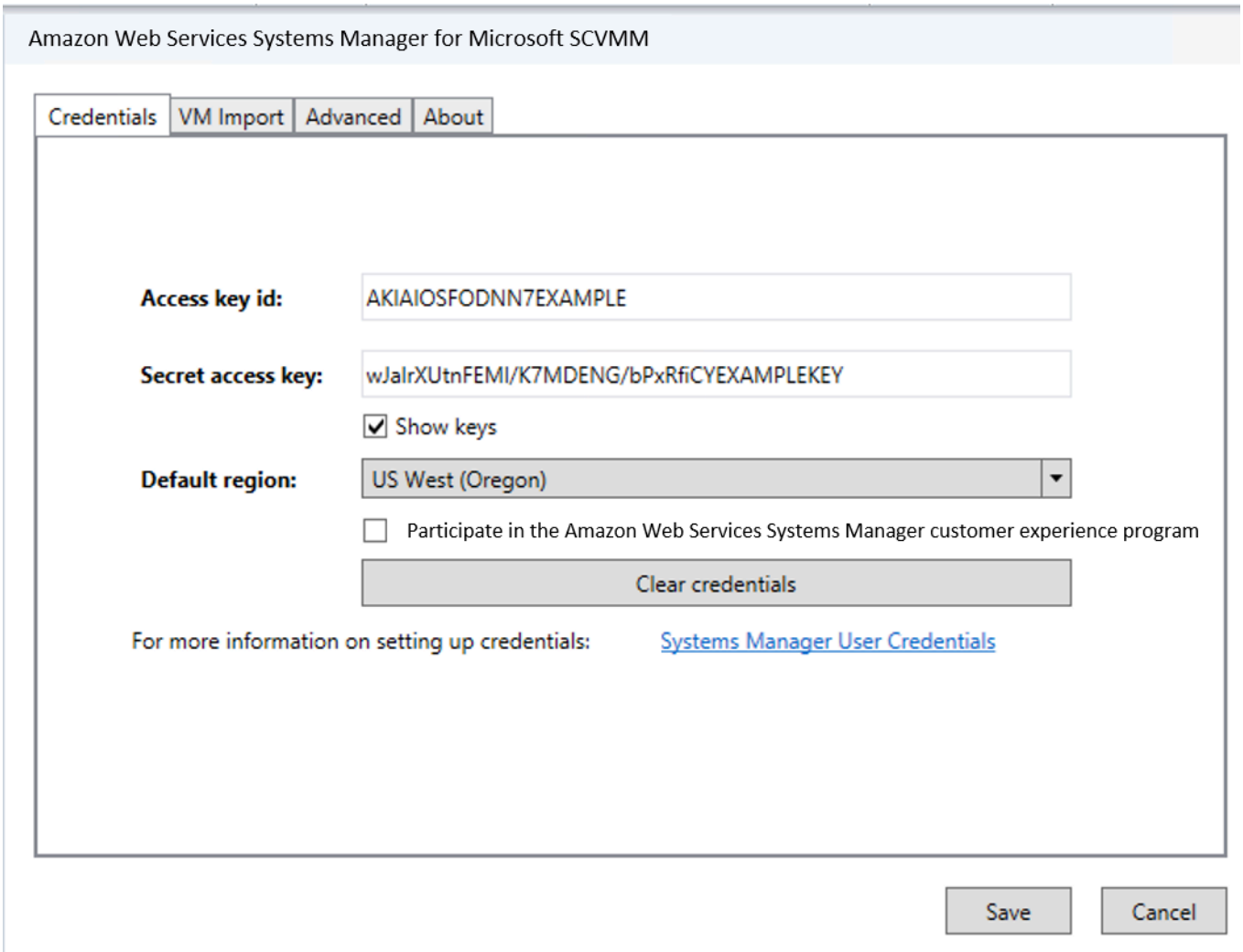
1. From your instance, go to AWS Systems Manager for Microsoft System Center Virtual Machine Manager and click **SCVMM**. Save the `aws-systems-manager-1.5.zip` file to your instance.
2. Open the VMM console.
3. In the navigation pane, click **Settings** and then click **Console Add-Ins**.
4. On the ribbon, click **Import Console Add-in**.
5. On the **Select an Add-in** page, click **Browse** and select the `aws-systems-manager-1.5.zip` file for the add-in that you downloaded.
6. Ignore any warnings that there are assemblies in the add-in that are not signed by a trusted authority. Select **Continue installing this add-in anyway** and then click **Next**.
7. On the **Summary** page, click **Finish**.
8. When the add-in is imported, the status of the job is `Completed`. You can close the **Jobs** window.

Provide your AWS credentials

When you use the Systems Manager for the first time, you must provide your AWS credentials. Your access keys identify you to AWS. There are two types of access keys: access key IDs (for example, AKIAIOSFODNN7EXAMPLE) and secret access keys (for example, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY). You should have stored your access keys in a safe place when you received them.

To provide your AWS credentials

1. Open the VMM console.
2. In the navigation pane, click **VMs and Services**.
3. On the ribbon, click **Amazon EC2**.
4. On the **Credentials** tab, specify your AWS credentials, select a default region, and then click **Save**.



To change these credentials at any time, click **Configuration**.

Manage EC2 instances using AWS Systems Manager for Microsoft SCVMM

After you log in to the Systems Manager console using your AWS credentials, you can manage your EC2 instances.

Tasks

- Create an EC2 Instance
- View your instances
- Connect to your instance

- Reboot your instance
- Stop your instance
- Start your instance
- Terminate your instance

Create an EC2 Instance

The permissions that you've been granted by your administrator determine whether you can create instances.

Prerequisites

- A virtual private cloud (VPC) with a subnet in the Availability Zone where you'll launch the instance. For more information about creating a VPC, see the Amazon VPC Getting Started Guide.

To create an EC2 instance

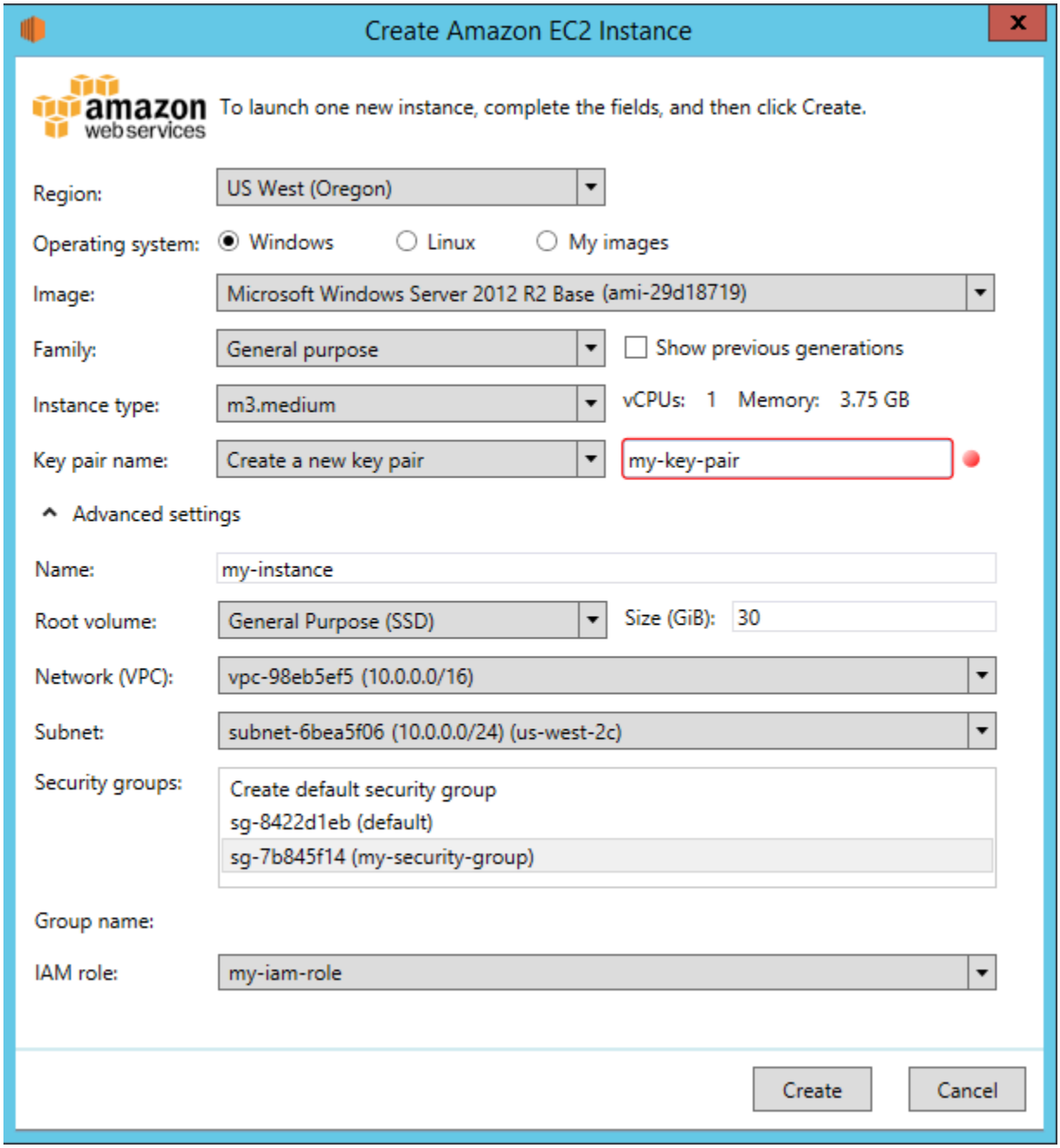
1. Open SCVMM.
2. On the ribbon, click **Create Amazon EC2 Instance**.
3. Complete the **Create Amazon EC2 Instance** dialog box as follows:
   a. Select a Region for your instance. By default, we select the Region that you configured as your default Region.
   b. Select a template (known as an AMI) for your instance. To use an AMI provided by Amazon, select **Windows** or **Linux** and then select an AMI from **Image**. To use an AMI that you created, select **My images** and then select the AMI from **Image**.
   c. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations.** For more information, see Amazon EC2 Instances and Previous Generation Instances.
   d. Create or select a key pair. To create a key pair, select `Create a new key pair` from **Key pair name** and enter a name for the key pair in the highlighted field (for example, `my-key-pair`).
   e. (Optional) Under **Advanced settings**, specify a display name for the instance.

f. (Optional) Under **Advanced settings**, select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this Region using the Amazon VPC console.

g. (Optional) Under **Advanced settings**, select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.

h. (Optional) Under **Advanced settings**, create a security group or select one or more security groups. If you select `Create default security group`, we create a security group that grants RDP and SSH access to everyone, which you can modify using the Amazon EC2 or Amazon VPC console. You can enter a name for this security group in the **Group name** box.

i. (Optional) Under **Advanced settings**, select an IAM role. If this list is empty, you can create a role using the IAM console.



4. Click **Create**. If you are creating a key pair, you are prompted to save the `.pem` file. Save this file in a secure place; you'll need it to log in to your instance. You'll receive confirmation that the instance has launched. Click **Close**.

After you've created your instance, it appears in the list of instances for the Region in which you launched it. Initially, the status of the instance is `pending`. After the status changes to `running`, your instance is ready for use.

You can manage the lifecycle of your instance using Systems Manager, as described on this page. To perform other tasks, such as the following, you must use the AWS Management Console:

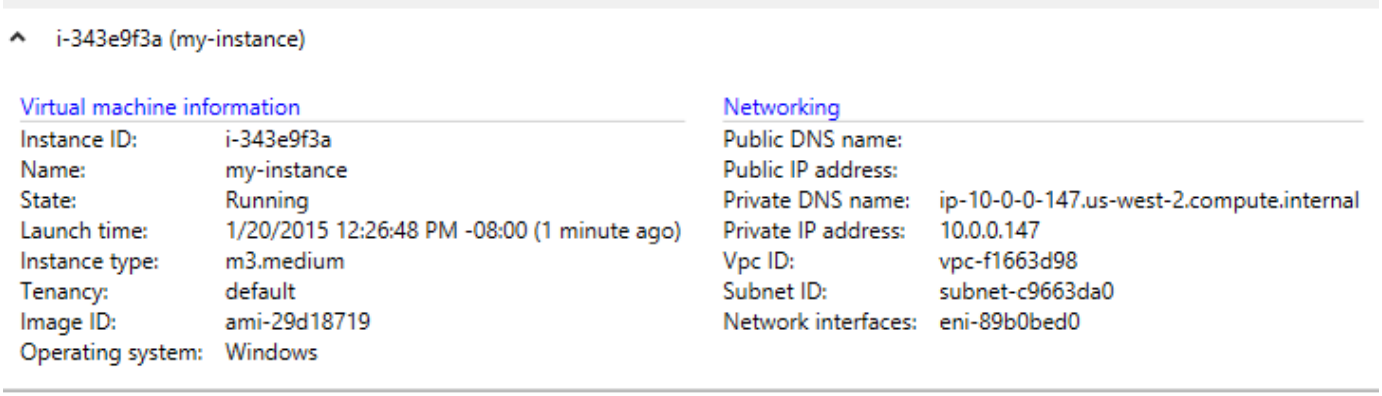
- Attach an Amazon EBS volume to your instance
- Associate an Elastic IP address with your instance
- Enable termination protection

View your instances

The permissions that your administrator grants you determine whether you can view instances and get detailed information about them.

To view your instances and get detailed information

1. Open the AWS Systems Manager console.
2. From the list of Regions, select a Region.
3. From the list of instances, select one or more instances.
4. In the lower pane, click the down arrow next to each instance to view detailed information about the instance.



Connect to your instance

You can log in to an EC2 instance if you have the private key (`.pem` file) for the key pair that was specified when launching the instance. The tool that you'll use to connect to your instance depends on whether the instance is a Windows instance or a Linux instance.

To connect to a Windows EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance, right-click, and then click **Retrieve Windows Password**.
3. In the **Retrieve Default Windows Administrator Password** dialog box, click **Browse**. Select the private key file for the key pair and then click **Open**.
4. Click **Decrypt Password**. Save the password or copy it to the clipboard.
5. Select the instance, right-click, and then click **Connect via RDP**. When prompted for credentials, use the name of the administrator account and the password that you saved in the previous step.
6. Because the certificate is self-signed, you might get a warning that the security certificate is not from a trusted certifying authority. Click **Yes** to continue.

   If the connection fails, see Troubleshoot Windows instances in the *Amazon EC2 User Guide for Windows Instances*.

To connect to a Linux EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. In the lower pane, click the down arrow next to the instance ID to view detailed information about the instance.
4. Locate the public DNS name. You'll need this information to connect to your instance.
5. Connect to the instance using PuTTY. For step-by-step instructions, see Connect to your Linux instance from Windows using PuTTY in the *Amazon EC2 User Guide for Linux Instances*.

Reboot your instance

The permissions that you've been granted by your administrator determine whether you can reboot instances.

To reboot your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.

3. Right-click the instance, and then click **Reset (Reboot)**.
4. When prompted for confirmation, click **Yes**.

Stop your instance

The permissions that you've been granted by your administrator determine whether you can stop instances.

To stop your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Shut Down (Stop)**.
4. When prompted for confirmation, click **Yes**.

Start your instance

The permissions that you've been granted by your administrator determine whether you can start instances.

To start your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Power On (Start)**.
4. When prompted for confirmation, click **Yes**.

If you get a quota error when you try to start an instance, you have reached your concurrent running instance limit. The default limit for your AWS account is 20. If you need additional running instances, complete the form at Request to Increase Amazon EC2 Instance Limit.

Terminate your instance

The permissions that you've been granted by your administrator determine whether you can terminate instances.

To terminate your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Delete (Terminate)**.
4. When prompted for confirmation, click **Yes**.

Import your virtual machine using AWS Systems Manager for Microsoft SCVMM

You can launch an EC2 instance from a virtual machine that you import from SCVMM to Amazon EC2.

> **Important**
>
> You can't import Linux virtual machines from SCVMM to Amazon EC2.

Contents

- Prerequisites
- Import your virtual machine
- Check the import task status
- Back up your imported instance

Prerequisites

- Ensure that your VM is ready. For more information, see Prepare Your VM in the *VM Import/Export User Guide*.
- In AWS Systems Manager, click **Configuration**, select the **VM Import** tab, and review the following settings:
  - **S3 bucket prefix**: We create a bucket for disk images to be uploaded before they are imported. The name of the bucket starts with the prefix listed here and includes the Region (for example, `us-east-2`). To delete the disk images after they are imported, select **Clean up S3 bucket after import**.

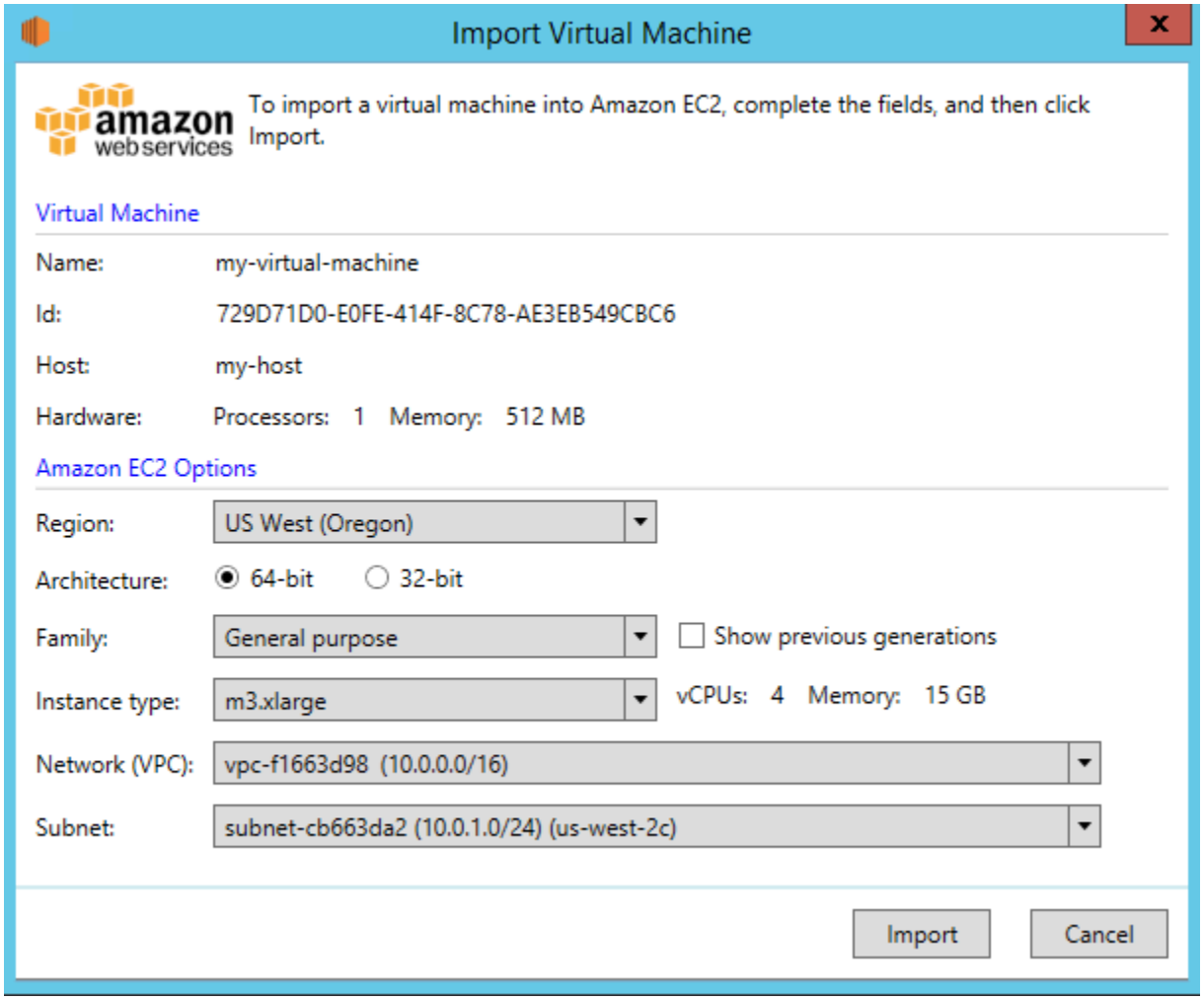
- **VM image export path**: A location for the disk images exported from the VM. To delete the disk images after they are imported, select **Clean up export path after import**.
- **Alternate Hyper-V PowerShell module path**: The location of the Hyper-V PowerShell module, if it's not installed in the standard location. For more information, see Installing the Hyper-V Management Tools.

Import your virtual machine

The permissions that you've been granted by your administrator determine whether you can import HyperV Windows virtual machines from SCVMM to AWS.

To import your virtual machine

1. Open SCVMM.
2. On the ribbon, click **VMs**. Select your virtual machine from the list.
3. On the ribbon, click **Import VM to Amazon EC2**.
4. Complete the **Import Virtual Machine** dialog box as follows:
   a. Select a Region for the instance. By default, we select the Region that you configured as your default Region.
   b. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations.** For more information, see Amazon EC2 Instances and Previous Generation Instances.
   c. Select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.
   d. Select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.



5. Click **Import**. If you haven't specified the required information in the **VM Import** tab, you'll receive an error asking you to provide the required information. Otherwise, you'll receive confirmation that the import task has started. Click **Close**.

Check the import task status

The import task can take several hours to complete. To view the current status, open AWS Systems Manager and click **Notifications**.

You'll receive the following notifications as the import task progresses:

- `Import VM: Created Import VM Task`
- `Import VM: Export VM Disk Image Done`
- `Import VM: Upload to S3`

- `Import VM: Image Conversion Starting`
- `Import VM: Image Conversion Done`
- `Import VM: Import Complete`

Note that you'll receive the `Import VM: Upload to S3`, `Import VM: Image Conversion Starting`, and `Import VM: Image Conversion Done` notifications for each disk image converted.

If the import task fails, you'll receive the notification `Import VM: Import Failed`. For more information about troubleshooting issues with import tasks, see Errors importing a virtual machine.

Back up your imported instance

After the import operation completes, the instance runs until it is terminated. If your instance is terminated, you can't connect to or recover the instance. To ensure that you can start a new instance with the same software as an imported instance if needed, create an Amazon Machine Image (AMI) from the imported instance. For more information, see Create a custom Windows AMI.

Troubleshoot AWS Systems Manager for Microsoft SCVMM

The following are common errors and troubleshooting steps.

Contents

- Error: Add-in cannot be installed
- Installation errors
- Check the log file
- Errors importing a virtual machine
- Uninstall the add-in

Error: Add-in cannot be installed

If you receive the following error, try installing KB2918659 on the computer running the VMM console. For more information, see Description of System Center 2012 SP1 Update Rollup 5. Note that you don't need to install all the updates listed in this article to address this issue, just KB2918659.

```
Add-in cannot be installed
The assembly "Amazon.Scvmm.Addin" referenced to by add-in component "AWS Systems
 Manager for
Microsoft SCVMM" could not be found in the add-in package. This could be due to the
 following
reasons:
1. The assembly was not included with the add-in package.
2. The AssemblyName attribute for the add-in does not match the name of the add-in
 assembly.
3. The assembly file is corrupt and cannot be loaded.
```

Installation errors

If you receive one of the following errors during installation, it is likely due to an issue with SCVMM:

```
Could not update managed code add-in pipeline due to the following error:
Access to the path 'C:\Program Files\Microsoft System Center 2012\Virtual Machine
 Manager
\Bin\AddInPipeline\PipelineSegments.store' is denied.
```

```
Could not update managed code add-in pipeline due to the following error:
The required folder 'C:\Program Files\Microsoft System Center 2012\Virtual Machine
 Manager
\Bin\AddInPipeline\HostSideAdapters' does not exist.
```

```
Add-in cannot be installed
The assembly "Microsoft.SystemCenter.VirtualMachineManager.UIAddIns.dll" referenced by
 the
add-in assembly "Amazon.Scvmm.AddIn" could not be found in the add-in package. Make
 sure
that this assembly was included with the add-in package.
```

Try one of the following steps to work around this issue:

- Grant authenticated users permission to read and run the `C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddInPipeline` folder. In Windows Explorer, right-click the folder, select **Properties**, and then select the **Security** tab.
- Close the SCVMM console and start it one time as an administrator. From the **Start** menu, locate SCVMM, right-click, and then select **Run as administrator**.

Check the log file

If you have a problem using the add-in, check the generated log file, `%APPDATA%\Amazon\SCVMM\ec2addin.log`, for useful information.

Errors importing a virtual machine

The log file, `%APPDATA%\Amazon\SCVMM\ec2addin.log`, contains detailed information about the status of an import task. The following are common errors that you might see in the log file when you import your VM from SCVMM to Amazon EC2.

Error: Unable to extract Hyper-V VirtualMachine object

Solution: Configure the path to the Hyper-V PowerShell module.

Error: You do not have permission to perform the operation

This error usually occurs when Hyper-V can't save the VM image into the configured path. To resolve this issue, do the following.

1. Create a directory on the Hyper-V server. For example: C:\vmimages.
2. Share the directory you just created in Hyper-V. Any user running SCVMM should be given access to the directory.
3. In the plugin, set the export path to `\\hyperv\vmimages`.
4. Perform the export.

The image will be exported to a local directory on the Hyper-V server. The SCVMM plugin will pull it from Hyper-V, and upload into Amazon S3.

Uninstall the add-in

If you need to uninstall the add-in, use the following procedure.

To uninstall the add-in

1. Open the VMM console.
2. Select the **Settings** workspace, and then click **Console Add-Ins**.
3. Select **AWS Systems Manager for Microsoft SCVMM**.

4. On the ribbon, click **Remove**.
5. When prompted for confirmation, click **Yes**.

If you reinstall the add-in after uninstalling it and receive the following error, delete the path as suggested by the error message.

```
Error (27301)
There was an error while installing the add-in. Please ensure that the following path
 does not
exist and then try the installation again.

C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddInPipeline
\
AddIns\EC2WINDOWS...
```

Related information

The following related resources can help you as you work with this service.

Windows on AWS

- **Windows on AWS** – Overview of Windows on AWS workloads and services.
- **Amazon Web Services and Microsoft: Frequently Asked Questions** – Frequently asked questions specific to running Microsoft software on AWS.
- **Microsoft Licensing on AWS: Options for using Microsoft software licenses on the AWS Cloud** – Options for using Microsoft software licenses on the AWS Cloud.
- **AWS Migration Acceleration Program for Windows** – AWS services, best practices, and tools to help you save costs and accelerate migrations of Windows workloads to AWS.
- **AWS Optimization and Licensing Assessment** – Evaluate your Windows environment to reduce costs and optimize compute.
- **AWS Launch Wizard** – AWS Launch Wizard guides you through the sizing, configuration, and deployment of applications on AWS following the AWS Well-Architected Framework.
- **Microsoft SQL Server on AWS** – Overview of Microsoft SQL Server on AWS workloads and services.
- **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
Tag new AMIs and snapshots when copying	When you copy an AMI, you can tag the new AMI and the new snapshots with the same tags, or you can tag them with different tags.	March 7, 2024
Remove AWS Management Pack pages	The AWS Management Pack was primarily used with Windows Server 2012 and prior. Those legacy OS platform versions are no longer supported. To manage and troubleshoot your fleet of servers running on AWS and on-premises, see AWS Systems Manager Fleet Manager.	February 12, 2024
EC2 Instance Connect 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	November 28, 2023

	case, workload type, and CPU manufacturer preference, as well as how you prioritiz e 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.	
EC2 Free Tier	You can track your 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-t o-Code records your console actions, and then uses generative AI to suggest code in your preferred infrastructu re-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
AWSDataLifecycleMa nagerSSMFullAccess AWS managed policy	Updated the policy to support application-consis tent snapshots for SAP HANA using the `AWSSystem sManagerSAP-Create DLMSnapshotForSAPH ANA` 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.	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
AWS managed policy for Amazon Data Lifecycle Manager	Added the AWSDLMDefaultSnapshotPoliciesServiceRolePolicy and AWSDLMDefaultAMIPoliciesServiceRolePolicy AWS managed policies to support Amazon Data Lifecycle Manager default policies.	November 16, 2023

Amazon EBS snapshot lock	You can lock your Amazon EBS snapshots to protect them against accidental or malicious deletions, or to or store them in WORM format for a specific duration.	November 15, 2023
Amazon EBS performance updates	Updated Amazon EBS performance for C6in, M6in, M6idn, R6in, and R6idn instances.	November 15, 2023
Amazon EBS performance updates	Updated Amazon EBS performance for C6in, M6in, M6idn, R6in, and R6idn instances.	November 15, 2023
Instance topology	You can use the DescribeI nstanceTopology API to detect the location of your instances, and then use this information to optimize HPC and ML jobs by running them on instances that are physicall y closer to each other.	November 13, 2023
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-consis tent snapshots.	November 7, 2023

AWSDataLifecycleManagerSSMFullAccess AWS managed policy	Amazon Data Lifecycle Manager added the AWSDataLifecycleManagerSSMFullAccess AWS managed policy.	November 7, 2023
Windows fast launch shared AMI support	You can now enable Windows fast launch on an AMI that's shared with you. When you enable Windows fast launch on a shared AMI, the pre-provisioned snapshots for faster launching are created in your account.	November 6, 2023
Capacity Blocks for ML	You can now reserve GPU instances on a future date to support your short duration machine learning (ML) workloads.	October 31, 2023
New bare metal instances	`.metal-16xl` and `.metal-32xl` bare metal instances for R7iz.	October 30, 2023
New bare metal instances	`.metal-24xl` and `.metal-48xl` 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

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 Sca lable processors.	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	`r7a.metal-48xl` 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 Microsoft Windows Server 2022	Hibernate your newly-launched instances that were launched from the Microsoft Windows Server 2022 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
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 *block public access for AMIs* 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 net work 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 Sca lable processors.	August 2, 2023
Amazon EBS performance updates	Updated Amazon EBS performance for R6a instances.	June 29, 2023
Dedicated Hosts	You can allocate Dedicated Hosts on specific hardware assets on an Outpost.	June 20, 2023

EC2 Instance Connect Endpoint	You can now connect to an instance via SSH or RDP without requiring the instance to have a public IPv4 address.	June 13, 2023
IMDS Package Analyzer	You can now use the IMDS Packet Analyzer to identify sources of IMDSv1 calls on your EC2 instances.	June 1, 2023
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

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
Windows Virtualization-based security - Credential Guard	You can enable Credential Guard, a Virtualization-based security (VBS) feature, on supported Amazon EC2 instances.	January 31, 2023
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 mod ifications 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 `priceCapacityOptim ized` 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-opt imized` 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 `rbin:Request/ResourceType` and `rbin:Attribute/ResourceType` 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 protectio n 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-i n-depth that helps customers secure software from threats that persist across reboots.	May 10, 2022

NitroTPM	Nitro Trusted Platform Module (NitroTPM) is a virtual device that is provided by the AWS Nitro System and conforms to the TPM 2.0 specification.	May 10, 2022
AMI state change events	Amazon EC2 now generates an event when an AMI changes state. You can use Amazon EventBridge to detect and react to these events.	May 9, 2022
Describe public keys	You can query the public key and creation date of an Amazon EC2 key pair.	April 28, 2022
Create key pairs	You can specify the key format (PEM or PPK) when creating a new key pair.	April 28, 2022
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 `lastLaunchedTime` 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
New Local Zones added	Add Local Zones in Atlanta, Phoenix, and Seattle.	January 11, 2022
Configure Windows AMIs for faster launching	Configure Windows AMIs to launch instances up to 65% faster, using pre-provisioned snapshots.	January 10, 2022
Instance tags in instance metadata	You can access an instance' s tags from the instance metadata.	January 6, 2022
Capacity Reservations in cluster placement groups	You can create Capacity Reservations in cluster placement groups.	January 6, 2022
Recycle Bin for Amazon EBS snapshots	Recycle Bin for Amazon EBS snapshots is a snapshot recovery feature that enables you to restore accidentally del eted snapshots.	November 29, 2021
M6a instances	New general purpose instances powered by AMD 3rd Generation EPYC processors.	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 rar ely-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
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
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
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 AWSDataLifecycleManagerServiceRoleForAMIManagement 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 interface s	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 Availabil ity 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 connectio n to the serial port of an instance.	March 30, 2021
Boot modes	Amazon EC2 now supports UEFI boot on selected AMD- and Intel-based EC2 instances .	March 22, 2021
Amazon EBS local snapshots on Outposts	You can now use Amazon 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
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`) vol umes can range in size from 125 GiB to 16 TiB.	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 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
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
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
Fast snapshot restore	You can enable fast snapshot restore for snaphots that are shared with you.	July 21, 2020
EC2Launch v2	You can use EC2Launch v2 to perform tasks during instance startup, if an instance is stopped and later started, if an instance is restarted, and on demand. EC2Launch v2 supports all versions of Windows Server and replaces EC2Launch and EC2Config.	June 30, 2020
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
Launch instances using a Systems Manager parameter	You can specify a AWS Systems Manager parameter instead of an AMI when you launch an instance.	May 5, 2020
Customize scheduled event notifications	You can customize scheduled event notifications to include tags in the email notification.	May 4, 2020
Windows Server on Dedicated Hosts	You can use Windows Server AMIs provided by Amazon to run the latest versions of Windows Server on Dedicated Hosts.	April 7, 2020
Stop and start a Spot Instance	Stop your Spot Instances backed by Amazon EBS and start them at will, instead of relying on the stop interrupt ion behavior.	January 13, 2020
Resource tagging	You can tag egress-only internet gateways, local gateways, local gateway route tables, local gateway virtual interfaces, local gateway virtual interface groups, local gateway route table VPC associations, and local gateway route table virtual interface group associations.	January 10, 2020

Connect to your instance using Session Manager	You can start a Session Manager session with an instance from the Amazon EC2 console.	December 18, 2019
Dedicated Hosts and host resource groups	Dedicated Hosts can now be used with host resource groups.	December 2, 2019
Dedicated Host sharing	You can now share your Dedicated Hosts across AWS accounts.	December 2, 2019
Default credit specification at the account level	You can set the default credit specification per burstable performance instance family at the account level per AWS Region.	November 25, 2019
Instance type discovery	You can find an instance type that meets your needs.	November 22, 2019
Dedicated Hosts	You can now configure a Dedicated Host to support multiple instance types in an instance family.	November 21, 2019
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 performan ce.	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
Hibernation support for On-Demand Windows instances	You can hibernate On-Demand Windows instances.	October 14, 2019
Queued purchases of Reserved Instances	You can queue the purchase of a Reserved Instance up to three years in advance.	October 4, 2019
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 blue screen/stop error.	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
Resource tagging	Launch templates on creation.	July 24, 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
VSS application-consistent snapshots	Take application-consistent snapshots of all Amazon EBS volumes attached to your Windows instances using AWS Systems Manager Run Command.	May 13, 2019
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
Windows Automated Upgrade	Perform automated upgrades of EC2 Windows instances using AWS Systems Manager.	May 6, 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 Reservati ons 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	2016-11-1 5	Partition placement groups spread instances across logical partitions, ensuring that instanc es in one partition do not share underlying hardware with instances in other partitions.	20 December 2018

Feature	API version	Description	Release date
		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
Instances featuring 100 Gbps of network bandwidth	2016-11-15	New C5n instances can utilize up to 100 Gbps of network bandwidth.	26 November 2018
Spot console recommends a fleet of instances	2016-11-15	The Spot console recommends a fleet of instances based on Spot best practice (instance diversification) to meet the minimum hardware specifications (vCPUs, memory, and storage) for your application need. For more information, see Create a Spot Fleet request.	20 November 2018
New EC2 Fleet request type: `instant`	2016-11-15	EC2 Fleet now supports a new request type, `instant`, that you can use to synchronously provision capacity across instance types and purchase models. The `instant` request returns the launched instances in the API response, and takes no further action, enabling you to control if and when instances are launched. For more information, see EC2 Fleet request types.	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

Feature	API version	Description	Release date
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
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

Feature	API version	Description	Release date
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. For more information, see Memory optimized instances.	27 September 2018
f1.4xlarge instances	2016-11-15	Expands the range of the accelerated-computing F1 instance family with the introduction of f1.4xlarge instances.	25 September 2018
Console support for scheduled scaling for Spot Fleet	2016-11-15	Increase or decrease the current capacity of the fleet based on the date and time. For more information, see Scale Spot Fleet using scheduled scaling.	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

Feature	API version	Description	Release date
Allocation strategies for EC2 Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Allocation strategies for Spot Instances.	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. For more information, see Memory optimized instances.	25 July 2018
z1d instances	2016-11-15	These instances are designed for applications that require high per-core performance with a large amount of memory, such as electronic design automation (EDA) and relational databases. These instances come with NVME instance store volumes. For more information, see Memory optimized instances.	25 July 2018

Feature	API version	Description	Release date
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. For more information, see Storage optimized instances.	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 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 Windows 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 informati on, 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-perf ormance big data workloads. For more information, see Storage optimized instances.	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
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

Feature	API version	Description	Release date
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. For more information, see Memory optimized instances.	28 November 2017
C5 instances	2016-11-15	C5 instances are designed for compute-heavy applications. For more information, see Compute optimized instances.	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. For more information, see Windows accelerated computing instances.	25 October 2017
Modify VPC tenancy	2016-11-15	You can change the instance tenancy attribute of a VPC from `dedicated` to `default`. For more information, see Change the tenancy of a VPC.	16 October 2017

Feature	API version	Description	Release date
Stop on interruption	2016-11-15	You can specify whether Amazon EC2 should stop or terminate Spot Instances when they are interrupted. For more information, see Interruption behavior.	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
Elastic Graphics	2016-11-15	Attach Elastic Graphics accelerators to your instances to accelerate the graphics performance of your applications. For more information, see Amazon Elastic Graphics.	29 August 2017
Recover Elastic IP addresses	2016-11-15	If you release an Elastic IP address for use in a VPC, you might be able to recover it. For more information, see 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
G3 instances	2016-11-15	G3 instances provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. G3 instances also provide NVIDIA GRID Virtual Workstation features, supporting 4 monitors with resolutions up to 4096x2160. For more information, see Windows accelerated computing instances.	13 July 2017

Feature	API version	Description	Release date
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. For more information, see 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
IPv6 support	2016-11-15	You can associate an IPv6 CIDR with your VPC and subnets, and assign IPv6 addresses to instances in your VPC. For more information, see Amazon EC2 instance IP addressing.	1 December 2016

Feature	API version	Description	Release date
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. For more information, see Memory optimized instances	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. For more information, see Windows accelerated computing instances.	29 September 2016
`m4.16xlarge` instances	2016-04-01	Expands the range of the general-purpose M4 family with the introduction of `m4.16xlarge` instances, with 64 vCPUs and 256 GiB of RAM.	6 September 2016
Automatic scaling for Spot Fleet		You can now set up scaling policies for your Spot Fleet. For more information, see Automatic scaling for Spot Fleet.	1 September 2016

Feature	API version	Description	Release date
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. For more information, see Memory optimized instances.	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
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

Feature	API version	Description	Release date
CloudWatch metrics for Spot Fleet		You can now get CloudWatch metrics for your Spot Fleet. For more information, see CloudWatch metrics for Spot Fleet.	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
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

Feature	API version	Description	Release date
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
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

Feature	API version	Description	Release date
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 *VM Import/Export User Guide* .	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. For more information, see Storage optimized instances.	24 March 2015
Systems Manager		Systems Manager enables you to configure and manage your EC2 instances.	17 February 2015
Systems Manager for Microsoft SCVMM 1.5		You can now use Systems Manager for Microsoft SCVMM to launch an instance and to import a VM from SCVMM to Amazon EC2. For more information, see Create an EC2 Instance and Import your virtual machine.	21 January 2015

Feature	API version	Description	Release date
Automatic recovery for EC2 instances		You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, IP addresses, and all instance metadata. For more information, see Recover your instance.	12 January 2015
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. For more information, see Compute optimized instances.	11 January 2015

Feature	API version	Description	Release date
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
Spot Instance termination notices		The best way to protect against Spot Instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot Instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot Instance. For more information, see Spot Instance interruption notices.	5 January 2015
Systems Manager for Microsoft SCVMM		Systems Manager for Microsoft SCVMM provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. For more information, see AWS Systems Manager for Microsoft System Center VMM.	29 October 2014
`DescribeVolumes` pagination support	2014-09-01	The `DescribeVolumes` API call now supports the pagination of results with the `MaxResults` and `NextToken` parameters. For more information, see DescribeVolumes in the *Amazon EC2 API Reference*.	23 October 2014

Feature	API version	Description	Release date
Added support for Amazon CloudWatch Logs		You can use Amazon CloudWatch Logs to monitor, store, and access your system, application, and custom log files from your instances or other sources. You can then retrieve the associated log data from CloudWatch Logs using the Amazon CloudWatch console, the CloudWatch Logs commands in the AWS CLI, or the CloudWatch Logs SDK.	10 July 2014
T2 instances	2014-06-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances.	30 June 2014
New **EC2 Service Limits** page		Use the **EC2 Service Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.	19 June 2014
Amazon EBS General Purpose SSD Volumes	2014-05-01	General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose SSD volumes can range in size from 1 GiB to 1 TiB.	16 June 2014

Feature	API version	Description	Release date
Windows Server 2012 R2		AMIs for Windows Server 2012 R2 use the new AWS PV drivers. For more information, see AWS PV drivers.	3 June 2014
AWS Management Pack		AWS Management Pack now supports for System Center Operations Manager 2012 R2.	22 May 2014
Amazon EBS encryption	2014-05-01	Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using AWS managed keys. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage.	21 May 2014
R3 instances	2014-02-01	Memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances. For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.	9 April 2014

Feature	API version	Description	Release date
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 `m3.medium` and `m3.large` are now supported. For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.	20 January 2014
I2 instances	2013-10-15	These instances provide very high IOPS. I2 instances also support enhanced networking that delivers improve inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see Storage optimized instances.	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, `m3.xlarge` and `m3.2xlarge` now support instance store with SSD volumes.	19 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 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	Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics. For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.	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. For more information, see Windows accelerated computing instances.	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 Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a Region.	11 September 2013
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see 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
AWS Management Pack		The AWS Management Pack links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager.	8 May 2013
New **Tags** page		There is a new **Tags** page in the Amazon EC2 console. For more information, see Tag your Amazon EC2 resources.	04 April 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: `c1.xlarge` , `m2.2xlarge` , `m3.xlarge` , and `m3.2xlarge` .	19 March 2013
PV Drivers		To learn how to upgrade the paravirtualized (PV) drivers on your Windows AMI, see Upgrade PV drivers on Windows instances.	March 2013

Feature	API version	Description	Release date
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 nondefaul t 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/Ma pReduce, and parallel file systems.	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs).	17 December 2012

Feature	API version	Description	Release date
Updated EBS metrics and status checks for Provisioned IOPS SSD volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS SSD volumes. Also added new status checks for Provisioned IOPS SSD volumes.	20 November 2012

Feature	API version	Description	Release date
Support for Windows Server 2012		Amazon EC2 now provides you with several pre-configured Windows Server 2012 AMIs. These AMIs are immediately available for use in every region and for every 64-bit instance type. The AMIs support the following languages: • English • Chinese Simplified • Chinese Traditional • Chinese Traditional Hong Kong • Japanese • Korean • Portuguese • Portuguese Brazil • Czech • Dutch • French • German • Hungarian • Italian •	19 November 2012

Feature	API version	Description	Release date
		Polish • Russian • Spanish • Swedish • Turkish	
M3 instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.	31 October 2012
Spot Instance request status	2012-10-01	Spot Instance request status makes it easy to determine the state of your Spot requests.	14 October 2012
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012

Feature	API version	Description	Release date
Provisioned IOPS SSD for Amazon EBS	2012-07-20	Provisioned IOPS SSD volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times.	31 July 2012
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: • 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 interrupt ion.		You can now manage your Spot Instances as follows: • Specify the amount you are willing to pay for Spot Instances using Auto Scaling launch configurations, and set up a schedule for specifying the amount you are willing to pay for Spot Instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the *Amazon EC2 Auto Scaling User Guide*. • Get notifications when instances are launched or terminated. • Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources.	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for exporting Windows Server instances that you originally imported into EC2. Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	25 May 2012

Feature	API version	Description	Release date
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for `cc2.8xlarge` instances in a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
Medium instances, support for 64-bit on all AMIs	2011-12-15	Added support for a new instance type and 64-bit information.	7 March 2012
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic network interfaces.	21 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitor the status of your instances.	16 November 2011

Feature	API version	Description	Release date
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (cc2.8xlarge) to Amazon EC2.	14 November 2011
Spot Instances in Amazon VPC	2011-07-15	Added information about the support for Spot Instances in Amazon VPC. With this update, users can launch Spot Instances a virtual private cloud (VPC). By launching Spot Instances in a VPC, users of Spot Instances can enjoy the benefits of Amazon VPC.	11 October 2011
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of `ImportInstance` and `ImportVolume`, which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of `ResumeImport`, users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide.	24 August 2011
Support for Windows Server 2003 R2		VM Import now supports Windows Server 2003 (R2). With this release, VM Import supports all versions of Windows Server supported by Amazon EC2.	24 August 2011

Feature	API version	Description	Release date
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 (Connecto r). This update includes proxy support for Internet access, better error handling, i mproved task progress bar accuracy, and several bug fixes.	27 June 2011
Spot Instances Availability Zone pricing changes	2011-05-1 5	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

Feature	API version	Description	Release date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Dedicated Instances.	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances.	27 March 2011
Support for Windows Server 2008 R2		Amazon EC2 now provides you with several pre-configured Windows Server 2008 R2 AMIs. These AMIs are immediately available for use in every region and in most 64-bit instance types, excluding t1.micro and HPC families. The AMIs will support multiple languages.	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance metadata and user data and Instance metadata categories.	11 March 2011

Feature	API version	Description	Release date
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
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.	19 September 2010

Feature	API version	Description	Release date
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
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For detailed instance type specifications, see the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.	12 July 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes.	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 the Amazon EC2 Instance Types Guide. For pricing information, see Amazon EC2 Instance Types.	22 February 2010
Reserved Instances with Windows		Amazon EC2 now supports Reserved Instances with Windows.	22 February 2010