aws

User Guide

Amazon Elastic Compute Cloud

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

Amazon Elastic Compute Cloud: User Guide

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 ... 1
Related services ... 2
Access EC2 ... 4
Pricing ... 5
Estimates, billing, and cost optimization ... 6
Resources ... 6
Get started tutorial ... 8
Step 1: Launch an instance ... 10
Step 2: Connect to your instance ... 11
Step 3: Clean up your instance ... 15
Next steps ... 15
Best practices ... 17
Amazon Machine Images ... 19
AMI characteristics ... 21
Launch permissions ... 21
Root device type ... 21
Determine the AMI root device type ... 23
Virtualization types ... 24
Find an AMI ... 26
Paid AMIs in the AWS Marketplace ... 34
Sell your AMI in the AWS Marketplace ... 35
Find a paid AMI ... 35
Purchase a paid AMI ... 37
Retrieve the product code ... 38
Use paid support ... 39
Bills for paid and supported AMIs ... 39
Manage your subscriptions ... 39
AMI lifecycle ... 40
Create an AMI ... 41
Create an instance store-backed AMI ... 49
Create an AMI using Windows Sysprep ... 89
Copy an AMI ... 106
Store and restore an AMI ... 117

Check when an AMI was last used ... 127
Deprecate an AMI ... 128
Disable an AMI ... 136
Deregister an AMI ... 142
Boot modes ... 149
Requirements for UEFI boot mode ... 151
AMI boot mode parameter ... 152
Instance type boot mode ... 154
Instance boot mode ... 159
Operating system boot mode ... 161
Set AMI boot mode ... 163
UEFI variables ... 168
UEFI Secure Boot ... 169
AMI encryption ... 184
Instance-launching scenarios ... 184
Image-copying scenarios ... 188
Shared AMIs ... 190
Verified provider ... 190
Find shared AMIs ... 191
Prepare to use shared AMIs for Linux ... 194
Make your AMI public ... 195
Understand block public access ... 199
Shared AMI use with organizations and OUs ... 209
Share an AMI with specific AWS accounts ... 220
Cancel having an AMI shared with your account ... 224
Recommendations for creating shared Linux AMIs ... 226
Monitor AMI events ... 231
Event details ... 233
available events ... 233
failed events ... 234
deregistered events ... 235
disabled events ... 235
Understand AMI billing ... 236
AMI billing fields ... 236
Find AMI billing information ... 239
Verify AMI charges on your bill ... 241

AMI quotas ... 242
Request a quota increase for AMIs ... 243
Instances ... 244
Instance types ... 245
Available instance types ... 246
Hardware specifications ... 247
AMI virtualization types ... 249
Find an instance type ... 250
EC2 instance type finder ... 255
Compute Optimizer recommendations ... 258
Instance type changes ... 261
Burstable performance instances ... 269
GPU instances ... 321
Mac instances ... 335
EBS optimization ... 361
CPU options ... 439
AMD SEV-SNP ... 571
Processor state control ... 577
Billing and purchasing options ... 580
On-Demand Instances ... 581
Reserved Instances ... 583
Spot Instances ... 650
Dedicated Hosts ... 744
Dedicated Instances ... 799
Capacity Reservations ... 807
Launch templates ... 892
Restrictions ... 893
Permissions ... 894
Control launching instances ... 901
Create ... 903
Modify (manage versions) ... 919
Delete ... 924
Launch an instance ... 926
Instance parameter reference ... 928
Launch using the launch instance wizard ... 943
Launch using a launch template ... 946

Launch from an existing instance ... 953
Launch from an AWS Marketplace AMI ... 955
Connect to your instance ... 959
Get the required instance details ... 960
Locate the private key and set permissions ... 962
(Optional) Get the instance fingerprint ... 963
Connect to your Linux instance using SSH ... 964
Connect to your Windows instance using RDP ... 980
Connect using Session Manager ... 990
Connect using EC2 Instance Connect ... 991
Connect using EC2 Instance Connect Endpoint ... 1027
Instance state changes ... 1053
Billing by instance state ... 1054
Pending instances ... 1055
Stopped instances ... 1056
Hibernated instances ... 1056
Rebooting instances ... 1057
Terminated instances ... 1057
Differences between instance states ... 1058
Stop and start ... 1060
Hibernate ... 1069
Reboot ... 1098
Terminate ... 1100
Retire ... 1110
Instance resiliency ... 1115
Instance metadata ... 1124
Instance metadata categories ... 1125
Dynamic data categories ... 1140
Access instance metadata ... 1140
Configure IMDS options ... 1177
Run commands at launch ... 1204
Example: AMI launch index value ... 1228
Detect whether a host is an EC2 instance ... 1233
Inspect the instance identity document ... 1233
Inspect the system UUID ... 1233
Inspect the system virtual machine generation identifier ... 1235

Instance identity documents ... 1240
Retrieve the instance identity document ... 1241
Verify instance identity document ... 1243
Public certificates ... 1254
Clock synchronization ... 1307
Leap seconds ... 1308
Use the local Amazon Time Sync Service ... 1309
Use the public Amazon Time Sync Service ... 1321
Compare timestamps for your Linux instances ... 1323
Change the time zone of your instance ... 1324
Manage device drivers ... 1327
Network drivers ... 1327
Graphics drivers ... 1328
Storage device drivers ... 1328
AMD drivers ... 1328
NVIDIA drivers ... 1334
Install the ENA driver on Windows ... 1371
Windows PV drivers ... 1388
AWS Windows NVMe drivers ... 1421
Configure Windows instances ... 1429
Windows-specific system settings ... 1430
Windows launch agents ... 1431
EC2 Fast Launch for Windows ... 1587
Change the Windows Administrator password ... 1610
Add Windows System components ... 1611
Install WSL on Windows ... 1616
Upgrade Windows instances ... 1618
Perform an in-place upgrade ... 1619
Perform an automated upgrade ... 1623
Migrate to a current generation instance type ... 1634
Troubleshoot an upgrade ... 1643
Tutorial: Connect EC2 instance to RDS database ... 1643
Tutorial objective ... 1643
Context ... 1644
Architecture ... 1644
Considerations ... 1646

Time to complete the tutorial ... 1647
Costs ... 1647
Option 1: Automatically connect using EC2 console ... 1647
Option 2: Automatically connect using RDS console ... 1659
Option 3: Manually connect ... 1669
Fleets ... 1679
Features and benefits ... 1679
Which fleet method to use? ... 1680
Configuration options ... 1682
Request types ... 1683
Spending limit ... 1712
Attribute-based instance type selection ... 1714
Instance weighting ... 1748
Allocation strategies ... 1751
Capacity Rebalancing ... 1758
Capacity Reservations ... 1763
Work with EC2 Fleet ... 1764
EC2 Fleet request states ... 1765
Create an EC2 Fleet ... 1766
Tag an EC2 Fleet ... 1780
Describe an EC2 Fleet ... 1782
Modify an EC2 Fleet ... 1786
Delete an EC2 Fleet ... 1788
Work with Spot Fleet ... 1792
Spot Fleet request states ... 1793
Create a Spot Fleet ... 1794
Tag a Spot Fleet ... 1811
Describe a Spot Fleet ... 1821
Modify a Spot Fleet request ... 1821
Cancel (delete) a Spot Fleet request ... 1823
Automatic scaling for Spot Fleet ... 1825
Monitor your fleet ... 1835
Monitor your fleet using CloudWatch ... 1836
Monitor your fleet using EventBridge ... 1839
Tutorials ... 1857
Tutorial: Configure EC2 Fleet to use instance weighting ... 1859

Tutorial: Configure EC2 Fleet to use On-Demand Instances as the primary capacity 1862
Tutorial: Configure EC2 Fleet to launch On-Demand Instances using targeted Capacity Reservations 1864
Tutorial: Configure your EC2 Fleet to launch instances into Capacity Blocks 1870
Example CLI configurations for EC2 Fleet 1873
Example 1: Launch Spot Instances as the default purchasing option 1874
Example 2: Launch On-Demand Instances as the default purchasing option 1874
Example 3: Launch On-Demand Instances as the primary capacity 1875
Example 4: Launch On-Demand Instances using multiple Capacity Reservations 1875
Example 5: Launch On-Demand Instances using Capacity Reservations when the total target capacity exceeds the number of unused Capacity Reservations 1879
Example 6: Launch On-Demand Instances using targeted Capacity Reservations 1883
Example 7: Configure Capacity Rebalancing to launch replacement Spot Instances 1886
Example 8: Launch Spot Instances in a capacity-optimized fleet 1888
Example 9: Launch Spot Instances in a capacity-optimized fleet with priorities 1889
Example 10: Launch Spot Instances in a price-capacity-optimized fleet 1891
Example 11: Configure attribute-based instance type selection 1892
Example CLI configurations Spot Fleet 1893
Example 1: Launch Spot Instances using the lowest-priced Availability Zone or subnet in the Region 1894
Example 2: Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list 1895
Example 3: Launch Spot Instances using the lowest-priced instance type in a specified list 1897
Example 4. Override the price for the request 1899
Example 5: Launch a Spot Fleet using the diversified allocation strategy 1900
Example 6: Launch a Spot Fleet using instance weighting 1903
Example 7: Launch a Spot Fleet with On-Demand capacity 1905
Example 8: Configure Capacity Rebalancing to launch replacement Spot Instances 1906
Example 9: Launch Spot Instances in a capacity-optimized fleet 1908
Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities 1909
Example 11: Launch Spot Instances in a priceCapacityOptimized fleet 1910
Example 12: Configure attribute-based instance type selection 1911
Fleet quotas 1912
Request a quota increase for target capacity 1914
Networking 1915

Regions and Zones ... 1916
Regions ... 1917
Availability Zones ... 1920
Local Zones ... 1923
Wavelength Zones ... 1925
AWS Outposts ... 1926
Instance IP addressing ... 1928
Private IPv4 addresses ... 1929
Public IPv4 addresses ... 1930
Public IPv4 address optimization ... 1931
IPv6 addresses ... 1933
EC2 instance hostnames ... 1934
Link-local addresses ... 1934
IPv4 addresses ... 1935
IPv6 addresses ... 1938
Multiple IP addresses ... 1941
Multiple IPv4 addresses on Windows ... 1951
Instance hostname types ... 1958
Types of EC2 hostnames ... 1959
Where to find resource names and IP names ... 1960
Choosing between resource names and IP names ... 1962
Change resource based naming options ... 1963
Bring your own IP addresses ... 1964
BYOIP definitions ... 1965
Requirements and quotas ... 1966
Regional availability ... 1967
Local Zone availability ... 1967
Prerequisites ... 1968
Onboard your address range ... 1976
Use your address range ... 1985
Elastic IP addresses ... 1986
Elastic IP address pricing ... 1987
Elastic IP address basics ... 1987
Elastic IP address quota ... 1988
Associate an Elastic IP address ... 1988
Transfer an Elastic IP address ... 1993

Release an Elastic IP address ... 1999
Use reverse DNS for email applications ... 2000
Network interfaces ... 2003
Network interface concepts ... 2004
Network cards ... 2006
IP addresses per network interface ... 2007
Create a network interface ... 2009
Manage IP addresses ... 2015
Modify network interface attributes ... 2017
Multiple network interfaces ... 2020
Requester-managed network interfaces ... 2023
Prefix delegation ... 2025
Delete a network interface ... 2032
Network bandwidth ... 2033
Available instance bandwidth ... 2034
Monitor instance bandwidth ... 2036
Enhanced networking ... 2037
Elastic Network Adapter (ENA) ... 2038
ENA Express ... 2053
Intel 82599 VF ... 2075
Monitor network performance ... 2087
Troubleshoot ENA on Linux ... 2098
Troubleshoot ENA on Windows ... 2112
Improve network latency on Linux ... 2130
Nitro performance considerations ... 2133
Optimize network performance on Windows ... 2140
Elastic Fabric Adapter ... 2142
EFA basics ... 2143
Supported interfaces and libraries ... 2144
Supported instance types ... 2144
Supported operating systems ... 2146
EFA limitations ... 2147
EFA pricing ... 2147
EFA on accelerated instances ... 2147
Get started with EFA and MPI ... 2152
Get started with EFA and NCCL ... 2169

Create and attach an EFA ... 2193
Detach and delete an EFA ... 2195
Monitor an EFA ... 2196
Verify the EFA installer ... 2197
Instance topology ... 2209
How it works ... 2210
Prerequisites ... 2214
Examples ... 2215
Placement groups ... 2227
Placement strategies ... 2228
Create a placement group ... 2234
Change instance placement ... 2235
Delete a placement group ... 2237
Share a placement group ... 2238
Placement groups on AWS Outposts ... 2243
Network MTU ... 2244
Jumbo frames (9001 MTU) ... 2245
Path MTU Discovery ... 2246
Set the MTU for your instances ... 2247
Troubleshoot ... 2253
Virtual private clouds ... 2254
Your default VPCs ... 2254
Nondefault VPCs ... 2255
Internet access ... 2256
Shared subnets ... 2256
IPv6-only subnets ... 2257
Security ... 2258
Data protection ... 2259
Amazon EBS data security ... 2260
Encryption at rest ... 2260
Encryption in transit ... 2261
Infrastructure security ... 2263
Network isolation ... 2264
Isolation on physical hosts ... 2264
Controlling network traffic ... 2265
Resilience ... 2267

Compliance validation ... 2268
Identity and access management ... 2269
Identity-based policies ... 2270
Example policies for the API ... 2281
Example policies for the console ... 2323
AWS managed policies ... 2335
IAM roles ... 2339
Update management ... 2350
Best practices for Windows instances ... 2351
High-level security best practices ... 2351
Update management ... 2352
Configuration management ... 2354
Change management ... 2355
Audit and accountability for Amazon EC2 Windows instances ... 2356
Key pairs ... 2357
Create a key pair ... 2358
Tag a key pair ... 2366
Describe your key pairs ... 2369
Delete your key pair ... 2377
Add or replace a public key on your Linux instance ... 2378
Verify the fingerprint ... 2380
Security groups ... 2383
Overview ... 2383
Create a security group ... 2384
Change security groups for your instance ... 2386
Delete a security group ... 2389
Connection tracking ... 2390
Security group rules for different use cases ... 2396
NitroTPM ... 2403
Requirements ... 2404
Enable a Linux AMI for NitroTPM ... 2406
Verify that an AMI is enabled for NitroTPM ... 2407
Enable or stop using NitroTPM ... 2408
Verify that an instance is enabled for NitroTPM ... 2408
Retrieve the public endorsement key ... 2410
Credential Guard for Windows instances ... 2411

Prerequisites ... 2411
Launch a supported instance ... 2412
Disable memory integrity ... 2413
Turn on Credential Guard ... 2414
Verify that Credential Guard is running ... 2416
AWS PrivateLink ... 2417
Create an interface VPC endpoint ... 2418
Create an endpoint policy ... 2418
Storage ... 2420
AWS Storage pricing ... 2421
Amazon EBS ... 2421
EBS volume limits ... 2422
Amazon EC2 instance store ... 2426
Data persistence ... 2428
Instance store limits ... 2430
SSD instance store volumes ... 2432
Add instance store volumes ... 2436
Enable swap volume for M1 and C1 instances ... 2442
Initialize instance store volumes ... 2445
Root volumes ... 2447
Amazon EBS-backed instances ... 2447
Instance store-backed instances (Linux instances only) ... 2449
Keep root volume after instance termination ... 2450
Replace a root volume ... 2454
Device names for volumes ... 2464
Available device names ... 2465
Device name considerations ... 2467
Block device mappings ... 2468
Block device mapping concepts ... 2468
Add block device mapping to AMI ... 2472
Add block device mapping to instance ... 2476
How volumes are attached and mapped for Windows instances ... 2483
Map NVME disks to volumes ... 2484
Map non-NVME disks to volumes ... 2490
Torn write prevention ... 2500
Supported block sizes ... 2501

Requirements ... 2502
Check instance support ... 2502
Configure workload ... 2504
Windows VSS EBS snapshots ... 2505
What is VSS? ... 2506
How the VSS based Amazon EBS snapshot solution works ... 2507
VSS prerequisites ... 2508
Create VSS snapshots ... 2520
Troubleshoot VSS snapshots ... 2529
Restore EBS volumes ... 2534
Version history ... 2535
Object storage, file storage, and file caching ... 2538
Amazon S3 ... 2539
Amazon EFS ... 2541
Amazon FSx ... 2545
Amazon File Cache ... 2550
Manage resources ... 2552
Select a Region for your resources ... 2552
Find your resources ... 2553
Console steps ... 2554
CLI and API steps ... 2560
Global View (cross-Region) ... 2563
Amazon EC2 Global View ... 2563
Tag your resources ... 2566
Tag basics ... 2567
Tag your resources ... 2568
Tag restrictions ... 2569
Tags and access management ... 2570
Tag your resources for billing ... 2570
Tag resource permissions ... 2571
Add and remove tags ... 2574
Filter resources by tag ... 2577
View tags using instance metadata ... 2579
Service quotas ... 2583
View your current quotas ... 2584
Request an increase ... 2585

Restriction on email sent using port 25 ... 2585
Monitor resources ... 2587
Monitor the status of your instances ... 2588
Status checks ... 2589
State change events ... 2596
Scheduled events ... 2599
Monitor your instances using CloudWatch ... 2631
Instance alarms ... 2632
Manage detailed monitoring ... 2633
CloudWatch metrics ... 2636
Install and configure the CloudWatch agent ... 2657
Statistics for metrics ... 2662
View monitoring graphs ... 2671
Create an alarm ... 2672
Create alarms that stop, terminate, reboot, or recover an instance ... 2673
Automate using EventBridge ... 2686
Amazon EC2 event types ... 2686
Log API calls using CloudTrail ... 2687
Amazon EC2 API management events in CloudTrail ... 2689
Amazon EC2 API event examples ... 2689
Audit connections made using EC2 Instance Connect ... 2690
Monitor .NET and SQL Server applications ... 2692
Track your Free Tier usage ... 2693
Troubleshoot ... 2696
Instance launch issues ... 2696
Invalid device name ... 2697
Instance limit exceeded ... 2698
Insufficient instance capacity ... 2698
The requested configuration is currently not supported. Please check the documentation for supported configurations. ... 2699
Instance terminates immediately ... 2699
Insufficient permissions ... 2701
High CPU usage shortly after Windows starts (Windows instances only) ... 2702
Instance stop issues ... 2702
Force stop an instance ... 2703
(Optional) Create a replacement instance ... 2704

Instance termination issues ... 2706
Instance terminates immediately ... 2706
Delayed instance termination ... 2706
Terminated instance still displayed ... 2707
Error: The instance may not be terminated. Modify its 'disableApiTermination' instance attribute ... 2707
Instances automatically launched or terminated ... 2707
Unreachable instances ... 2708
Instance reboot ... 2708
Instance console output ... 2708
Capture a screenshot of an unreachable instance ... 2709
Common screenshots for Windows instances ... 2711
Instance recovery when a host computer fails ... 2721
Linux instance SSH issues ... 2721
Common causes for connection issues ... 2722
Error connecting to your instance: Connection timed out ... 2724
Error: unable to load key ... Expecting: ANY PRIVATE KEY ... 2727
Error: User key not recognized by server ... 2728
Error: Permission denied or connection closed by [instance] port 22 ... 2730
Error: Unprotected private key file ... 2732
Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----" ... 2733
Error: Server refused our key *or* No supported authentication methods available ... 2734
Cannot ping instance ... 2735
Error: Server unexpectedly closed network connection ... 2735
Error: Host key validation failed for EC2 Instance Connect ... 2736
Can't connect to Ubuntu instance using EC2 Instance Connect ... 2738
I've lost my private key. How can I connect to my instance? ... 2738
Linux instance failed status checks ... 2745
Review status check information ... 2746
Retrieve the system logs ... 2747
Troubleshoot system log errors for Linux instances ... 2747
Out of memory: kill process ... 2749
ERROR: mmu_update failed (Memory management update failed) ... 2750
I/O error (block device failure) ... 2750
I/O ERROR: neither local nor remote disk (Broken distributed block device) ... 2752

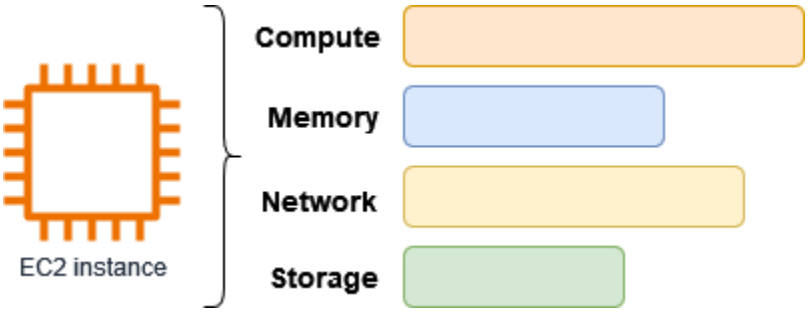
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions) ... 2753
"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch) ... 2754
"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules) ... 2755
ERROR Invalid kernel (EC2 incompatible kernel) ... 2757
fsck: No such file or directory while trying to open... (File system not found) ... 2758
General error mounting filesystems (failed mount) ... 2760
VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch) ... 2763
Error: Unable to determine major/minor number of root device... (Root file system/device mismatch) ... 2764
XENBUS: Device with no driver... ... 2765
... days without being checked, check forced (File system check required) ... 2767
fsck died with exit status... (Missing device) ... 2767
GRUB prompt (grubdom>) ... 2768
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address) ... 2771
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration) ... 2773
XENBUS: Timeout connecting to devices (Xenbus timeout) ... 2774
Linux instance boots from wrong volume ... 2775
Windows instance RDP issues ... 2777
Remote Desktop can't connect to the remote computer ... 2777
Error using the macOS RDP client ... 2781
RDP displays a black screen instead of the desktop ... 2782
Unable to remotely log on to an instance with a user that is not an administrator ... 2782
Troubleshooting Remote Desktop issues using AWS Systems Manager ... 2782
Enable Remote Desktop on an EC2 instance with remote registry ... 2786
I've lost my private key. How can I connect to my Windows instance? ... 2788
Windows instance start issues ... 2788
"Password is not available" ... 2789
"Password not available yet" ... 2790
"Cannot retrieve Windows password" ... 2790
"Waiting for the metadata service" ... 2790
"Unable to activate Windows" ... 2795
"Windows is not genuine (0x80070005)" ... 2797

"No Terminal Server License Servers available to provide a license" 2797
"Some settings are managed by your organization" 2797
Windows instance issues 2798
EBS volumes don't initialize on Windows Server 2016 and 2019 2799
Boot an EC2 Windows instance into Directory Services Restore Mode (DSRM) 2800
Instance loses network connectivity or scheduled tasks don't run when expected 2803
Unable to get console output 2803
Windows Server 2012 R2 not available on the network 2804
Disk signature collision 2804
Reset Windows administrator password 2805
Reset password using EC2Launch v2 2807
Reset password using EC2Launch 2812
Reset password using EC2Config 2817
Troubleshoot Sysprep issues 2823
EC2Rescue for Linux instances 2824
Install EC2Rescue 2825
Run EC2Rescue commands 2829
Develop EC2Rescue modules 2832
EC2Rescue for Windows instances 2839
Troubleshoot using EC2Rescue GUI 2840
Troubleshoot using EC2Rescue CLI 2846
Troubleshoot using EC2Rescue and Systems Manager 2854
EC2 Serial Console 2858
Prerequisites 2859
Configure access to the EC2 Serial Console 2866
Connect to the EC2 Serial Console 2875
Disconnect from the EC2 Serial Console 2884
Troubleshoot your instance using the EC2 Serial Console 2885
Send diagnostic interrupts 2894
Supported instance types 2895
Prerequisites 2895
Send a diagnostic interrupt 2899
Document history 2900
History for 2018 and earlier 2925

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

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



Features of Amazon EC2

Amazon EC2 provides the following high-level features:

Instances

Virtual servers.

Amazon Machine Images (AMIs)

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

Instance types

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

Amazon EBS volumes

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

Instance store volumes

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

Key pairs

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

Security groups

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

Amazon EC2 supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see PCI DSS Level 1.

Related services

Services to use with Amazon EC2

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

Amazon EC2 Auto Scaling

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

AWS Backup

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

Amazon CloudWatch

Monitor your instances and Amazon EBS volumes.

Elastic Load Balancing

Automatically distribute incoming application traffic across multiple instances.

Amazon GuardDuty

Detect potentially unauthorized or malicious use of your EC2 instances.

EC2 Image Builder

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

AWS Launch Wizard

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

AWS Systems Manager

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

Additional compute services

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

Amazon Lightsail

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

Amazon Elastic Container Service (Amazon ECS)

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

Amazon Elastic Kubernetes Service (Amazon EKS)

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

Access Amazon EC2

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

Amazon EC2 console

A simple web interface to create and manage Amazon EC2 instances and resources. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

AWS Command Line Interface

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

AWS CloudFormation

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

AWS SDKs

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

AWS Tools for PowerShell

A set of PowerShell modules that are built on the functionality exposed by the 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.

Query API

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

Pricing for Amazon EC2

Amazon EC2 provides the following pricing options:

Free Tier

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

On-Demand Instances

Pay for the instances that you use by the second, with a minimum of 60 seconds, with no long-term commitments or upfront payments.

Savings Plans

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

Reserved Instances

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

Spot Instances

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

Dedicated Hosts

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

On-Demand Capacity Reservations

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

Per-second billing

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

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

Estimates, billing, and cost optimization

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

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

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

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

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

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

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

Resources

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

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

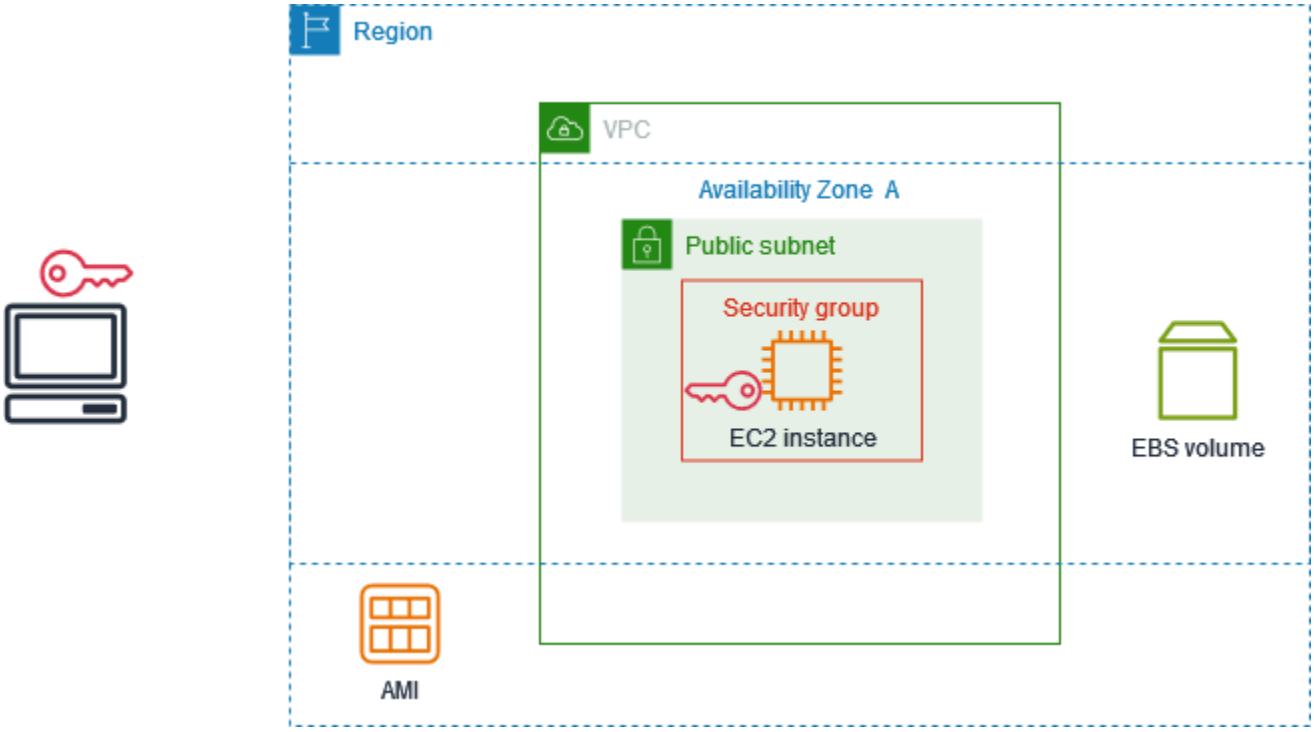
Get started with Amazon EC2

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

Overview

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

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



Cost for this tutorial

When you 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.

For instructions to determine whether you are eligible for the Free Tier, see the section called "Track your Free Tier usage".

Tasks

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

Step 1: Launch an instance

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

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, we display the current AWS Region — for example, **Ohio**. You can use the selected Region, or optionally select a Region that is closer to you.
3. From the EC2 console dashboard, in the **Launch instance** pane, choose **Launch instance**.
4. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
5. Under **Application and OS Images (Amazon Machine Image)**, do the following:
   a. Choose **Quick Start**, and then choose the operating system (OS) for your instance. For your first Linux instance, we recommend that you choose Amazon Linux.
   b. From **Amazon Machine Image (AMI)**, select an AMI that is marked **Free Tier eligible**.
6. Under **Instance type**, for **Instance type**, choose `t2.micro`, which is eligible for the Free Tier. In Regions where `t2.micro` is not available, `t3.micro` is eligible for the Free Tier.
7. Under **Key pair (login)**, for **Key pair name**, choose an existing key pair or choose **Create new key pair** to create your first key pair.

   > **Warning**
   >
   > If you choose **Proceed without a key pair (Not recommended)**, you won't be able to connect to your instance using the methods described in this tutorial.

8. Under **Network settings**, notice that we selected your default VPC, selected the option to use the default subnet in an Availability Zone that we choose for you, and configured a security group with a rule that allows connections to your instance from anywhere. For your first instance, we recommend that you use the default settings. Otherwise, you can update your network settings as follows:
   - (Optional) To use a specific default subnet, choose **Edit** and then choose a subnet.

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

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

Step 2: Connect to your instance

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

Linux instances

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

To connect to your instance using SSH

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and then choose **Connect**.

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

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

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

   The following is an example response:

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

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

   The following is an example response:

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

Windows instances

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

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.

To retrieve the initial administrator password

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

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

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

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

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

To connect to a Windows instance using an RDP client

1. On the **Connect to instance** page, choose **Download remote desktop file**. When the file download is finished, choose **Cancel** to return to the **Instances** page. The RDP file is downloaded to your `Downloads` folder.
2. Run `mstsc.exe` to open the RDP client.
3. Expand **Show options**, choose **Open**, and select the .rdp file from your `Downloads` folder.
4. By default, **Computer** is the public IPv4 DNS name of the instance and **User name** is the administrator account. To connect to the instance using IPv6 instead, replace the public IPv4 DNS name of the instance with its IPv6 address. Review the default settings and change them as needed.
5. Choose **Connect**. If you receive a warning that the publisher of the remote connection is unknown, choose **Connect** to continue.
6. Enter the password that you saved previously, and then choose **OK**.
7. Due to the nature of self-signed certificates, you might get a warning that the security certificate could not be authenticated. Do one of the following:
   - If you trust the certificate, choose **Yes** to connect to your instance.
   - [Windows] Before you proceed, compare the thumbprint of the certificate with the value in the system log to confirm the identity of the remote computer. Choose **View certificate** and then choose **Thumbprint** from the **Details** tab. Compare this value to the value of `RDPCERTIFICATE-THUMBPRINT` in **Actions**, **Monitor and troubleshoot**, **Get system log**.
   - [Mac OS X] Before you proceed, compare the fingerprint of the certificate with the value in the system log to confirm the identity of the remote computer. Choose **Show Certificate**, expand **Details**, and choose **SHA1 Fingerprints**. Compare this value to the value of `RDPCERTIFICATE-THUMBPRINT` in **Actions**, **Monitor and troubleshoot**, **Get system log**.
8. If the RDP connection is successful, the RDP client displays the Windows login screen and then the Windows desktop. If you receive an error message instead, see the section called "Remote Desktop can't connect to the remote computer". When you are finished with the RDP connection, you can close the RDP client.

Step 3: Clean up your instance

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up, see Next steps.

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

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

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Instance state**, **Terminate 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 explore the following next steps:

- Learn how to track your Amazon EC2 Free Tier usage using the console. For more information, see the section called “Track your Free Tier usage”.
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see Tracking your AWS Free Tier usage in the *AWS Billing User Guide*.

- Add an EBS volume. For more information, see Create an Amazon EBS volume in the *Amazon EBS User Guide*.
- Learn how to remotely manage your EC2 instance using the Run command. For more information, see AWS Systems Manager Run Command in the *AWS Systems Manager User Guide*.
- Learn about instance purchasing options. For more information, see Amazon EC2 billing and purchasing options.
- Get advice about instance types. For more information, see Get recommendations from EC2 instance type finder.

Best practices for Amazon EC2

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

Security

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

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see Root device type.
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination. For more information, see Preserve data when an instance is terminated.
- Use the instance store available for your instance to store temporary data. Remember that the data stored in instance store is deleted when you stop, hibernate, or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.
- Encrypt EBS volumes and snapshots. For more information, see Amazon EBS encryption in the *Amazon EBS User Guide*.

Resource management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see Use instance metadata to manage your EC2 instance and Tag your Amazon EC2 resources.
- View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see Amazon EC2 service quotas.
- Use AWS Trusted Advisor to inspect your AWS environment, and then make recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. For more information, see AWS Trusted Advisor in the *AWS Support User Guide*.

Backup and recovery

- Regularly back up your EBS volumes using Amazon EBS snapshots, and create an Amazon Machine Image (AMI) from your instance to save the configuration as a template for launching future instances. For more information about AWS services that help achieve this use case, see AWS Backup and Amazon Data Lifecycle Manager.
- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.
- Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see Amazon EC2 instance IP addressing.
- Monitor and respond to events. For more information, see Monitor Amazon EC2 resources.
- Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see Elastic network interfaces. For an automated solution, you can use Amazon EC2 Auto Scaling. For more information, see the Amazon EC2 Auto Scaling User Guide.
- Regularly test the process of recovering your instances and Amazon EBS volumes to ensure data and services are restored successfully.

Networking

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

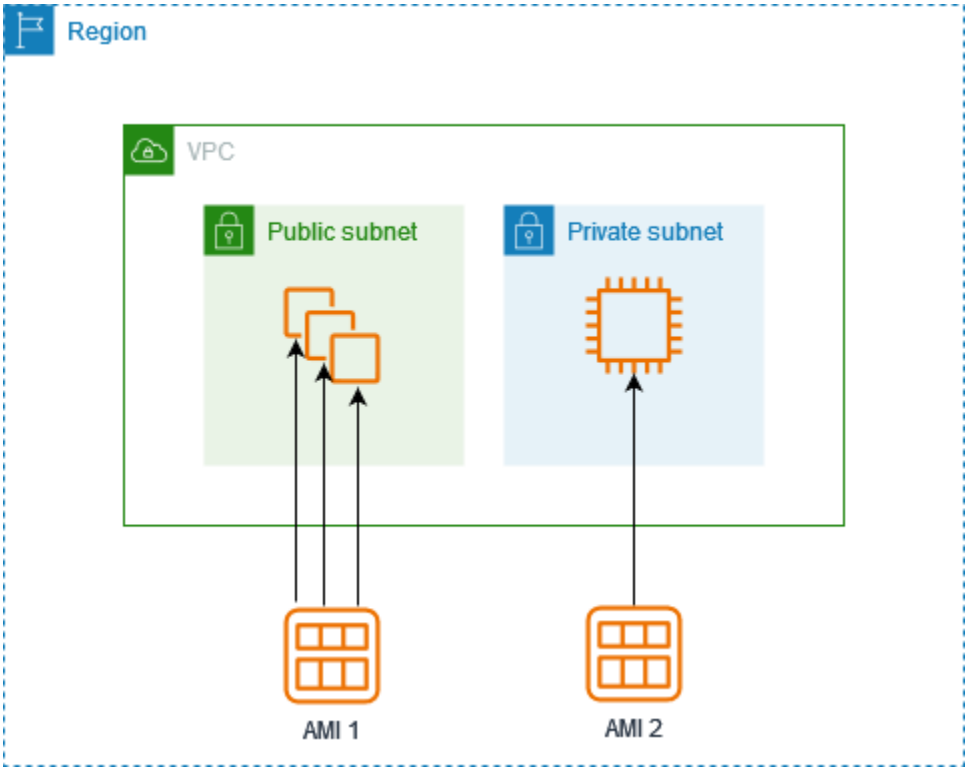
Amazon Machine Images in Amazon EC2

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

An AMI is specific to the following:

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

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



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

Contents

- AMI types and characteristics in Amazon EC2
- Find an AMI that meets the requirements for your EC2 instance
- Paid AMIs in the AWS Marketplace for Amazon EC2 instances
- Amazon EC2 AMI lifecycle
- Instance launch behavior with Amazon EC2 boot modes
- Use encryption with EBS-backed AMIs
- Understand shared AMI usage in Amazon EC2
- Monitor AMI events using Amazon EventBridge
- Understand AMI billing information
- AMI quotas in Amazon EC2

AMI types and characteristics in Amazon EC2

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

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

Launch permissions

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

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

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

Root device type

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

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

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

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

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

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
	You're charged for instance usage, EBS volume usage, and storing your AMI as an EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools

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

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

Determine the root device type of your AMI

The AMI that you use to launch an EC2 instance determines the type of the root volume. The root volume of an EC2 instance is either an EBS volume or an instance store volume. Current generation instance types support only EBS root volumes. The only instance types that support instance store root volumes are C1, C3, D2, I2, M1, M2, M3, R3, and X1.

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

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**, and select the AMI.
3. On the **Details** tab, check the value of **Root device type** as follows:
   - `ebs` – This is an EBS-backed AMI.
   - `instance store` – This is an an instance store-backed AMI.

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

You can use one of the following commands.

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

Virtualization types

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

The following table compares HVM and PV AMIs.

Characteristic	HVM	PV
Description	HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.	PV AMIs boot with a special boot loader called PV-GRUB, which starts the boot cycle and then chain loads the kernel specified in the `menu.lst` file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization. For more information about PV-GRUB and its use in Amazon EC2, see User provided kernels.
Supported instance types	All current generation instance types support HVM AMIs.	The following previous generation instance types support PV AMIs: C1, C3, M1, M3, M2, and T1. Current generation instance types do not support PV AMIs.

Characteristic	HVM	PV
Support for hardware extensions	HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. They are required to use enhanced networking and GPU processing. To pass through instructions to specialized network and GPU devices, the OS must have access to the native hardware platform, and HVM virtualization provides this access. For more information, see Enhanced networking on Amazon EC2 instances.	No, they can't take advantage of special hardware extensions such as enhanced networking or GPU processing.
How to find	Verify that the virtualization type of the AMI is set to `hvm`, using the console or the describe-images command.	Verify that the virtualization type of the AMI is set to `paravirtual` , using the console or the describe-images command.

PV on HVM

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

Find an AMI that meets the requirements for your EC2 instance

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

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

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

To find an Amazon Linux 2023 AMI, see AL2023 on Amazon EC2 in the *Amazon Linux 2023 User Guide*.

To find an Ubuntu AMI, see Amazon EC2 AMI Locator on the *Canonical Ubuntu website*.

To find a RHEL AMI, see Red Hat Enterprise Linux Images (AMI) Available on Amazon Web Services (AWS) on the *Red Hat website*.

There are various ways to find an AMI that meets your needs. You can find an AMI using the Amazon EC2 console, AWS CLI, AWS Tools for Windows PowerShell, and AWS Systems Manager.

Find an AMI using the Amazon EC2 console

You can find 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.

To find an AMI using the launch instance wizard

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location. AMI IDs are unique to each AWS Region.

3. From the console dashboard, choose **Launch instance**.
4. (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. .

To find an AMI using the AMIs page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location. AMI IDs are unique to each AWS Region.
3. In the navigation pane, choose **AMIs**.
4. (Optional) Use the filter and search options to scope the list of displayed AMIs to see only the AMIs that match your criteria.

   For example, to list all AMIs provided by AWS, choose **Public images**. Then use the search options to further scope the list of displayed AMIs. Choose the **Search** bar and, from the menu, choose **Owner alias**, then the **=** operator, and then the value **amazon**. To find AMIs that match a specific platform, for example Linux or Windows, choose the **Search** bar again to choose **Platform**, then the **=** operator, and then the operating system from the list provided.
5. (Optional) Choose the **Preferences** icon to select which image attributes to display, such as the root device type. Alternatively, you can select an AMI from the list and view its properties on the **Details** tab.
6. Before you select an AMI, it's important that you check whether it's backed by instance store or by Amazon EBS and that you are aware of the effects of this difference. For more information, see Root device type.
7. To launch an instance from this AMI, select it and then choose **Launch instance from image**. For more information about launching an instance using the console, see Launch an EC2 instance using the launch instance wizard in the console. If you're not ready to launch the instance now, make note of the AMI ID for later.

Find an AMI using the AWS CLI

You can use the describe-images AWS CLI command to list only the AMIs that match your requirements. After locating an AMI that matches your requirements, make note of its ID so that you can use it to launch instances. For more information, see Launch your instance in the *AWS Command Line Interface User Guide*.

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

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

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

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

You can add the following filter to the previous command to display only AMIs backed by Amazon EBS.

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

> **Important**
>
> If you omit the `--owners` parameter from the `describe-images` command, all images are returned for which you have launch permissions, regardless of ownership.

Find an AMI using the AWS Tools for Windows PowerShell

You can use PowerShell cmdlets to list only the Windows AMIs that match your requirements. For information and examples, see Find an Amazon Machine Image Using Windows PowerShell in the *AWS Tools for Windows PowerShell User Guide*.

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 Amazon EC2 Instance Using Windows PowerShell in the *AWS Tools for Windows PowerShell User Guide*.

Find an AMI using a Systems Manager parameter

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

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

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

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 the following permissions to your IAM policy:

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

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

Limitations

AMIs and Systems Manager parameters are Region specific. To use the same Systems Manager parameter name across Regions, create a Systems Manager parameter in each Region with the

same name (for example, `golden-ami`). In each Region, point the Systems Manager parameter to an AMI in that Region.

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 an 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 EC2 instance using the launch instance wizard in the console.

Old console

To find an 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).