Contents

[1.0 EC2 i](#_Toc523482131)

[1.1 Accessing Amazon EC2 i](#_Toc523482132)

[1.2 Instances and AMIs i](#_Toc523482133)

[1.2.1 Storage for Your Instance ii](#_Toc523482134)

[1.2.2 Security Best Practices ii](#_Toc523482135)

[1.2.3 AMIs ii](#_Toc523482136)

[1.2.4 Regions and Availability Zones ii](#_Toc523482137)

[1.2.5 To find the Regions and Availity Zones using command line ii](#_Toc523482138)

[1.2.6 Launching Instances in an Availability Zone iii](#_Toc523482139)

[1.2.7 Migrating an Instance to Another Availability Zone iii](#_Toc523482140)

[1.2.8 Amazon EC2 Root Device Volume iii](#_Toc523482141)

[1.2.9 Root Device Storage Concepts iii](#_Toc523482142)

[1.2.10 Choosing an AMI by Root Device Type v](#_Toc523482143)

[1.2.11 Changing the Root Device Volume to Persist v](#_Toc523482144)

<https://aws.amazon.com/documentation/>

# EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster.

## Accessing Amazon EC2

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

For command line interface you have the following two options

* AWS Command Line Interface (CLI)
* AWS Tools for Windows PowerShell

## Instances and AMIs

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). . From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

You can launch different types of instances from a single AMI. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities.

### Storage for Your Instance

The root device for your instance contains the image used to boot the instance. Additionally, your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping. After these volumes have been added to and mapped on your instance, they are available for you to mount and use. To keep important data safe, you should use a replication strategy across multiple instances, or store your persistent data in Amazon S3 or Amazon EBS volumes.

### Security Best Practices

* Use AWS Identity and Access Management (IAM) to control access to your AWS resources, including your instances.
* Restrict access by only allowing trusted hosts or networks to access ports on your instance
* Review the rules in your security groups regularly, and ensure that you apply the principle of least privilege—only open up permissions that you require.
* Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

### AMIs

Amazon Web Services (AWS) publishes many Amazon Machine Images (AMI) that contain common software configurations for public use.

All AMIs are categorized as either backed by Amazon EBS, which means that the root device for an instance launched from the AMI is an Amazon EBS volume, or backed by instance store, which means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

### Regions and Availability Zones

Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones. Amazon EC2 provides you the ability to place resources, such as instances, and data in multiple locations.

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links

### To find the Regions and Availity Zones using command line

* describes regions for your account
  + aws ec2 describe-regions
  + Get-EC2Region
* describe-availability-zones
  + aws ec2 describe-availability-zones --region region-name
  + Get-EC2AvailabilityZone -Region region-name

### Launching 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 you have. By launching your instances in separate Availability Zones, you can protect your applications from the failure of a single location.

### Migrating an Instance to Another Availability Zone

The migration process involves creating an AMI from the original instance, launching an instance in the new Availability Zone, and updating the configuration of the new instance.

Steps:

* Create an AMI from the instance
* 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
* Launch an instance from the AMI that you just created, specifying the new Availability Zone or subnet.
* If the original instance has an associated Elastic IP address, associate it with the new instance.
* If the original instance is a Reserved Instance, change the Availability Zone for your reservation.

### Amazon EC2 Root Device Volume

When you launch an instance, the root device volume contains the image used to boot the instance.

When we introduced Amazon EC2, all AMIs were backed by Amazon EC2 instance store, which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

After we introduced Amazon EBS, we introduced AMIs that are backed by Amazon EBS. This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.

So now, you can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS. We recommend that you use AMIs backed by Amazon EBS, because they launch faster and use persistent storage.

### Root Device Storage Concepts

You can launch an instance from either an instance store-backed AMI or an Amazon EBS-backed AMI.

*Instance Store Backed-up Instance*

Instances that use instance stores for the root device automatically have one or more instance store volumes available, with one volume serving as the root device volume. When an instance is launched, the image that is used to boot the instance is copied to the root volume. Note that you can optionally use additional instance store volumes, depending on the instance type.

Any data on the instance store volumes persists as long as the instance is running, but this data is deleted when the instance is terminated.



*Amazon EBS-backed Instance.*

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. When you launch an Amazon EBS-backed instance, we create an Amazon EBS volume for each Amazon EBS snapshot referenced by the AMI you use.



### Choosing an AMI by Root Device Type

* Open the Amazon EC2 console.
* In the navigation pane, choose AMIs
* From the filter lists, select the image type (such as Public images).
* In the search bar choose Platform to select the operating system (such as Amazon Linux
* Select the Root Device Type: could be EBS Image or Inastance Store.
* Choose an AMI and write down its AMI ID.

### Changing the Root Device Volume to Persist

*Using User Interface*

By default, the root device volume for an AMI backed by Amazon EBS is deleted when the instance terminates. To change the default behavior, set the DeleteOnTermination attribute to false using a block device mapping.

*Console*

# Setting Up with Amazon EC2

<https://aws.amazon.com/>

## Create an IAM User

* Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at <https://console.aws.amazon.com/iam/>
* In the navigation pane of the console, choose Users, and then choose Add user.
* For User name, type Administrator
* Select the check box next to AWS Management Console access, select Custom password, and then type the new user's password in the text box.
* Choose **Next: Permissions**
* On the **Set permissions** page, choose **Add user to group**
* Choose Create group.
* In the Create group dialog box, for Group name type Administrators
* For Filter policies, select the check box for AWS managed - job function
* In the policy list, select the check box for AdministratorAccess. Then choose Create group.
* Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.
* Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where your\_aws\_account\_id is your AWS account number

[**https://*your\_aws\_account\_id*.signin.aws.amazon.com/console/**](https://your_aws_account_id.signin.aws.amazon.com/console/)

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM console, choose Dashboard in the navigation pane. From the dashboard, choose Customize and enter an alias such as your company name. To sign in

[**https://*your\_account\_alias*.signin.aws.amazon.com/console/**](https://your_account_alias.signin.aws.amazon.com/console/)

## 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 when you log in using SSH.

* Sign in to AWS using the URL that you created in the previous section
* From the AWS dashboard, choose **EC2** to open the Amazon EC2 console.
* From the navigation bar, select a region for the key pair
* In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
* Choose create a key pair.
* Enter a new of the key-pair,
* Private key is automatically doenloaded, with an extension .pem. Save the proviate key.
* Change the permission of the key if needed.

**chmod 400 *your\_user\_name*-key-pair-*region\_name*.pem**

## Connecting to your instance using the Key-pair.

* Download and install putty. Make sure you install the entire suit.
* Start Puttygen
* Under Type of key to generate, choose RSA.
* Choose Load. Locate the .pem file
* Save the private key. This will save the private key with the extension .ppk

## Create a Virtual Private Cloud

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. To create a non-default VPC.

* Open the Amazon VPC at <https://console.aws.amazon.com/vpc/>
* Select the region from the navigaton bar.
* Choose launch VPC platform.
* On the Step 1: Select a VPC Configuration page, ensure that VPC with a Single Public Subnet is selected, and choose Select
* On the Step 2: VPC with a Single Public Subnet page, enter a friendly name for your VPC in the VPC name field. Leave the other default configuration settings, and choose Create VPC. On the confirmation page, choose OK.

<https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>

### Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Amazon VPC is the networking layer for Amazon EC2.

### VPCs and Subnets

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud.

A **subnet** is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.

To create a non-default VPC

* Open Amazon console.
* Select the region
* Lunch VPC dashboard
* On the Step 1: Select a VPC Configuration page, ensure that VPC with a Single Public Subnet is selected, and choose Select.
* On the Step 2: VPC with a Single Public Subnet page, enter a friendly name for your VPC in the VPC name field. Leave the other default configuration settings, and choose Create VPC.

To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

<https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>

### Create a security group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level.

If you paln to lunch the instance in multiple regions, you’ll need to crete security groups in multiple regions.

You will need the public IPv4 address of your local computer.

To create a security group

* Open Amazon EC2 console
* Select the regions from the naviagation bar
* Choose Create Security group
* Enter the name of the security group
* Select the VPC, in case of default VPC, the default is already selected.