

AWS Management Console

AWS Management Console is a central place to create your cloud infrastructure. You can access all the services available in Amazon Web Services(AWS) from the management console. Using these services you can recreate your traditional IT infrastructure in the cloud. By creating your infrastructure in cloud, you can gain the benefits of high-availability, low running costs, scalability, and security etc.

For example, you can create your own virtual server (called as an EC2 instance) with a Linux operating system and then access it using secure-shell(SSH) from Windows or Mac. You can also create other things like serverless apps, virtual private cloud (VPC), and functional database etc.

This article explains how you can create a virtual server with Linux OS and then connect to it using Terminal in Mac.

EC2 Instances or Virtual Servers

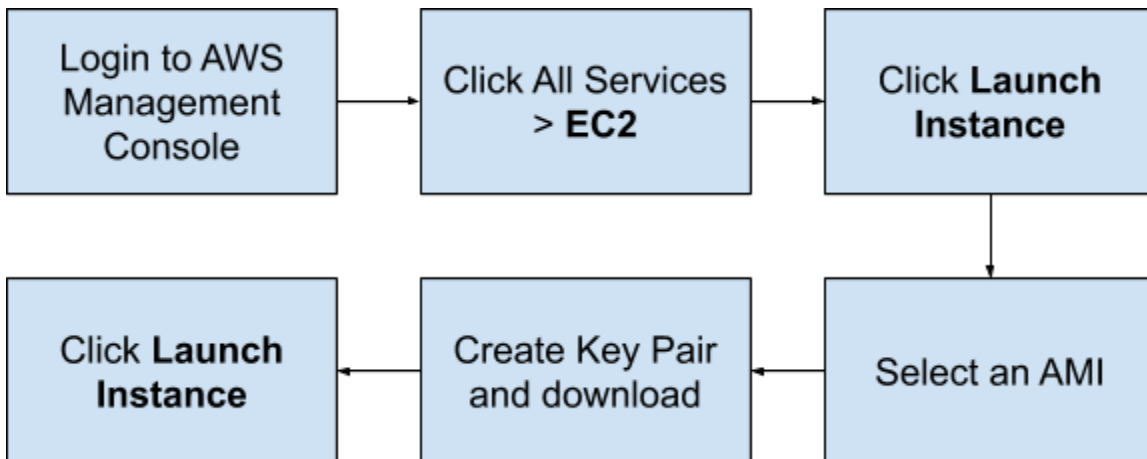
The starting point of creating your IT infrastructure is to create/launch EC2 instances. Amazon Elastic Compute Cloud (EC2) instances are simply virtual servers. You can launch and configure as many EC2 instances as you need in the AWS Cloud. The EC2 instances have an excellent mechanism to scale the resources like storage and computing capacity on-demand. The EC2 instances also have preconfigured templates that contain the operating system and additional software.

EBS or Storage Volumes

Amazon Elastic Block Store (EBS) in the AWS cloud is equivalent to a hard disk in a physical machine. EBS represents the storage part for your virtual server (EC2 instance). You can use an EBS block and mount it as a volume in an EC2 instance. You can also dynamically change the configuration of these blocks when required.

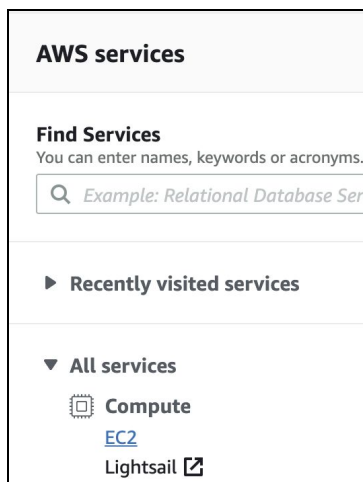
Launch an Amazon Machine Image(AMI) with Linux OS

The following image shows the process of launching an Linux AMI from AWS Management Console.



To launch an Amazon Machine Image (AMI) from AWS Management Console:

1. Login to AWS Management Console.
2. Under All Services, click **EC2**. The EC2 Dashboard appears.



3. Under the **Launch Instance** section, click **Launch Instance**. The Launch Instance Wizard appears.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Search by Systems Manager parameter

Quick Start 1 to 40 of 40 AMIs

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only

Amazon Linux
Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0603cbe34fd08cb81 (64-bit x86) / ami-07b66785f4f646ccf (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

☒ 64-bit (x86)
☐ 64-bit (Arm)

Amazon Linux
Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-04fcd96153cb57194

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl,

Select

64-bit (x86)

- Click **Select** to select an AMI from the list. The Choose an Instance Type screen appears.
- Select an instance type from the available list and click **Review and Launch**.
- Review the information on the **Review and Launch** screen.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details Edit AMI

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0603cbe34fd08cb81

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

Instance Type Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups Edit security groups

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2020-09-13T13:59:51.915+05:30

Type	Protocol	Port Range	Source	Description
------	----------	------------	--------	-------------

Cancel Previous **Launch**

- Click **Launch**. The instance will be launched and available for use.

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
Firstlinuxinstance	i-0cdade14cd46a5703	t2.micro	us-east-2a	stopped		None		

- Select **Create a new key pair** and assign a name. The key pair file (.pem) will download automatically.
 - Ensure to save this in a safe place as you can use it later to log in to the instance.

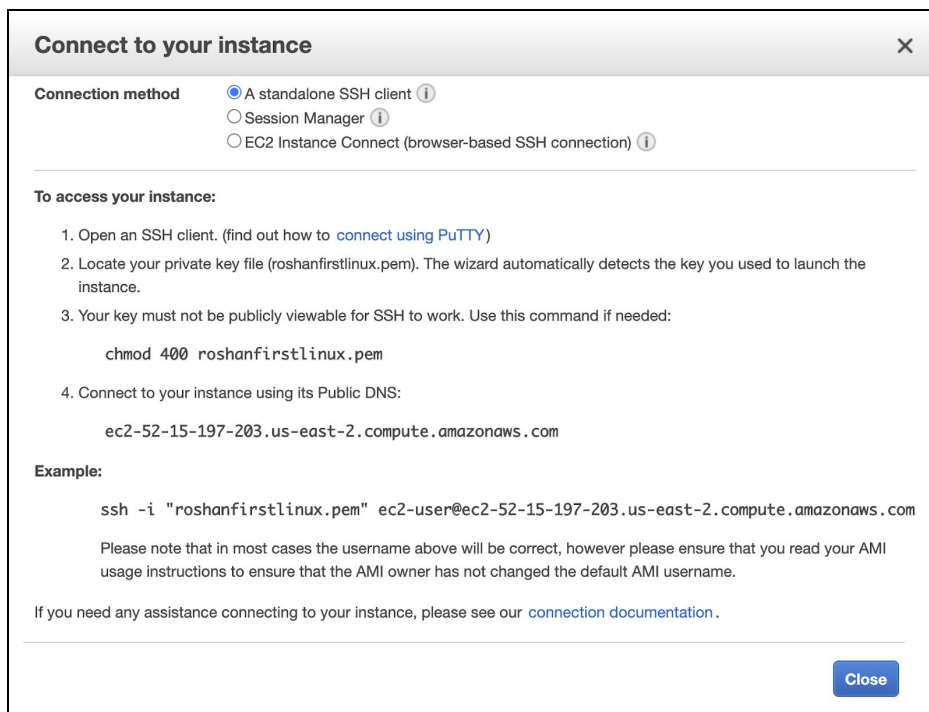
9. Click **Launch Instances** to complete the set up. After you run the instance you can connect using SSH or Putty and work with the Linux server.

Connect to Linux AMI using SSH

Once you have the Linux AMI running in AWS cloud, you can connect to the instance directly from the Mac Terminal.

To connect from the Terminal to Linux AMI:

1. In the AWS console, start your EC2 instance.
2. Click **Connect**. The **Connect to your instance** dialog appears.



3. Copy the command below **Example:** section. For example, the command can be like

```
ssh -i "roshanfirstlinux.pem"  
ec2-user@ec2-52-15-197-203.us-east-2.compute.amazonaws.com.
```
4. In Mac, open Terminal.
5. Navigate to the folder where the .pem file is placed.
6. Run the command starting with `ssh`.

```
Downloads — ec2-user@ip-172-31-8-20:~ — ssh -i roshanfirstlinux.pem ec2-user@ec2-52-15-197-203.us-east-2.compute.amazonaws.com
Last login: Sun Sep 13 12:06:32 on console
roshanchittoorrahim@Roshans-MacBook-Pro ~ % cd Downloads
roshanchittoorrahim@Roshans-MacBook-Pro Downloads % ssh -i "roshanfirstlinux.pem" ec2-user@ec2-52-15-197-203.us-east-2.compute.amazonaws.com
Warning: Permanently added 'ec2-52-15-197-203.us-east-2.compute.amazonaws.com,52.15.197.203' (ECDSA)
Last login: Sat Sep 12 17:49:16 2020 from 157.48.175.74
ec2-user@ip-172-31-8-20:~$
https://aws.amazon.com/amazon-linux-2/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-8-20 ~]$
```

Connect to your instance

Connection method: A standalone SSH client

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (roshanfirstlinux.pem). The wizard automatically locates the key file.
3. Your key must not be publicly viewable for SSH to work. Use this command to check: `chmod 400 roshanfirstlinux.pem`
4. Connect to your instance using its Public DNS: `ec2-52-15-197-203.us-east-2.compute.amazonaws.com`

7. Type **yes** to confirm the connection. A confirmation message appears and shows a success message.
8. The command line shows a prompt “ec2-user” and you can confirm that you are connected to the linux AMI instance.