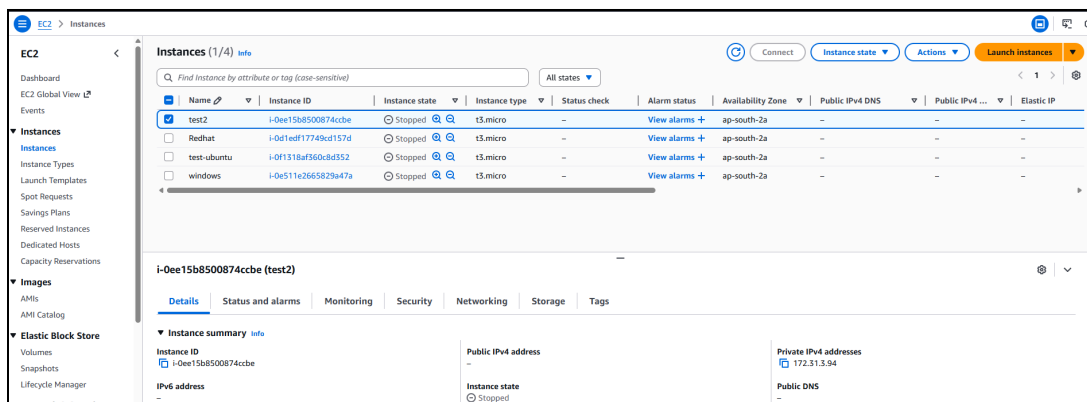


Week 2 : Assignment

How to Deploy and Connect to Amazon Linux, Ubuntu, and Windows EC2 Instances

Amazon EC2 (Elastic Compute Cloud) allows users to deploy scalable virtual machines in the AWS cloud. In this article, we will discuss how to launch and connect to three types of EC2 instances—**Amazon Linux**, **Ubuntu**, and **Windows Server**—along with detailed steps on **Security Group** configuration, **Key Pairs**, and remote access tools such as **EC2 Instance Connect**, **MobaXterm**, **PuTTY**, and **RDP**.

1. Deploying and Connecting to an Amazon Linux EC2 Instance

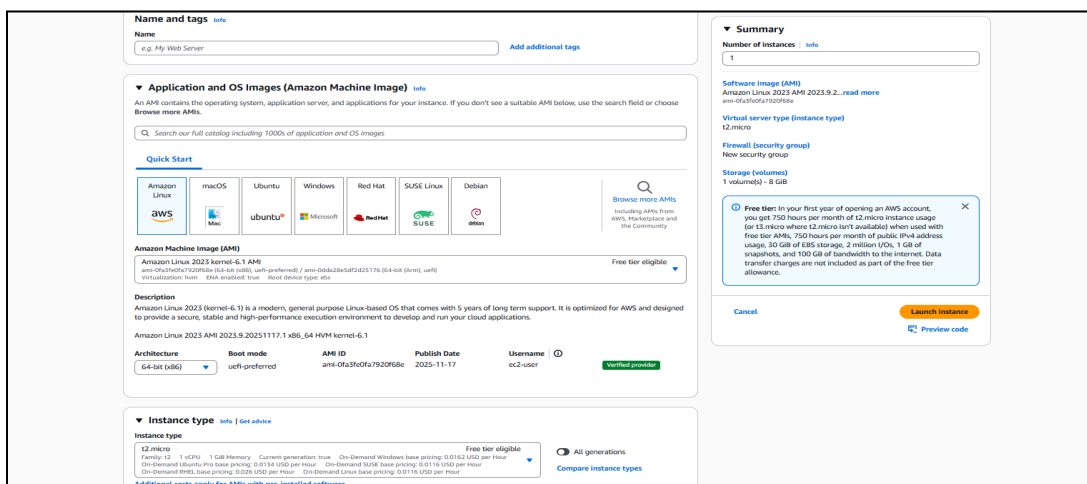


The screenshot displays the AWS Management Console's EC2 Instances page. At the top, there's a search bar and buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below this is a table listing instances. The instance 'test2' is selected, and its details are shown below the table. The details include the instance ID, name, state (Stopped), instance type (t3.micro), and various addresses (Public IPv4, Private IPv4, Public DNS).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4	Elastic IP
test2	i-0ee15b8500874ccb2	Stopped	t3.micro	–	View alarms +	ap-south-2a	–	–	–
Redhat	i-0d1edf17749cd157d	Stopped	t3.micro	–	View alarms +	ap-south-2a	–	–	–
test-ubuntu	i-0f1318af560cd8d552	Stopped	t3.micro	–	View alarms +	ap-south-2a	–	–	–
windows	i-0e511e2665829a47a	Stopped	t3.micro	–	View alarms +	ap-south-2a	–	–	–

test2 Details:

- Instance ID: i-0ee15b8500874ccb2
- Instance name: test2
- Instance state: Stopped
- Instance type: t3.micro
- Public IPv4 address: –
- Private IPv4 address: 172.31.3.94
- Public DNS: –



The screenshot shows the 'Launch instances' wizard in the AWS Management Console, specifically the 'Application and OS Images' step. It displays a search bar for AMIs, a list of quick start options (Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian), and a detailed view of the selected Amazon Linux 2023 AMI. The summary on the right shows the instance type (t3.micro), security group, and storage.

Application and OS Images (Amazon Machine Image)

Search our full catalog including 1000s of application and OS images

Quick Start

- Amazon Linux
- macOS
- Ubuntu
- Windows
- Red Hat
- SUSE Linux
- Debian

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0f3b0f7a72095de
Virtualization: hvm
Enclave: enabled
True
Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.9.2023.1117.1 x86_64 HVM kernel-6.1

Architecture: x86_64
Boot mode: uefi-preferred
AMI ID: ami-0f3b0f7a72095de
Publish Date: 2023-11-17
Username: ec2-user
Verified provider: Yes

Instance type

Instance type: t3.micro
Family: t3
vCPU: 1
Memory: 1 GB
Current generation: true
On-Demand Windows base pricing: 0.012 USD per hour
Free tier eligible: Yes
On-Demand Linux base pricing: 0.0118 USD per hour
On-Demand Linux base pricing: 0.0118 USD per hour
On-Demand Linux base pricing: 0.0118 USD per hour
Additional costs apply for AMIs with pre-installed software

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2...read more
ami-0f3b0f7a72095de

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t3.micro instance usage (or t3.micro where t3.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance.

Cancel **Launch instances** **Preview code**

1.1 Launching the Instance

1. Log in to **AWS Management Console** → Go to **EC2 Dashboard**.
2. Click **Launch Instance**.
3. Enter a name (e.g., *Amazon-Linux-Server*).
4. Select an AMI → **Amazon Linux 2023**.
5. Choose instance type → ex: **t2.micro** (Free Tier).
6. Create or select an existing **Key Pair** (for SSH login).
7. Configure **Network Settings** → Select VPC & Subnet.
8. Configure **Security Group** (explained below).
9. Launch the instance.

1.2 Security Group Configuration

sg-034e712f7eec653d3 - launch-wizard-1

Actions

Details

Security group name

launch-wizard-1

Security group ID

sg-034e712f7eec653d3

Description

launch-wizard-1 created 2025-12-03T14:49:52.722Z

VPC ID

vpc-0b115f16bcb92fcb8

Owner

900001851712

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (1)

Manage tags

Edit inbound rules

Search

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-01fd963a6cb07286d	IPv4	SSH	TCP	22	0.0.0.0/0	-

Inbound Rules (Allow Traffic In)

Type	Protocol	Port	Source	Description
SSH	TCP	22	Your IP	To connect via SSH

Outbound Rules (Allow Traffic Out)

Type	Protocol	Port	Destination
All traffic	All	All	0.0.0.0/0

1.3 Connecting to Amazon Linux

1.3.1 Using EC2 Instance Connect (Browser-based SSH)

1. Select the instance → Click **Connect**.
2. Choose **EC2 Instance Connect**.
3. Click **Connect** → Opens browser-based terminal.

1.3.2 Connecting via SSH (MobaXterm / PuTTY)

Using MobaXterm

1. Open MobaXterm → Session → SSH.
2. Enter Public IP of instance.
3. Username: **ec2-user**
4. Browse and select your **.pem key**.
5. Connect.

Using PuTTY

PuTTY requires converting **.pem** to **.ppk**.

1. Open **PuTTYgen** → Load **.pem** → Save as **.ppk**.
 2. Open PuTTY → Hostname = Public IP.
 3. Go to SSH → Auth → Browse **.ppk** file.
 4. Connect with username **ec2-user**.
-
-

2. Deploying and Connecting to an Ubuntu EC2 Instance

2.1 Launching the Instance

Steps are same as Amazon Linux except:

1. Log in to **AWS Management Console** → Go to **EC2 Dashboard**.
2. Click **Launch Instance**.
3. Enter a name (e.g., Ubuntu-Server).
4. Choose **Ubuntu Server AMI** (e.g., Ubuntu 22.04 LTS).
5. Choose instance type → ex: **t2.micro** (Free Tier).
6. Create or select an existing **Key Pair** (for SSH login).
7. Configure **Network Settings** → Select VPC & Subnet.
8. Configure **Security Group** (explained below).
9. Launch the instance.

2.2 Security Group Configuration

Inbound Rules

Type	Protocol	Port	Source
SSH	TCP	22	My IP

Outbound Rules

Type	Protocol	Port	Destination
All traffic	All	All	0.0.0.0/0

2.3 Connecting to Ubuntu

2.3.1 Using EC2 Instance Connect

1. Select the instance → Click **Connect**.
2. Choose **EC2 Instance Connect**.
3. Click **Connect** → Opens browser-based terminal.

2.3.2 Using MobaXterm

1. SSH session → IP address.
2. Username: **ubuntu**
3. Load **.pem** key.
4. Connect.

2.3.3 Using PuTTY

PuTTY requires converting **.pem** to **.ppk**.

- Open **PuTTYgen** → Load **.pem** → Save as **.ppk**.
 - Open PuTTY → Hostname = Public IP.
 - Go to SSH → Auth → Browse **.ppk** file.
 - Connect with username **ubuntu**.
-
-

3. Deploying and Connecting to a Windows Server EC2 Instance

3.1 Launching the Instance

1. Go to EC2 → Launch Instance.
 2. Select **Windows Server AMI** (Example: Windows Server 2022).
 3. Instance type → **t2.micro**.
 4. Create/select **Key Pair** (used to decrypt Windows password).
 5. Configure networking.
 6. Set **Security Group** (port 3389).
 7. Launch the instance.
-

3.2 Security Group Configuration

Inbound Rules

Type	Protocol	Port	Source	Description
RDP	TCP	3389	My IP	Remote Desktop Access

Outbound Rules

Default → All traffic allowed.

3.3 Connecting to Windows Instance with RDP

Step 1: Get Windows password

1. Select the instance → Click **Connect** → Choose **RDP Client**.
2. Click **Get Password**.
3. Upload your **.pem** key → Decrypt password.
4. Copy the password.

Step 2: Connect using RDP

On your local PC:

Windows

- Open **Remote Desktop Connection (mstsc)**.
- Enter Public IP → Connect.
- Username: **Administrator**
- Paste password.

Mac/Linux

- Use Microsoft Remote Desktop app.

4. Key Pair Explanation

- A **Key Pair** is used for secure login to EC2 servers.
- Contains:
 - **Private Key (.pem)** → stored by user.

- **Public Key** → stored in AWS instance.
- Linux servers use key pairs for **SSH authentication**.
- Windows servers use the key pair to **decrypt the Administrator password**.

Important rules:

- Never share `.pem` file.
- Keep backup securely.
- If lost → cannot retrieve → must create a new instance.

Remember

Instance Type	Connect Method	Default User	Port
Amazon Linux	SSH, EC2 Instance Connect	ec2-user	22
Ubuntu	SSH, EC2 Instance Connect	ubuntu	22
Windows Server	RDP	Administrator	3389
