

# 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 shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for Dashboard, Events, Instances (with sub-links for Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager, Networks & Security). The main content area has a header 'Instances (1/4) Info' with filters for 'Find instance by attribute or tag (case-sensitive)' and 'All states'. A table lists four instances: 'test2' (Stopped, t3.micro, ap-south-2a, 172.31.3.94, publicdns), 'Redhat' (Stopped, t3.micro, ap-south-2a, 172.31.3.94, publicdns), 'test-ubuntu' (Stopped, t3.micro, ap-south-2a, 172.31.3.94, publicdns), and 'windows' (Stopped, t3.micro, ap-south-2a, 172.31.3.94, publicdns). Below the table, a modal window is open for 'i-0ee15b8500874ccbe (test2)'. It shows tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under 'Details', it shows the Instance ID (i-0ee15b8500874ccbe), Public IPv4 address (172.31.3.94), and Instance state (Stopped). It also lists Private IPv4 addresses (172.31.3.94) and Public DNS (publicdns).

**Name and tags** Info  
Name: e.g. My Web Server  Add additional tags

**Application and OS Images (Amazon Machine Image)** Info  
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.  
 Search our full catalog including 1000s of application and OS images

**Quick Start**  
        Browse more AMIs including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**  
Amazon Linux 2023 kernel-6.1 AMI  
AMI identifier: ami-0f3fe0fa7920f6bc Free tier eligible  
Architecture: 64-bit (x86) Boot mode: uefi-preferred AMI ID: ami-0f3fe0fa7920f6bc Publish Date: 2025-11-17 Username: ec2-user Verified provider

**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.

**Instance type** Info Get advice  
Instance type: t2.micro Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Windows base pricing: 0.014 USD per Hour On-Demand Linux base pricing: 0.014 USD per Hour On-Demand Pre-emptible base pricing: 0.014 USD per Hour On-Demand T4G base pricing: 0.0118 USD per Hour On-Demand Linux base pricing: 0.0118 USD per Hour All generations Compare Instance types 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  
arn:aws:image:amazonlinux:2023.9.20251117.1

**Virtual server type (instance type)** (Optional)

**Firewall (Security group)**  
New security group

**Storage (Volume(s))**  
1 volume(s) - 8 GiB

**Free tier**: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage for free. You can also receive up to 1000 hours per month of 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 150 GiB of bandwidth to the internet. Data transfers to and from the internet are not included as part of the free tier allowance.

**Launch instance** 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

The screenshot shows the AWS Security Groups console for the security group **sg-034e712f7eec653d3 - launch-wizard-1**. The **Details** tab is selected, displaying information such as the security group name (**launch-wizard-1**), security group ID (**sg-034e712f7eec653d3**), owner (**900001851712**), description (**launch-wizard-1 created 2025-12-03T14:49:52Z**), and VPC ID (**vpc-0b115f16cb92fcbb**). The **Inbound rules** tab is active, showing one rule: **sgr-01fd963a6cb07286d** (IPv4), Type: **SSH**, Protocol: **TCP**, Port range: **22**, Source: **0.0.0.0/0**. There are also tabs for **Outbound rules**, **Sharing - new**, **VPC associations - new**, and **Tags**.

### 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

---