Mastering AWS EC2: A Beginner's Journey

If you're just starting with AWS, launching an **EC2 instance** is one of the first things you'll need to do. Think of EC2 as your **virtual computer in the cloud**—you can use it to host websites, run applications, or even set up a personal development environment.

In this guide, I'll walk you through everything you need to know, from **what EC2 is** to **setting up networking and security** and finally **connecting to your instance**. Let’s get started!

◇ What is Amazon EC2?

Amazon **Elastic Compute Cloud (EC2)** is a cloud-based virtual server that allows you to run applications without having to buy or maintain physical hardware. You can **start, stop, or scale** these instances as needed, paying only for what you use.

Step 1: Log in to AWS

1. Head over to **AWS Console**
2. Sign in to your AWS account
3. Navigate to the **EC2 Dashboard**

If you don’t have an AWS account yet, you’ll need to **create one** first. AWS provides a **free tier**, which lets you run a small EC2 instance for free for 12 months!

Step 2: Launch an EC2 Instance

Click on the **Launch Instance** button to start the setup.

You’ll need to enter a **name** for your instance (e.g., "MyFirstEC2"). This helps you identify your instances later.

Step 3: Choose an Amazon Machine Image (AMI)

An **AMI (Amazon Machine Image)** is like a **blueprint** for your instance. It determines what operating system (OS) and pre-installed software your EC2 instance will have.

Here are some common AMIs:

- **Amazon Linux** (Recommended for AWS users)
- **Ubuntu** (Good for developers)
- **Windows Server** (For Windows-based applications)
- **Red Hat Enterprise Linux (RHEL)** (Used in corporate environments)

For beginners, **Amazon Linux 2023** or **Ubuntu 22.04** is a solid choice.

Step 4: Select an Instance Type

AWS offers different **EC2 instance types** depending on your needs.

For example:

- **t2.micro** (1 vCPU, 1GB RAM) – Free tier eligible, great for small applications
- **t3.medium** (2 vCPUs, 4GB RAM) – More power for testing
- **m5.large** (2 vCPUs, 8GB RAM) – Good for production workloads

If you're using AWS Free Tier, go with **t2.micro**.

Step 5: Configure Network Settings (VPC, Subnet, and Security Group)

Now, we set up **networking**, which controls how your instance communicates with the internet.

- **VPC (Virtual Private Cloud)** – AWS will select the default VPC
- **Subnet** – Choose a region where your instance will be hosted
- **Auto-assign Public IP** – Enable this if you want to access your instance from the internet

🔥 Security Group (Firewall Settings)

A **security group** is like a firewall that controls traffic to and from your instance.

💡 **Add these inbound rules:**

- **SSH (Port 22, Source: My IP)** – Allows remote login

- **HTTP (Port 80, Source: Anywhere)** – Needed for web servers
- **HTTPS (Port 443, Source: Anywhere)** – For secure website access

Step 6: Create a Key Pair (For Secure Access)

AWS uses a **key pair** for security instead of passwords.

- If you don't have a key pair, click **Create New Key Pair**
- Choose **RSA** and **.pem** format
- **Download the key** and **store it safely** (you won't be able to download it again!)

💡 You'll need this key to **SSH into your instance** later.

Step 7: Configure Storage

Your EC2 instance needs storage to hold the OS and your files.

AWS provides **EBS (Elastic Block Store)**, and you can customize the size and type.

- **8GB gp3 (SSD)** – Good for most applications
- **16GB or more** – If you're running databases or high-storage apps

You can **increase storage later**, so don't worry too much about this step.

Step 8: Review and Launch! 🚀

Before launching, double-check:

✅ Correct **AMI** and **Instance Type**

✅ Proper **Network & Security** settings

✅ **Key Pair** is downloaded

Click **Launch Instance** and wait a few seconds for AWS to set it up.

Step 9: Connect to Your EC2 Instance

Once your instance is running, you’ll want to **log in using SSH**.

For Linux/macOS Users:

1. Open a terminal
2. Navigate to the directory where your key file is stored
3. Run this command:

```
ssh -i "your-key.pem" ubuntu@your-instance-public-i
```

For Windows Users (Using PuTTY):

a. Convert your .pem key to .ppk using **PuTTYgen**
b. Open **PuTTY**, enter your instance’s **Public IP**
c. Under **SSH → Auth**, load the .ppk file
d. Click **Open**, and log in as ubuntu

Now you have full access to your EC2 instance!

Learn how to launch an Amazon EC2 instance in AWS from start to finish. This guide covers essential steps including signing into AWS, choosing an Amazon Machine Image (AMI), selecting an instance type, configuring networking and security settings, creating a key pair for secure access, configuring storage, and connecting to your instance via SSH or PuTTY. Perfect for beginners, the guide walks you through each step to help you set up your virtual server efficiently.