

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. It allows users to run virtual servers (instances) on-demand, enabling scalable computing for a variety of applications, from web hosting to big data processing.

Key Features

- **Scalability:** Easily scale your capacity up or down as your computing requirements change.
- **Variety of Instance Types:** Choose from a wide range of instance types optimized for different use cases, including compute-optimized, memory-optimized, storage-optimized, and GPU instances.
- **Pay-As-You-Go Pricing:** Only pay for the compute capacity you use, with options for on-demand, reserved, or spot instances.
- **Integration with AWS Services:** Seamlessly integrate with other AWS services, such as Amazon S3, RDS, and VPC.

EC2 Instance Types

EC2 instances are categorized into several families, each designed for specific workloads:

- **General Purpose:** Balanced compute, memory, and networking resources (e.g., T3, M5).
- **Compute Optimized:** High-performance processors for compute-intensive tasks (e.g., C5).
- **Memory Optimized:** Optimized for memory-intensive applications (e.g., R5, X1).
- **Storage Optimized:** Designed for high storage throughput and low latency (e.g., I3, D2).
- **Accelerated Computing:** Includes GPU instances for high-performance computing tasks (e.g., P3, G4).

- **High I/O:** Ideal for applications that require high input/output operations per second (IOPS), such as large databases or data-intensive applications (e.g., I3en).

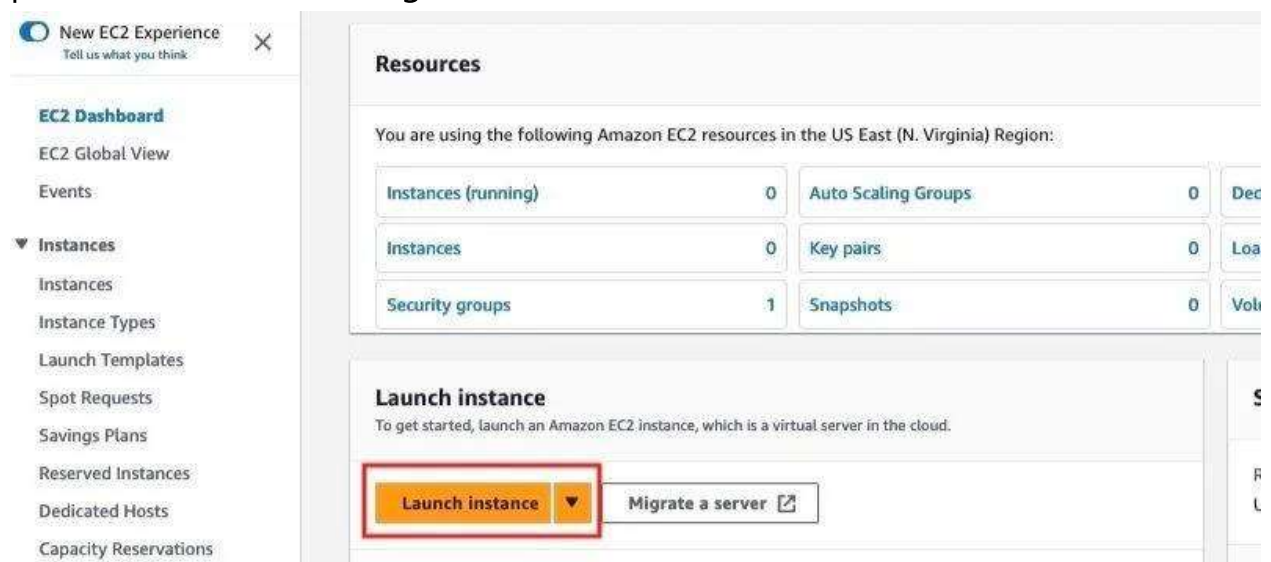
Creating an EC2 Instance: Step-by-Step Guide

Step 1: Sign In to AWS

- Log in to your AWS Management Console at [AWS Management Console](#) using your AWS account credentials.
- From the AWS Management Console, locate the **Services** dropdown and select **EC2** under the **Compute** section.

Step 2: Launch an Instance

- In the EC2 Dashboard, click the **Instances** link in the left navigation pane, then click the orange **Launch Instance** button.



Step 3: Choose an Amazon Machine Image (AMI)

- Choose an AMI that matches your requirements.

- AMIs are pre-configured templates that include an operating system and other software. Most servers use Linux because it's open-source, reliable, and efficient, commonly used for web servers, databases, and more. For this guide, we'll choose a Linux AMI.

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
my-ec2-instance [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE L

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type Free tier eligible

ami-0f409bae3775dc8e5 (64-bit (x86)) / ami-0f0f7b386be96ec2d (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

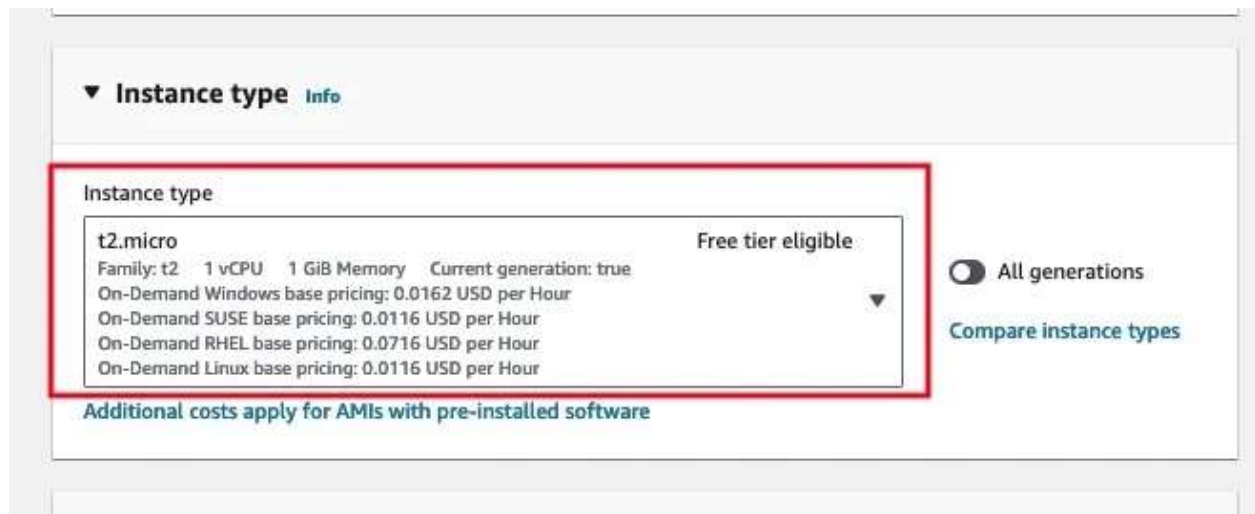
Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230822.0 x86_64 HVM gp2

Architecture: 64-bit (x86) AMI ID: ami-0f409bae3775dc8e5 Verified provider

Step 4: Choose an Instance Type

- Select the instance type based on your computing needs. Consider factors like CPU, memory, storage, and network capacity.



Step 5: Create or Select a Key Pair

- You need a key pair to securely connect to your instance using SSH. Create a new key pair or select an existing one.
- Make sure to safely store the private key file, as it's the key to accessing your server. If you lose it, you won't be able to access the instance. You can add other key pairs later if needed.
- Select **Create a new key pair**, give a name to your .pem key, and click the **Create key pair** orange button.
- A .pem key file will be downloaded to your local computer.

Create key pair [X]

Key pair name
Key pairs allow you to connect to your instance securely.

my-key-pair

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ **RSA**
RSA encrypted private and public key pair

☐ **ED25519**
ED25519 encrypted private and public key pair

Private key file format

☒ **.pem**
For use with OpenSSH

☐ **.ppk**
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel **Create key pair**

Step 6: Configure Network Details

- Choose the network settings (like the VPC and subnet). For now, we'll use the default VPC and subnet. Think of VPC as a private network, and we'll dive deeper into it later.

▼ **Network settings** [Info](#)

VPC - required [Info](#)

vpc-0007c146447af0ddf (default) ▼

172.31.0.0/16

Subnet [Info](#)

No preference ▼

Auto-assign public IP [Info](#)

Enable ▼

[Create new subnet](#)

Step 7: Configure Security Group

- Security groups act as virtual firewalls for your instance. Define inbound and outbound rules to control network traffic. For now, use the default security group. We can delve deeper into security groups later.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups [Info](#)

Select security groups ▼

default sg-092eb3afa24a11ac6 X

VPC: vpc-0007c146447af0ddf

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Step 8: Add Storage

- Configure the amount and type of storage for your instance. You can add additional storage volumes if needed.

▼ **Configure storage** [Info](#) Advanced

1x 8 GiB gp2 Root volume (Encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

[Add new volume](#)

0 x File systems [Edit](#)

Step 9: Review and Launch

- Review your instance configuration settings. If everything looks good, click the **Launch** button.

Step 10: Launch Status

- Once you click **Launch**, your instance will start launching. You'll see its status change to **running** in the EC2 Dashboard.
- As you delve deeper into the EC2 service, you will gradually grasp and gain insights into additional settings, expanding your understanding and knowledge over time.

| <input type="checkbox"/> | Name | Instance ID | Instance state | Instance type |
|--------------------------|-----------------|---------------------|----------------|---------------|
| <input type="checkbox"/> | my-ec2-instance | i-03b2cf8cd1cc7448c | Running | t2.micro |

Accessing Your Instance

Access Your Instance using SSH

- Once your instance is running, go to the EC2 Dashboard, select your instance, and click the **Connect** button. You'll find instructions for both Linux and Windows connections below:

Connecting from Mac and Linux:

- Open a terminal on your local machine.
- Navigate to the directory where you saved your private key file (.pem).
- Use the following command to set the appropriate permissions on the key file:

Bash Code :

```
chmod 400 <your-key-file>.pem
```

- Copy the SSH command from the EC2 instance connect page.
- Paste the command into your terminal and press Enter.

Connecting from Windows:

- Download and install an SSH client such as **PuTTY**.
- Convert your .pem key file to a .ppk key file using **PuTTYgen**.
- Open **PuTTY** and enter the public IP address of your instance in the "Host Name" field.
- Load your .ppk key file in the "Connection > SSH > Auth" settings.
- Click **Open** to start the SSH session.

Access Your Instance using AWS Session Manager

- You can access your instance through the AWS console using **Session Manager**. This provides a browser-based session similar to EC2 Instance Connect, eliminating the need to open port 22.