

# Amazon Elastic Compute Cloud (Amazon EC2) Report

**Lab:** Introducing Amazon Elastic Compute Cloud(EC2)

## Objectives

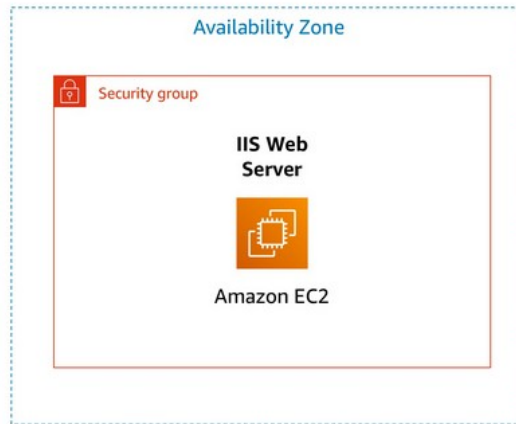
- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale and enable stop protection
- Explore EC2 limits
- Test stop protection
- Stop your EC2 instance

## Lab Environment Setup

- This lab provides you with a basic overview of launching, resizing, managing, and monitoring an Amazon EC2 instance.
- **Amazon Elastic Compute Cloud (Amazon EC2)** is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
- Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.
- Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

# Lab 3: Introduction to Amazon EC2

## Lab overview and objectives



## Task 1: Launch Your Amazon EC2 Instance

- In the **AWS Management Console** choose **Services**, choose **Compute** and then choose **EC2**.
- Choose the **Launch instance** menu and select **Launch instance**.

### Step 1: Name and tags

- Name the instance **Web Server**, which creates a tag with key **Name** and value **Web Server**.

### Step 2: AMI selection

- Keep the default **Amazon Linux 2023** AMI selected from the Quick Start list.

### Step 3: Instance type

- Keep the default **t2.micro** instance type with 1 vCPU and 1 GiB memory.

### Step 4: Key pair

- Select the **vokey** key pair for secure login authentication.

### Step 5: Network settings

- Select **Lab VPC**, use **PublicSubnet1**, create a security group named **Web Server security group**, and remove the default inbound rule.

## Step 6: Configure storage

- Keep the default **8 GiB EBS root volume** unchanged.

## Step 7: Advanced details

- Enable **termination protection** and add the provided user-data script to install and start the Apache web server.

## Step 8: Launch the instance

- Choose **Launch instance**, then view your instance and wait until it reaches **Running** with **2/2 status checks passed**.

### 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

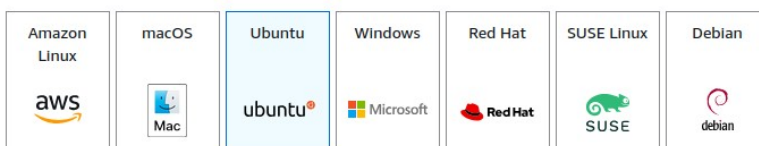
[Add additional tags](#)

### ▼ 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**.

Recents

**Quick Start**



[Browse more AMIs](#)

Including AMIs from  
AWS, Marketplace and  
the Community

#### Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))  
Virtualization: hvm    ENA enabled: true    Root device type: ebs

Free tier eligible ▼

#### Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86) ▼

AMI ID

ami-0ecb62995f68bb549

Publish Date

2025-10-22

Username

ubuntu



Verified provider

### ▼ Instance type [Info](#) | [Get advice](#)

#### Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

#### Key pair name - required

vockey

[Create new key pair](#)

### ▼ Network settings [Info](#)

#### VPC - required [Info](#)

vpc-0146cdfc86b20f546 (Lab VPC)  
10.0.0.0/16

[Create new VPC](#)

#### Subnet [Info](#)

subnet-00ca9a38f9763dc47 PublicSubnet1  
VPC: vpc-0146cdfc86b20f546 Owner: 975050223293 Availability Zone: us-east-1a (use1-az1)  
Zone type: Availability Zone IP addresses available: 1 CIDR: 10.0.1.0/28

[Create new subnet](#)

#### Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

#### 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

#### Security group name - required

websg1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . \_ - / () # , @ [] + = & ; {} ! \$ \*

#### Description - required [Info](#)

launch-wizard-1 created 2025-11-18T07:12:50.161Z

**Inbound Security Group Rules**

▼ Security group rule 1 (TCP, 22, 45.112.148.130/32) Remove

<b>Type</b>   <a href="#">Info</a>	<b>Protocol</b>   <a href="#">Info</a>	<b>Port range</b>   <a href="#">Info</a>
ssh ▼	TCP	22
<b>Source type</b>   <a href="#">Info</a>	<b>Name</b>   <a href="#">Info</a>	<b>Description - optional</b>   <a href="#">Info</a>
My IP ▼	<input type="text" value="Add CIDR, prefix list or security group"/> <input type="text" value="45.112.148.130/32"/>	e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0) Remove

<b>Type</b>   <a href="#">Info</a>	<b>Protocol</b>   <a href="#">Info</a>	<b>Port range</b>   <a href="#">Info</a>
HTTP ▼	TCP	80
<b>Source type</b>   <a href="#">Info</a>	<b>Source</b>   <a href="#">Info</a>	<b>Description - optional</b>   <a href="#">Info</a>
Anywhere ▼	<input type="text" value="Add CIDR, prefix list or security group"/> <input type="text" value="0.0.0.0/0"/>	e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ×

Add security group rule

► **Advanced network configuration**

## In Terminal Execution:

```
ubuntu@ip-10-0-1-8:~$ git clone https://github.com/sreepathysois/Cafe_Dynamic_Website.git
ubuntu@ip-10-0-1-8:~$ sudo apt-get install apache2 php php-mysql mysql-server mysql-client
libapache2-mod-php
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopcafe$ sudo apt-get update
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopcafe$ sudo cp -rf * /var/www/html/
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopcafe$ sudo rm -rf
/var/www/html/index.html
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopcafe$ sudo systemctl restart apache2
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopcafe$ sudo mysql -u root -p
mysql> create database cafedb;
mysql> create user 'msis'@'localhost' identified by 'msis@123';
mysql> grant all privileges on cafedb.* to 'msis'@'localhost';
mysql> exit
```

```
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website$ cd mompopdb/
```

```
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopdb$ mysql -u msis -p
```

```
mysql> use cafedb;
```

```
mysql> source create-db.sql
```

```
mysql> exit
```

```
ubuntu@ip-10-0-1-8:~/Cafe_Dynamic_Website/mompopdb$ cd
```

```
ubuntu@ip-10-0-1-8:~$ sudo nano /var/www/html/getAppParameters.php
```

```
<?php
// Get the application environment parameters from the Parameter Store.
//include ('getAppParameters.php');
$showServerInfo = "false";
$timeZone = "America/New_York";
$currency = "$";
$db_url = "localhost";
$db_name = "cafedb";
$db_user = "msis";
$db_password = "msis@123";

// Display the server metadata information if the showServerInfo parameter is
//include('serverInfo.php');
?>
```

```
ubuntu@ip-10-0-1-8:~$ sudo systemctl restart apache2
```

```
ubuntu@ip-10-0-1-8:~$ sudo systemctl restart mysql
```



In Web Browser using EC2 Public ip

