

Cloud Computing Lab

Experiment-3

Create an Amazon EC2 instance and set up a web-server on the instance and associate an IP address with the instance.

Amazon Web Services (AWS) provides a powerful cloud computing platform with a wide range of services. One of the core services offered by AWS is Amazon Elastic Compute Cloud (EC2), which allows users to launch virtual servers in the cloud.

An **Amazon EC2(Elastic Compute Cloud)** instance is a virtual server in the cloud that allows users to run applications and workloads on AWS. It's essentially renting a computer in a data center, providing on-demand computing resources that can be scaled up or down as needed. EC2 instances offer various configurations of CPU, memory, storage, and networking, allowing users to choose the best fit for their specific requirements.

The **AWS Management Console** is a centralized platform designed for accessing and managing AWS cloud services. It offers a graphical user interface (GUI) that simplifies the management of various AWS resources, such as compute power (EC2), storage (S3), databases (RDS), and more.

Creating an Amazon EC2 instance involves several steps within the AWS Management Console:

1. Log in to your AWS account

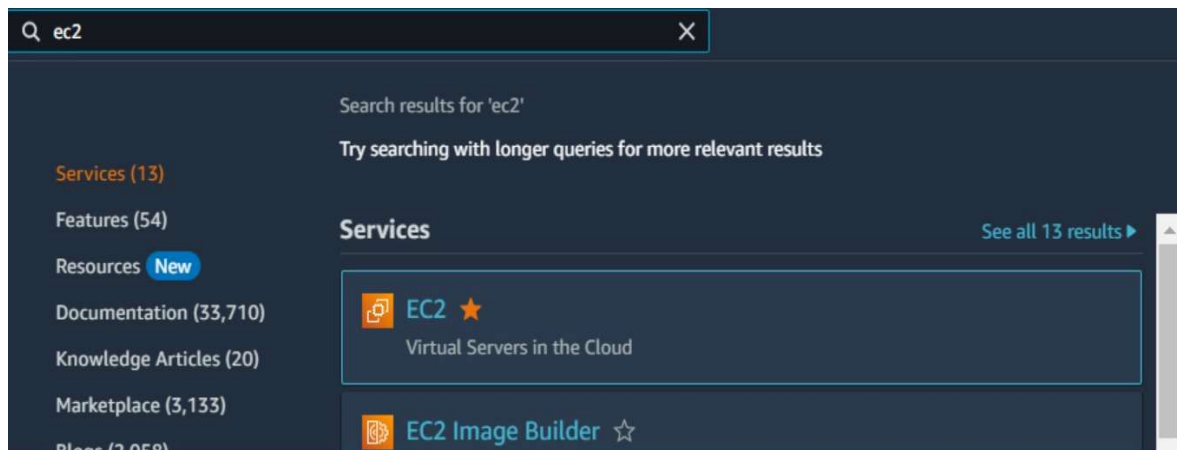
Log in to the [AWS Management Console](#) and set up your root account. If you don't already have an account, you will be prompted to create one.

- With the [AWS Free Tier](#), you can get 750 hours/month of select EC2 instances for free.

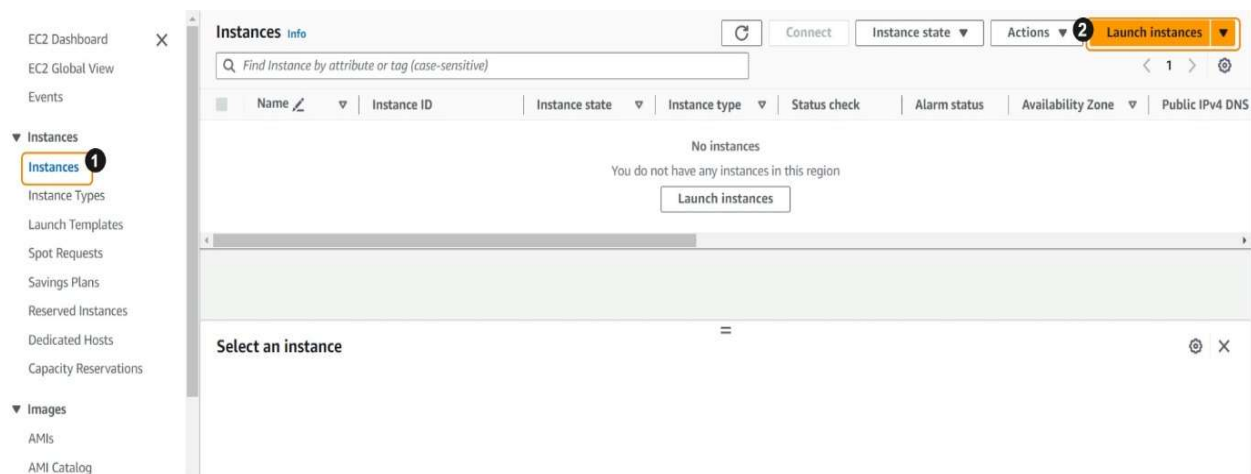
2. Launch your instance

Identify which [instance type](#) is best for your workload. For your first instance, we recommend a low-cost, general-purpose instance type—[t2.micro](#)—and Amazon Machine Image (AMI)—Amazon Linux 2 AMI—which are both [free-tier eligible](#).

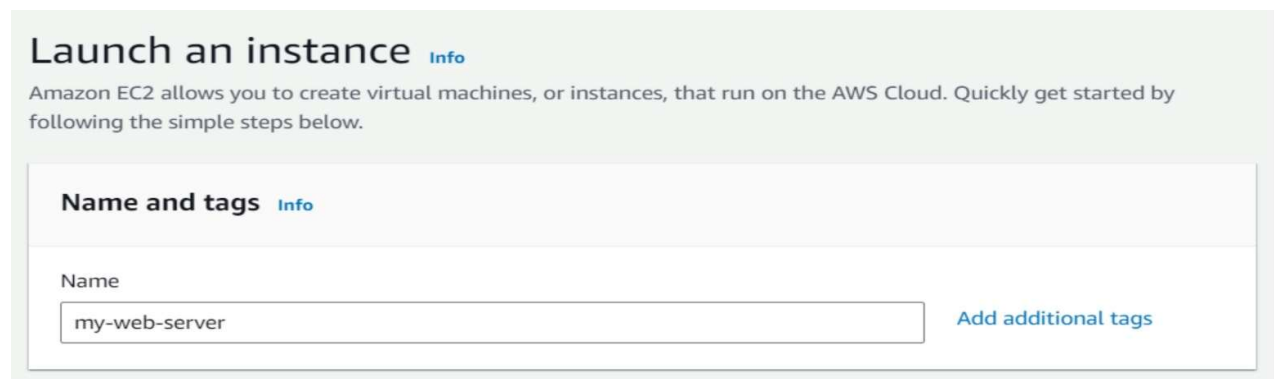
Step-1: Search ‘ec2’ in the AWS Management Console search bar. Click **EC2** on the search results.



Step-2: In the left window pane, select **Instances**, then click the **Launch instances** option.



Step-3: Name the instance ‘*my-web-server*’ or any name that you prefer.





Step-4: Under the **Application and OS Images** section, click the default **Amazon Linux AMI**.


▼ 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

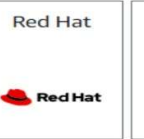
Quick Start

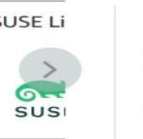
**Amazon Linux**


**macOS**

**Ubuntu**

**Windows**

**Red Hat**

**SUSE Linux**

**Browse more AMIs**
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-0dbc3d7bc646e8516 (64-bit (x86)) / ami-055859c8e0f361065 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description
Amazon Linux 2023 AMI 2023.2.20231018.2 x86_64 HVM kernel-6.1

Step-5: Under the **Instance Type** section, select **t2.micro**.

▼ Instance type [Info](#)

Instance type

t2.micro **Free tier eligible**
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 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](#)


Step-6: Under the **Key Pair** section, click **Create new key pair**.

▼ 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

▼

 **Create new key pair**

3.Configure your instance

Here are some guidelines when setting up your first instance:

- **Security group:** Create your own firewall rules or select the default VPC security group.
- **Storage:** EC2 offers both magnetic disk and SSD storage. We recommend starting with Amazon EBS gp2 volumes.
- Choose "Launch Instances" to complete the setup.

** Note: We will use the key pair file (.pem) later.*

Step-1: Enter a key pair name and follow the configurations below. Then, click **Create key pair**.

The screenshot shows the 'Create key pair' dialog box in AWS. It has a title bar 'Create key pair' with a close button. Below the title bar, there's a section 'Key pair name' with a text input field containing 'web-server-key-pair'. A small black circle with the number '1' is next to the input field. Below the input field, there's a note: 'The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.' Below this, there's a section 'Key pair type' with two options: 'RSA' (selected, indicated by a blue dot) and 'ED25519'. A small black circle with the number '2' is next to the 'RSA' option. Below this, there's a section 'Private key file format' with two options: '.pem' (selected, indicated by a blue dot) and '.ppk'. A small black circle with the number '3' is next to the '.pem' option. Below the options, there's a yellow warning box with a triangle icon and text: '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'. At the bottom right, there are two buttons: 'Cancel' and 'Create key pair'. A small black circle with the number '4' is next to the 'Create key pair' button.

Create key pair

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

web-server-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

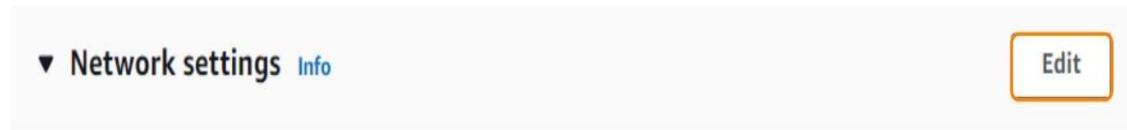
☐ .ppk
For use with PuTTY

⚠ 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

After creating a key pair, the private key will be downloaded to your computer. Remember to note the location of this file, as you'll need it later to SSH to your EC2 instance.

Step-2. Under the **Network settings** section, click **Edit**.



Step-3. Scroll down the **Firewall (Security Groups)** option.

- a. Enter '**WebServerSG**' for the security group name.
- b. For Description, enter '*Allows SSH and HTTP access*'.

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*

WebServerSG

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](#)

Allows SSH and HTTP access

Step-4: Add two inbound security group rules with the following configuration.

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 15/32, Allow SSH from my IP) Remove			
Type Info	Protocol Info	Port range Info	
1 ssh	TCP	22	
Source type Info	Name Info	Description - optional Info	
2 My IP	<input type="text" value="Add CIDR, prefix list or security"/>	<input type="text" value="Allow SSH from my IP"/>	
<input type="text" value="15/32"/>			
▼ Security group rule 2 (TCP, 80, 0.0.0.0/0, Allow web traffic on port 80) Remove			
Type Info	Protocol Info	Port range Info	
3 HTTP	TCP	80	
Source type Info	Source Info	Description - optional Info	
4 Anywhere	<input type="text" value="Add CIDR, prefix list or security"/>	<input type="text" value="Allow web traffic on port 80"/>	
<input type="text" value="0.0.0.0/0"/>			
<input type="button" value="Add security group rule"/>			

Inbound rule 1

Type	Source Type
SSH	My IP

Inbound rule 2

Type	Source Type
HTTP	Anywhere (0.0.0.0/0)

Step-5. In the right window pane, at the bottom section, click **Launch instance**.

Setting up the web server.

4.Connect to your instance

After launching your instance, you can connect to it and use it the way you'd use a computer sitting in front of you. There are several ways to [connect to the console](#) depending on the operating system. We recommend using [EC2 Instance Connect](#), an easy-to-use browser-based client.

- Select the EC2 instance that you created and choose "Connect."
- Select "EC2 Instance Connect."
- Choose "Connect." A window opens, and you are connected to your instance.

Step-1: After the instance is created successfully, click the **instance ID**.



Step-2: Tick the checkbox next to your instance name. Then, copy the Public IP address of your instance and paste it somewhere you can easily retrieve it later.

The screenshot shows the AWS Management Console interface for EC2 instances. At the top, there's a search bar and a filter for 'Instance ID = i-0b554dc7418b6dd89'. Below this, a table lists instances. The first instance, 'my-web-server', is selected (checkbox checked) and its details are shown below. The instance is in the 'Running' state, has a 't2.micro' instance type, and its public IPv4 address is '34.229.194.201'. The private IPv4 address is '192.168.5.22'. The public IP address is highlighted with an orange box and labeled with a '2'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
my-web-server	i-0b554dc7418b6dd89	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b

Instance: i-0b554dc7418b6dd89 (my-web-server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID: i-0b554dc7418b6dd89 (my-web-server)

Public IPv4 address: 34.229.194.201 | open address

Private IPv4 addresses: 192.168.5.22

In this lab, we'll be using the SSH utility from OpenSSH. It usually comes built-in with Windows 10 and 11, Mac, and most Linux distributions. If your operating system doesn't have it pre-installed, ensure you install it first before proceeding.

Step-3: Open up a terminal, then run the command below to connect to your instance via SSH.

```
ssh -i /path/to/YOUR-KEY.pem ec2-user@YOUR-EC2-PUBLIC-IP
```

Ensure that you reference the correct path to your private key pair and that you use the correct public IP of your EC2 instance.

Once connected, your shell prompt should change to something similar to `ec2-user@ip-192-168-5-22:~$`, confirming that you're now connected to your EC2 instance.

In the next steps, you will configure the necessary settings to set up a web server on the EC2 instance.

Step-4: Run the command below to update the system.

```
sudo yum update -y
```

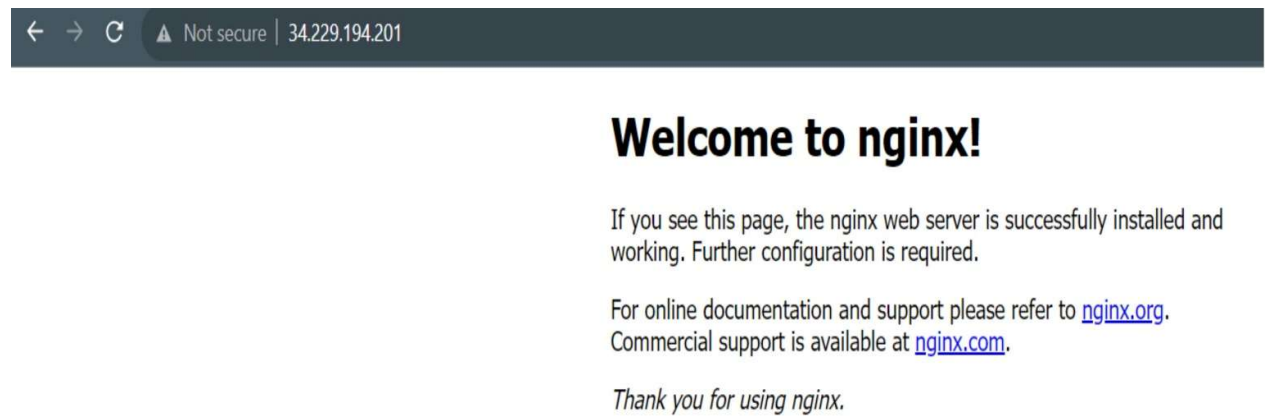
Step-5: Once the update is completed, install Nginx.

```
sudo yum install nginx -y
```

Step-6: Start the Nginx Service.

```
sudo service nginx start
```

Step-7: Enter your EC2 instance's public IP in your browser. The default Nginx welcome page should be displayed.



Now, let's replace the welcome page with a custom one.

Step-8: Go to the /usr/share/nginx/html/

```
cd /usr/share/nginx/html/
```

Step-9: Create a custom HTML page.

```
echo '<h1>Welcome to my web page!</h1>' | sudo tee mypage.html > /dev/null
```

Step-10: Let's override the default Nginx configuration by creating a new configuration file in the /etc/nginx/conf.d/

```
sudo vi /etc/nginx/conf.d/server.conf
```


Step-11: Press i to enter Insert mode in Vi and paste the following configuration.

```
server {  
    listen 80 default_server;  
    server_name _;  
    root /usr/share/nginx/html;  
  
    location / {  
        index mypage.html;  
    }  
}
```

Step-12: Press the Escape button and enter :wq! to exit and save your changes.

Step-13: Reload Nginx for the changes to take effect.

```
Sudo nginx -t && sudo service nginx reload
```

Step-7: Reload your browser to see the changes you've made.

Congratulations! You've successfully set up a web server on an Amazon EC2 instance using Nginx. You've also hosted a custom web page, giving you foundational skills in web hosting on the cloud. This is just the beginning. As you continue to explore, you can experiment with different configurations, host more complex web applications, and even integrate databases.

5. Terminate your instance

Amazon EC2 is free to start ([learn more](#)), but it is important that you terminate your instances to prevent additional charges. The EC2 instance and the data associated will be deleted.

Select the EC2 instance, choose "Actions," select "Instance State," and then select "Terminate."