

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.

Search results for 'ec2'

Try searching with longer queries for more relevant results

Services (13)

Features (54)

Resources New

Documentation (33,710)

Knowledge Articles (20)

Marketplace (3,133)

Blogs (2,058)

See all 13 results ▶

EC2 ★
Virtual Servers in the Cloud

EC2 Image Builder ★

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

EC2 Dashboard

EC2 Global View

Events

Instances

Instances 1

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Instances Info

Find Instance by attribute or tag (case-sensitive)

Name Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4 DNS

No instances

You do not have any instances in this region

Launch instances

Select an instance

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

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-web-server

Add additional tags

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

The screenshot shows the 'Application and OS Images (Amazon Machine Image)' section. It features a search bar at the top with the placeholder 'Search our full catalog including 1000s of application and OS images'. Below the search bar is a 'Quick Start' section with icons for various operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. To the right of this section is a 'Browse more AMIs' button with the text 'Including AMIs from AWS, Marketplace and the Community'. The main content area displays the 'Amazon Machine Image (AMI)' section for the 'Amazon Linux 2023 AMI'. The details shown are: Name: Amazon Linux 2023 AMI, ID: ami-0dbc3d7bc646e8516 (64-bit (x86)) / ami-055859c8e0f361065 (64-bit (Arm)), Virtualization: hvm, ENA enabled: true, Root device type: ebs. A 'Free tier eligible' badge is present. Below this, there is a 'Description' section with the text 'Amazon Linux 2023 AMI 2023.2.20231018.2 x86_64 HVM kernel-6.1'.

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

The screenshot shows the 'Instance type' section. It lists the 't2.micro' instance type, which is described as having 1 vCPU, 1 GiB Memory, and being part of the t2 family. It is marked as 'Free tier eligible'. Other details include On-Demand Windows base pricing, On-Demand SUSE base pricing, On-Demand RHEL base pricing, and On-Demand Linux base pricing. A note states 'Additional costs apply for AMIs with pre-installed software'. To the right of the instance type list are buttons for 'All generations' and 'Compare instance types'.

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

The screenshot shows the 'Key pair (login)' section. It contains a note about using a key pair for secure connection and a field for 'Key pair name - required' with a dropdown menu labeled 'Select'. To the right of the dropdown is a 'Create new key pair' button with a blue outline.

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**.

Create key pair

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

1 web-server-key-pair
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

2 RSA
RSA encrypted private and public key pair

3 ED25519
ED25519 encrypted private and public key pair

Private key file format

4 .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](#)

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**.

The screenshot shows a user interface for managing network settings. At the top left is a dropdown menu labeled "Network settings" with a blue "Info" link next to it. To the right is an "Edit" button, which is highlighted with a yellow border. The background is white with some light gray shadows.

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*'.

The screenshot shows the "Firewall (security groups)" configuration page. At the top left is a "Firewall (security groups)" label with a blue "Info" link. Below it is a brief description: "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." Two radio buttons are present: "Create security group" (selected, highlighted with a blue border) and "Select existing security group".

The "Security group name - required" field contains "WebServerSG". A note below it states: "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 ._-:/()#,@[]+=;&{}!\$*".

The "Description - required" field contains "Allows SSH and HTTP access".

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

Inbound Security Group Rules

The screenshot shows the "Inbound Security Group Rules" configuration page. It lists two security group rules:

1. **Security group rule 1 (TCP, 22, 15/32, Allow SSH from my IP)**:
 - Type: ssh (highlighted with a yellow border)
 - Protocol: TCP
 - Port range: 22
 - Source type: My IP (highlighted with a yellow border)
 - Name: 15/32
 - Description: Allow SSH from my IP

2. **Security group rule 2 (TCP, 80, 0.0.0.0/0, Allow web traffic on port 80)**:
 - Type: HTTP (highlighted with a yellow border)
 - Protocol: TCP
 - Port range: 80
 - Source type: Anywhere (highlighted with a yellow border)
 - Name: 0.0.0.0/0
 - Description: Allow web traffic on port 80

At the bottom left is a "Add security group rule" button.

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 EC2 Instances page. At the top, there's a search bar with placeholder text 'Find Instance by attribute or tag (case-sensitive)'. Below it, a filter bar has 'Instance ID = i-0b554dc7418b6dd89' selected. A 'Clear filters' button is also present. The main table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is highlighted with a yellow background, showing 'my-web-server' in the Name column, 'i-0b554dc7418b6dd89' in the Instance ID column, 'Running' in the Instance state column, 't2.micro' in the Instance type column, '2/2 checks passed' in the Status check column, 'No alarms' in the Alarm status column, and 'us-east-1b' in the Availability Zone column. A circled '1' is next to the 'my-web-server' entry in the Name column. Below the table, the instance details are shown: Instance ID: i-0b554dc7418b6dd89 (my-web-server), Public IPv4 address: 34.229.194.201 (with a circled '2' over the address), and Private IPv4 addresses: 192.168.5.22. A 'Details' tab is active, followed by Security, Networking, Storage, Status checks, Monitoring, and Tags.

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
```

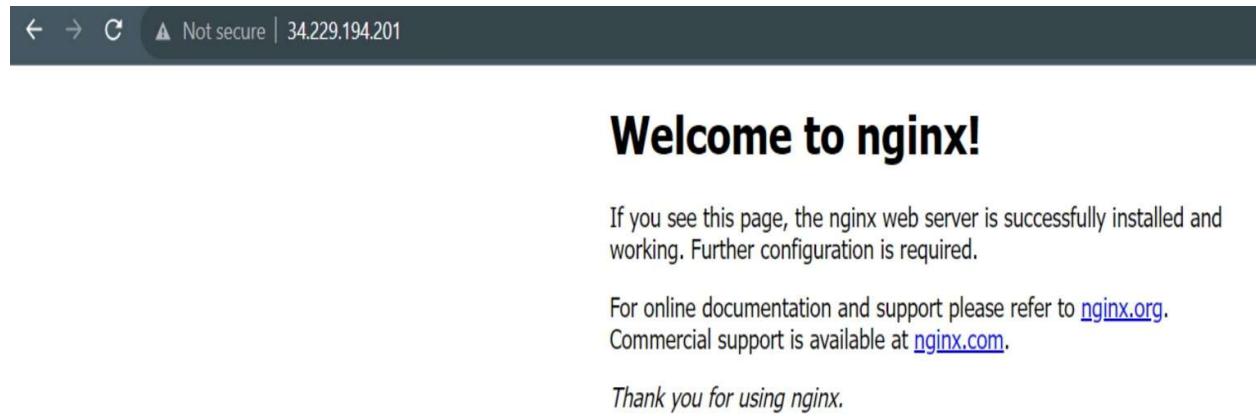
Step-5: Once the update is completed, install Nginx.

```
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."