

AMI-Based Apache Web Server Deployment on AWS EC2

Create a VPC

1. Go to **VPC Console** (search “VPC” in AWS Console).
2. Left menu → **Your VPCs** → **Create VPC**.
3. Choose **VPC only**.
4. Fill in:

Name tag: MyLab-VPC

IPv4 CIDR block: 10.0.0.0/16 (gives 65k IPs, fine for labs)

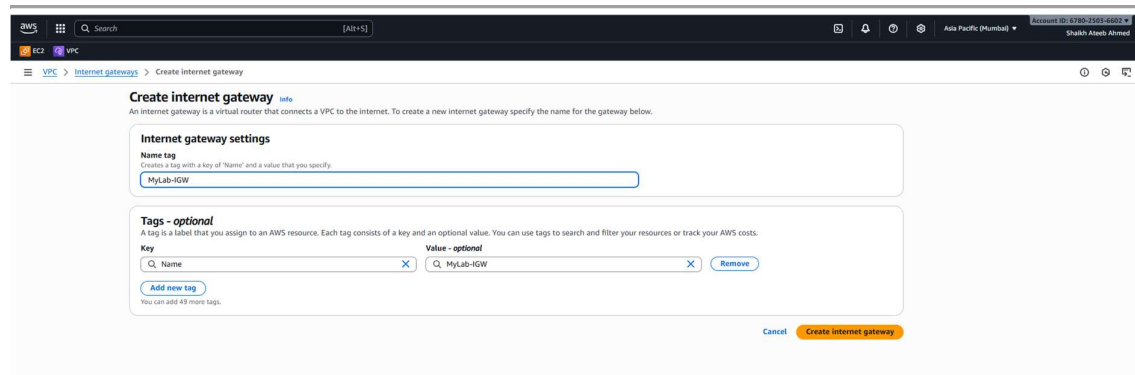
Leave other defaults.

5. Click **Create VPC**.

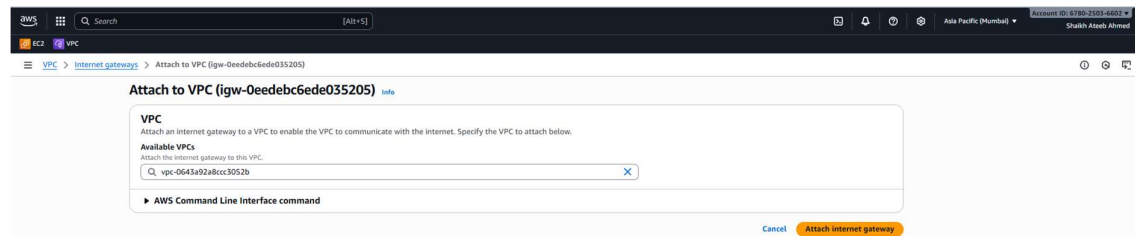
The screenshot shows the AWS VPC Console 'Create VPC' page. The 'VPC settings' section is active, showing 'VPC only' selected under 'Resources to create'. The 'Name tag' is set to 'MyLab-VPC'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected with the value '10.0.0.0/16'. Under 'IPv6 CIDR block', 'No IPv6 CIDR block' is selected. The 'Tenancy' is set to 'Default'. The 'Tags' section shows a key 'Q, Name' with a value 'Q, MyLab-VPC'. The 'Create VPC' button is highlighted in orange.

Create an Internet Gateway (IGW)

1. Left menu → **Internet gateways** → **Create internet gateway**.
2. Name: MyLab-IGW → Create.



3. Select it → **Actions** → **Attach to VPC** → choose MyLab-VPC.



Create a Public Subnet

1. Left menu → **Subnets** → **Create subnet.**

2. Choose:

VPC: MyLab-VPC

Subnet name: Public-Subnet-1

Availability Zone: e.g. ap-south-1a

IPv4 CIDR block: 10.0.1.0/24 (gives 256 IPs)

3. Click **Create subnet.**

Shaikh Ateeb Ahmed

07-08-2025

The screenshot shows the 'Create subnet' page in the AWS Management Console. The 'VPC' section shows 'VPC ID' as 'vpc-0643a92a8cc3052b (MyLab-VPC)'. The 'Subnet settings' section shows 'Subnet name' as 'Public-Subnet-1', 'Availability Zone' as 'Asia Pacific (Mumbai) / ap-south-1a', and 'IPv4 VPC CIDR block' as '10.0.0.0/16'. The 'IPv4 subnet CIDR block' is set to '10.0.1.0/24' with a '256 IP' indicator. There is a 'Tags - optional' section with a table containing one tag: 'Name' with value 'Public-Subnet-1'. At the bottom, there are 'Add new tag' and 'Remove' buttons.

4. Go to **Subnets** → select **Public-Subnet-1**.
5. Actions → **Modify subnet settings**.
6. Check **Enable auto-assign public IPv4 address** → Save.

The screenshot shows the 'Edit subnet settings' page in the AWS Management Console. The 'Subnet' section shows 'Subnet ID' as 'subnet-00779be86d3fc53c' and 'Name' as 'Public-Subnet-1'. The 'Auto-assign IP settings' section has 'Enable auto-assign public IPv4 address' checked. The 'Resource-based name (RBN) settings' section has 'Enable resource name DNS A record on launch' and 'Enable resource name DNS AAAA record on launch' both checked. The 'Hostname type' section has 'IP name' selected. The 'DNS64 settings' section has 'Enable DNS64' checked. At the bottom right, there are 'Cancel' and 'Save' buttons.

Create and Configure a Route Table

1. Left menu → **Route tables** → **Create route table**.

Name: Public-RT

VPC: MyLab-VPC → Create.

Shaikh Ateeb Ahmed

07-08-2025

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of "Name" and a value that you specify.
Public-RT

VPC
The VPC to use for this route table.
vpc-0643a92a8cc3052b (MyLab-VPC)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key
Name

Value - optional
Public-RT

Buttons: Cancel, Create route table

2. Select Public-RT → **Subnet associations** → **Edit subnet associations** → check Public-Subnet-1 → Save.

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Public-Subnet-1	subnet-00779be86d3fcc53c	10.0.1.0/24	-

Selected subnets
subnet-00779be86d3fcc53c / Public-Subnet-1

Buttons: Cancel, Save associations

3. Go to **Routes tab** → **Edit routes** → Add route:

Destination: 0.0.0.0/0

Target: your MyLab-IGW

Save changes.

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute

Buttons: Add route, Cancel, Preview, Save changes

Now your subnet is “public” (it routes traffic to the Internet).

Security Group

1. Choose **Create a new security group**.
2. Creating new, add rules:

SSH → Type: **SSH**, Protocol: TCP, Port: 22, Source: **My IP** (choose “My IP” to restrict).
HTTP → Type: **HTTP**, Protocol: TCP, Port: 80, Source: **0.0.0.0/0** (allows public access to web server).

Launch Ubuntu EC2 (Console)

1. EC2 → Instances → **Launch instances**.
2. Give name eg. EC2-Ubuntu-1
3. *Image*: choose **Ubuntu Server**.
4. *Instance type*: e.g. t3.micro (or whatever).
5. *Key pair*: select existing.
6. *Network / subnet*: pick a public subnet (ensure **Auto-assign Public IP = Enable**).
7. *Security group*: choose that we have created above.
8. Launch.

Shaikh Ateeb Ahmed

07-08-2025

Launch an instance

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

Name: EC2-Ubuntu-1

Application and OS Images (Amazon Machine Image)

Search for a full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-02d26659f0824f299 (64-bit x86) / ami-0b9093c0d00e0f992 (64-bit ARM)

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-02d26659f0824f299 | Publish Date: 2025-08-21 | Username: ubuntu | Verified provider

Instance type

Instance type: t3.micro

Family: t3 | 2 vCPU | 1 GiB Memory | Current generation: true | On-Demand Linux base pricing: 0.0112 USD per Hour | On-Demand SUSE base pricing: 0.0112 USD per Hour | On-Demand Windows base pricing: 0.0204 USD per Hour | On-Demand Ubuntu Pro base pricing: 0.0147 USD per Hour | On-Demand RHEL base pricing: 0.04 USD per Hour

Free tier eligible

Additional costs apply for AMIs with pre-installed software

Key pair (login)

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: keypair_2

Create new key pair

Network settings

VPC - required: vpc-0643a92a8cc3052b (MyLab-VPC)

Subnet: subnet-00739b8d8d3f0c53c

Auto-assign public IP: Enable

Firewall (security groups)

Create security group | Select existing security group: MyLab-SG

Common security groups

MyLab-SG sg-026900539ab7389e

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64, read more

Virtual server type (instance type): t3.micro

Firewall (security group): MyLab-SG

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

SSH in, install Apache, verify (commands to run on your machine after instance is running)

Replace <KEY.pem> and <PUBLIC_IP> with your values.

chmod 400 <KEY.pem>

ssh -i "<KEY.pem>" ubuntu@<PUBLIC_IP>

Shaikh Ateeb Ahmed

07-08-2025

```
ubuntu@ip-10-0-1-59: ~  
System information as of Thu Sep  4 09:55:00 UTC 2025  
  
System load:  0.03      Temperature:    ~273.1 C  
Usage of /:   25.6% of 6.71GB    Processes:      115  
Memory usage: 23%      Users logged in:  0  
Swap usage:   0%        IPv4 address for ens5: 10.0.1.59  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-10-0-1-59:~$ |
```

Once on the instance:

update packages

sudo apt upgrade -y

```
ubuntu@ip-10-0-1-59: ~  
System information as of Thu Sep  4 09:55:00 UTC 2025  
  
System load:  0.03      Temperature:    ~273.1 C  
Usage of /:   25.6% of 6.71GB    Processes:      115  
Memory usage: 23%      Users logged in:  0  
Swap usage:   0%        IPv4 address for ens5: 10.0.1.59  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-10-0-1-59:~$ sudo apt upgrade -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ubuntu@ip-10-0-1-59:~$ |
```

install Apache

sudo apt install -y apache2

Shaikh Ateeb Ahmed

07-08-2025

```
ubuntu@ip-10-0-1-59: ~$ sudo systemctl enable --now apache2
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.5) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-1-59:~$ sudo apt install -y apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-10-0-1-59:~$
```

enable and start Apache

sudo systemctl enable --now apache2

```
ubuntu@ip-10-0-1-59: ~$ sudo systemctl enable --now apache2
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.5) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-1-59:~$ sudo apt install -y apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-10-0-1-59:~$ sudo systemctl enable --now apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-10-0-1-59:~$
```

From your laptop/browser: open http://<PUBLIC_IP>/ — you should see the default Apache2 Ubuntu page.



Create an AMI of that instance

Console (simplest)

1. EC2 → Instances → select the Ubuntu instance that u have create above.
2. **Actions → Image and templates → Create image.**

Name: e.g. ubuntu-apache-AMI

Leave volumes as-is (root volume included).

3. Click **Create image.**

Shaikh Ateeb Ahmed

07-08-2025

Create image [info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Image details

Instance ID: [i-01990a421768bcc52](#) (EC2-Ubuntu-1)

Image name:

Maximum 127 characters. Can't be modified after creation.

Image description - optional:

Maximum 255 characters

☒ **Reboot instance**

When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/sd...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

[During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.](#)

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ **Tag image and snapshots together**
Tag the image and the snapshots with the same tag.

☐ **Tag image and snapshots separately**
Tag the image and the snapshots with different tags.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

4. EC2 → Images → AMIs → wait until the AMI state becomes **available**.

Terminate the original instance (Console)

- EC2 → Instances → select the instance → **Instance state** → **Terminate instance** → confirm.
- Wait for state to change to **terminated**.

Instances (1/1) [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	Public IPv4 ...	Elastic IP	IPv4
EC2-Ubuntu-1	i-01990a421768bcc52	Terminated	t3.micro	-	-	ap-south-1a	-	-	-	-

[View alarms](#)

i-01990a421768bcc52 (EC2-Ubuntu-1)

Instance summary [info](#)

Instance ID: [i-01990a421768bcc52](#)

IPv6 address: -

Hostname type: -

Answer private resource DNS name: -

Auto-assigned IP address: -

IAM Role: -

IMDSv2: Required

Operator: -

Instance details [info](#)

AMI ID: [ami-02d26659f082cf299](#)

AMI name: [ubuntu/images/hvm-sd-gp3/ubuntu-noble-24.04-amd64-server-20250821](#)

Public IPv4 address: -

Instance state: [Terminated](#)

Instance type: [t3.micro](#)

VPC ID: -

Subnet ID: -

Instance ARN: [arn:aws:ec2:ap-south-1:678025036602:instance/i-01990a421768bcc52](#)

Monitoring: disabled

Allowed image: -

Private IPv4 addresses: -

Public DNS: -

Elastic IP addresses: -

AWS Compute Optimizer finding

[Opt-in to AWS Compute Optimizer for recommendations.](#) [Learn more](#)

Auto Scaling Group name

-

Managed: false

Platform details

[Linux/UNIX](#)

Termination protection

Disabled

Terminating is permanent for that instance ID; AMI remains available in **Images** → **AMIs**.

Launch new instance from your AMI

1. EC2 → **Launch instances**.
2. Under **My AMIs** (left) → select the AMI you created.
3. Give name eg. EC2-Ubuntu-AMI
4. Set *Instance type* (e.g. t3.micro).
5. Configure network/subnet: choose as before (public subnet + auto-assign public IP if you want browser access).
6. Select or create Security Group (must allow SSH and HTTP).
7. Choose key pair (same or different).
8. Launch. EC2 will create instances from the same AMI (It will have Apache already installed and your default page present).

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices. [Take a walkthrough](#) [Do not show me this message again](#)

Launch an instance

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

Name: [Add additional tags](#)

Application and OS Images (Amazon Machine Image)

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](#).

[AMI from catalog](#) [Recents](#) [My AMIs](#) [Quick Start](#)

Name	Image ID	Username	Published	Architecture	Virtualization	Root device type	ENA Enabled
ubuntu-apache-AMI	ami-0d21d036558b1550e	ubuntu	2025-09-04T10:00:58.000Z	x86_64	hvm	ebs	Yes

[Free for eligible](#) [Browse more AMIs](#) (including AMIs from AWS, Marketplace and the Community)

Instance type

Summary

Number of instances:

Software image (AMI): [ubuntu-apache-AMI](#) (ami-0d21d036558b1550e)

Virtual server type (instance type): [t3.micro](#)

Firewall (security group): [MyLab-SG](#)

Storage (volumes): [1 volume\(s\) - 8 GiB](#)

[Cancel](#) [Launch instance](#) [Preview code](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Shaikh Ateeb Ahmed

07-08-2025

Instance type [info](#) [Get advice](#)

Instance type: **t3.micro** [Free tier eligible](#) [All generations](#) [Compare instance types](#)

Family: t3 - 2 vCPU - 1 GiB Memory - Current generation: true - On-Demand Linux base pricing: 0.0112 USD per Hour - On-Demand S3OL base pricing: 0.0112 USD per Hour - On-Demand Windows base pricing: 0.0204 USD per Hour - On-Demand Ubuntu Pro base pricing: 0.0147 USD per Hour - On-Demand RHEL base pricing: 0.04 USD per Hour

[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**: **keypair_2** [Create new key pair](#)

Network settings [info](#)

VPC - required [info](#): **vpc-0645a92a8cc3052b (MyLab-VPC)** [Create new VPC](#)

Subnet [info](#): **subnet-00779be6d3fc53c** [Create new subnet](#)

VPC: vpc-0645a92a8cc3052b - Owner: 678023036602 - Availability Zone: ap-south-1a (ap-s1-az1) - Zone type: Availability Zone - IP addresses available: 251 - CIDR: 10.0.1.0/24

Auto-assign public IP [info](#): **Enable**

Firewall (security group) [info](#): **MyLab-SG sg-026900539ab73899e** [Compare security group rules](#)

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](#): **MyLab-SG sg-026900539ab73899e** [Compare security group rules](#)

Select security groups: **MyLab-SG sg-026900539ab73899e**

[Advanced network configuration](#)

Summary [info](#)

Number of instances: **1**

Software Image (AMI): **ubuntu-apache-AMI** [ami-0a21a6555861550e](#)

Virtual server type (instance type): **t3.micro**

Firewall (security group): **MyLab-SG**

Storage (volumes): **1 volume(s) - 8 GiB**

[Cancel](#) [Launch instance](#) [Preview code](#)

Then test in browser: <http://<new-instance-public-ip>/> — you should see the same Apache default page.

