Web Hosting on AWS EC2 with Apache & nano

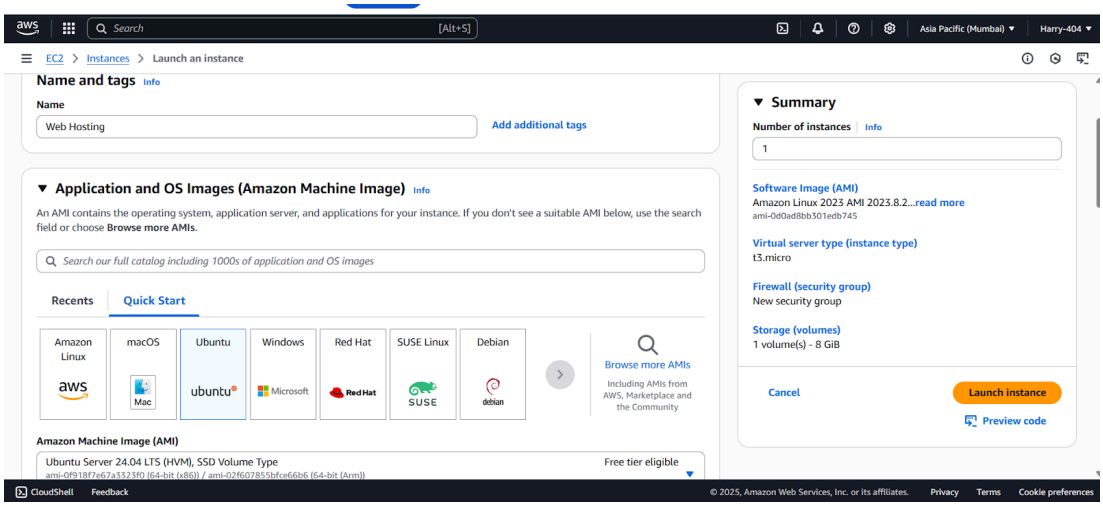
Step 1: Launch an EC2 Instance

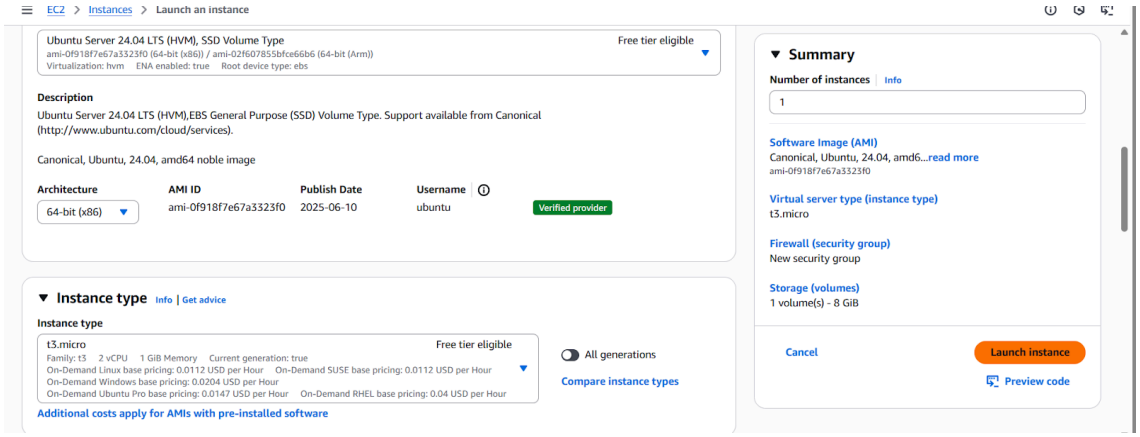
1. Open AWS Management Console

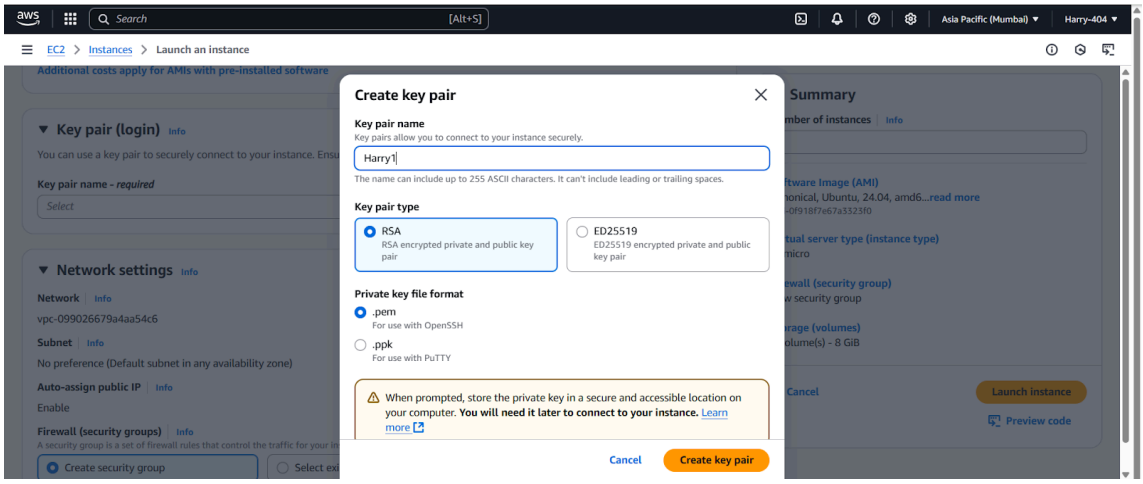
- Go to EC2 Dashboard.

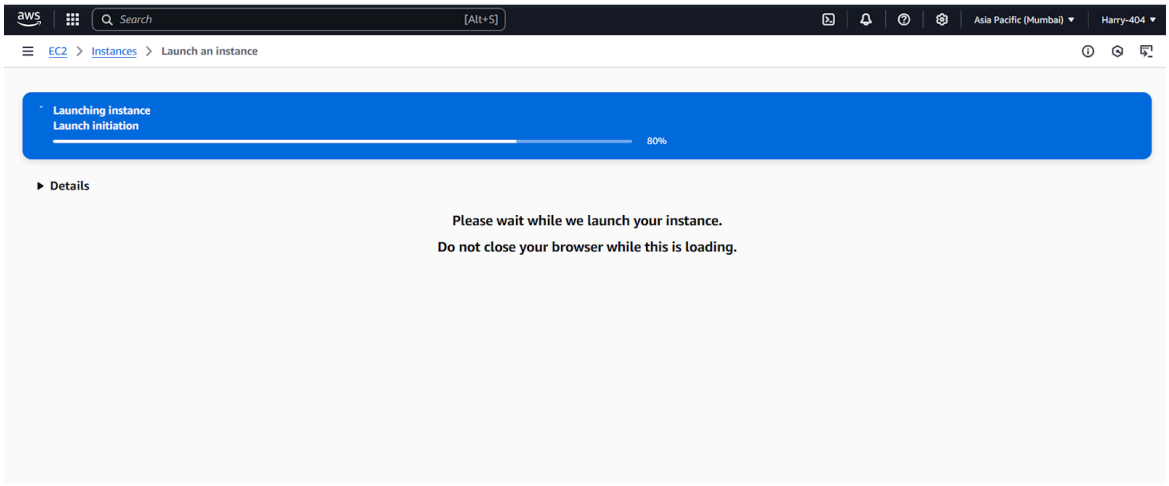
2. Launch a New Instance

- Click Instances > Launch Instances.
- Name: e.g., `Web Hosting`
- Amazon Machine Image (AMI):
  Select: `Ubuntu Server 24.04 LTS (HVM), SSD Volume Type`
- Instance Type:
  Choose: `t3.micro` (free tier eligible)
- Key Pair (login):
  - If you don't have a key, click Create new key pair.
  - Name: e.g., `Harry1`, select type (`RSA`, format `.pem`), download and save securely.
- Network settings:
  - Ensure default VPC/subnet.
  - Auto-assign public IP: Enabled.
  - Firewall (security group):
    - Create or select a security group that allows SSH (port 22) and HTTP (port 80) inbound.
- Storage: Leave default (usually 8 GiB).
- Launch Instance: Click the button and wait for success.


Search [Alt+S]
Asia Pacific (Mumbai)
Harry-404
EC2 > Instances > Launch an instance
Name and tags Info
Name
Web Hosting
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.
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-0d0ad8bb301edb745
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


EC2 > Instances > Launch an instance
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-0f918f7e67a3323f0 (64-bit (x86)) / ami-02f607855bfce66b6 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
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
AMI ID
Publish Date
Username
64-bit (x86)
ami-0f918f7e67a3323f0
2025-06-10
ubuntu
Verified provider
Instance type Info | Get advice
Instance type
t3.micro
Free tier eligible
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
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0f918f7e67a3323f0
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code

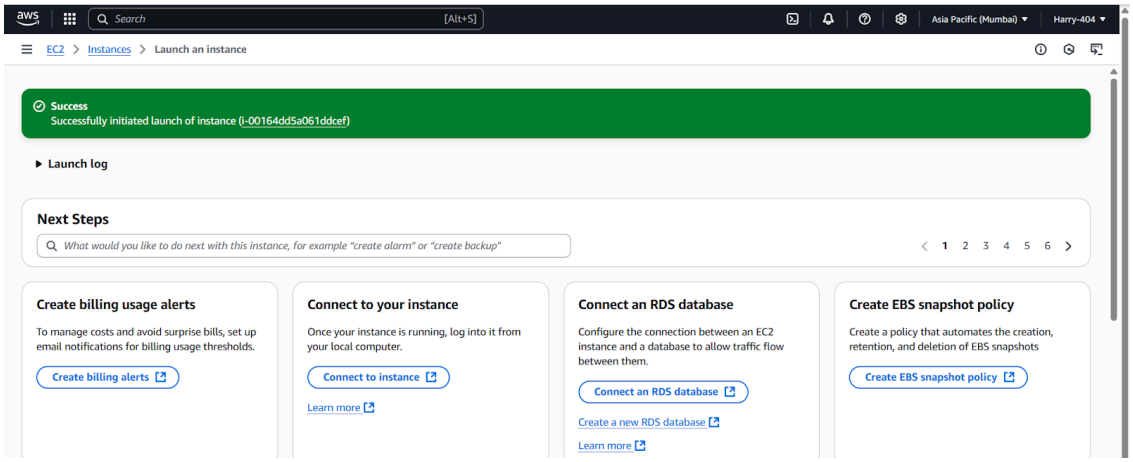

Search [Alt+S]
Asia Pacific (Mumbai)
Harry-404
EC2 > Instances > Launch an instance
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance.
Key pair name - required
Select
Network settings Info
Network Info
vpc-099026679a4aa54c6
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Firewall (security groups) Info
Create security group
Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
Harry1
The name can include up to 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
Summary
Launch instance
Preview code

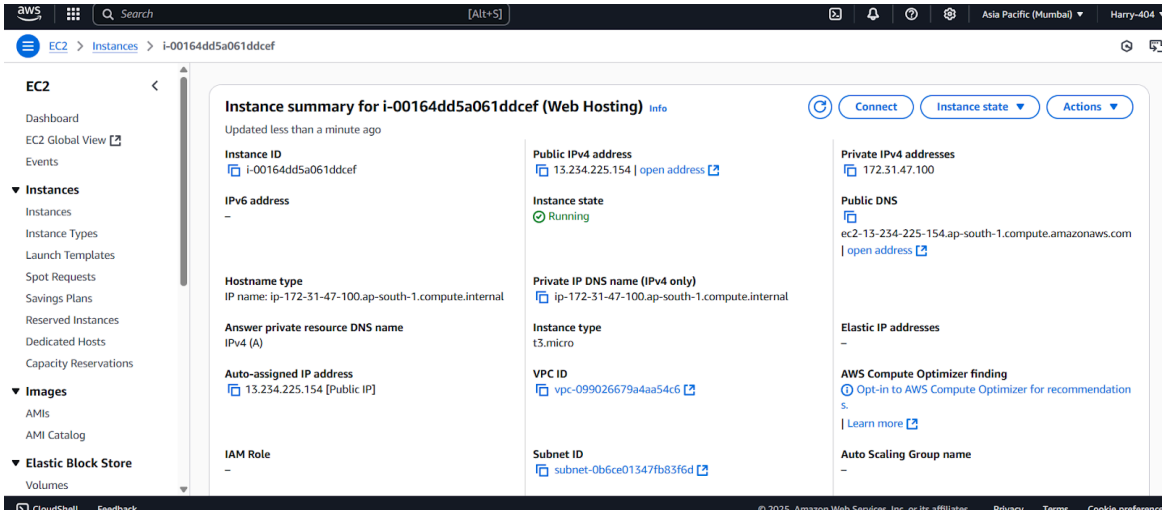


Step 2: Connect to Your Instance

1. Get Public IPv4 Address

- Find in EC2 > Instances, e.g., `13.234.225.154`.





2. Connect via SSH

- On your computer, open a terminal where your `.pem` is stored.

  ssh -i "Harry.pem" ubuntu@13.234.225.154

- Type `yes` if prompted to add the server to `known_hosts`.

```
C:\Users\Hope Foundation\Downloads>ssh -i "Harry.pem" ubuntu@13.234.225.154
The authenticity of host '13.234.225.154 (13.234.225.154)' can't be established.
ED25519 key fingerprint is SHA256:83siE1tcqteB9t7wh4UdCzbht25Bq+7YeaauE9w45ys.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.234.225.154' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Aug  4 06:07:17 UTC 2025

  System load:  0.0               Temperature:           -273.1 C
  Usage of /:   25.3% of 6.71GB   Processes:             114
  Memory usage: 22%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.47.100

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
```

Step 3: Update Packages & Install Apache

Update OS and Install Apache

```
sudo apt update
sudo apt install apache2 -y
```

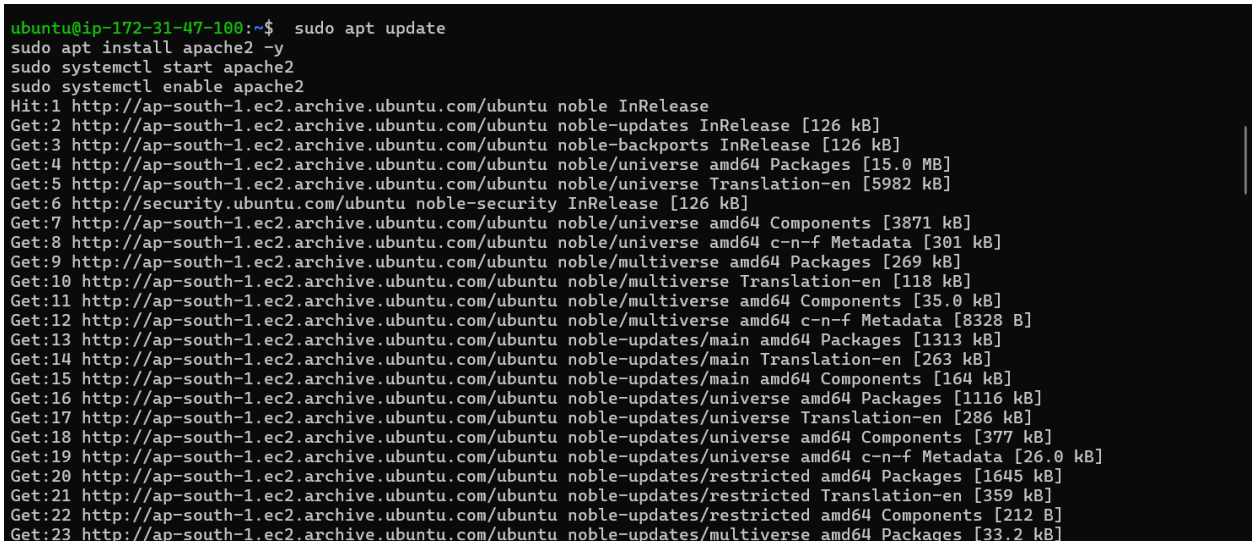
Start and Enable Apache

```
sudo systemctl start apache2
sudo systemctl enable apache2
```

Check Apache Status

```
sudo systemctl status apache2
```

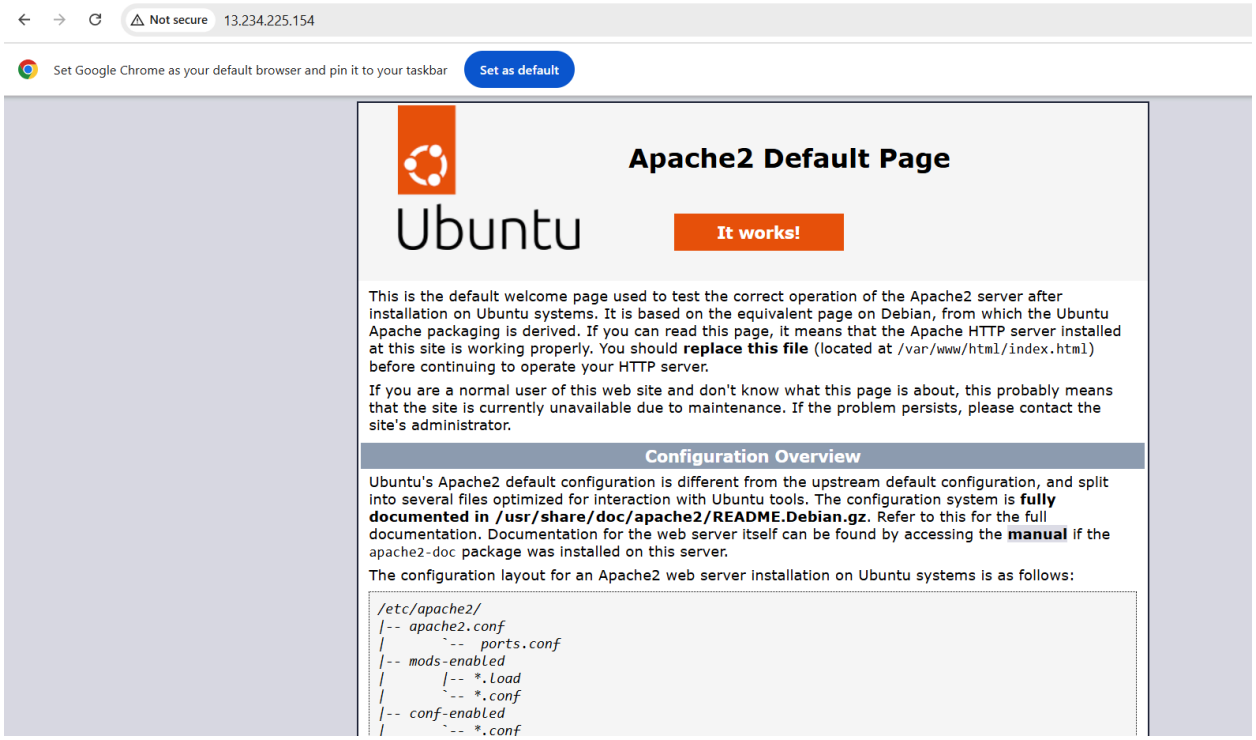
- Active (running) means Apache is working.



```
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [380 B]
Fetched 36.0 MB in 6s (6225 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
98 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 98 not upgraded.
Need to get 2086 kB of archives.
After this operation, 8090 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108
kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [1
1.2 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.7 [1330
 kB]
```

```
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-172-31-47-100:~$ sudo service apache2 start
ubuntu@ip-172-31-47-100:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-08-04 06:08:36 UTC; 1min 46s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2213 (apache2)
      Tasks: 55 (limit: 1072)
     Memory: 5.4M (peak: 5.6M)
        CPU: 40ms
     CGroup: /system.slice/apache2.service
             ├─2213 /usr/sbin/apache2 -k start
             ├─2216 /usr/sbin/apache2 -k start
             └─2217 /usr/sbin/apache2 -k start

Aug 04 06:08:36 ip-172-31-47-100 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 04 06:08:36 ip-172-31-47-100 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-47-100:~$
```

- Visit `http://<public-ip>` in a browser—you'll see the Apache default page.



Step 4: Create Your Web Page Using nano

(A) Use nano to Create/Edit Your HTML File

1. Create/Edit index.html in your home directory:

   ```
   nano index.html
   ```

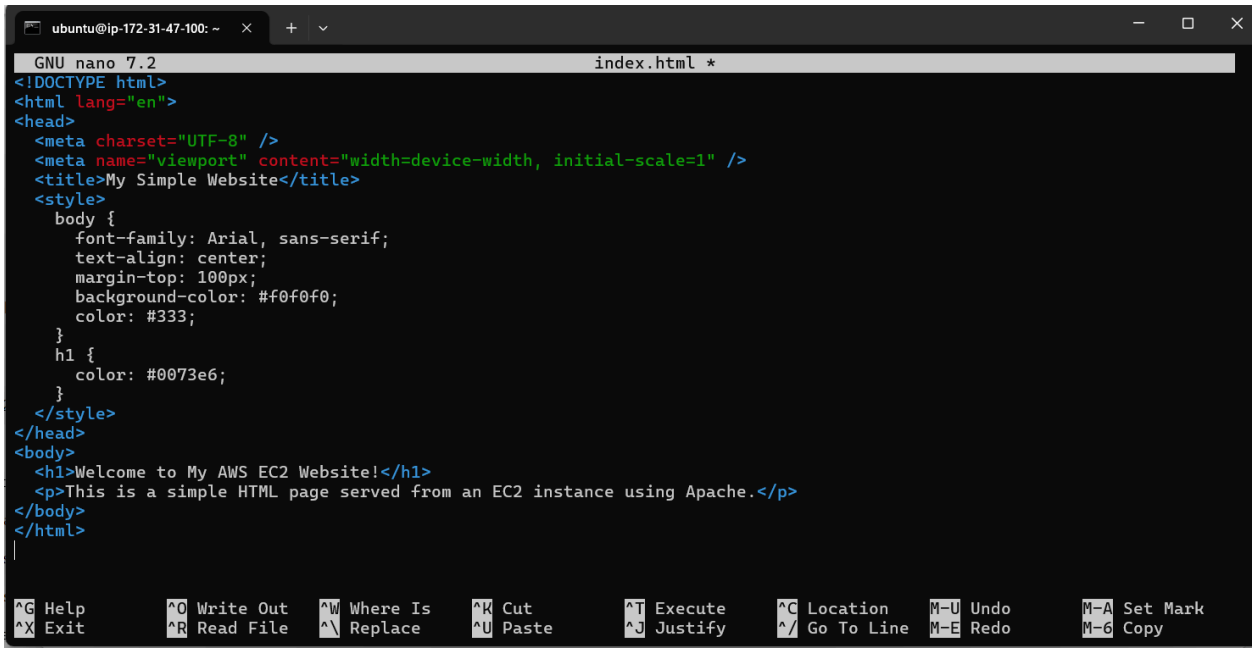
2. Write Your HTML (as in your screenshot):

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1" />
```

```
  <title>My Simple Website</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 100px;
      background-color: #f0f0f0;
      color: #333;
    }
    h1 { color: #0073e6; }
  </style>
</head>
<body>
  <h1>Welcome to My AWS EC2 Website!</h1>
  <p>This is a simple HTML page served from an EC2 instance
using Apache.</p>
</body>
</html>
```

3. Save & Exit:

    - Press `Ctrl + O` (write out), then `Enter` (save), then `Ctrl + X` (exit).



Step 6: (Optional) Check Apache Virtual Host Config

- To check which file is serving your site:
- `bash`

```
cat /etc/apache2/sites-enabled/000-default.conf
```

-
- Ensure `DocumentRoot /var/www/html` is set.

```
ubuntu@ip-172-31-47-100:~$ cat /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

(B) Replace Default Apache Page

1. Remove Default Apache Files:

   ```
   sudo rm -rf /var/www/html/*
   ```

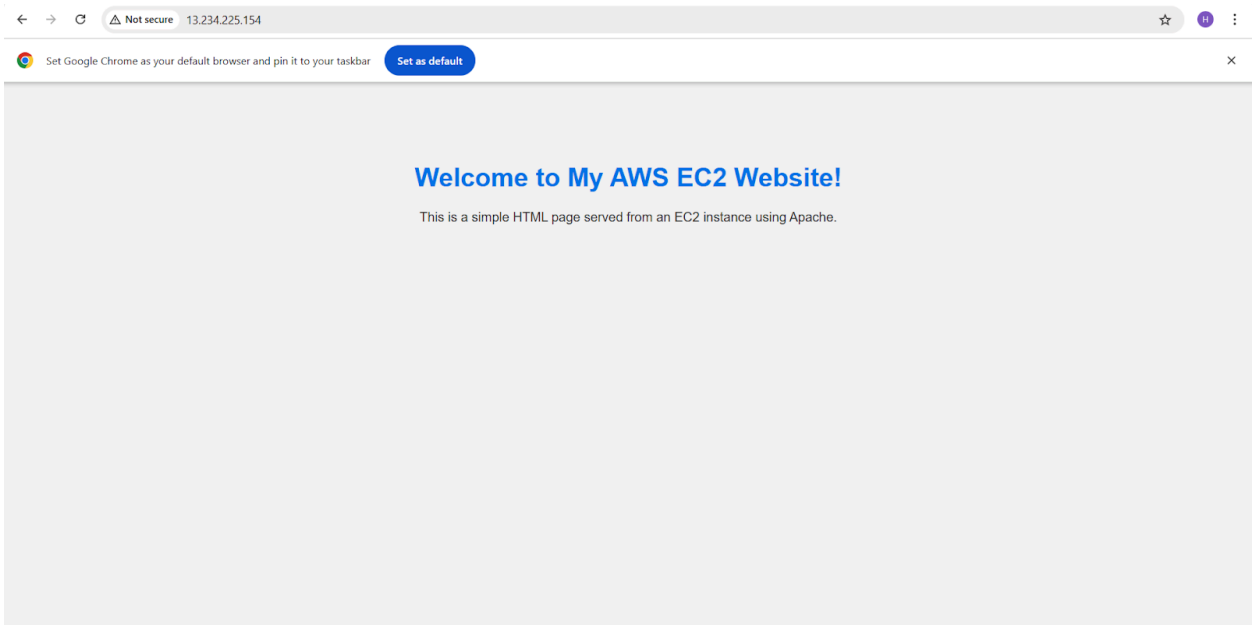
2. Copy Your `index.html` to Web Directory:

   ```
   sudo cp ~/index.html /var/www/html/
   ```

```
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
ubuntu@ip-172-31-47-100:~$ sudo rm -rf /var/www/html/* && sudo cp ~/index.html /var/www/html/
ubuntu@ip-172-31-47-100:~$
```

Step 5: Test Your Website

- Open a browser and go to:
  `http://<public-ip>` (e.g., `http://13.234.225.154`)
- You should see your custom webpage:
  "Welcome to My AWS EC2 Website!...etc"



Summary of Key Commands

```
# SSH into instance
ssh -i "Harry.pem" ubuntu@13.234.225.154

# Update and install apache2
sudo apt update
sudo apt install apache2 -y

# Start apache2
sudo systemctl start apache2
sudo systemctl enable apache2

# Create HTML file with nano
nano index.html

# Remove default files, copy your HTML
sudo rm -rf /var/www/html/*
sudo cp ~/index.html /var/www/html/
```