README.md

Question

“Imagine you are part of a startup team tasked with creating a prototype for a web application. Your first step is to provision a server and set up a simple landing page to demonstrate your team’s capabilities to potential investors. Your task is as follows:”

Tasks

1. **Provisioning the Server:**
    - Use any virtualization or cloud platform (AWS.) to set up a **Linux server**.
    - Install a Linux distribution of your choice (e.g., Ubuntu).
2. **Web Server Setup:**
    - Install a web server (e.g., Apache, Nginx) to serve web content.
3. **HTML Page Deployment:**
    - Create a simple HTML page with the following information:
        - Your name.
        - A project title: “Welcome to [Your Name] Landing Page.”
        - A brief description of your project.
        - Your full bio with every interesting information about you
    - Deploy the HTML page on the server.
4. **Networking:**
    - Configure the server to allow HTTP traffic (port 80).
    - Provide the public IP address (or URL if using DNS) so your page can be accessed from any browser.

Deliverables:

- The **public IP address or URL** of your web page.
- A screenshot showing your HTML page in a browser.
- Write clear, step-by-step documentation of how you provisioned the server, installed the web server, deployed the HTML page, and configured networking.

Bonus Tasks (Optional for Extra Credit):

- Configure HTTPS for your web server using a free SSL certificate (e.g., Let's Encrypt).

DOCUMENTATION

Stack used

1. Linux distribution: Ubuntu, using Termius as the SSH client and terminal emulator
2. Cloud virtualization platform: AWS
3. Web server: Apache2
4. IDE: VS code
5. Version control: Git

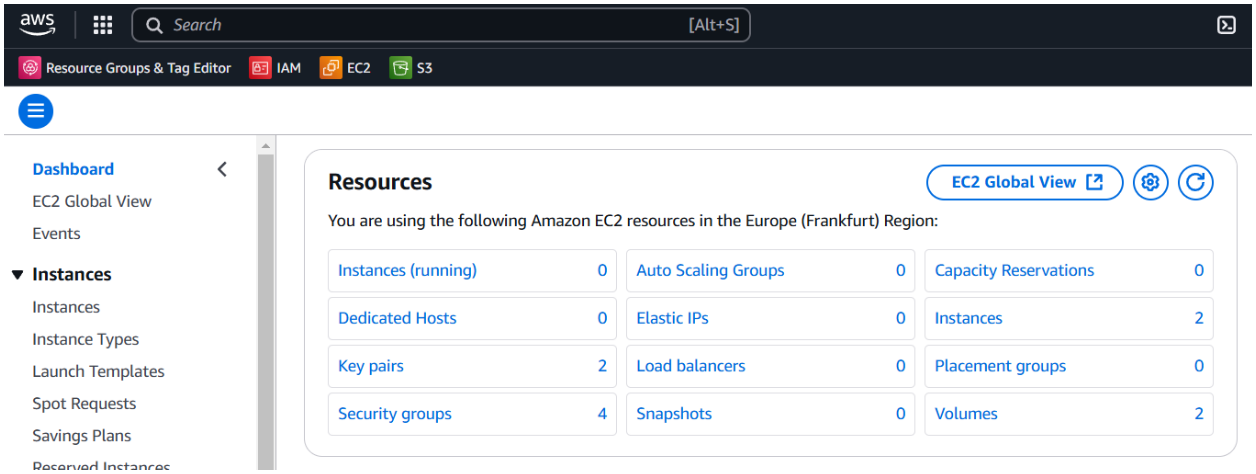
Introduction

The project assumes that user already has a github account set up and that commits are being made on the project repository at intermittent sections of the documentation. An AWS account is also needed to complete the project and a folder created on your local machine.

Provisioning the Server

Setting up the Linux server and installing Ubuntu on AWS: Once signed-in to the AWS platform, navigate to the EC2 service. This you can find by searching for 'EC2' on the search bar / `Alt + S`.

Click on 'Instances' on the below page.



Resources

You are using the following Amazon EC2 resources in the Europe (Frankfurt) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	2
Key pairs	2	Load balancers	0	Placement groups	0
Security groups	4	Snapshots	0	Volumes	2

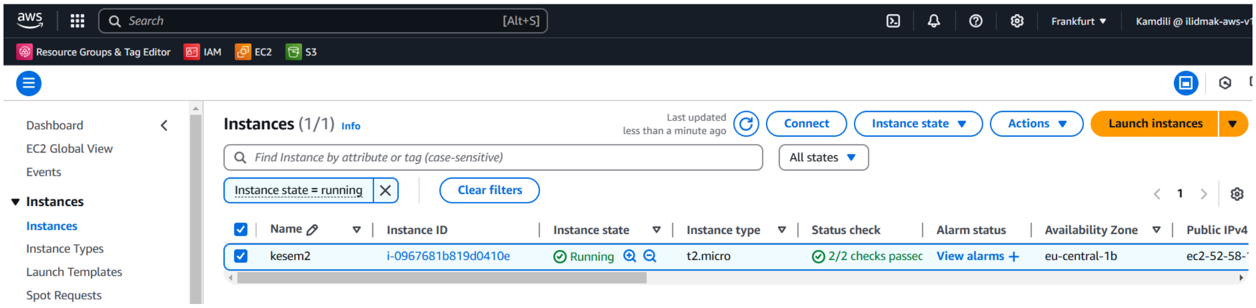
Servers are referred to as instances on AWS. Click on the 'Launch an instance' button to navigate to the page where you create a virtual machine that runs on the AWS cloud.

- Set the server name
- Select Ubuntu as the OS Image (AMI)
- Choose the architecture for your server [64-bit (x86) in this use case]
- Choose your instance type [t2 micro in this use case]
- Create a new key pair
  - enter a key pair name
  - RSA key pair type
  - .pem private key file format

This creates the new key pair for the project and downloads it to your local machine. Save the downloaded .pem file in your project folder.

- Toggle on `create a security group` and allow SSH traffic [from Anywhere 0.0.0.0/0 in this use case]
- Configure the amount of storage needed for the project [15 GiB in this use case]
- Click `Launch instance`

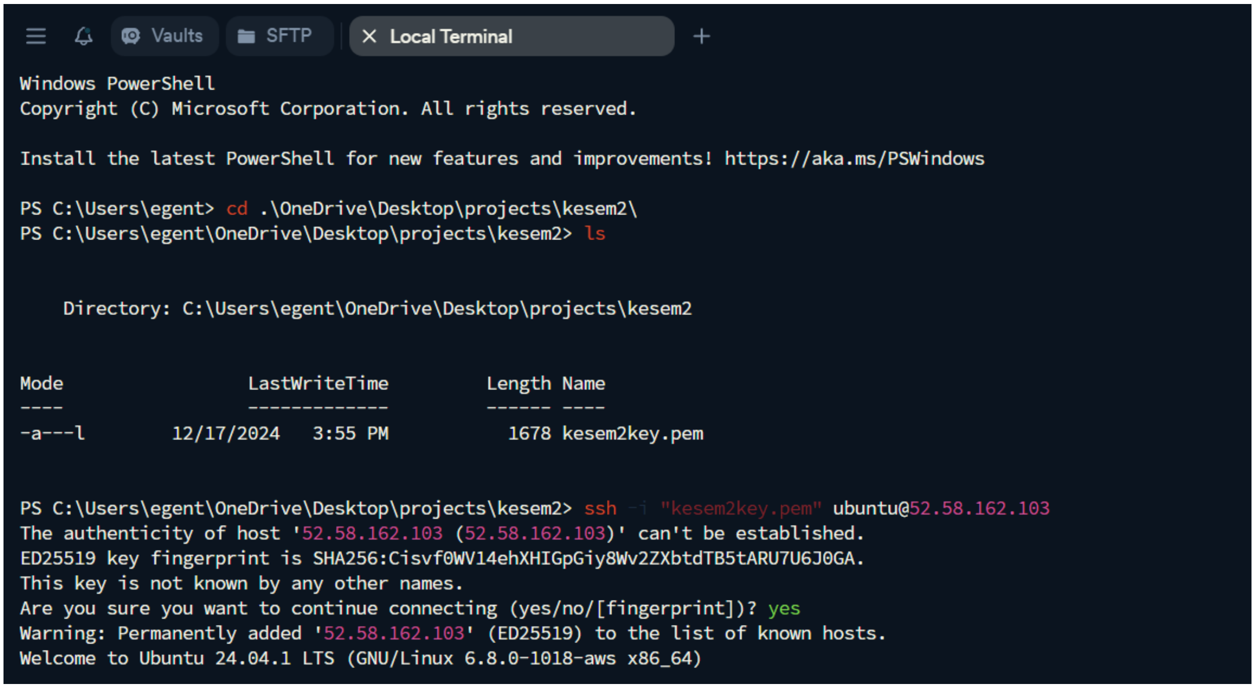
With the newly created linux server running, please go back to the EC2 page > Instances (running) > and then connect to your server.



Instances (1/1)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	kesem2	i-0967681b819d0410e	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	eu-central-1b	ec2-52-58-

Using the SSH client login page, follow the instructions given and login to your server with the public DNS or IPv4 address. Below is the login image using Termius; navigate to the directory with .pem file and ssh into your server.

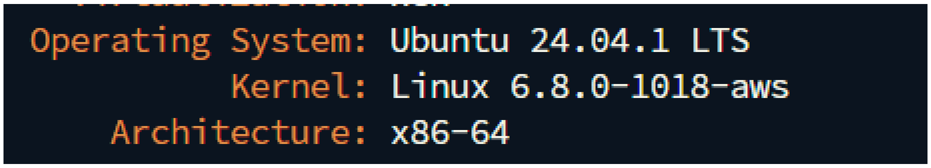


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\egent> cd .\OneDrive\Desktop\projects\kesem2\
PS C:\Users\egent\OneDrive\Desktop\projects\kesem2> ls


    Directory: C:\Users\egent\OneDrive\Desktop\projects\kesem2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---l        12/17/2024   3:55 PM           1678 kesem2key.pem


PS C:\Users\egent\OneDrive\Desktop\projects\kesem2> ssh -i "kesem2key.pem" ubuntu@52.58.162.103
The authenticity of host '52.58.162.103 (52.58.162.103)' can't be established.
ED25519 key fingerprint is SHA256:Cisvf0WV14ehXHIGpGiy8Wv2ZXbtdTB5tARU7U6J0GA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.58.162.103' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)
```

Run the `hostnamectl` command:

~~~
```bash
hostnamectl
```
~~~

This command displays system hostname and operating system information and shows the below to confirm the setup on AWS.



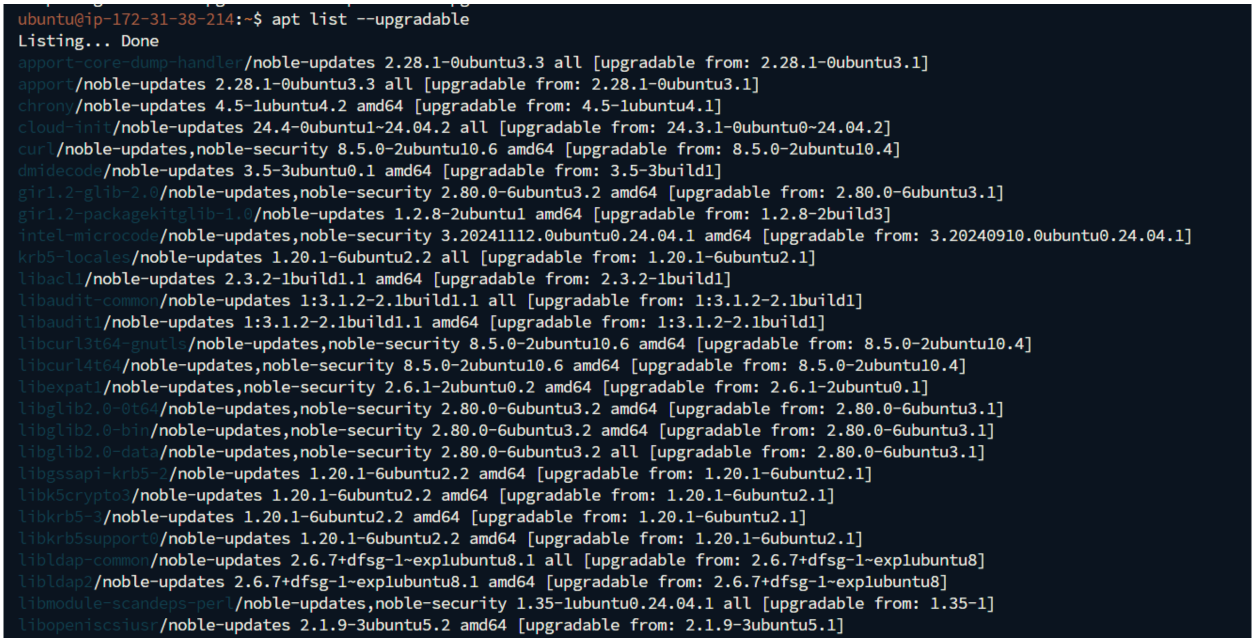
```
Operating System: Ubuntu 24.04.1 LTS
          Kernel: Linux 6.8.0-1018-aws
    Architecture: x86-64
```

Web Server Setup

Apache2 was installed to serve the html web content that would be generated. Run `sudo apt update` to ensure your package list is up to date.

Then run the `systemctl status apache2` or `apache2 -v` commands to check if you have apache running on your ubuntu system. If the CLI shows ‘not found’, Apache2 is not yet installed on your system.

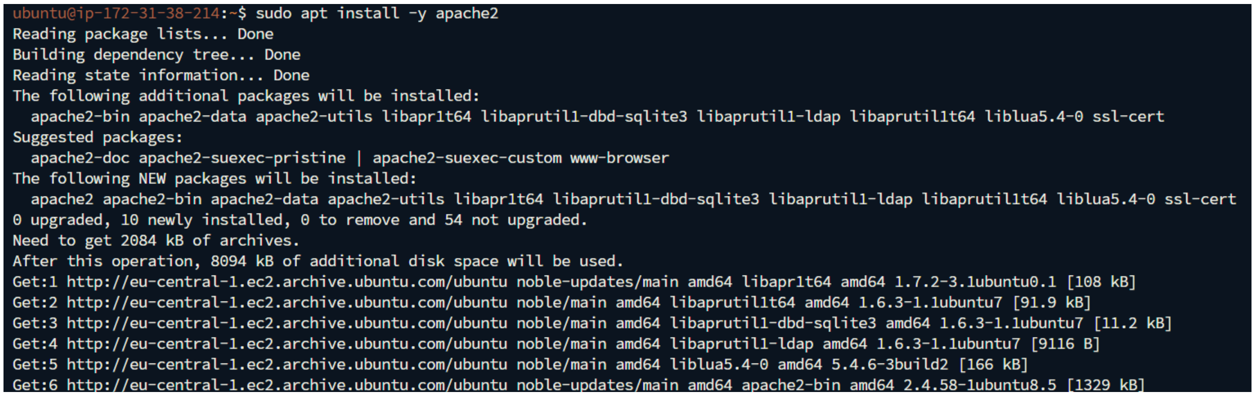
Additionally, run `apt list --upgradable` to see all packages currently available on your server.



```
ubuntu@ip-172-31-38-214:~$ apt list --upgradable
Listing... Done
apport-core-dump-handler/noble-updates 2.28.1-0ubuntu3.3 all [upgradable from: 2.28.1-0ubuntu3.1]
apport/noble-updates 2.28.1-0ubuntu3.3 all [upgradable from: 2.28.1-0ubuntu3.1]
chrony/noble-updates 4.5-1ubuntu4.2 amd64 [upgradable from: 4.5-1ubuntu4.1]
cloud-init/noble-updates 24.4-0ubuntu1~24.04.2 all [upgradable from: 24.3.1-0ubuntu0~24.04.2]
curl/noble-updates,noble-security 8.5.0-2ubuntu10.6 amd64 [upgradable from: 8.5.0-2ubuntu10.4]
dmidecode/noble-updates 3.5-3ubuntu0.1 amd64 [upgradable from: 3.5-3build1]
gir1.2-glib-2.0/noble-updates,noble-security 2.80.0-6ubuntu3.2 amd64 [upgradable from: 2.80.0-6ubuntu3.1]
gir1.2-packagekitglib-1.0/noble-updates 1.2.8-2ubuntu1 amd64 [upgradable from: 1.2.8-2build3]
intel-microcode/noble-updates,noble-security 3.20241112.0ubuntu0.24.04.1 amd64 [upgradable from: 3.20240910.0ubuntu0.24.04.1]
krb5-locales/noble-updates 1.20.1-6ubuntu2.2 all [upgradable from: 1.20.1-6ubuntu2.1]
libacl1/noble-updates 2.3.2-1build1.1 amd64 [upgradable from: 2.3.2-1build1]
libaudit-common/noble-updates 1:3.1.2-2.1build1.1 all [upgradable from: 1:3.1.2-2.1build1]
libaudit1/noble-updates 1:3.1.2-2.1build1.1 amd64 [upgradable from: 1:3.1.2-2.1build1]
libcurl3t64-gnutls/noble-updates,noble-security 8.5.0-2ubuntu10.6 amd64 [upgradable from: 8.5.0-2ubuntu10.4]
libcurl4t64/noble-updates,noble-security 8.5.0-2ubuntu10.6 amd64 [upgradable from: 8.5.0-2ubuntu10.4]
libexpat1/noble-updates,noble-security 2.6.1-2ubuntu0.2 amd64 [upgradable from: 2.6.1-2ubuntu0.1]
libglib2.0-0t64/noble-updates,noble-security 2.80.0-6ubuntu3.2 amd64 [upgradable from: 2.80.0-6ubuntu3.1]
libglib2.0-bin/noble-updates,noble-security 2.80.0-6ubuntu3.2 amd64 [upgradable from: 2.80.0-6ubuntu3.1]
libglib2.0-data/noble-updates,noble-security 2.80.0-6ubuntu3.2 all [upgradable from: 2.80.0-6ubuntu3.1]
libgssapi-krb5-2/noble-updates 1.20.1-6ubuntu2.2 amd64 [upgradable from: 1.20.1-6ubuntu2.1]
libk5crypto3/noble-updates 1.20.1-6ubuntu2.2 amd64 [upgradable from: 1.20.1-6ubuntu2.1]
libkrb5-3/noble-updates 1.20.1-6ubuntu2.2 amd64 [upgradable from: 1.20.1-6ubuntu2.1]
libkrb5support0/noble-updates 1.20.1-6ubuntu2.2 amd64 [upgradable from: 1.20.1-6ubuntu2.1]
libldap-common/noble-updates 2.6.7+dfsg-1~exp1ubuntu8.1 all [upgradable from: 2.6.7+dfsg-1~exp1ubuntu8]
libldap2/noble-updates 2.6.7+dfsg-1~exp1ubuntu8.1 amd64 [upgradable from: 2.6.7+dfsg-1~exp1ubuntu8]
libmodule-scandeps-perl/noble-updates,noble-security 1.35-1ubuntu0.24.04.1 all [upgradable from: 1.35-1]
libopeniscsiusr/noble-updates 2.1.9-3ubuntu5.2 amd64 [upgradable from: 2.1.9-3ubuntu5.1]
```

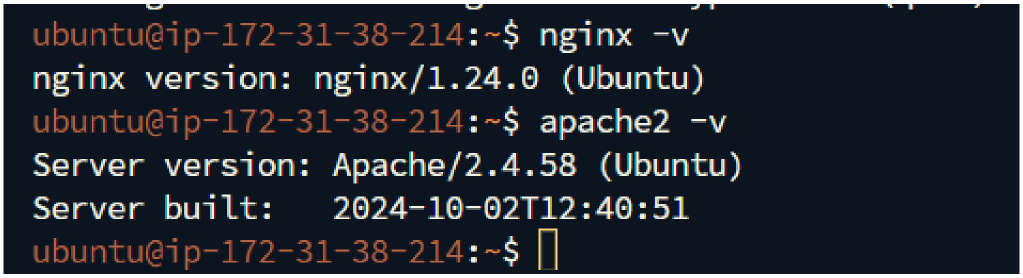
Run `apt search apache2` to search for the package on Ubuntu's default repositories before installation.

Installing Apache: To install apache web server, run `sudo apt install -y apache2` to install the latest version of apache.



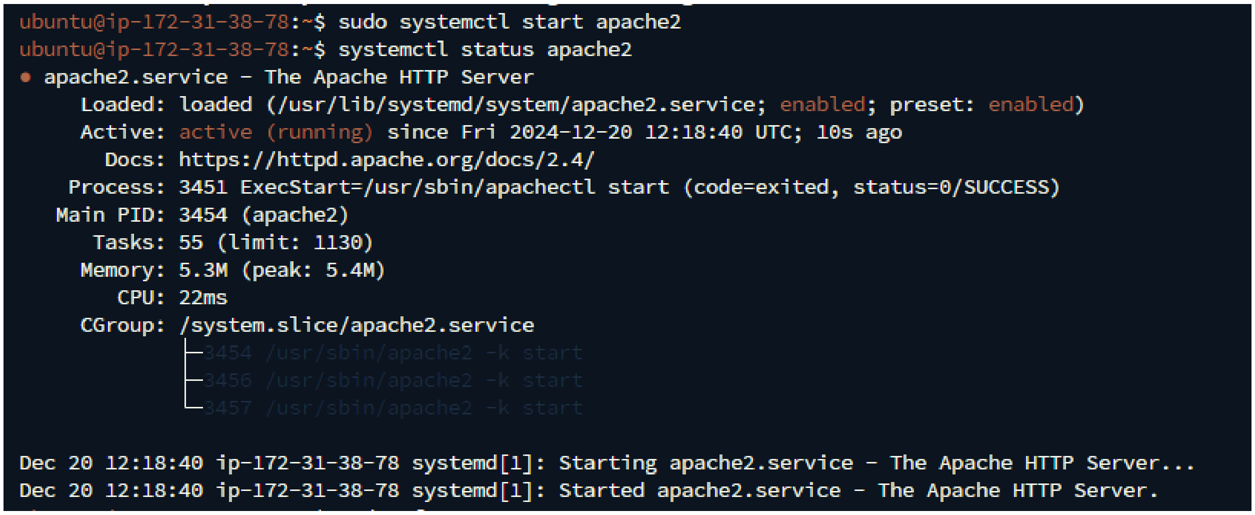
```
ubuntu@ip-172-31-38-214:~$ sudo apt install -y apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap 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 54 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Get:1 http://eu-central-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Get:2 http://eu-central-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://eu-central-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://eu-central-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://eu-central-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://eu-central-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.5 [1329 kB]
```

Using the `apache2 -v` command, confirm if the package is installed on the server.



```
ubuntu@ip-172-31-38-214:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-38-214:~$ apache2 -v
Server version: Apache/2.4.58 (Ubuntu)
Server built:   2024-10-02T12:40:51
ubuntu@ip-172-31-38-214:~$ 
```

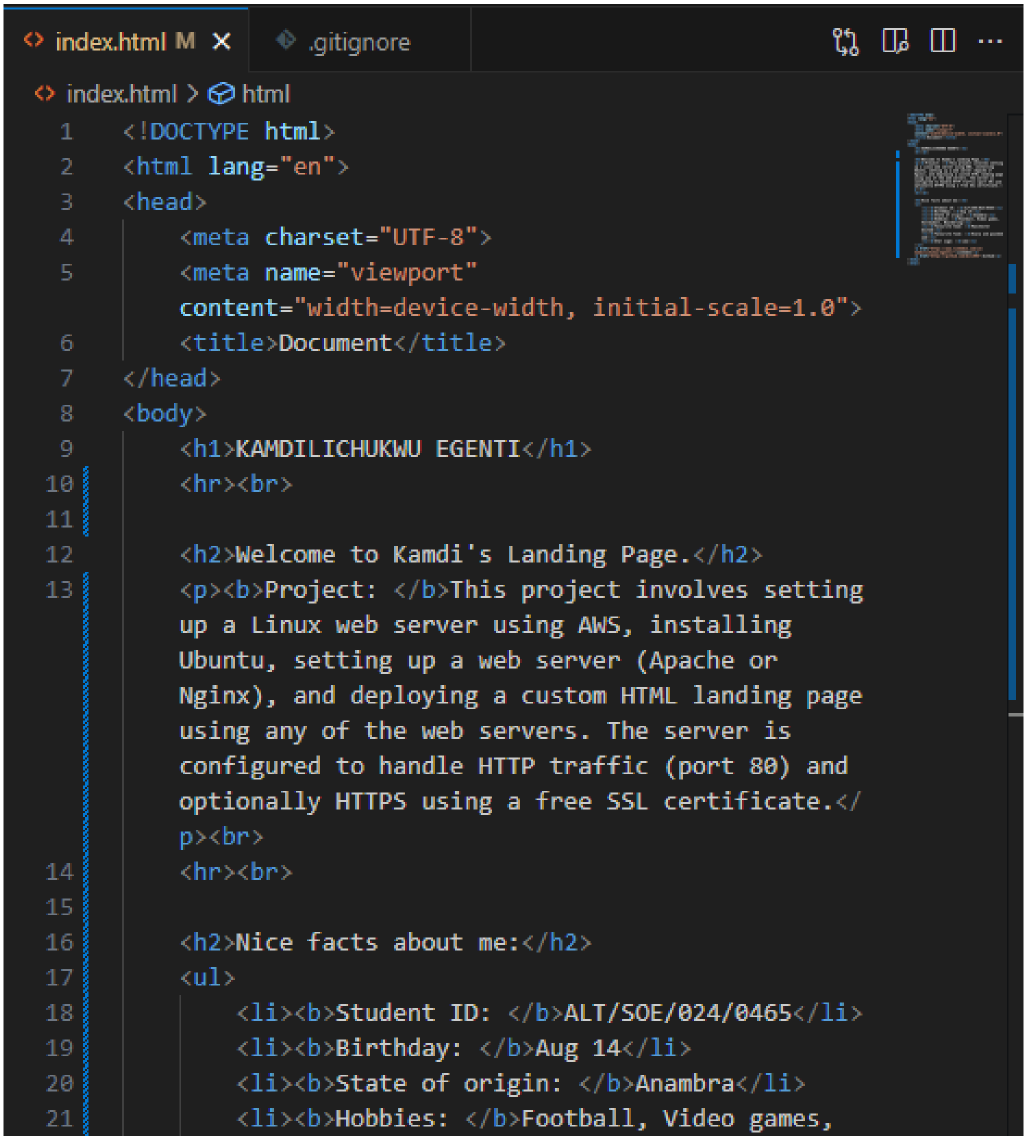
Use `sudo systemctl start apache2` to start running the apache web server on your instance and `systemctl status apache2` to confirm if the web server is active (running).



```
ubuntu@ip-172-31-38-78:~$ sudo systemctl start apache2
ubuntu@ip-172-31-38-78:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-12-20 12:18:40 UTC; 10s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3451 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 3454 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.3M (peak: 5.4M)
        CPU: 22ms
     CGroup: /system.slice/apache2.service
             ├─3454 /usr/sbin/apache2 -k start
             ├─3456 /usr/sbin/apache2 -k start
             └─3457 /usr/sbin/apache2 -k start

Dec 20 12:18:40 ip-172-31-38-78 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 20 12:18:40 ip-172-31-38-78 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

HTML Page Deployment

Create a simple HTML page: Created a simple `index.html` page on VS code IDE and below is a snapshot of the simple webpage used for the project.

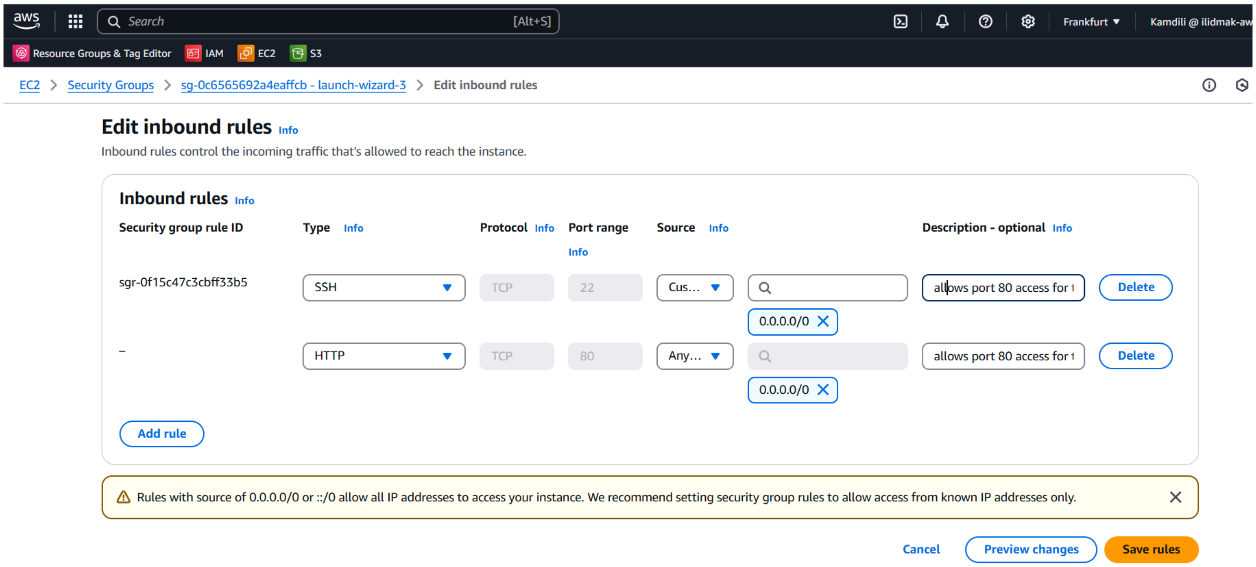


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>KAMDILICHUKWU EGENTI</h1>
    <hr><br>

    <h2>Welcome to Kamdi's Landing Page.</h2>
    <p><b>Project: </b>This project involves setting up a Linux web server using AWS, installing Ubuntu, setting up a web server (Apache or Nginx), and deploying a custom HTML landing page using any of the web servers. The server is configured to handle HTTP traffic (port 80) and optionally HTTPS using a free SSL certificate.</p><br>
    <hr><br>

    <h2>Nice facts about me:</h2>
    <ul>
        <li><b>Student ID: </b>ALT/SOE/024/0465</li>
        <li><b>Birthday: </b>Aug 14</li>
        <li><b>State of origin: </b>Anambra</li>
        <li><b>Hobbies: </b>Football, Video games,
```

To deploy above webpage to the web server, port 80 has to be added in configuration for HTTP access. This is either done through the command line interface or on your AWS security group which was created earlier while provisioning your instance.

Network Configuration

Configuring HTTP (Port 80): On the AWS console, go to the EC2 page and click on security groups. Click on the security group configured for the linux server and edit the inbound rules by adding HTTP (port 80).


aws
Search
[Alt+S]
Frankfurt ▼
Kamdili @ ilidmak-aw
Resource Groups & Tag Editor
IAM
EC2
S3
EC2 > Security Groups > sg-0c6565692a4eaffcb - launch-wizard-3 > Edit inbound rules

Edit inbound rules Info

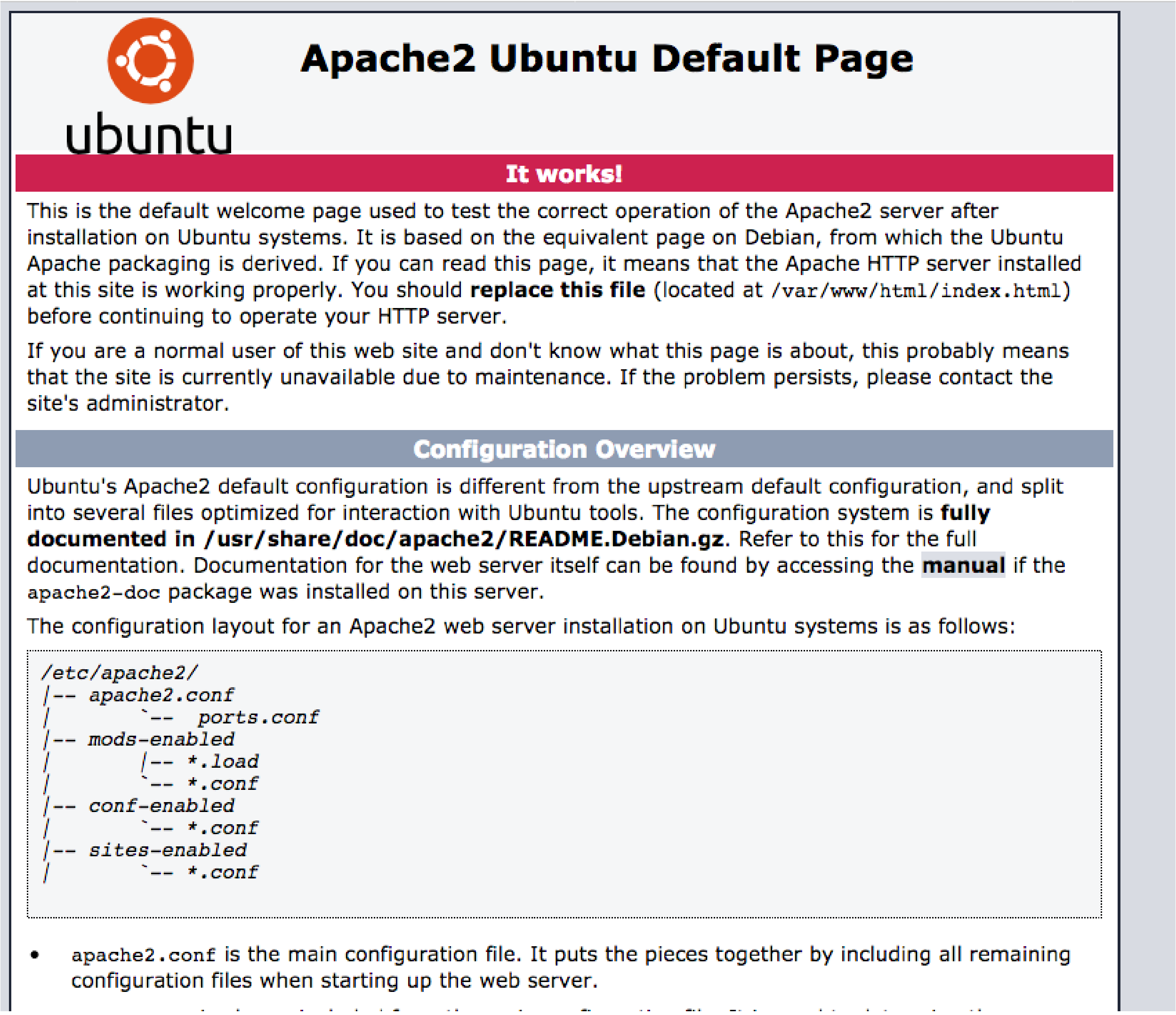
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info
sgr-0f15c47c3cbff33b5	SSH	TCP	22	Cus…	0.0.0.0/0	allows port 80 access for t
–	HTTP	TCP	80	Any…	0.0.0.0/0	allows port 80 access for t

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

Once the Port 80 has been added to the security group, search the HTTP for your IP address to confirm if the web server is running and notice below page should be displayed.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

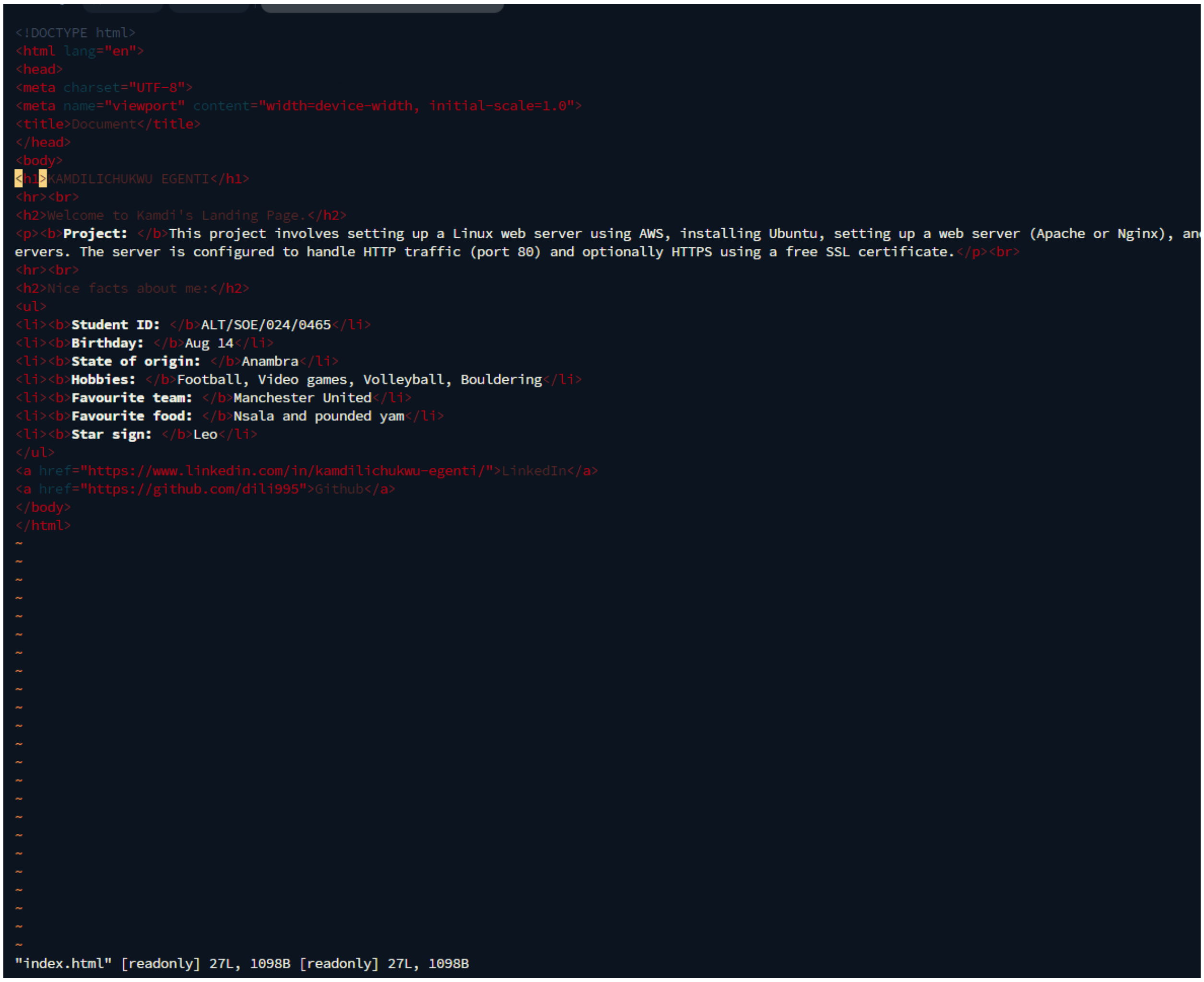
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

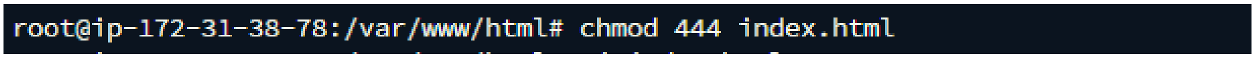
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

Then cd to `/var/www.html` directory, create an index.html file and paste in your code from VS code using nano or vim text editor.



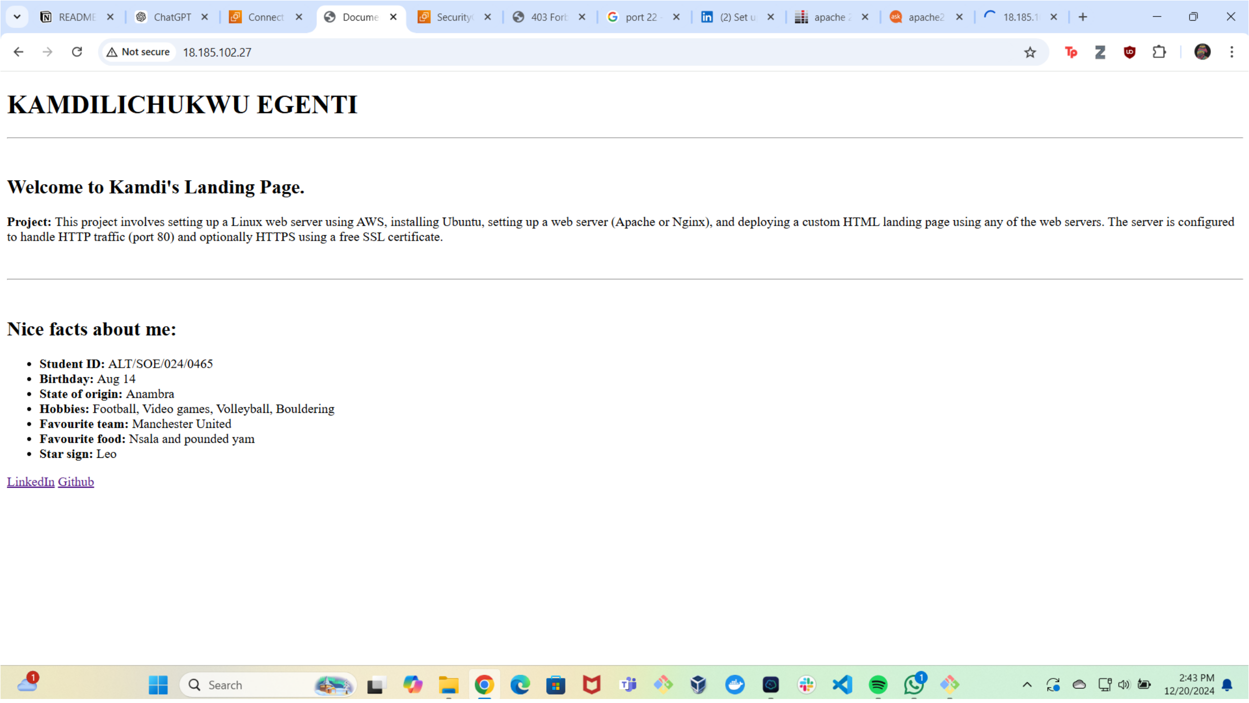
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<h1>KAMDILICHUKWU EGENTI</h1>
<hr><br>
<h2>Welcome to Kamdi's Landing Page.</h2>
<p><b>Project: </b>This project involves setting up a Linux web server using AWS, installing Ubuntu, setting up a web server (Apache or Nginx), an
ervers. The server is configured to handle HTTP traffic (port 80) and optionally HTTPS using a free SSL certificate.</p><br>
<hr><br>
<h2>Nice facts about me:</h2>
<ul>
<li><b>Student ID: </b>ALT/SOE/024/0465</li>
<li><b>Birthday: </b>Aug 14</li>
<li><b>State of origin: </b>Anambra</li>
<li><b>Hobbies: </b>Football, Video games, Volleyball, Bouldering</li>
<li><b>Favourite team: </b>Manchester United</li>
<li><b>Favourite food: </b>Nsala and pounded yam</li>
<li><b>Star sign: </b>Leo</li>
</ul>
<a href="https://www.linkedin.com/in/kamdilichukwu-egenti/">LinkedIn</a>
<a href="https://github.com/dili995">Github</a>
</body>
</html>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"index.html" [readonly] 27L, 1098B [readonly] 27L, 1098B
```

Set the file to be readable using chmod file permission.



```
root@ip-172-31-38-78:/var/www/html# chmod 444 index.html
```

After this, the html file will be accessible via the instances’s IP address in a browser:



KAMDILICHUKWU EGENTI

Welcome to Kamdi's Landing Page.

Project: This project involves setting up a Linux web server using AWS, installing Ubuntu, setting up a web server (Apache or Nginx), and deploying a custom HTML landing page using any of the web servers. The server is configured to handle HTTP traffic (port 80) and optionally HTTPS using a free SSL certificate.

Nice facts about me:

- **Student ID:** ALT/SOE/024/0465
- **Birthday:** Aug 14
- **State of origin:** Anambra
- **Hobbies:** Football, Video games, Volleyball, Bouldering
- **Favourite team:** Manchester United
- **Favourite food:** Nsala and pounded yam
- **Star sign:** Leo

LinkedIn Github

Public IP Address: 18.185.102.27