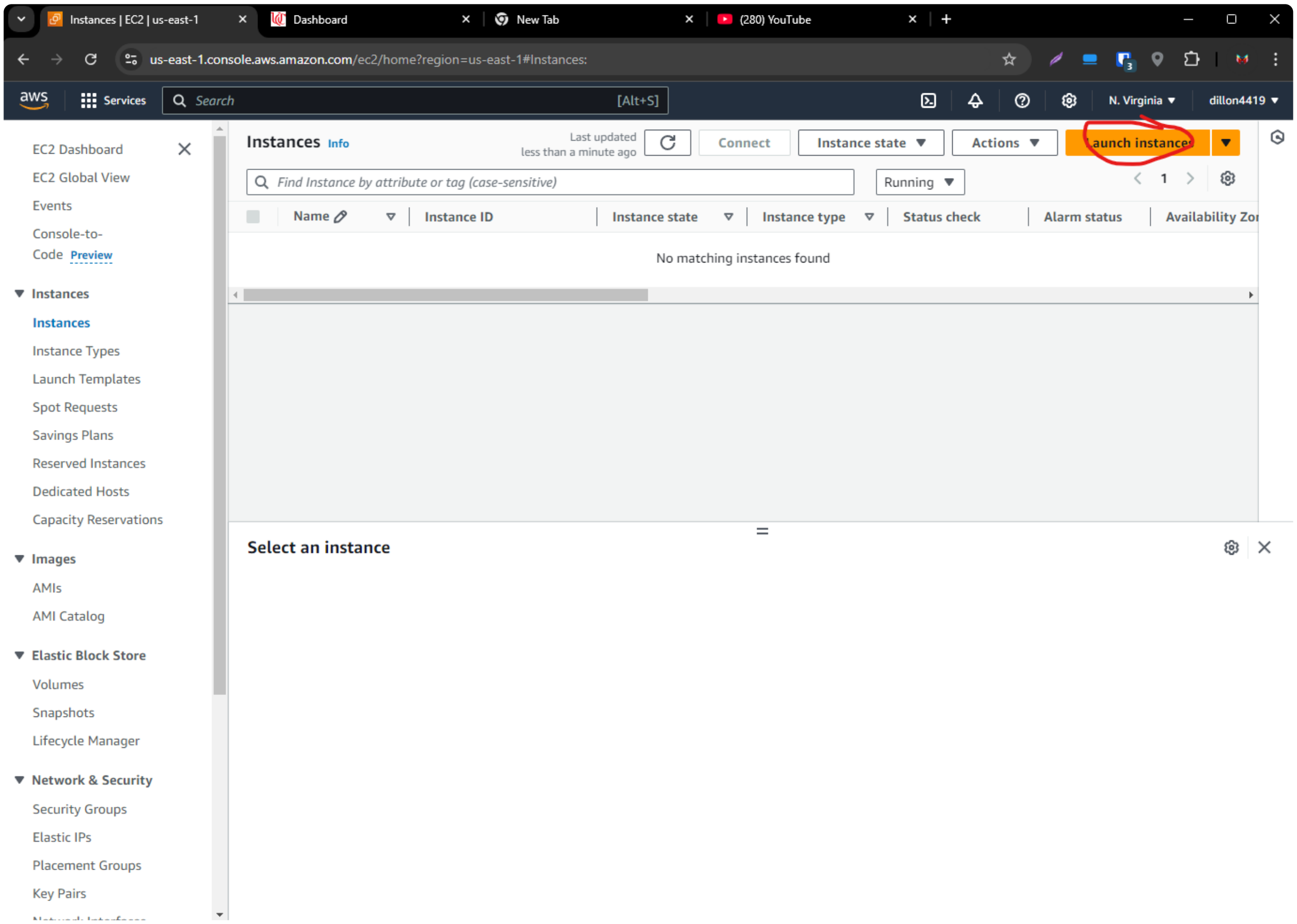
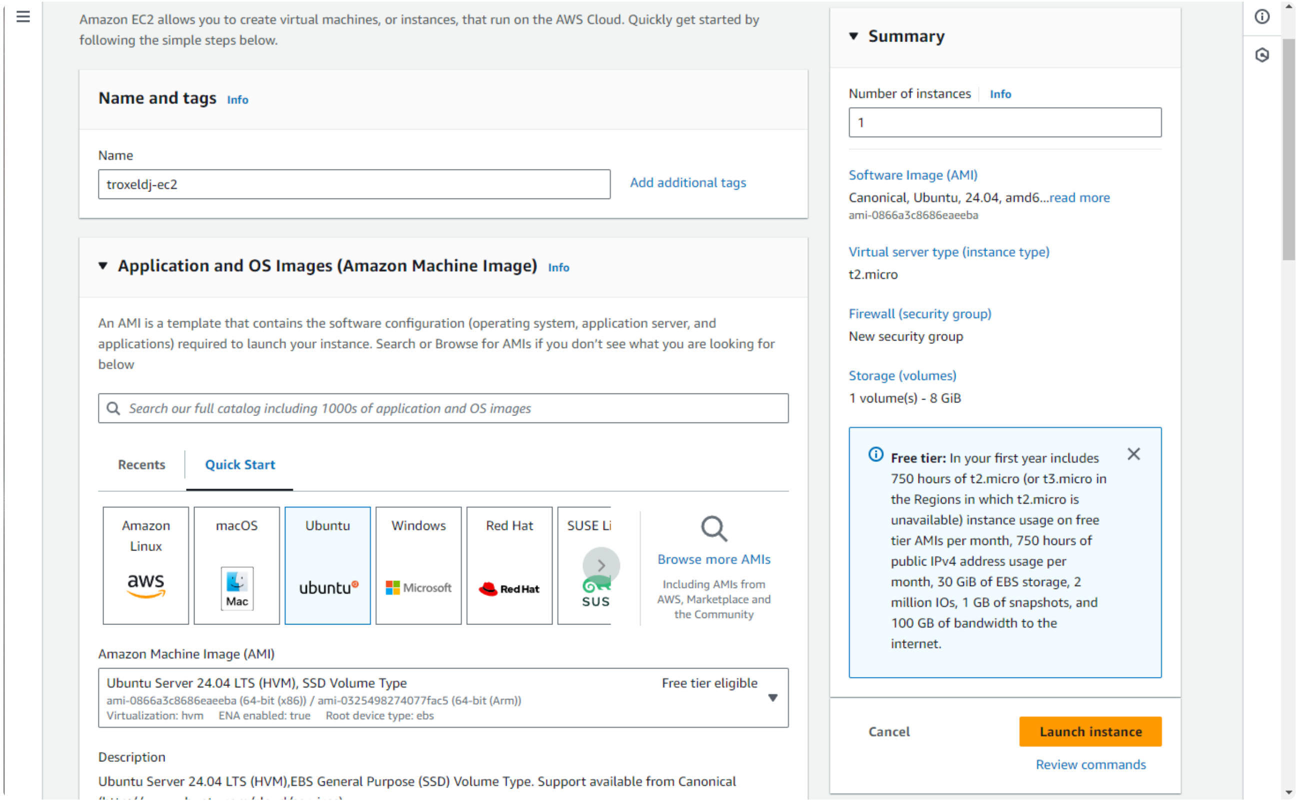
Cloud-Computing-Assignment_2

Step 1: Launch EC2 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 Info

Name: troxeldj-ec2

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

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (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

Summary

Number of instances Info: 1

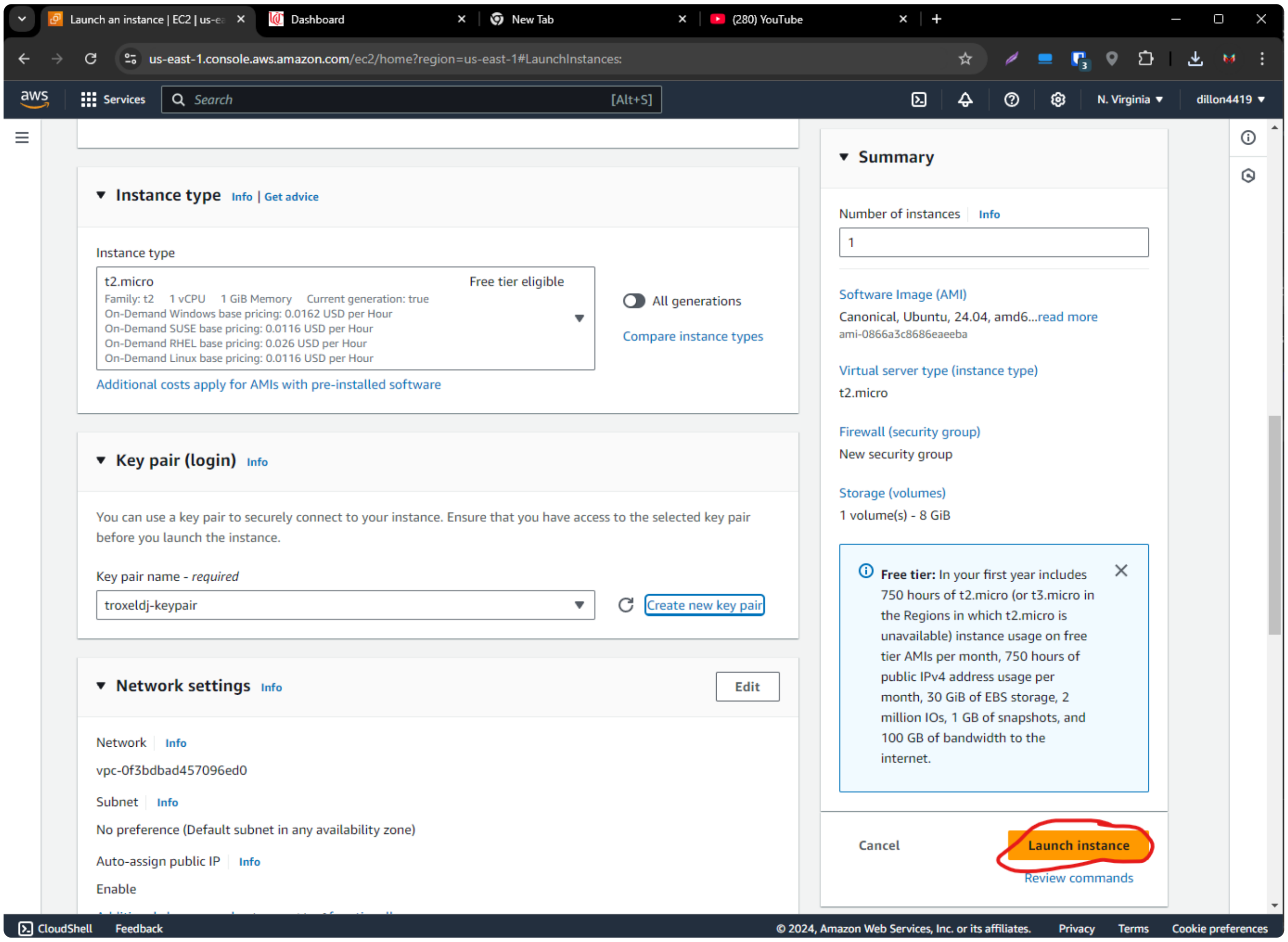
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Instance type Info | Get advice

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: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 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*: troxeldj-keypair

Network settings Info

Network Info: vpc-0f3bdbad457096ed0

Subnet Info: No preference (Default subnet in any availability zone)

Auto-assign public IP Info: Enable

Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba

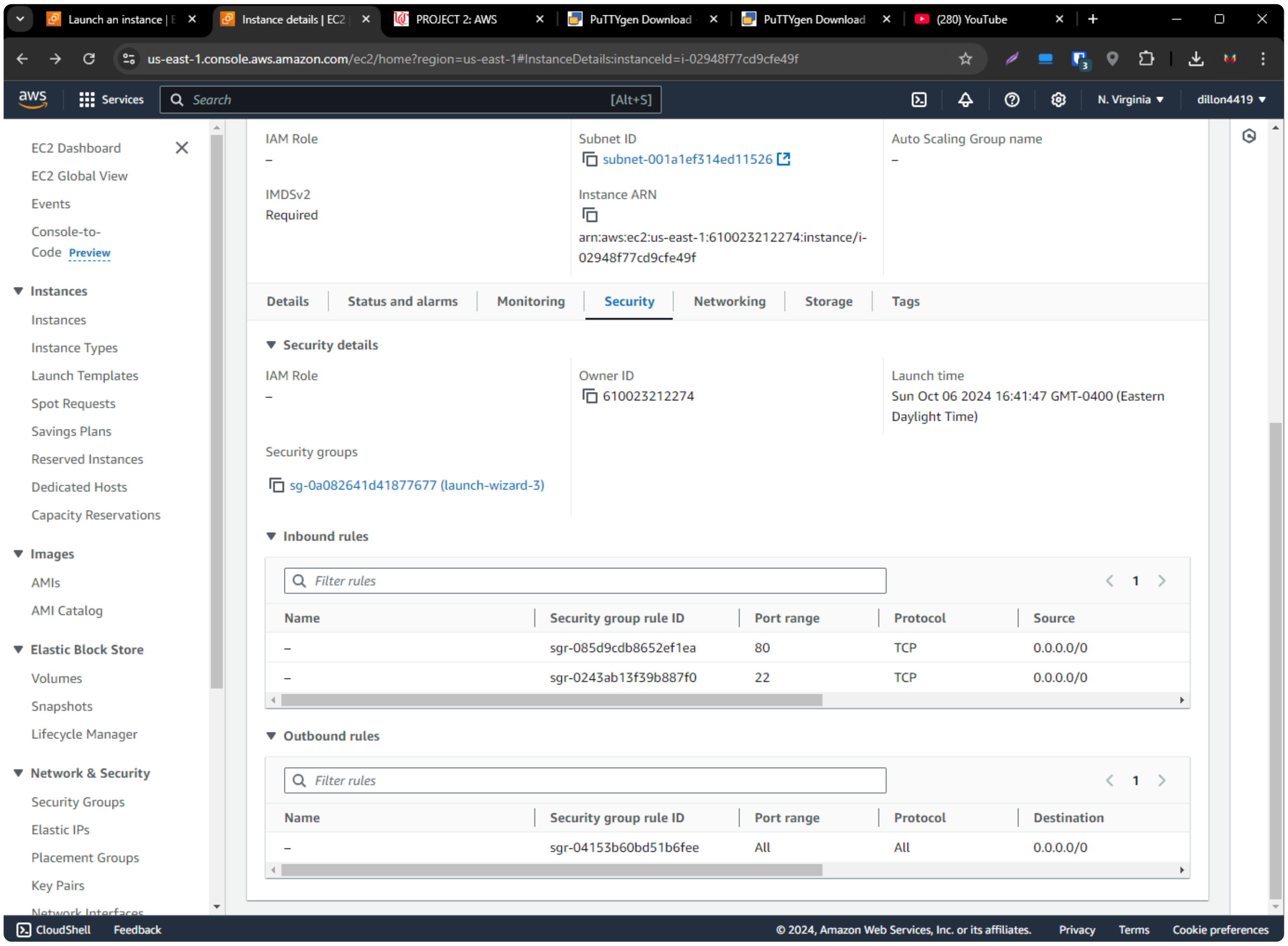
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Security Rules



IAM Role
–

Subnet ID
subnet-001a1ef314ed11526

Auto Scaling Group name
–

IMDSv2
Required

Instance ARN
arn:aws:ec2:us-east-1:610023212274:instance/i-02948f77cd9cfe49f

Security details

IAM Role
–

Owner ID
610023212274

Launch time
Sun Oct 06 2024 16:41:47 GMT-0400 (Eastern Daylight Time)

Security groups
sg-0a082641d41877677 (launch-wizard-3)

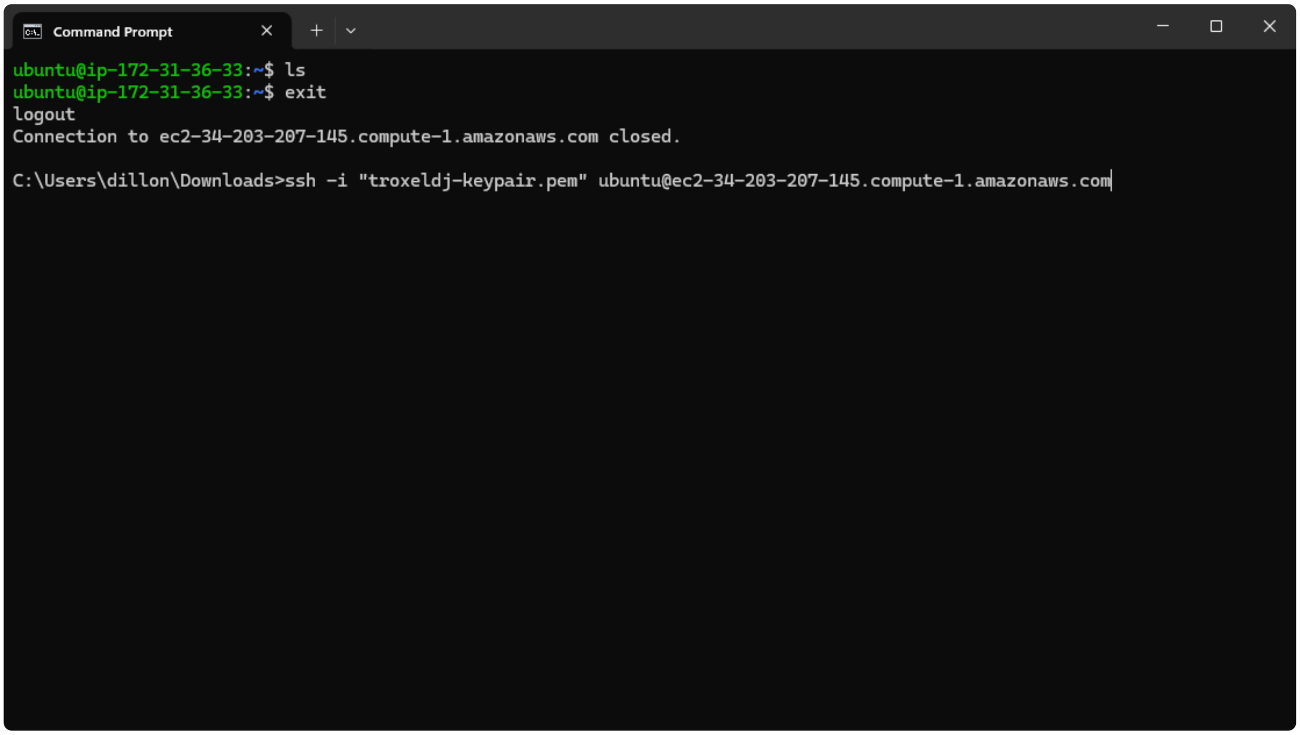
Inbound rules

Name	Security group rule ID	Port range	Protocol	Source
–	sgr-085d9cdb8652ef1ea	80	TCP	0.0.0.0/0
–	sgr-0243ab13f39b887f0	22	TCP	0.0.0.0/0

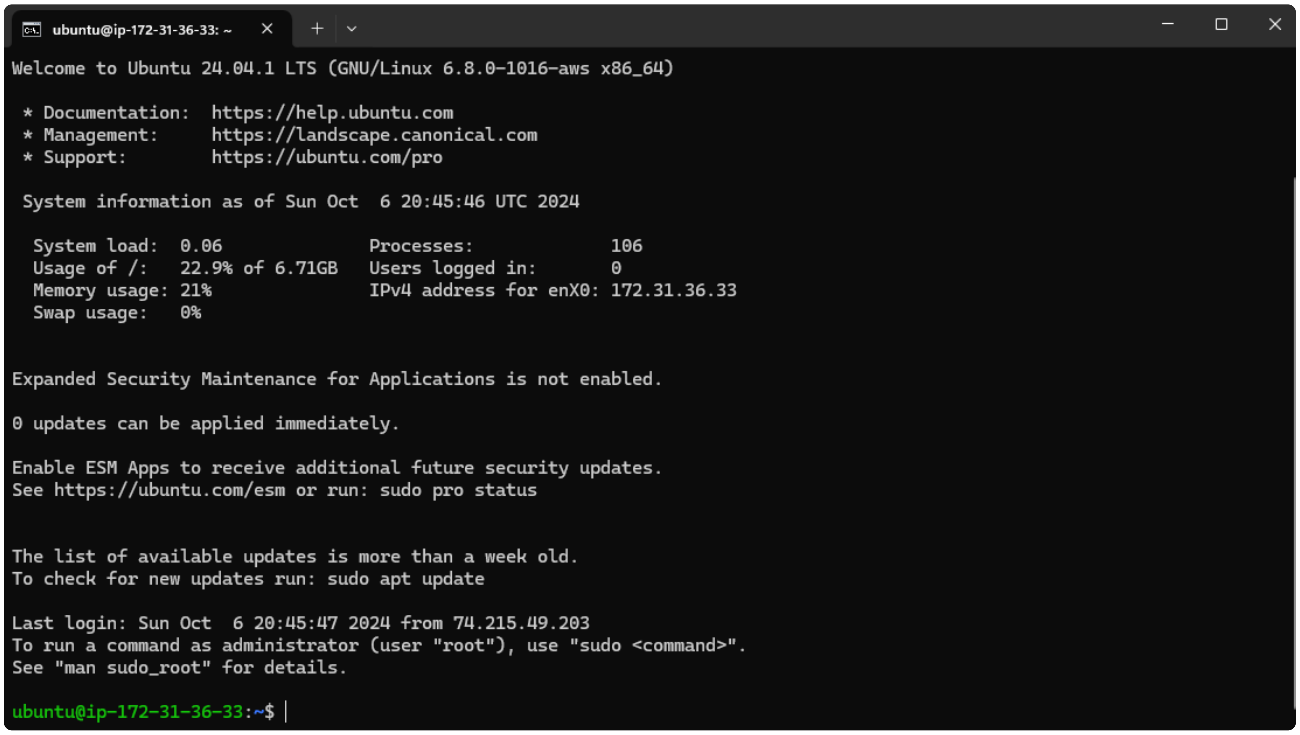
Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination
–	sgr-04153b60bd51b6fee	All	All	0.0.0.0/0

Step 2: Connect to EC2 Instance



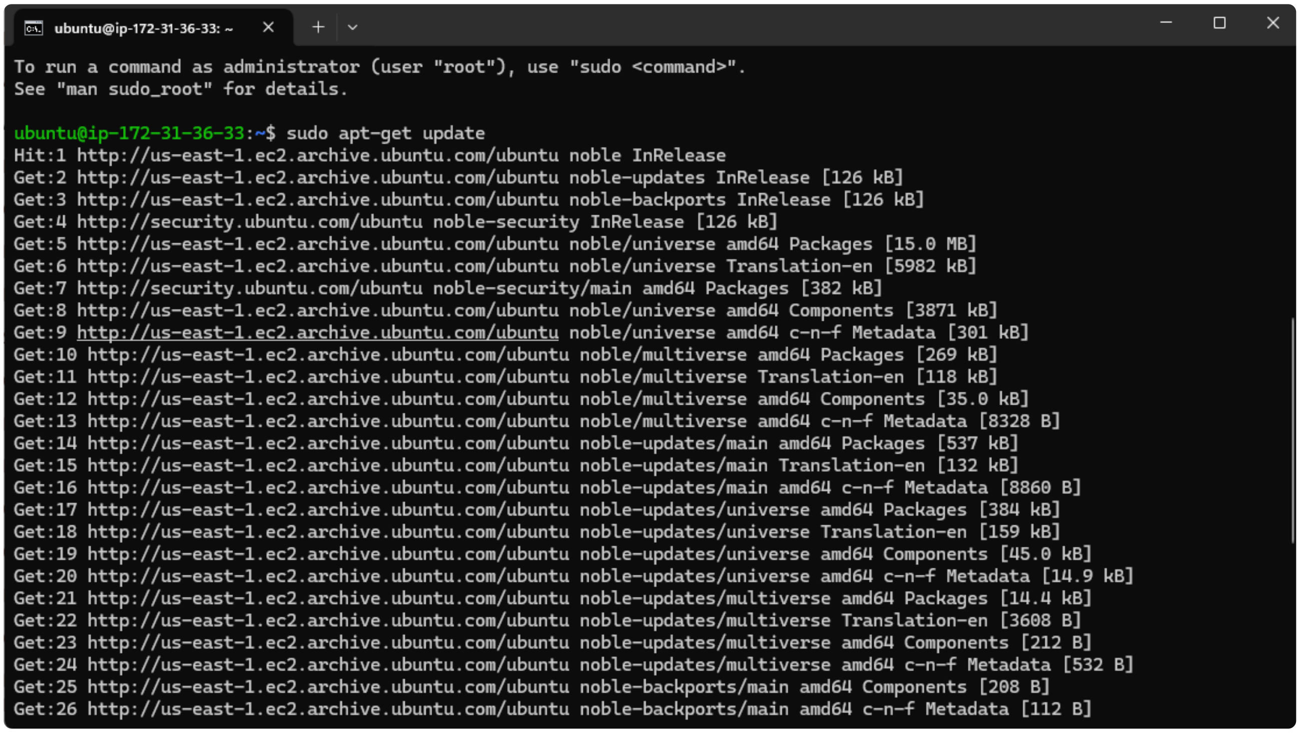
```
ubuntu@ip-172-31-36-33:~$ ls
ubuntu@ip-172-31-36-33:~$ exit
logout
Connection to ec2-34-203-207-145.compute-1.amazonaws.com closed.

C:\Users\dillon\Downloads>ssh -i "troxeldj-keypair.pem" ubuntu@ec2-34-203-207-145.compute-1.amazonaws.com
```



```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

 System information as of Sun Oct  6 20:45:46 UTC 2024

  System load:  0.06              Processes:             106
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 172.31.36.33
  Swap usage:   0%


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

Last login: Sun Oct  6 20:45:47 2024 from 74.215.49.203
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-36-33:~$
```

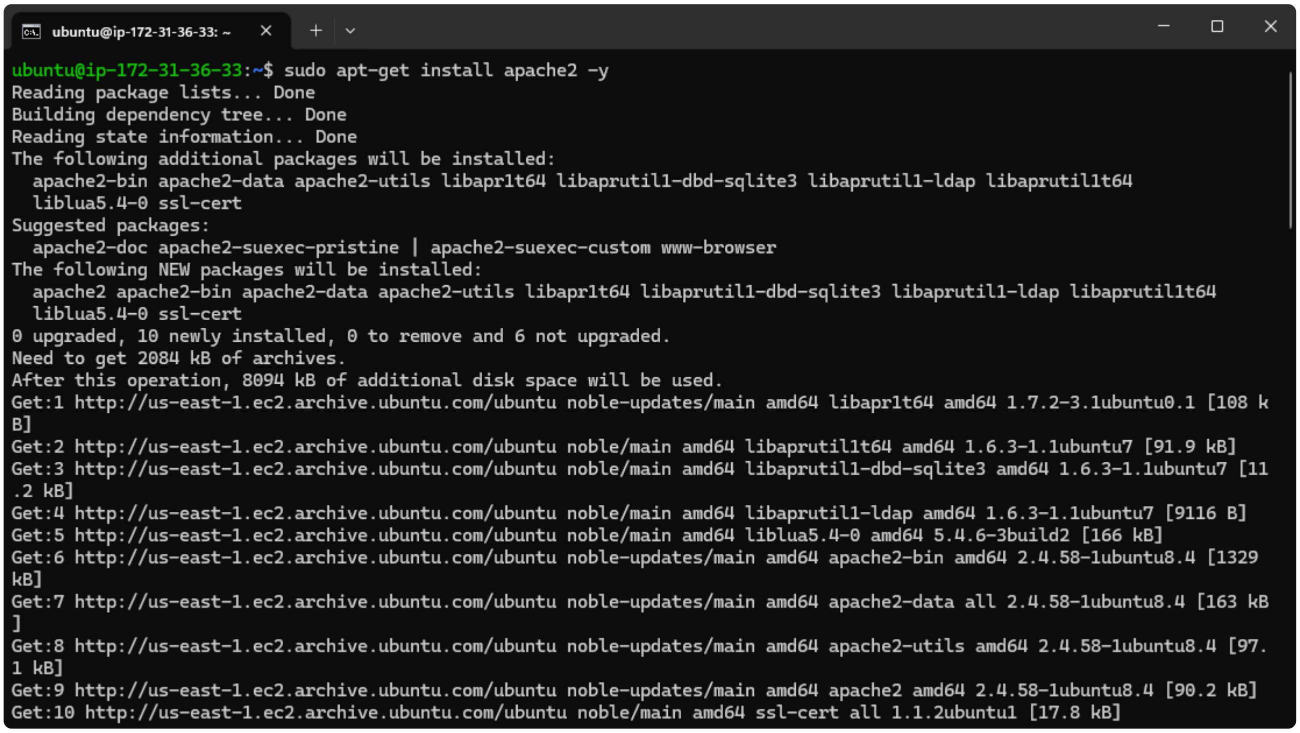
Step 3: Install Apache, mod_wgsi, Python3, and pip

```
sudo apt-get update
```



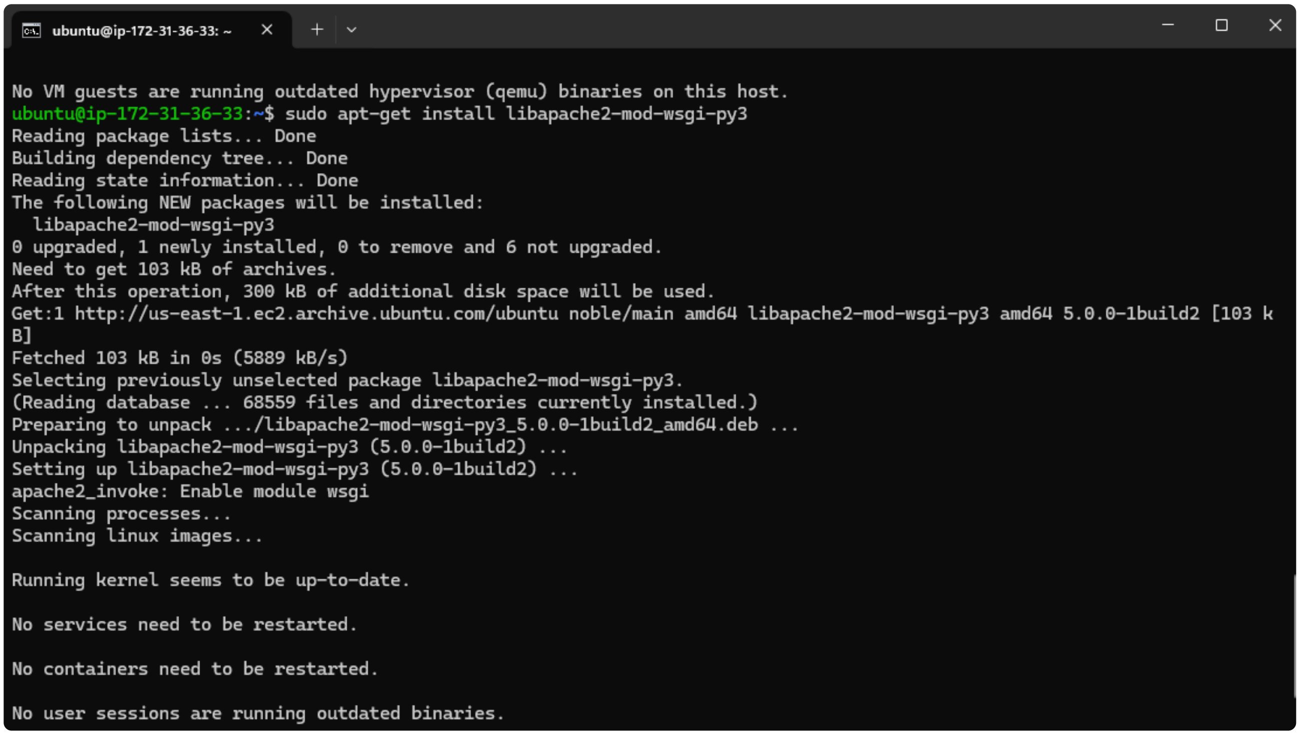
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-36-33:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [159 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
```

```
sudo apt-get install apache2 -y
```



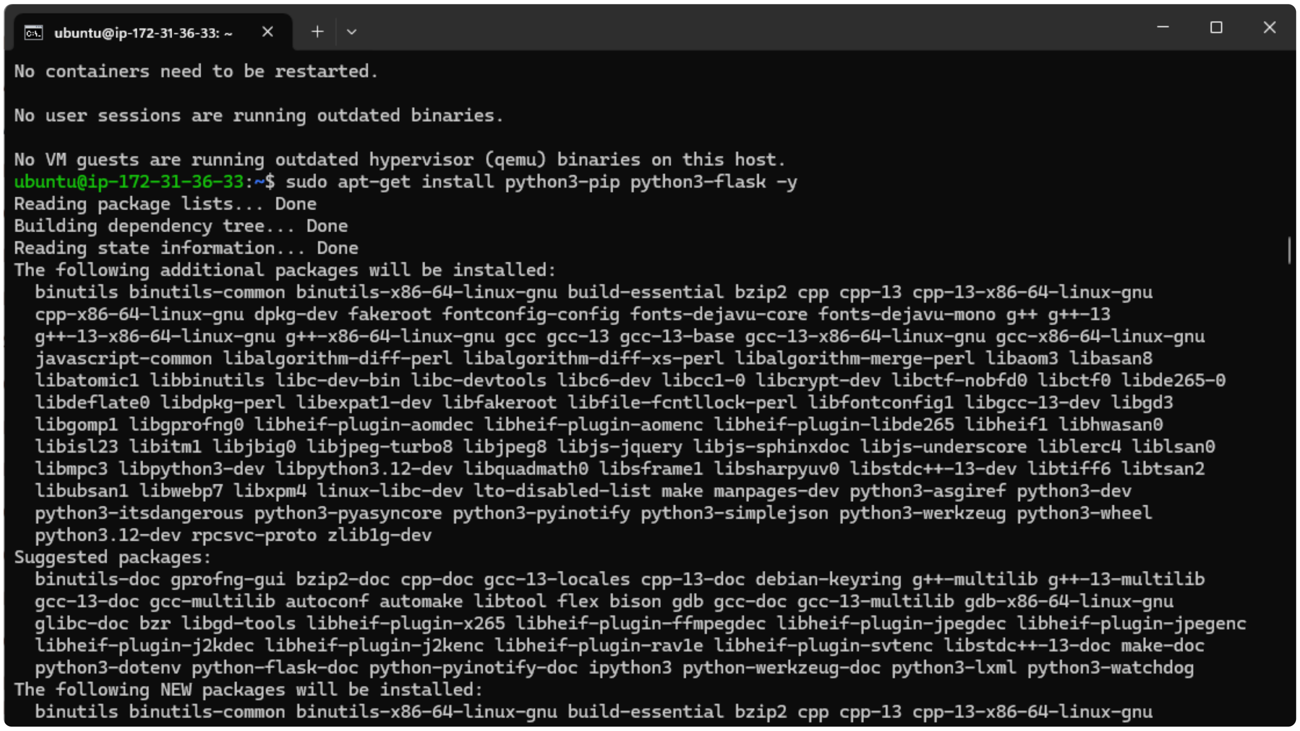
```
ubuntu@ip-172-31-36-33:~$ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap 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 6 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 k
B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11
.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329
kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB
]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.
1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
```

```
sudo apt-get install libapache2-mod-wsgi-py3 -y
```



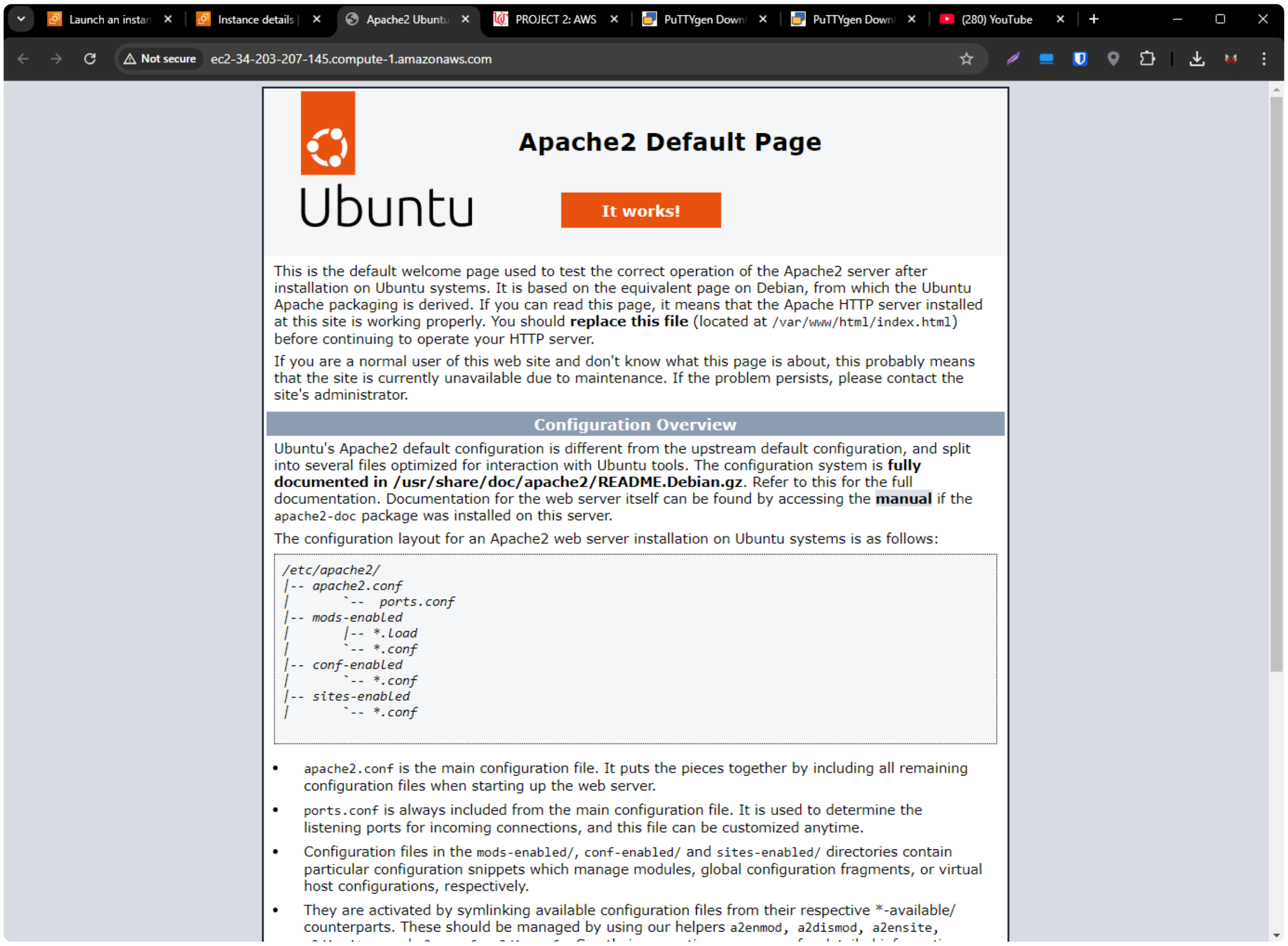
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-36-33:~$ sudo apt-get install libapache2-mod-wsgi-py3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libapache2-mod-wsgi-py3
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 103 kB of archives.
After this operation, 300 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapache2-mod-wsgi-py3 amd64 5.0.0-1build2 [103 k
B]
Fetched 103 kB in 0s (5889 kB/s)
Selecting previously unselected package libapache2-mod-wsgi-py3.
(Reading database ... 68559 files and directories currently installed.)
Preparing to unpack .../libapache2-mod-wsgi-py3_5.0.0-1build2_amd64.deb ...
Unpacking libapache2-mod-wsgi-py3 (5.0.0-1build2) ...
Setting up libapache2-mod-wsgi-py3 (5.0.0-1build2) ...
apache2_invoke: Enable module wsgi
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.
```

```
sudo apt-get install python3-pip python3-flask -y
```



```
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-172-31-36-33:~$ sudo apt-get install python3-pip python3-flask -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu
  cpp-x86-64-linux-gnu dpkg-dev fakeroot fontconfig-config fonts-dejavu-core fonts-dejavu-mono g++ g++-13
  g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu
  javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3 libasan8
  libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0
  libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-13-dev libgd3
  libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0
  libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-underscore liblerc4 liblsan0
  libmpc3 libpython3-dev libpython3.12-dev libquadmath0 libsframe1 libsharpyuv0 libstdc++-13-dev libtiff6 libtsan2
  libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list make manpages-dev python3-asgiref python3-dev
  python3-itsdangerous python3-pyasyncore python3-pyinotify python3-simplejson python3-werkzeug python3-wheel
  python3.12-dev rpcsvc-proto zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debian-keyring g++-multilib g++-13-multilib
  gcc-13-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gdb-x86-64-linux-gnu
  glibc-doc bzr libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc
  libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e libheif-plugin-svtenc libstdc++-13-doc make-doc
  python3-dotenv python-flask-doc python-pyinotify-doc ipython3 python-werkzeug-doc python3-lxml python3-watchdog
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu
```

Test Apache



Apache2 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.

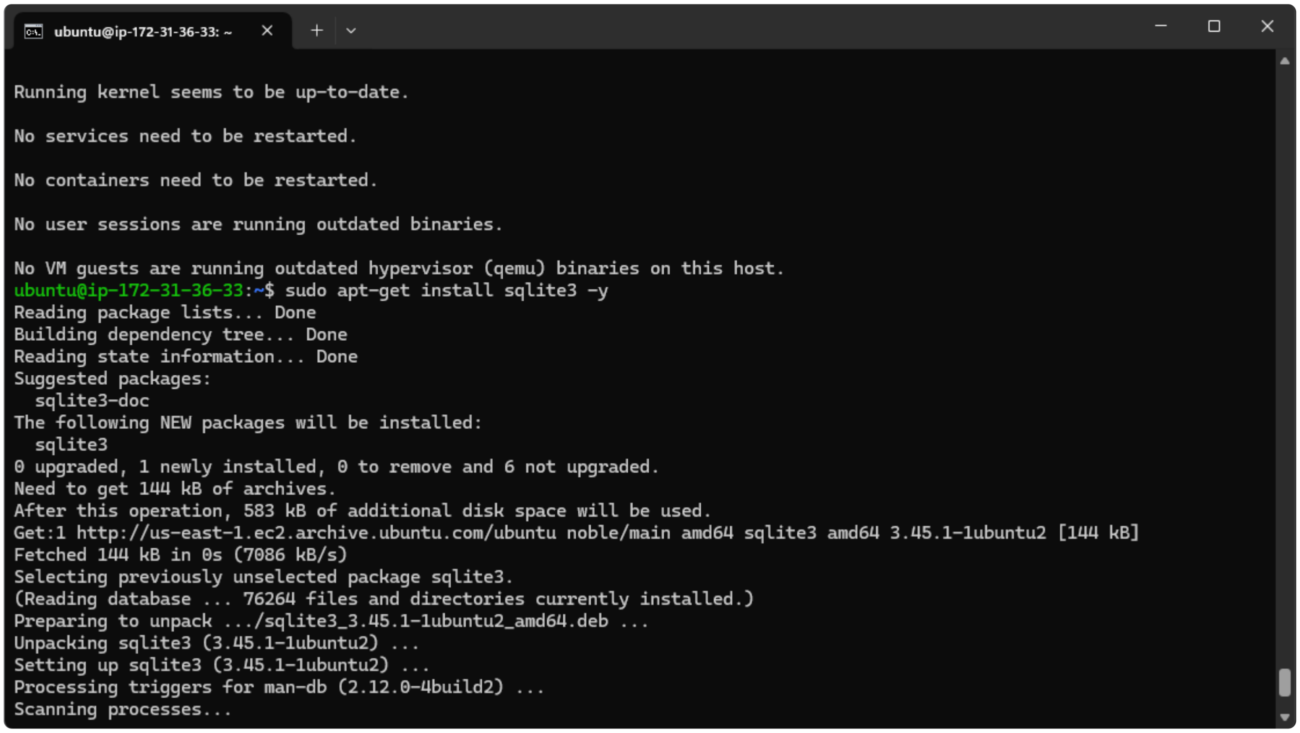
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/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.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`,

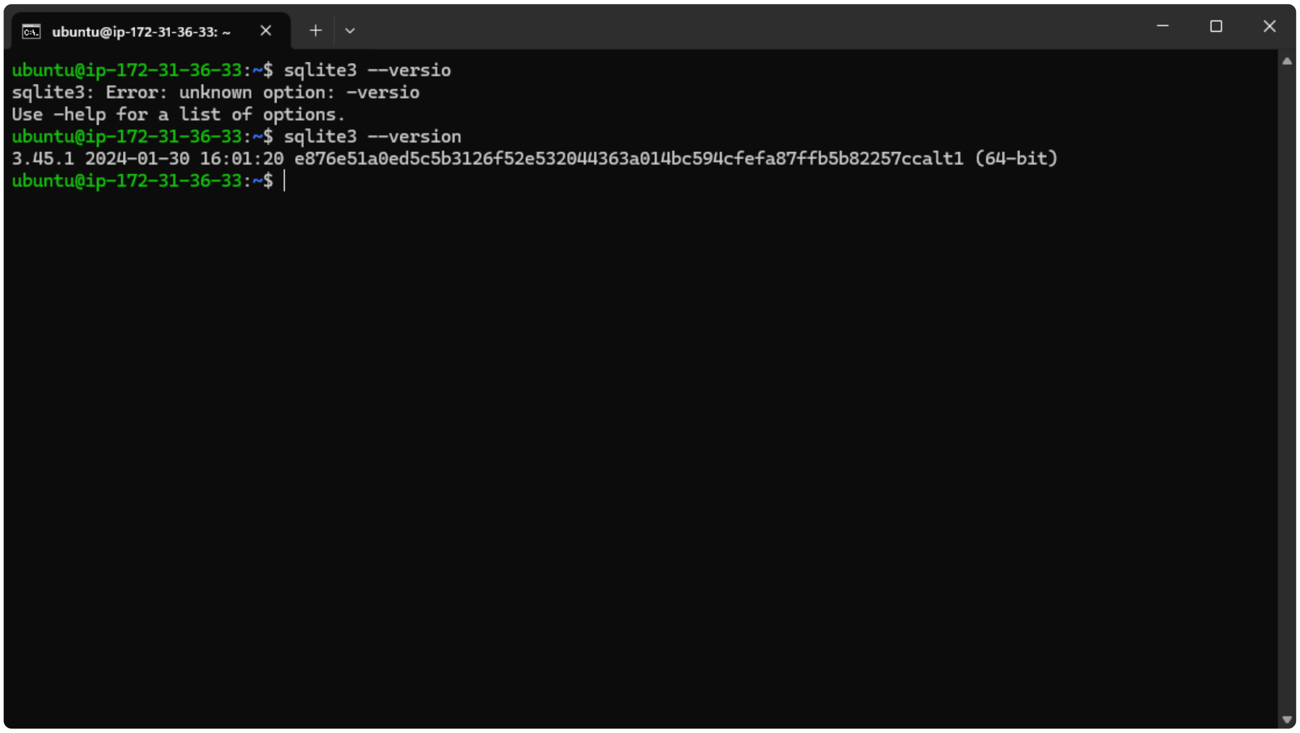
Step 4: Install SQLite3

```
sudo apt-get install sqlite3 -y
```



```
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-172-31-36-33:~$ sudo apt-get install sqlite3 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  sqlite3
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 144 kB of archives.
After this operation, 583 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 sqlite3 amd64 3.45.1-1ubuntu2 [144 kB]
Fetched 144 kB in 0s (7086 kB/s)
Selecting previously unselected package sqlite3.
(Reading database ... 76264 files and directories currently installed.)
Preparing to unpack .../sqlite3_3.45.1-1ubuntu2_amd64.deb ...
Unpacking sqlite3 (3.45.1-1ubuntu2) ...
Setting up sqlite3 (3.45.1-1ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
```

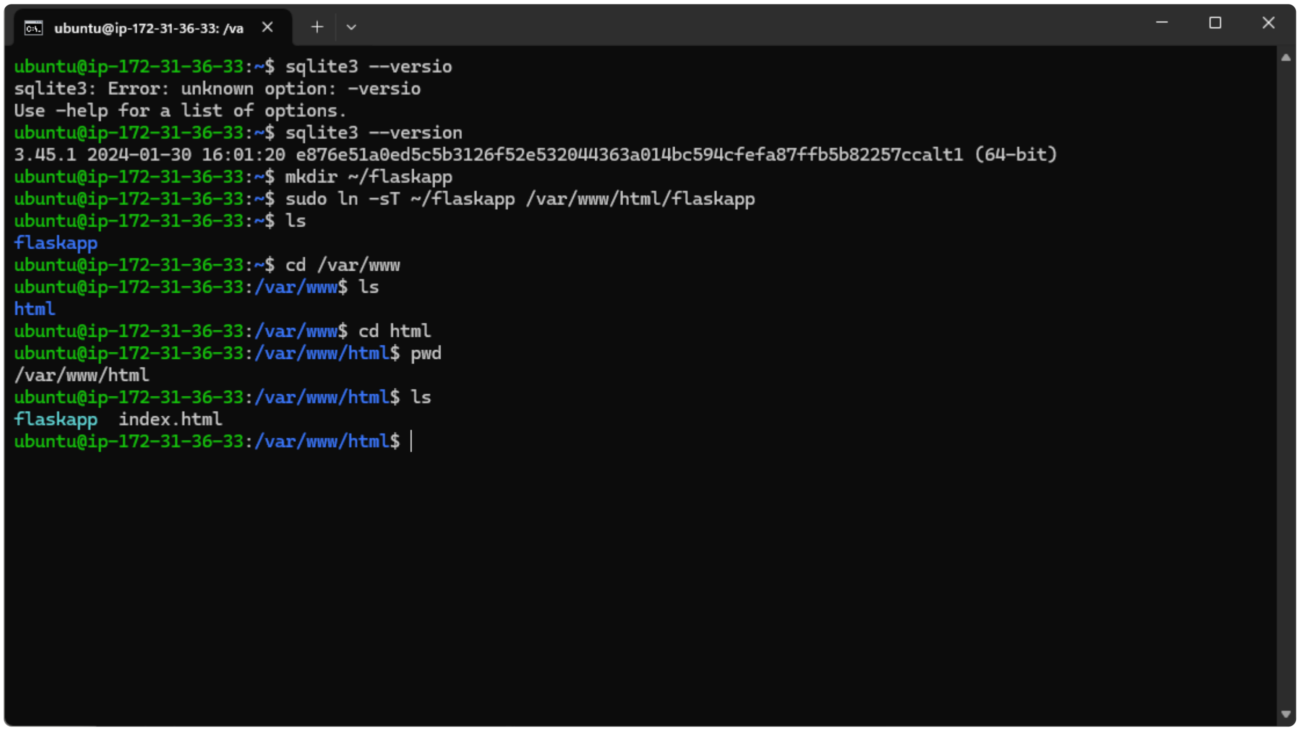
```
sqlite3 --version
```



```
ubuntu@ip-172-31-36-33:~$ sqlite3 --versio
sqlite3: Error: unknown option: -versio
Use -help for a list of options.
ubuntu@ip-172-31-36-33:~$ sqlite3 --version
3.45.1 2024-01-30 16:01:20 e876e51a0ed5c5b3126f52e532044363a014bc594cfefa87ffb5b82257ccalt1 (64-bit)
ubuntu@ip-172-31-36-33:~$
```

Step 5: Create Directory for Flask App

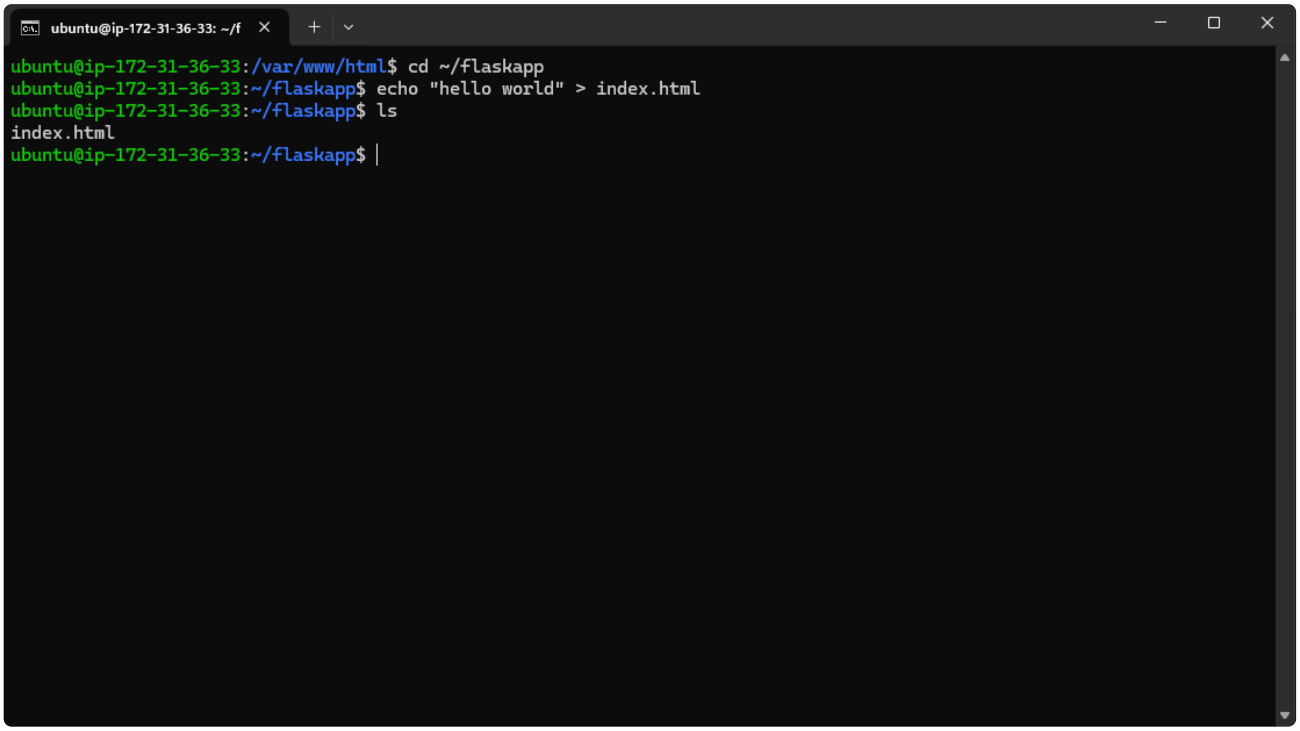
```
mkdir ~/flaskapp
sudo ln -sT ~/flaskapp /var/www/html/flaskapp
```



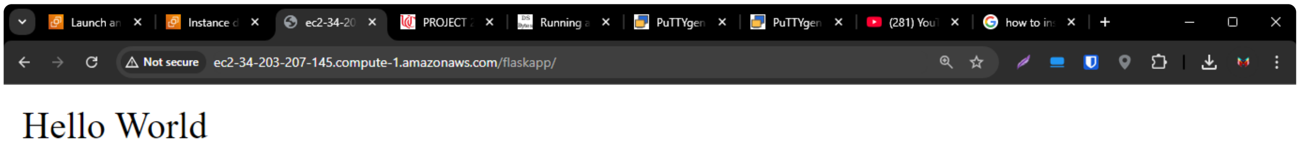
```
ubuntu@ip-172-31-36-33:~$ sqlite3 --versio
sqlite3: Error: unknown option: -versio
Use -help for a list of options.
ubuntu@ip-172-31-36-33:~$ sqlite3 --version
3.45.1 2024-01-30 16:01:20 e876e51a0ed5c5b3126f52e532044363a014bc594cfefa87ffb5b82257ccalt1 (64-bit)
ubuntu@ip-172-31-36-33:~$ mkdir ~/flaskapp
ubuntu@ip-172-31-36-33:~$ sudo ln -sT ~/flaskapp /var/www/html/flaskapp
ubuntu@ip-172-31-36-33:~$ ls
flaskapp
ubuntu@ip-172-31-36-33:~$ cd /var/www
ubuntu@ip-172-31-36-33:/var/www$ ls
html
ubuntu@ip-172-31-36-33:/var/www$ cd html
ubuntu@ip-172-31-36-33:/var/www/html$ pwd
/var/www/html
ubuntu@ip-172-31-36-33:/var/www/html$ ls
flaskapp  index.html
ubuntu@ip-172-31-36-33:/var/www/html$ 
```

Test Flask with simple hello world

```
cd ~/flaskapp
echo "hello world" > index.html
```



```
ubuntu@ip-172-31-36-33:/var/www/html$ cd ~/flaskapp
ubuntu@ip-172-31-36-33:~/flaskapp$ echo "hello world" > index.html
ubuntu@ip-172-31-36-33:~/flaskapp$ ls
index.html
ubuntu@ip-172-31-36-33:~/flaskapp$ |
```



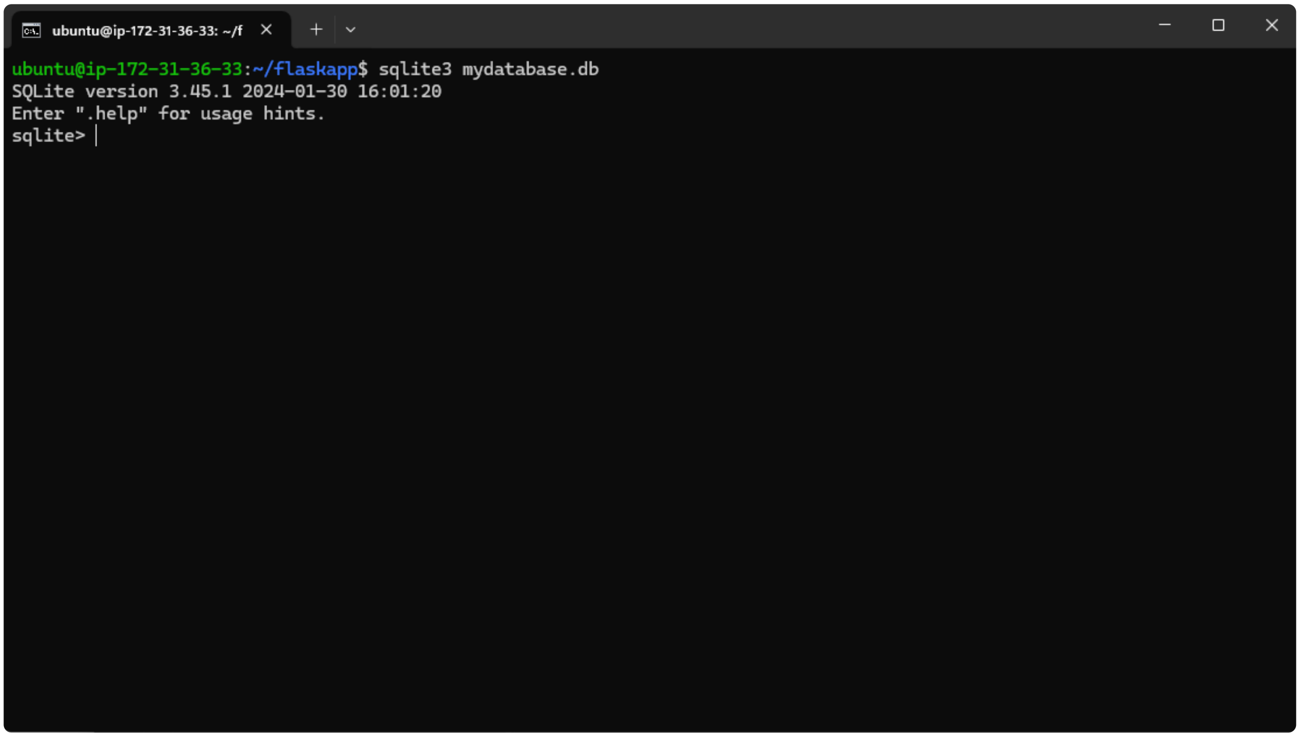
Hello World

Step 6: Create SQLite3 Database

Create Database

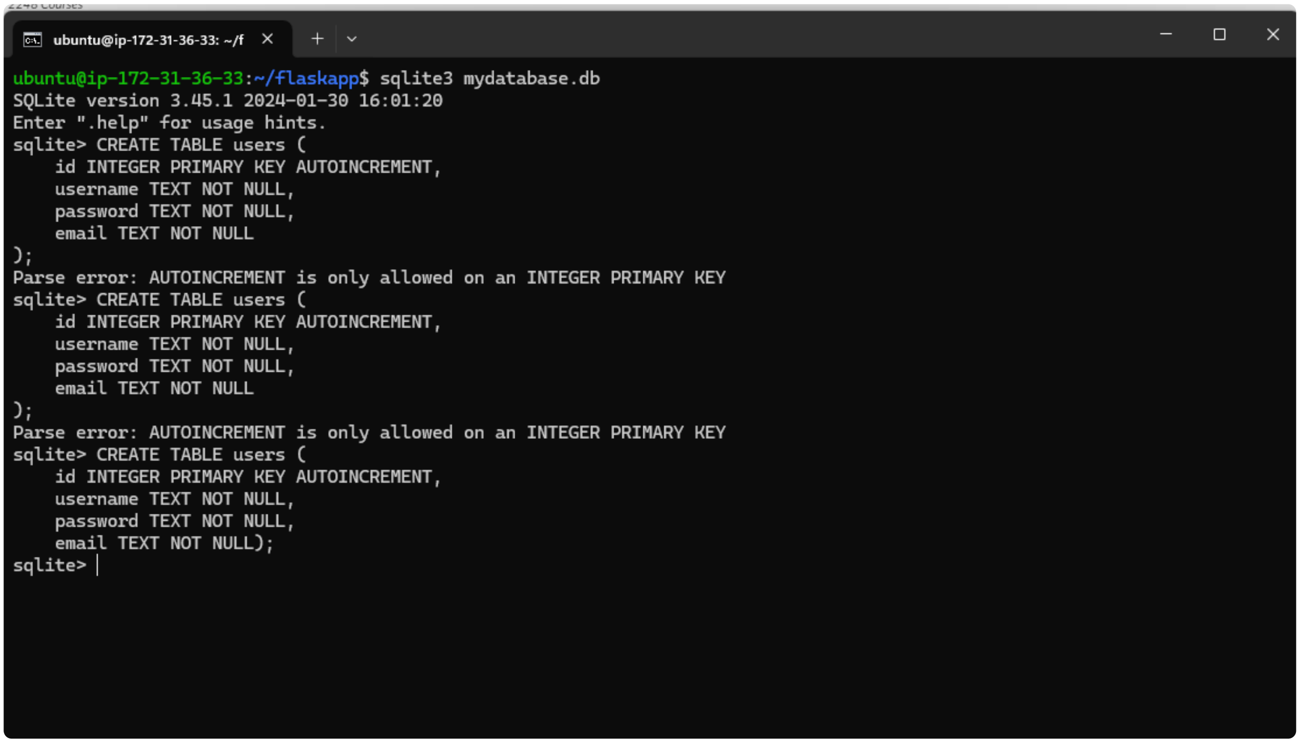
```
sqlite3 mydatabase.db
```

Add Table



```
ubuntu@ip-172-31-36-33:~/flaskapp$ sqlite3 mydatabase.db
SQLite version 3.45.1 2024-01-30 16:01:20
Enter ".help" for usage hints.
sqlite>
```

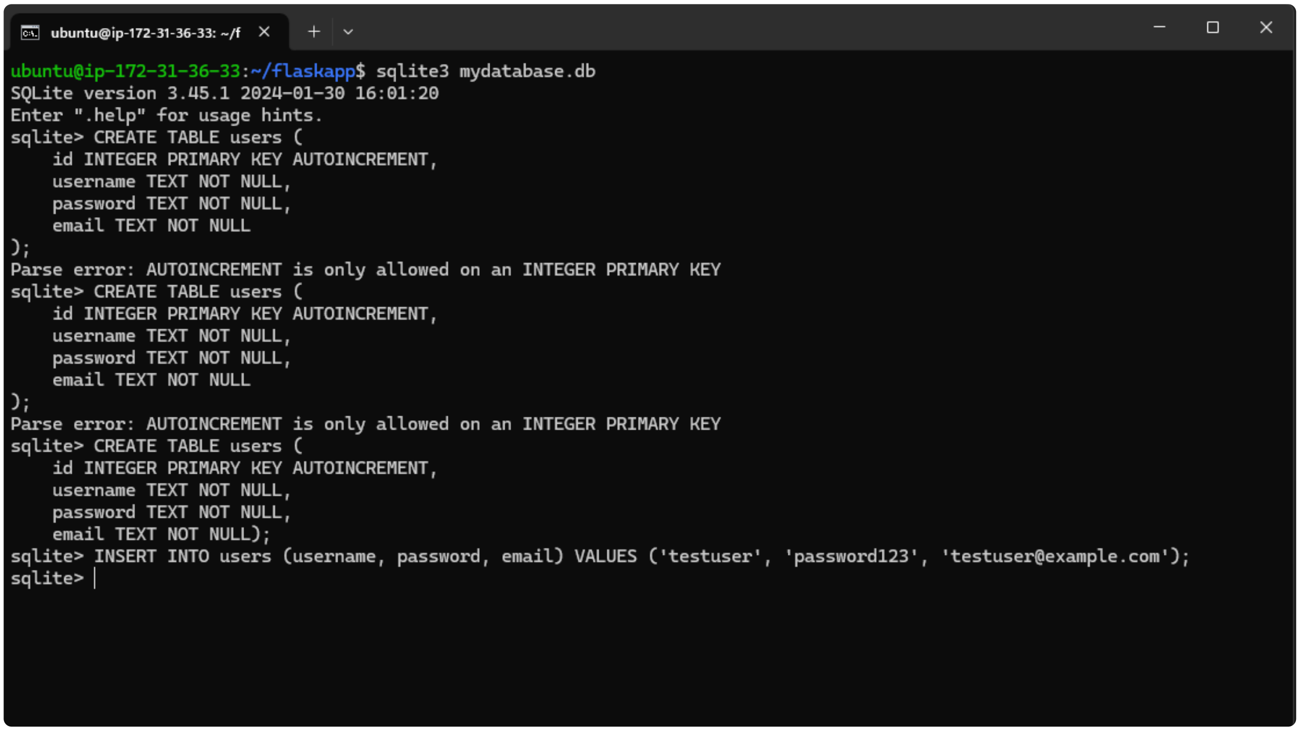
```
CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL);
```



```
ubuntu@ip-172-31-36-33:~/flaskapp$ sqlite3 mydatabase.db
SQLite version 3.45.1 2024-01-30 16:01:20
Enter ".help" for usage hints.
sqlite> CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL
);
Parse error: AUTOINCREMENT is only allowed on an INTEGER PRIMARY KEY
sqlite> CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL
);
Parse error: AUTOINCREMENT is only allowed on an INTEGER PRIMARY KEY
sqlite> CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL);
sqlite> |
```

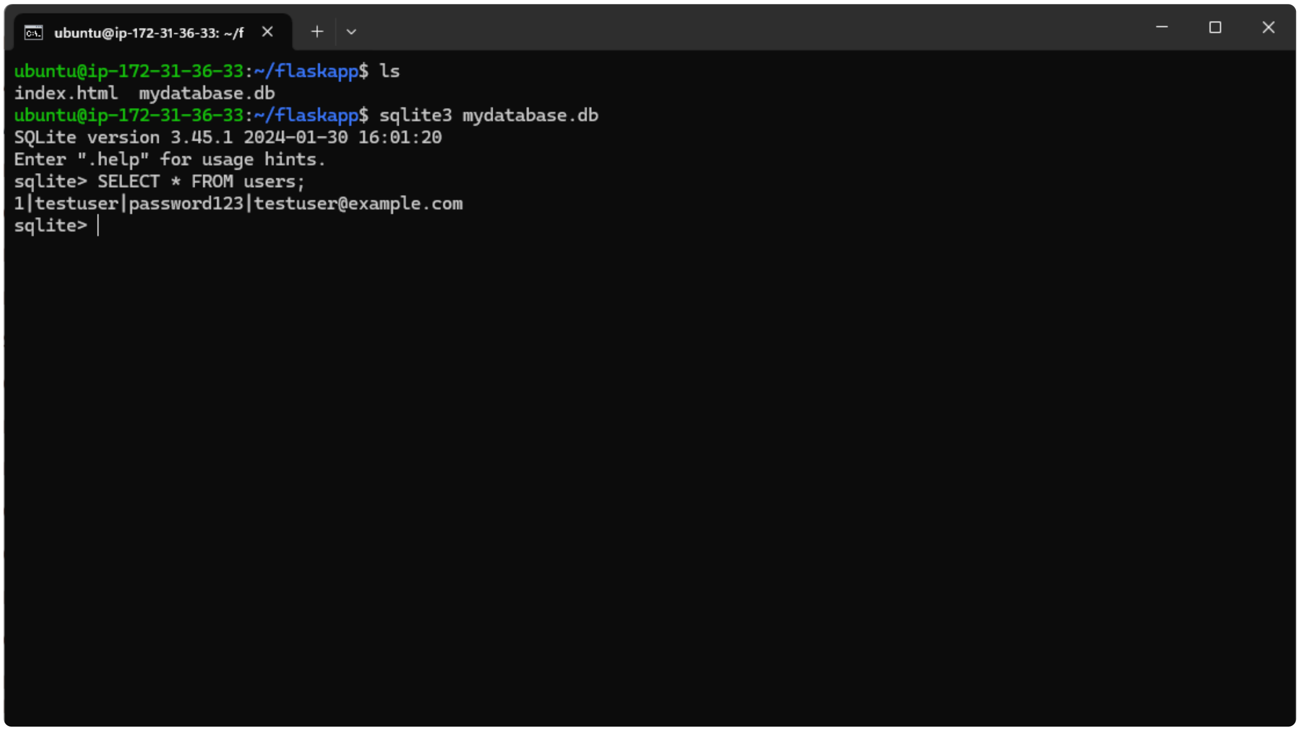
Add Test Entry

```sql
INSERT INTO users (username, password, email) VALUES ('testuser', 'password123', 'testuser@example.com');
```



```
ubuntu@ip-172-31-36-33:~/flaskapp$ sqlite3 mydatabase.db
SQLite version 3.45.1 2024-01-30 16:01:20
Enter ".help" for usage hints.
sqlite> CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL
);
Parse error: AUTOINCREMENT is only allowed on an INTEGER PRIMARY KEY
sqlite> CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL
);
Parse error: AUTOINCREMENT is only allowed on an INTEGER PRIMARY KEY
sqlite> CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    password TEXT NOT NULL,
    email TEXT NOT NULL);
sqlite> INSERT INTO users (username, password, email) VALUES ('testuser', 'password123', 'testuser@example.com');
sqlite> |
```

View Table Contents



```
ubuntu@ip-172-31-36-33:~/flaskapp$ ls
index.html  mydatabase.db
ubuntu@ip-172-31-36-33:~/flaskapp$ sqlite3 mydatabase.db
SQLite version 3.45.1 2024-01-30 16:01:20
Enter ".help" for usage hints.
sqlite> SELECT * FROM users;
1|testuser|password123|testuser@example.com
sqlite> 
```

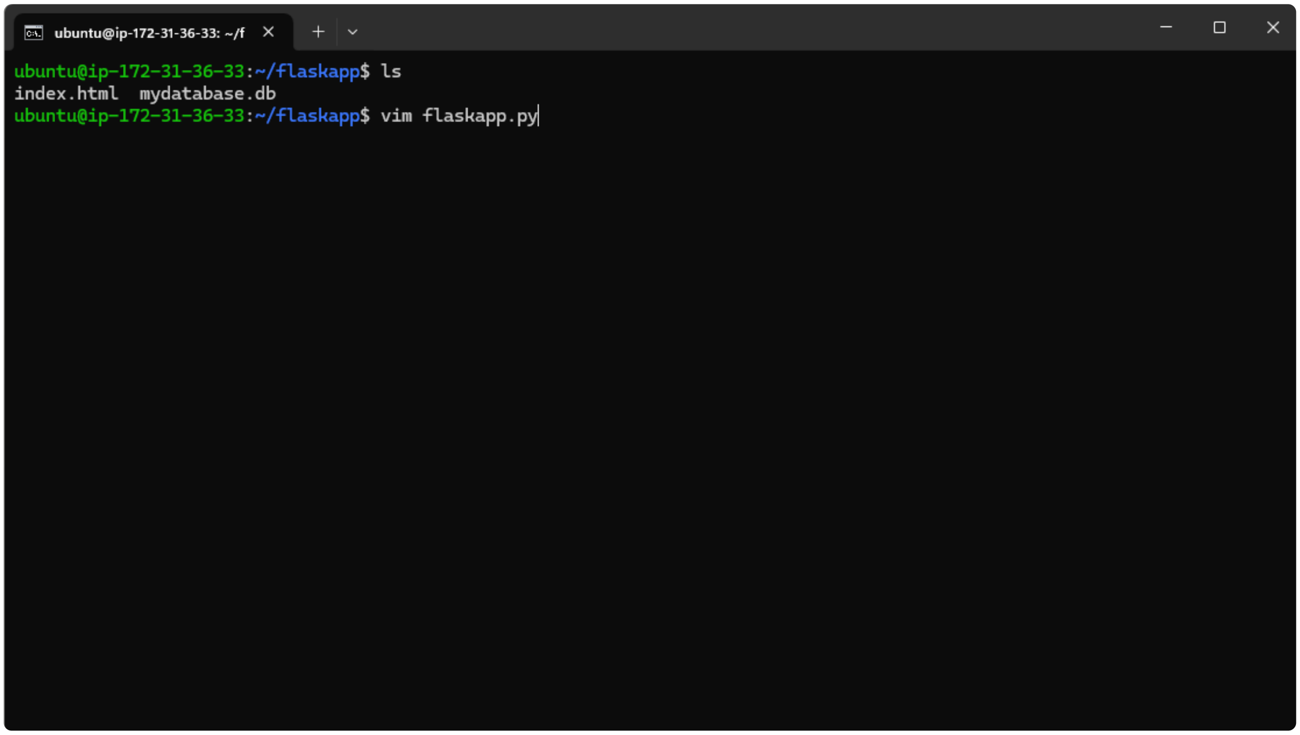
Step 7: Creating More Advanced "Test" Flask Application (using wsgi)

Add to ~/flaskapp/flaskapp.py

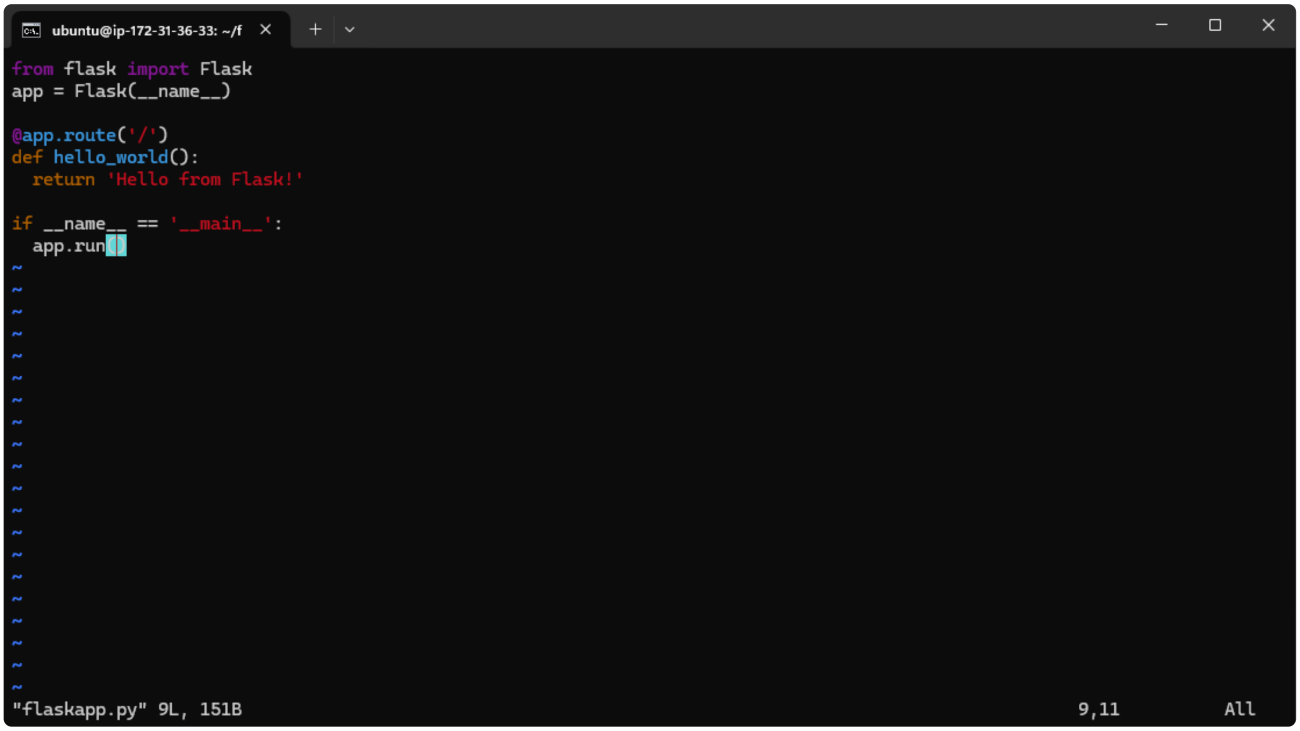
flaskapp.py

```python
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
  return 'Hello from Flask!'

if __name__ == '__main__':
  app.run()
```

```
cd ~/flaskapp
vim flaskapp.py
```



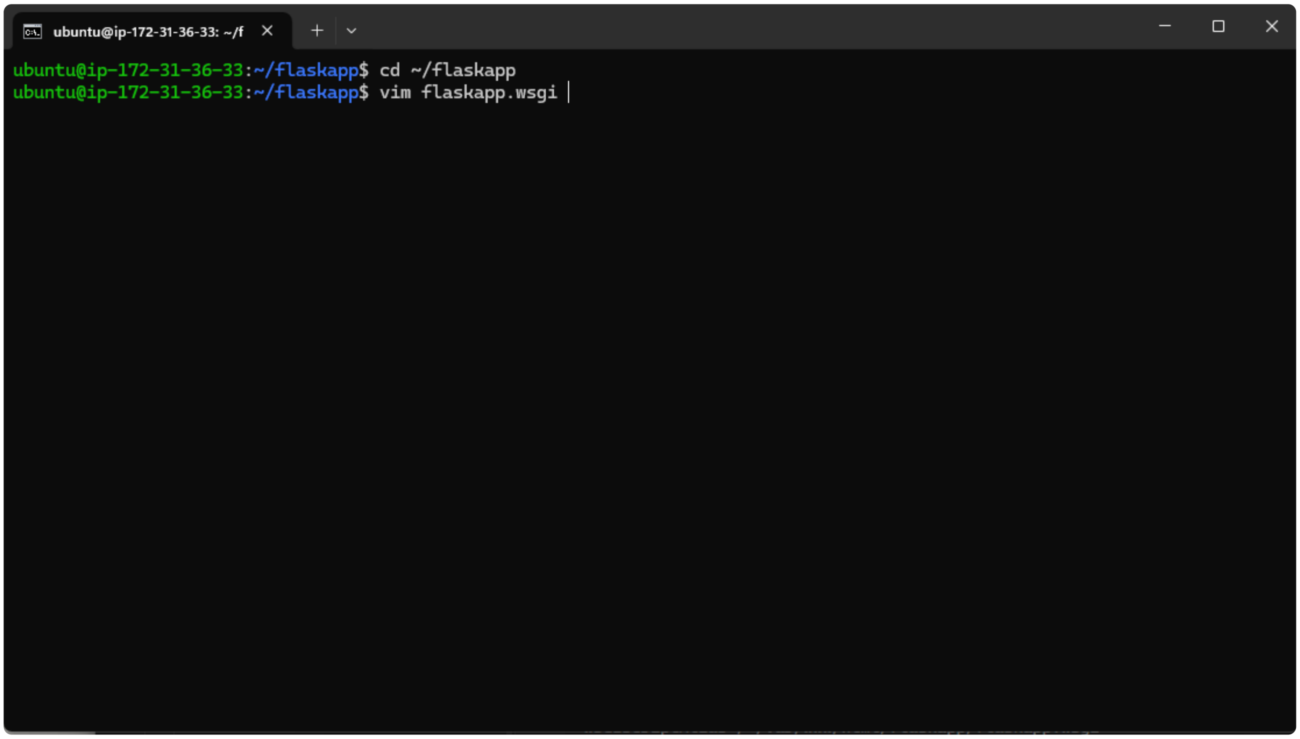
```
ubuntu@ip-172-31-36-33:~/flaskapp$ ls
index.html  mydatabase.db
ubuntu@ip-172-31-36-33:~/flaskapp$ vim flaskapp.py
```



```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
  return 'Hello from Flask!'

if __name__ == '__main__':
  app.run()
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"flaskapp.py" 9L, 151B                                                     9,11          All
```

Create flaskapp.wsgi

```
cd ~/flaskapp
vim flaskapp.wsgi
```

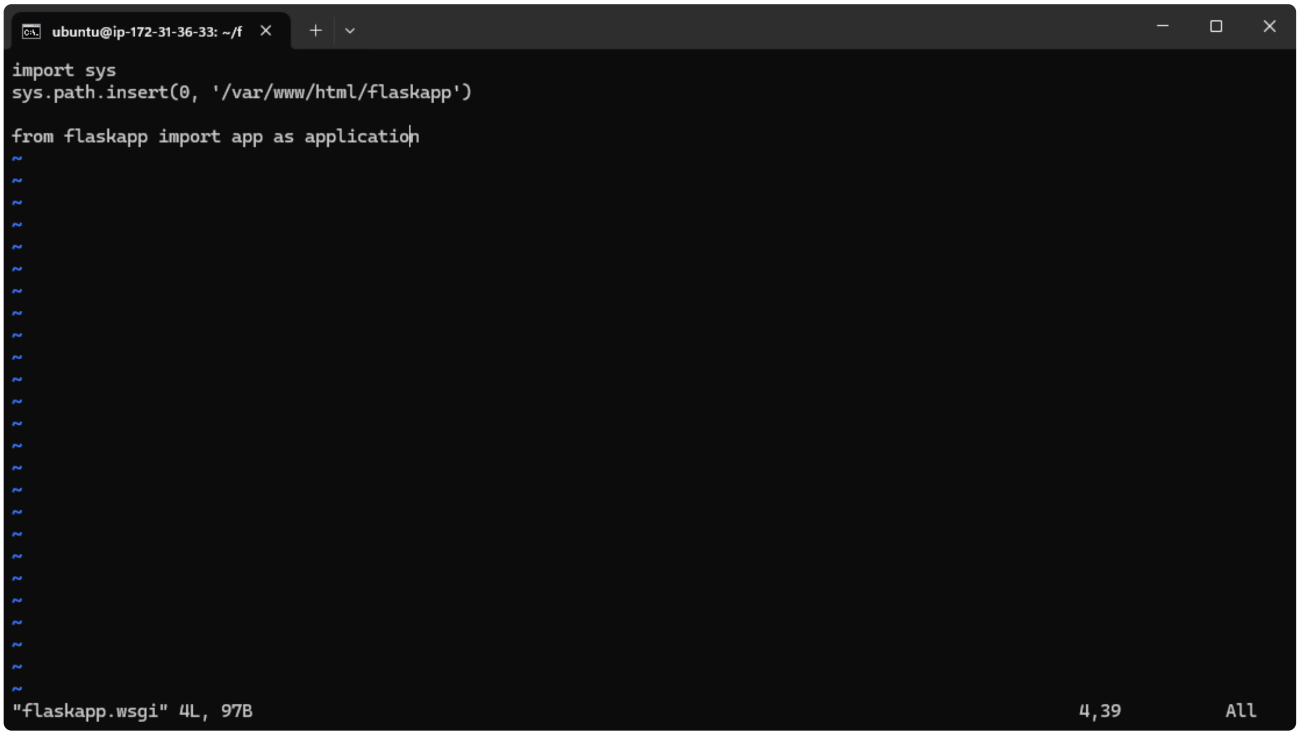


```
ubuntu@ip-172-31-36-33:~/flaskapp$ cd ~/flaskapp
ubuntu@ip-172-31-36-33:~/flaskapp$ vim flaskapp.wsgi
```

Copy to flaskapp.wsgi

flaskapp.wsgi

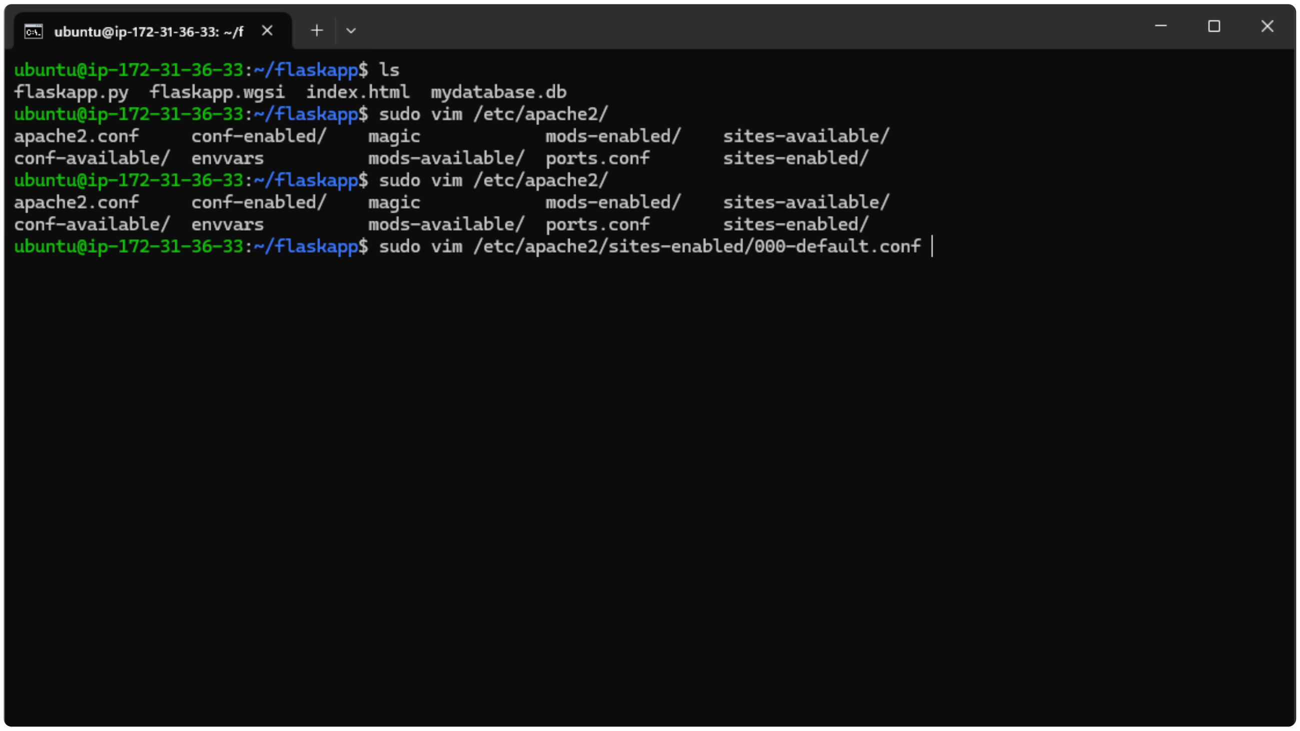
```python
import sys
sys.path.insert(0, '/var/www/html/flaskapp')

from flaskapp import app as application
```



```
import sys
sys.path.insert(0, '/var/www/html/flaskapp')

from flaskapp import app as application
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"flaskapp.wsgi" 4L, 97B                                                              4,39          All
```

Enable wsgi

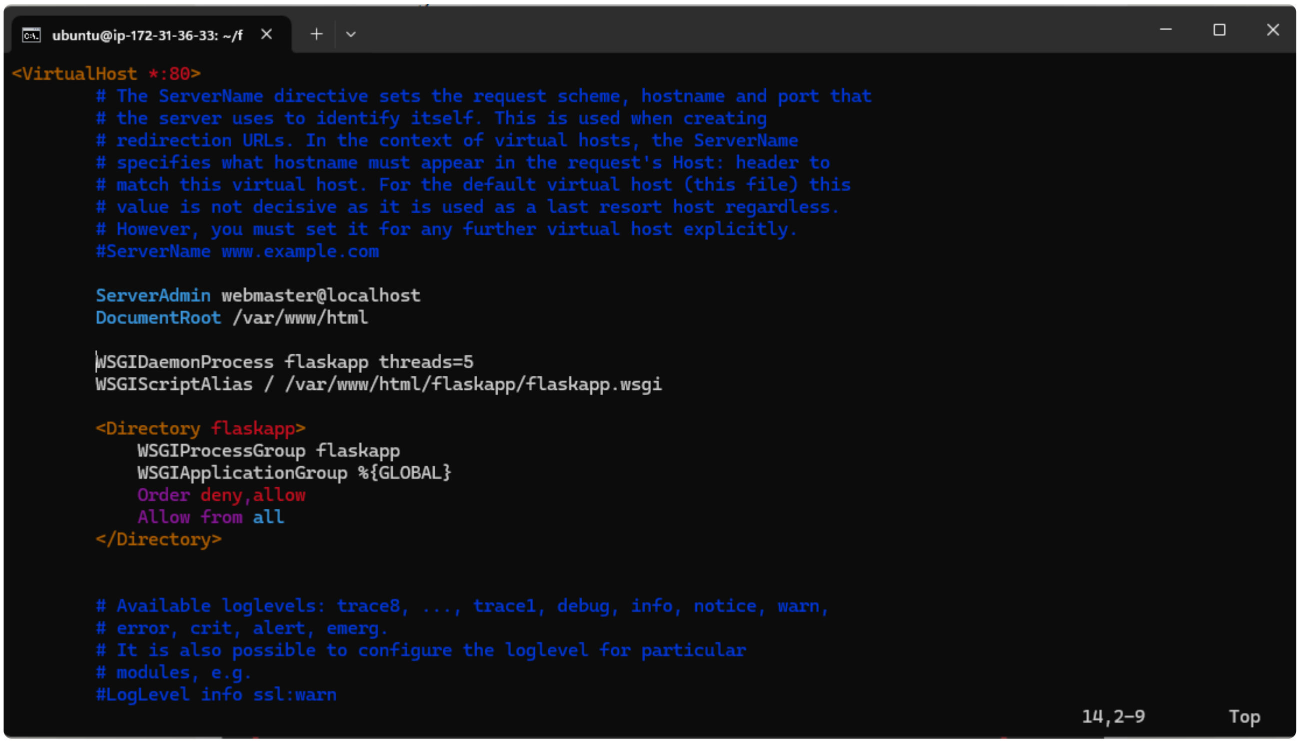
```
sudo vim /etc/apache2/sites-enabled/000-default.conf
```



```
ubuntu@ip-172-31-36-33:~/flaskapp$ ls
flaskapp.py  flaskapp.wgsi  index.html  mydatabase.db
ubuntu@ip-172-31-36-33:~/flaskapp$ sudo vim /etc/apache2/
apache2.conf     conf-enabled/    magic            mods-enabled/    sites-available/
conf-available/  envvars          mods-available/  ports.conf       sites-enabled/
ubuntu@ip-172-31-36-33:~/flaskapp$ sudo vim /etc/apache2/
apache2.conf     conf-enabled/    magic            mods-enabled/    sites-available/
conf-available/  envvars          mods-available/  ports.conf       sites-enabled/
ubuntu@ip-172-31-36-33:~/flaskapp$ sudo vim /etc/apache2/sites-enabled/000-default.conf
```

Copy to /etc/apache2/sites-enabled/000-default.conf

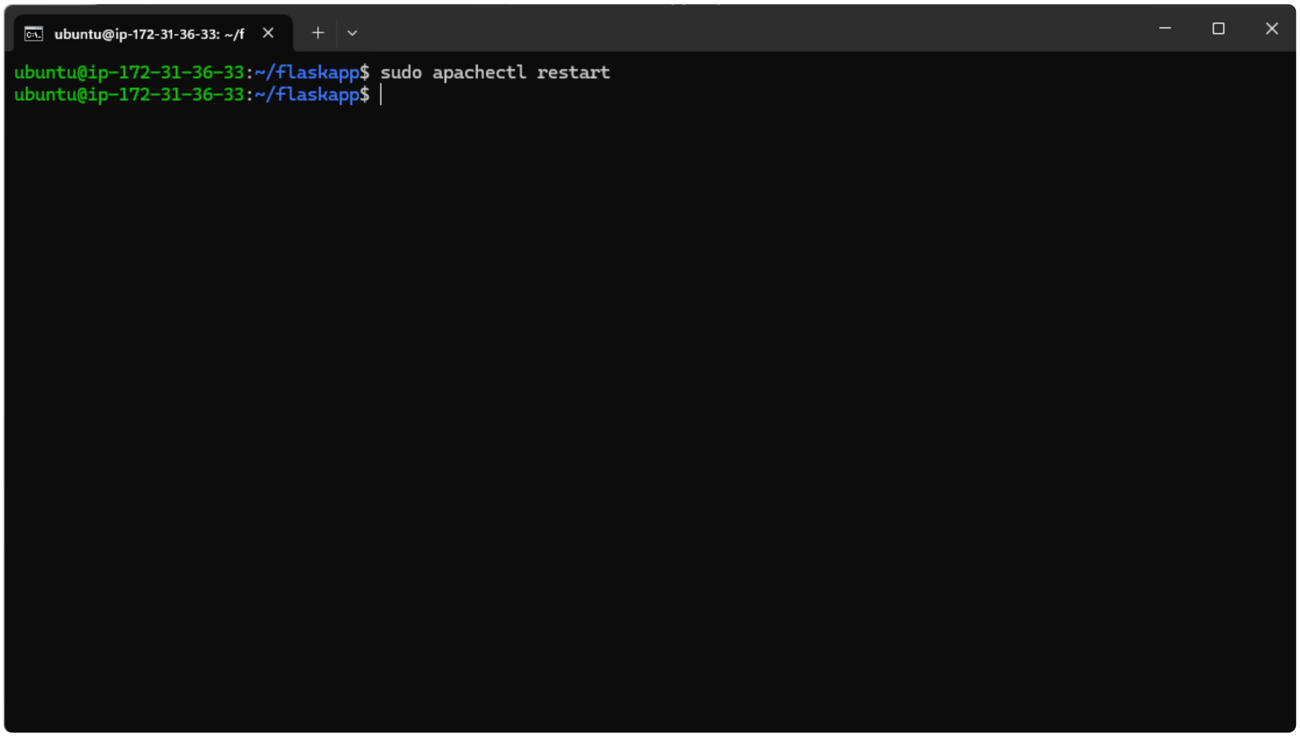
```
WSGIDaemonProcess flaskapp threads=5
WSGIScriptAlias / /var/www/html/flaskapp/flaskapp.wsgi

<Directory flaskapp>
        Header set Access-Control-Allow-Origin "*"
    WSGIProcessGroup flaskapp
    WSGIApplicationGroup %{GLOBAL}
    Order deny,allow
    Allow from all
</Directory>
```



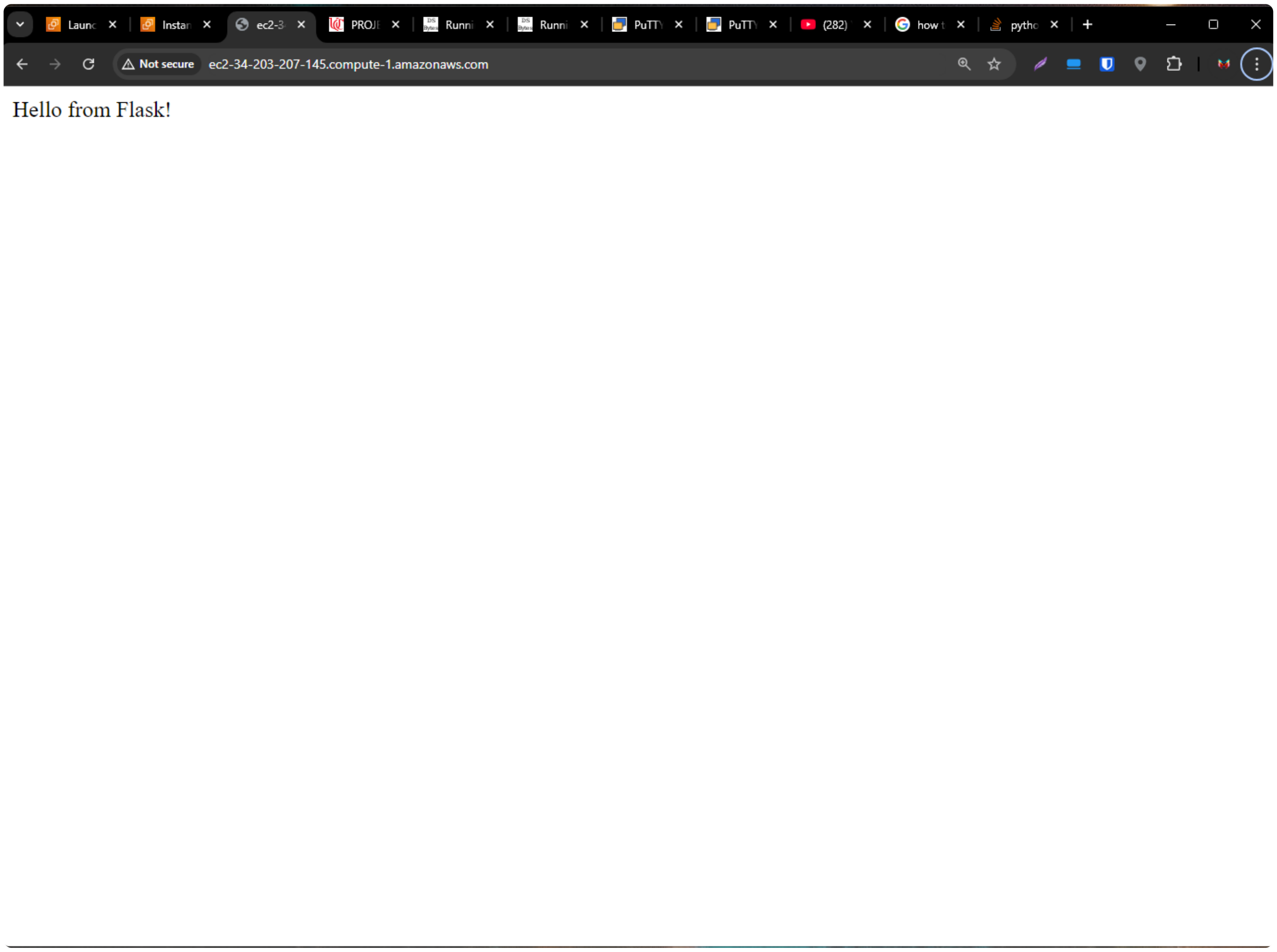
```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear 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

        WSGIDaemonProcess flaskapp threads=5
        WSGIScriptAlias / /var/www/html/flaskapp/flaskapp.wsgi

        <Directory flaskapp>
            WSGIProcessGroup flaskapp
            WSGIApplicationGroup %{GLOBAL}
            Order deny,allow
            Allow from all
        </Directory>


        # 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
                                                                                14,2-9        Top
```

Restart Apache

```
sudo apachectl restart
```



```
ubuntu@ip-172-31-36-33:~/flaskapp$ sudo apachectl restart
ubuntu@ip-172-31-36-33:~/flaskapp$ |
```



Hello from Flask!

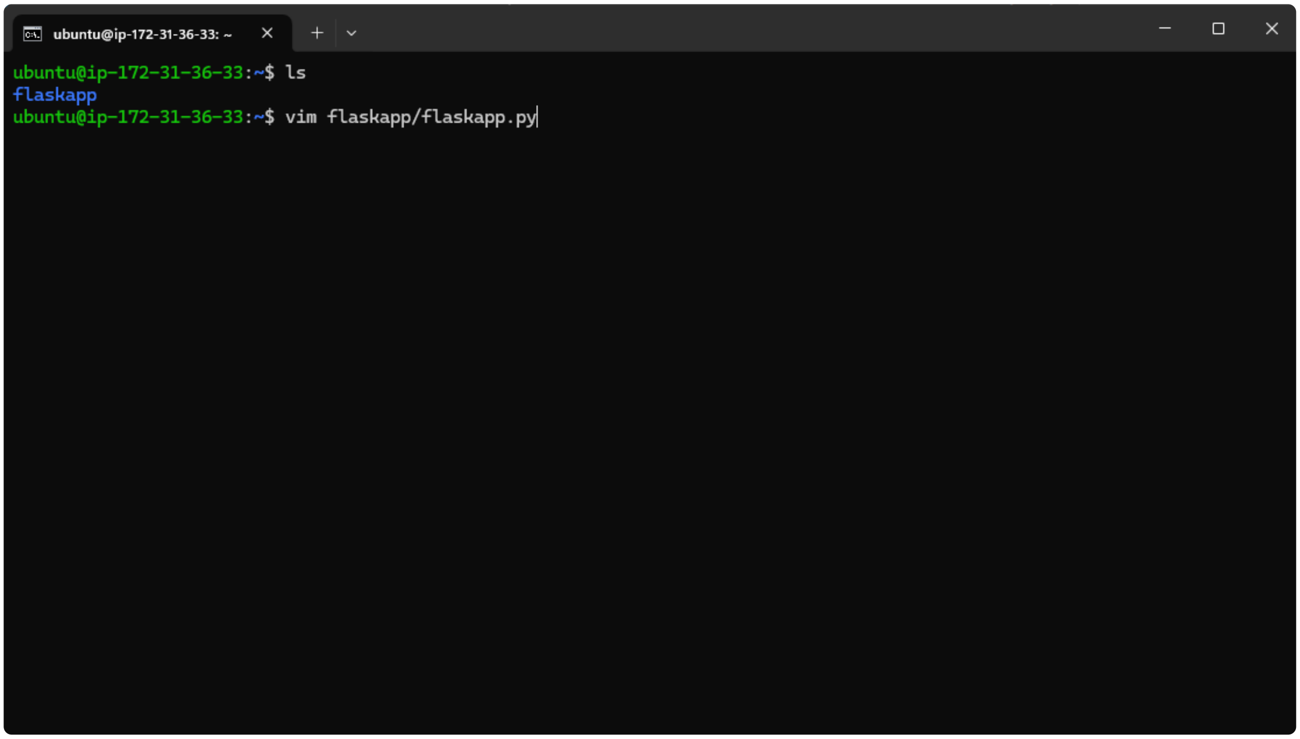
Step 9: Adjusting Flask Application / Create Users Flask App

Copy to ~/flaskapp/flaskapp.py

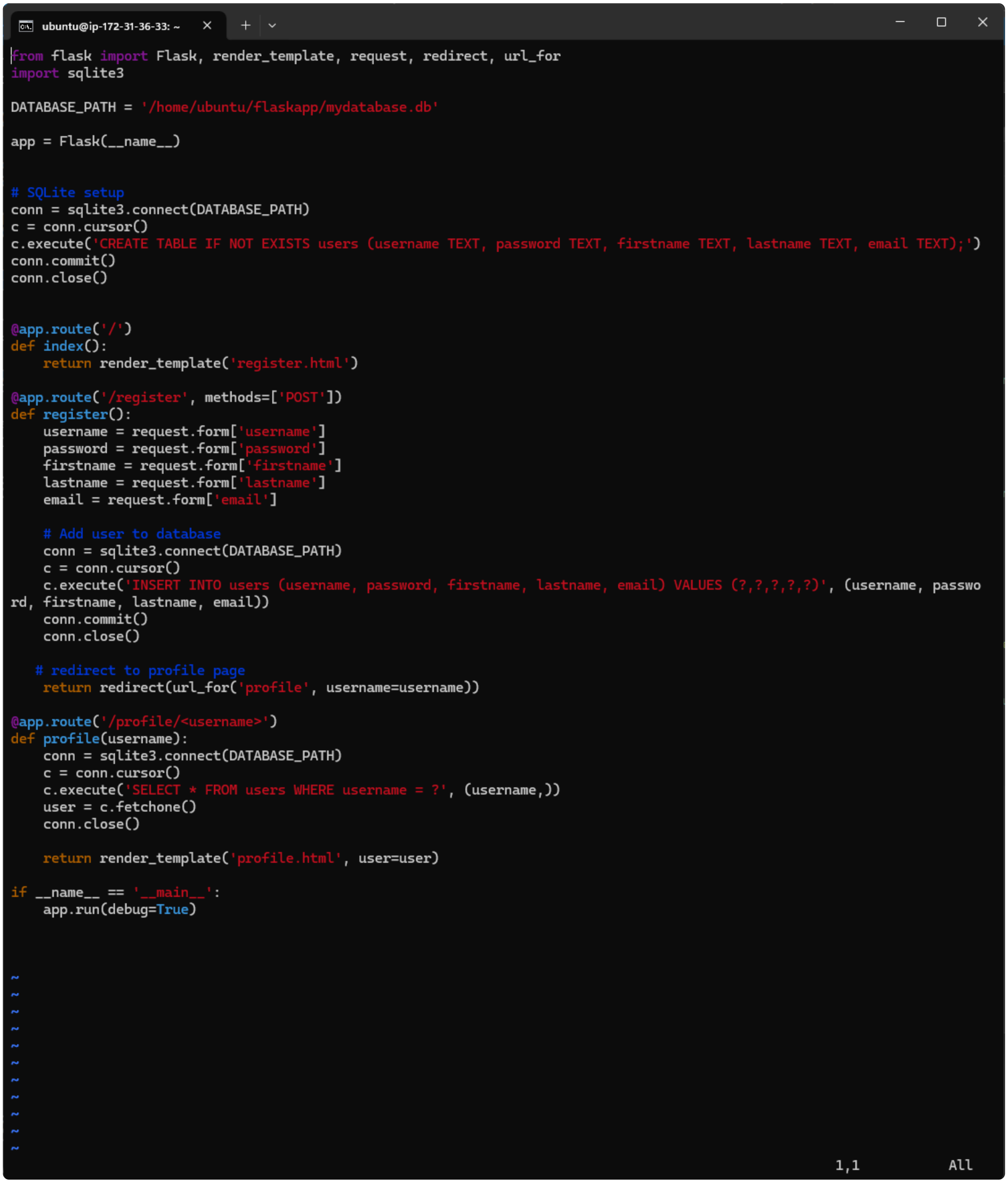
~/flaskapp/flaskapp.py

```python
from flask import Flask, render_template, request, redirect, url_for, session
import sqlite3

DATABASE_PATH = '/home/ubuntu/flaskapp/mydatabase.db'

app = Flask(__name__)
app.secret_key = 'SUPERSECRETSECRET'

# SQLite setup
conn = sqlite3.connect(DATABASE_PATH)
c = conn.cursor()
c.execute('CREATE TABLE IF NOT EXISTS users (username TEXT, password TEXT, firstname TEXT, lastname TEXT, email TEXT);')
conn.commit()
conn.close()

@app.route('/')
def index():
    return render_template('register.html')

@app.route('/register', methods=['POST'])
def register():
    username = request.form['username']
    password = request.form['password']
    firstname = request.form['firstname']
    lastname = request.form['lastname']
    email = request.form['email']

    # Add user to database
    conn = sqlite3.connect(DATABASE_PATH)
    c = conn.cursor()
    c.execute('INSERT INTO users (username, password, firstname, lastname, email) VALUES (?,?,?,?,?)', (username, password, firstname, lastname, email))
    conn.commit()
    conn.close()

   # redirect to profile page
    return redirect(url_for('login'))

@app.route('/login')
```

```
def login():
    return render_template('login.html')

@app.route('/login', methods=['POST'])
def login_post():
    username = request.form['username']
    password = request.form['password']
    conn = sqlite3.connect(DATABASE_PATH)
    c = conn.cursor()
    c.execute('SELECT * FROM users WHERE username = ?', (username,))
    user = c.fetchone()
    conn.close()

    if user and (str(user[1]) == password):
        session['username'] = username
        session['password'] = password
        return redirect(url_for('profile', username=username))
    else:
        return redirect(url_for('index'))

@app.route('/profile/<username>/')
def profile(username):
    if session.get('username') == username and session.get('password'):
        # Connect to the database and fetch user details
        conn = sqlite3.connect(DATABASE_PATH)
        c = conn.cursor()
        c.execute('SELECT * FROM users WHERE username = ?', (username,))
        user = c.fetchone()
        conn.close()

        # Check if the password in session matches the database password
        if user and user[1] == session['password']:
            return render_template('profile.html', user=user)
        else:
            return redirect(url_for('login'))

    return redirect(url_for('login'))

if __name__ == '__main__':
    app.run(debug=True)
```



```
ubuntu@ip-172-31-36-33:~$ ls
flaskapp
ubuntu@ip-172-31-36-33:~$ vim flaskapp/flaskapp.py
```

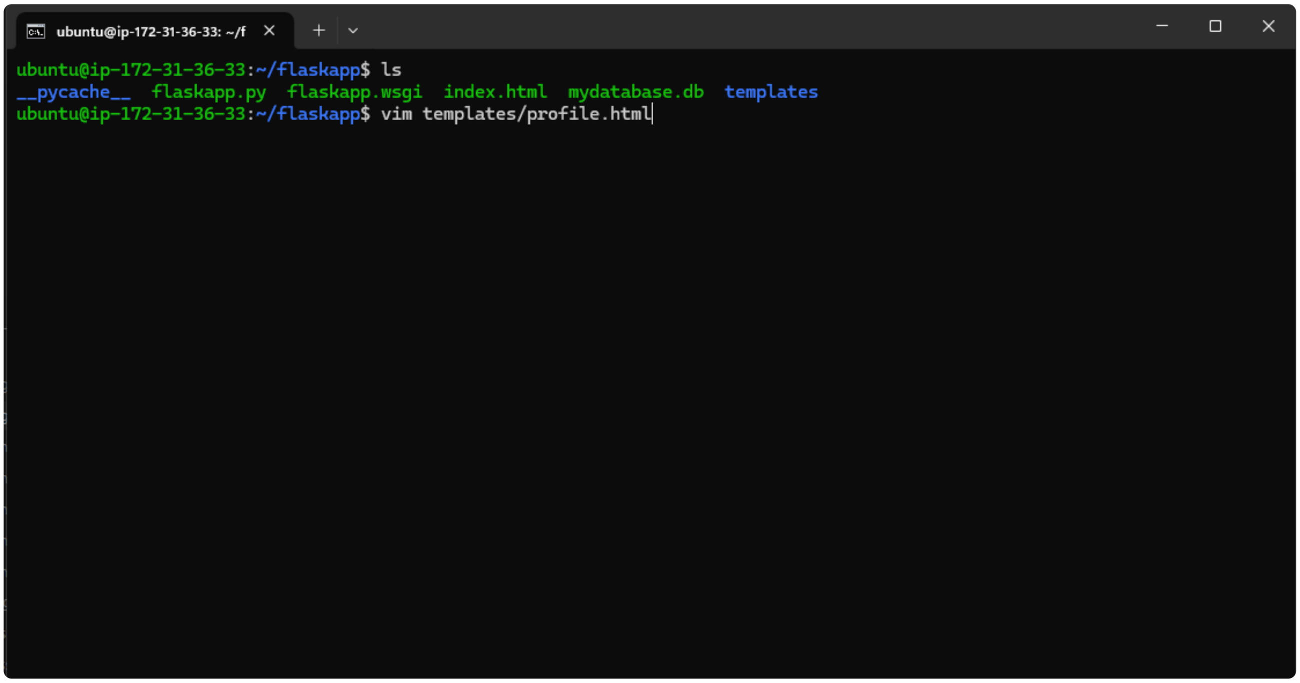


```
from flask import Flask, render_template, request, redirect, url_for
import sqlite3

DATABASE_PATH = '/home/ubuntu/flaskapp/mydatabase.db'

app = Flask(__name__)


# SQLite setup
conn = sqlite3.connect(DATABASE_PATH)
c = conn.cursor()
c.execute('CREATE TABLE IF NOT EXISTS users (username TEXT, password TEXT, firstname TEXT, lastname TEXT, email TEXT);')
conn.commit()
conn.close()


@app.route('/')
def index():
    return render_template('register.html')

@app.route('/register', methods=['POST'])
def register():
    username = request.form['username']
    password = request.form['password']
    firstname = request.form['firstname']
    lastname = request.form['lastname']
    email = request.form['email']

    # Add user to database
    conn = sqlite3.connect(DATABASE_PATH)
    c = conn.cursor()
    c.execute('INSERT INTO users (username, password, firstname, lastname, email) VALUES (?,?,?,?,?)', (username, passwo
rd, firstname, lastname, email))
    conn.commit()
    conn.close()

   # redirect to profile page
    return redirect(url_for('profile', username=username))

@app.route('/profile/<username>')
def profile(username):
    conn = sqlite3.connect(DATABASE_PATH)
    c = conn.cursor()
    c.execute('SELECT * FROM users WHERE username = ?', (username,))
    user = c.fetchone()
    conn.close()

    return render_template('profile.html', user=user)

if __name__ == '__main__':
    app.run(debug=True)




~
~
~
~
~
~
~
~
~
~
~
                                                                                                        1,1           All
```

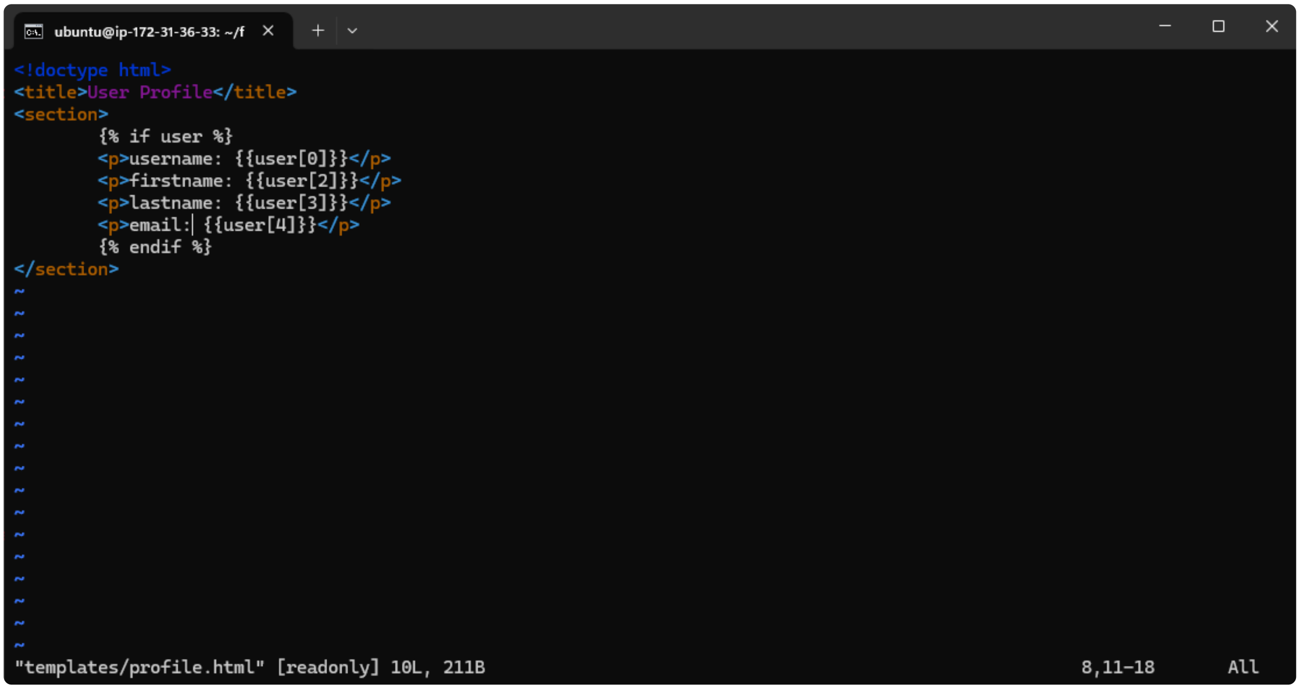
Copy to ~/flaskapp/templates/profile.html

```
<!doctype html>
<title>User Profile</title>
<section>
        {% if user %}
```

```
                <p>username: {{user[0]}}</p>
                <p>firstname: {{user[2]}}</p>
                <p>lastname: {{user[3]}}</p>
                <p>email: {{user[4]}}</p>
        {% endif %}
</section>
```



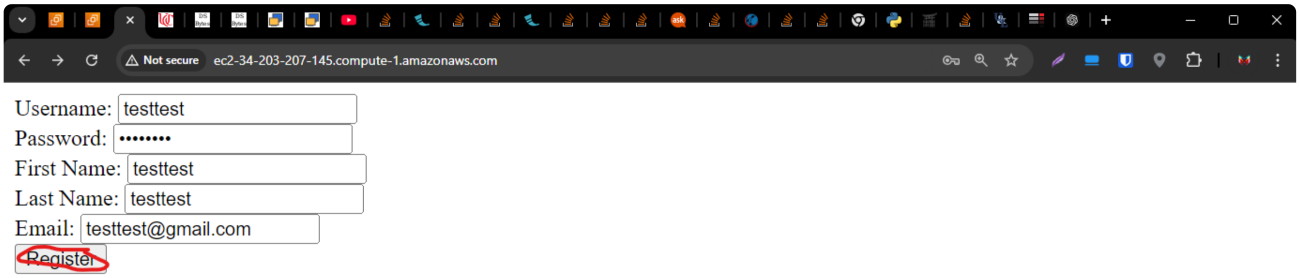
```
ubuntu@ip-172-31-36-33:~/flaskapp$ ls
__pycache__  flaskapp.py  flaskapp.wsgi  index.html  mydatabase.db  templates
ubuntu@ip-172-31-36-33:~/flaskapp$ vim templates/profile.html
```



```
<!doctype html>
<title>User Profile</title>
<section>
        {% if user %}
        <p>username: {{user[0]}}</p>
        <p>firstname: {{user[2]}}</p>
        <p>lastname: {{user[3]}}</p>
        <p>email: {{user[4]}}</p>
        {% endif %}
</section>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"templates/profile.html" [readonly] 10L, 211B                     8,11-18       All
```

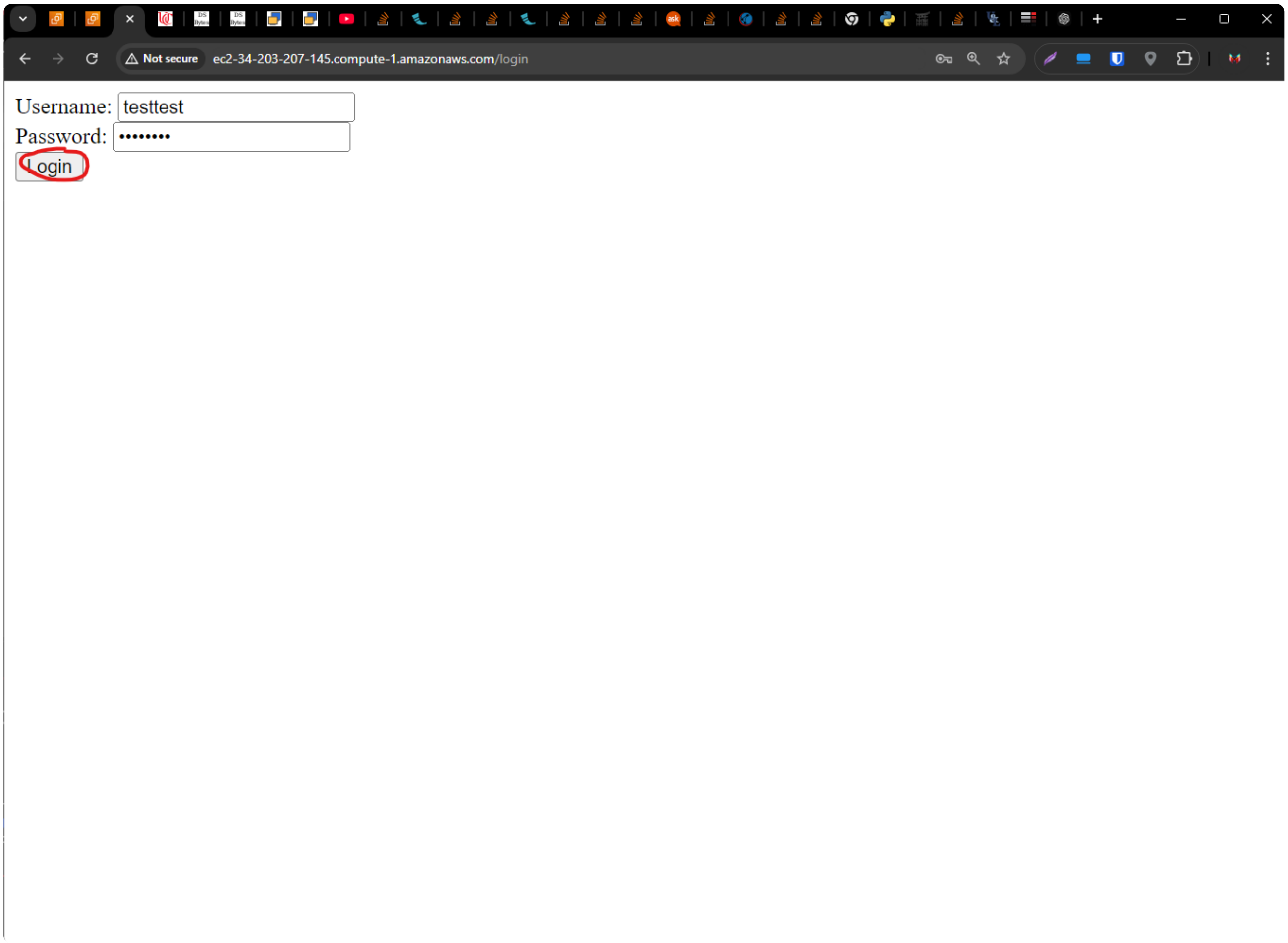
Demo

Register



Username: testtest
Password: ••••••••
First Name: testtest
Last Name: testtest
Email: testtest@gmail.com

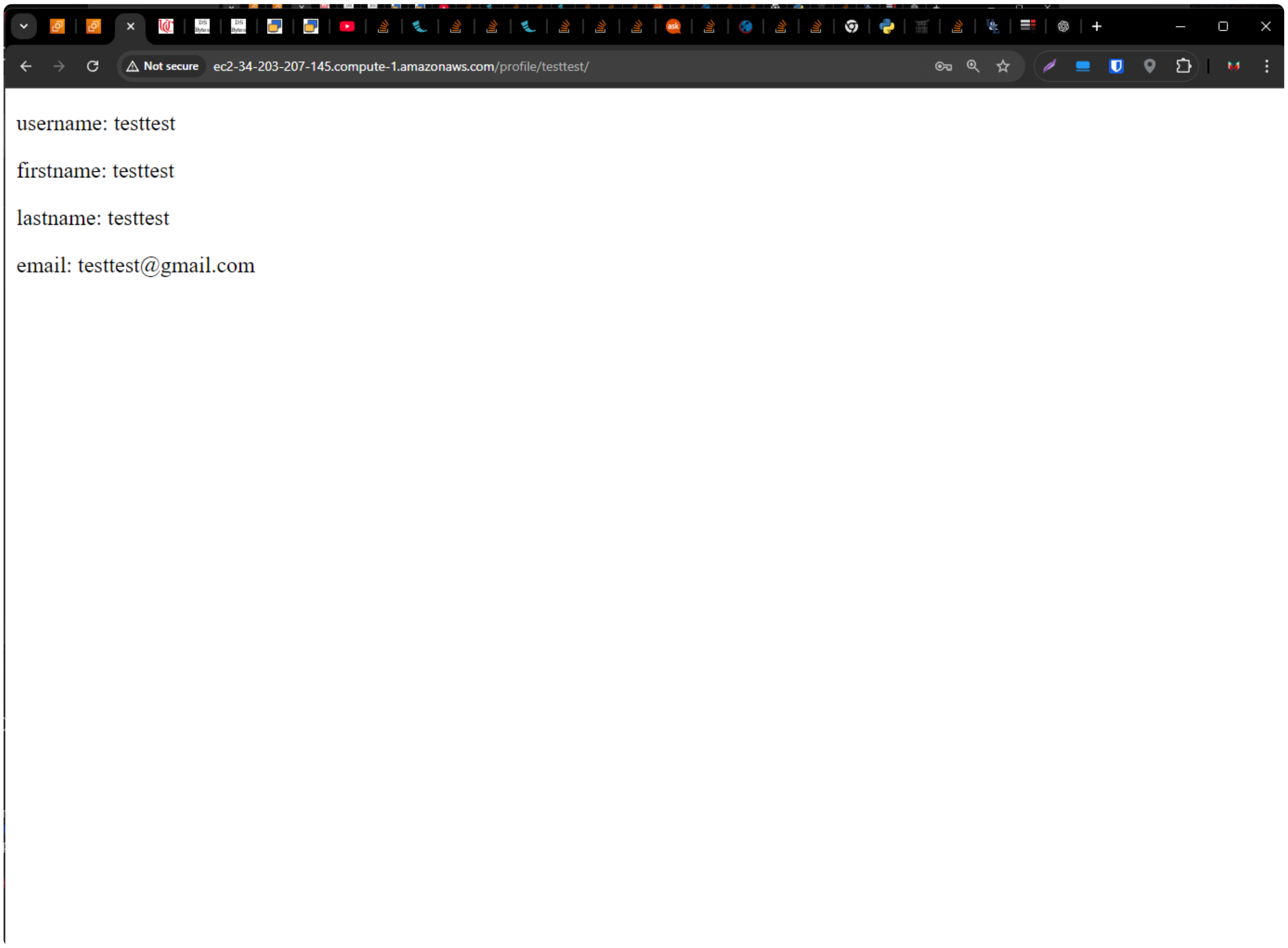
Login



Username: testtest

Password: ••••••••

Profile (After Login)



Not secure
ec2-34-203-207-145.compute-1.amazonaws.com/profile/testtest/

username: testtest

firstname: testtest

lastname: testtest

email: testtest@gmail.com