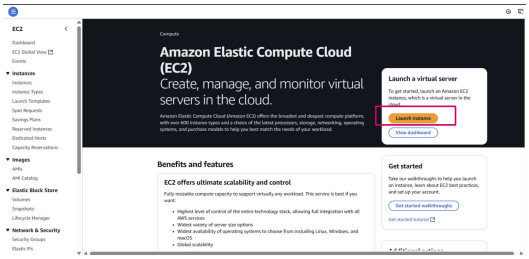
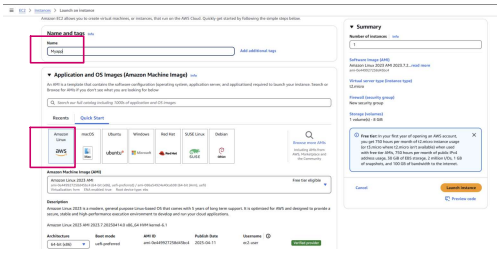
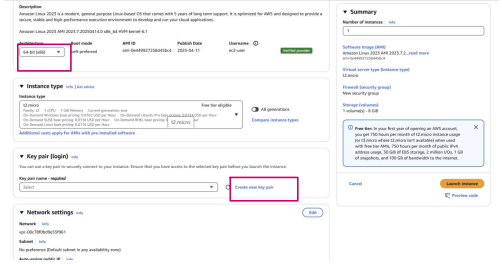
EC2- Hosting without Port

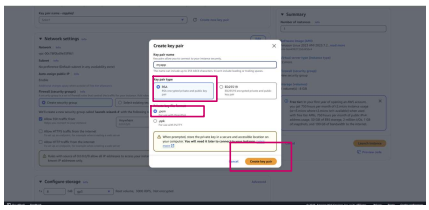
27 April 2025 20:15

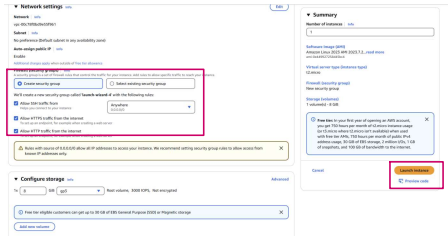
Step 1: Go to EC2 Service

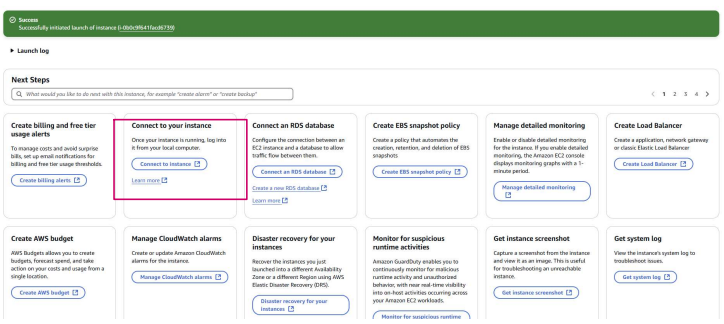


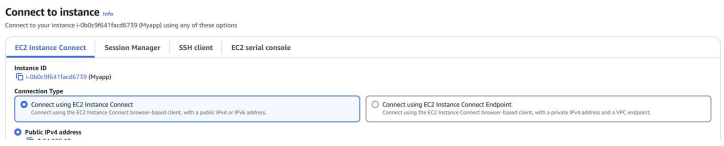


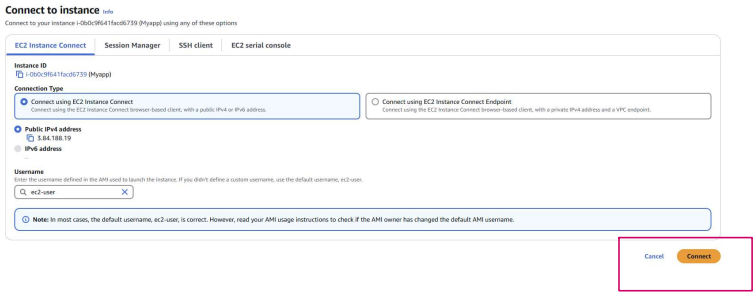


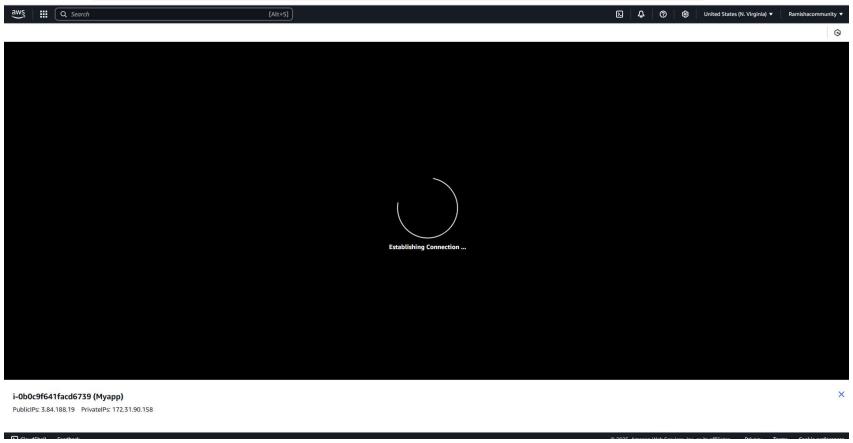


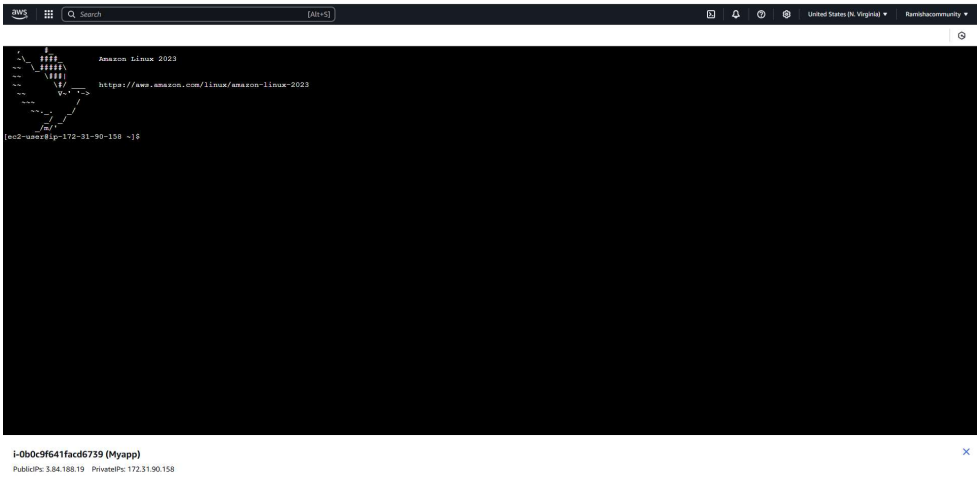






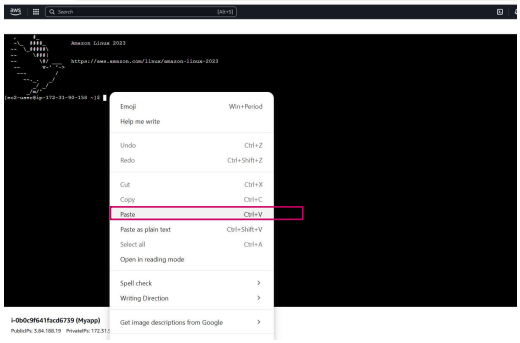


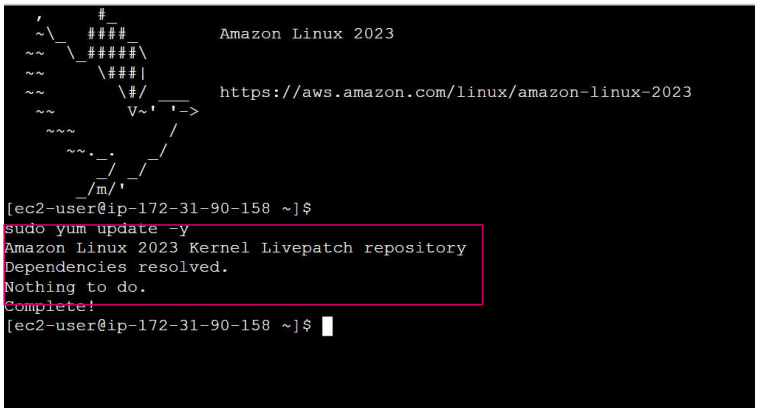




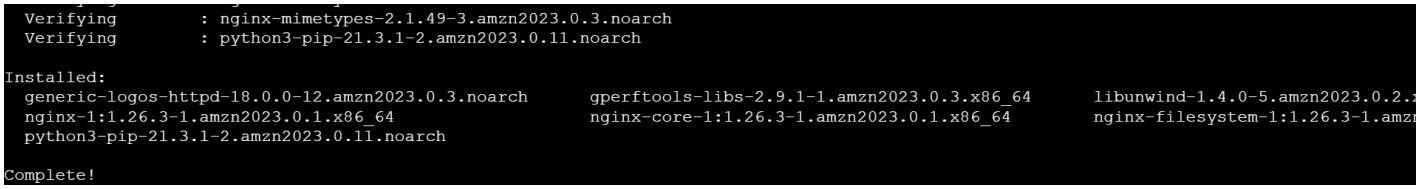
sudo yum update -y

Copy the command and Right Click it and paste it

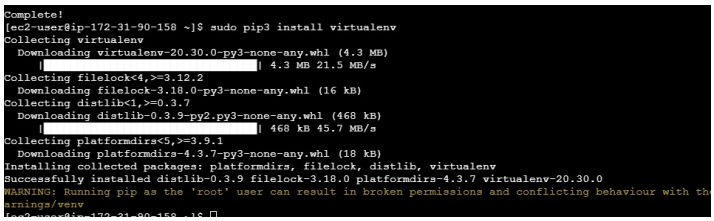




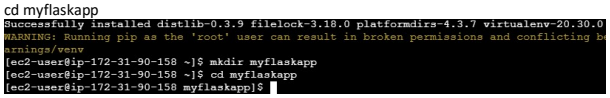
sudo yum install python3 python3-pip nginx -y



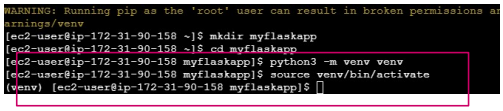
sudo pip3 install virtualenv



mkdir myflaskapp
cd myflaskapp



python3 -m venv venv
source venv/bin/activate



pip install flask gunicorn

```
Downloading virtualenv-20.30.0-py3-none-any.whl (4.3 MB)
   |████████████████████████████████| 4.3 MB 21.5 MB/s
ollecting filelock<4,>=3.12.2
 Downloading filelock-3.18.0-py3-none-any.whl (16 kB)
ollecting distlib<1,>=0.3.7
 Downloading distlib-0.3.9-py2.py3-none-any.whl (468 kB)
   |████████████████████████████████| 468 kB 45.7 MB/s
ollecting platformdirs<5,>=3.9.1
 Downloading platformdirs-4.3.7-py3-none-any.whl (18 kB)
nstalling collected packages: platformdirs, filelock, distlib, virtualenv
uccessfully installed distlib-0.3.9 filelock-3.18.0 platformdirs-4.3.7 virtualenv-20.30.0
ARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
rnings/venv
c2-user@ip-172-31-90-158 ~]$ mkdir myflaskapp
c2-user@ip-172-31-90-158 ~]$ cd myflaskapp
c2-user@ip-172-31-90-158 myflaskapp]$ python3 -m venv venv
c2-user@ip-172-31-90-158 myflaskapp]$ source venv/bin/activate
env) [ec2-user@ip-172-31-90-158 myflaskapp]$ pip install flask gunicorn
ollecting flask
 Downloading flask-3.1.0-py3-none-any.whl (102 kB)
   |████████████████████████████████| 102 kB 5.2 MB/s
ollecting gunicorn
 Downloading gunicorn-23.0.0-py3-none-any.whl (85 kB)
   |████████████████████████████████| 85 kB 6.5 MB/s
ollecting blinker>=1.9
 Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
ollecting importlib-metadata>=3.6
 Downloading importlib_metadata-8.7.0-py3-none-any.whl (27 kB)
ollecting itsdangerous>=2.2
 Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
ollecting Jinja2>=3.1.2
 Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
   |████████████████████████████████| 134 kB 82.1 MB/s
ollecting Werkzeug>=3.1
 Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
   |████████████████████████████████| 224 kB 97.9 MB/s
ollecting click>=8.1.3
 Downloading click-8.1.8-py3-none-any.whl (98 kB)
   |████████████████████████████████| 98 kB 15.0 MB/s
ollecting packaging
 Downloading packaging-25.0-py3-none-any.whl (66 kB)
   |████████████████████████████████| 66 kB 8.3 MB/s
ollecting zipp>=3.20
 Downloading zipp-3.21.0-py3-none-any.whl (9.6 kB)
ollecting MarkupSafe>=2.0
 Downloading MarkupSafe-3.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
nstalling collected packages: zipp, MarkupSafe, Werkzeug, packaging, Jinja2, itsdangerous, importlib-metadata, click, blinker, gunicorn, flask
uccessfully installed Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 flask-3.1.0 gunicorn-23.0.0 importlib-metadata-8.7.0 itsdanger
ARNING: You are using pip version 21.3.1; however, version 25.1 is available.
ou should consider upgrading via the '/home/ec2-user/myflaskapp/venv/bin/python3 -m pip install --upgrade pip' command.
env) [ec2-user@ip-172-31-90-158 myflaskapp]$
```

sudo yum install git -y

```
  Installing       : git-2.47.1-1.amzn2023.0.2.x86_64
8
  Running scriptlet: git-2.47.1-1.amzn2023.0.2.x86_64
8
  Verifying        : git-2.47.1-1.amzn2023.0.2.x86_64
8
  Verifying        : git-core-2.47.1-1.amzn2023.0.2.x86_64
8
  Verifying        : git-core-doc-2.47.1-1.amzn2023.0.2.noarch
8
  Verifying        : perl-Error-1:0.17029-5.amzn2023.0.2.noarch
8
  Verifying        : perl-File-Find-1.37-477.amzn2023.0.6.noarch
8
  Verifying        : perl-Git-2.47.1-1.amzn2023.0.2.noarch
8
  Verifying        : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64
8
  Verifying        : perl-lib-0.65-477.amzn2023.0.6.x86_64
8

Installed:
  git-2.47.1-1.amzn2023.0.2.x86_64                      git-core-2.47.1-1.amzn2023.0.2.x86_64

  perl-File-Find-1.37-477.amzn2023.0.6.noarch           perl-Git-2.47.1-1.amzn2023.0.2.noarch

Complete!
```

git clone https://github.com/RamishaRaniK/flasktesting.git
git clone https://github.com/RamishaRaniK/awslocalmodelhosting.git

```
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ git clone https://github.com/RamishaRaniK/flasktesting.git
Cloning into 'flasktesting'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$
```

cd flasktesting

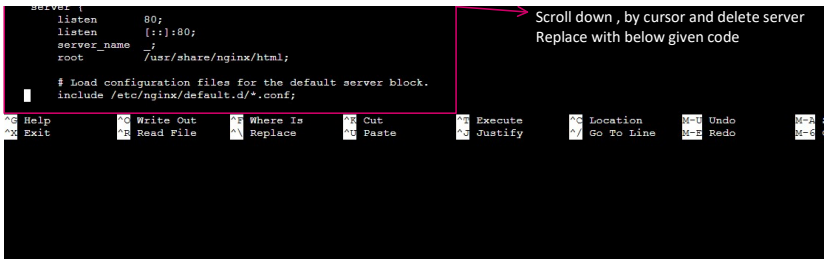
```
Receiving objects: 100% (3/3), done.
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$
```

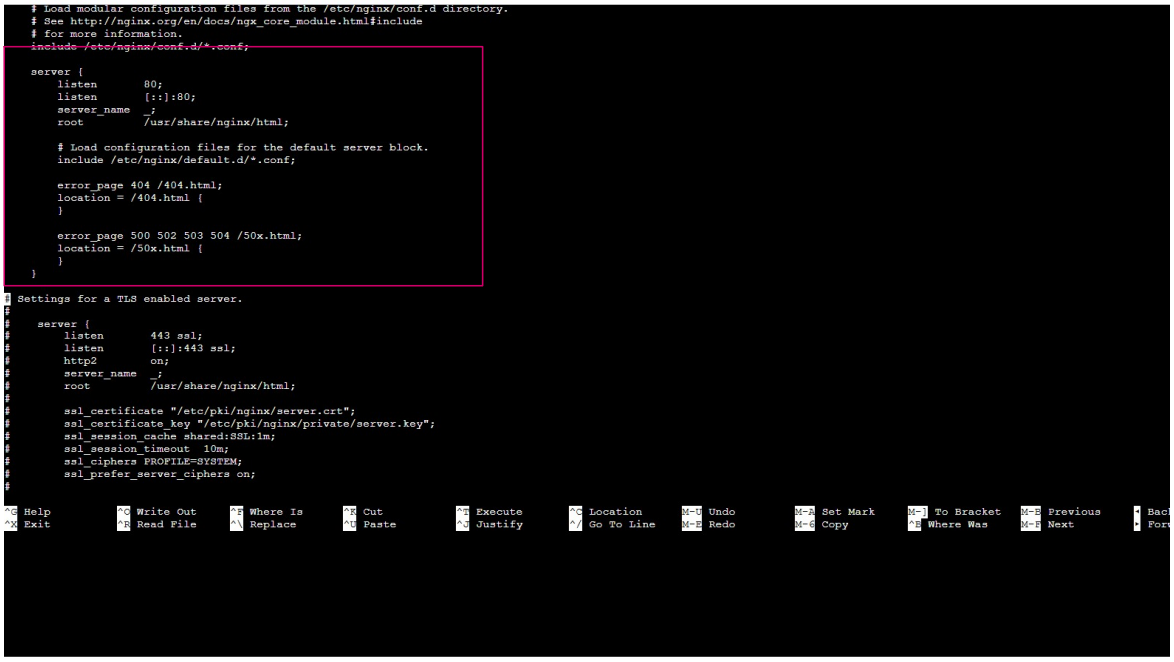
pip install -r requirements.txt

sudo nano /etc/nginx/nginx.conf

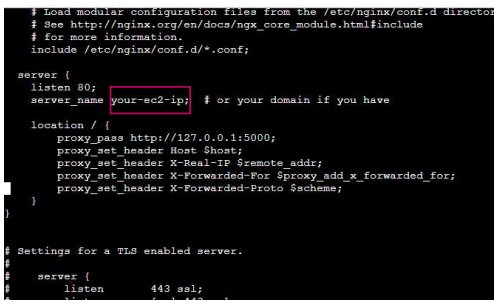

```
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80;
        listen       [::]:80;
        server_name  _;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;
```

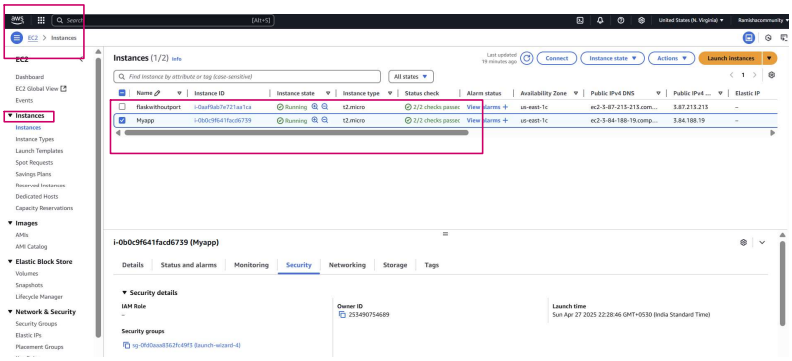
Scroll down , by cursor and delete server
Replace with below given code

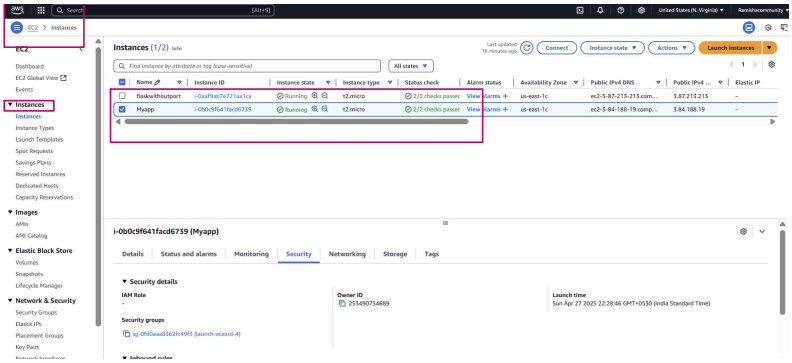


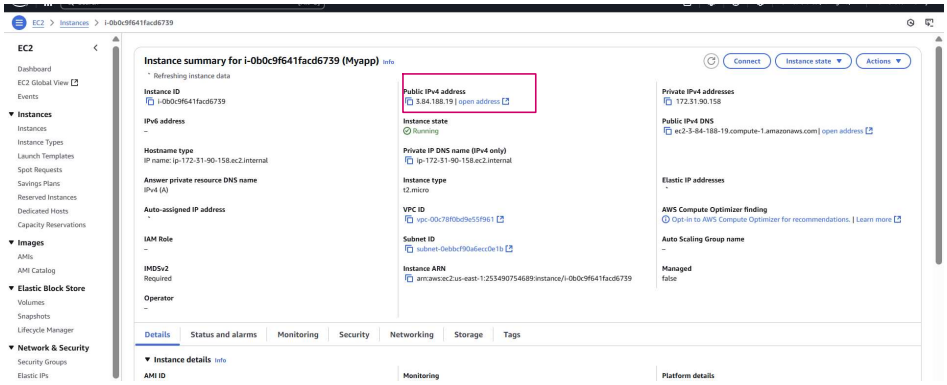


```
server {
   listen 80;
   server_name your-ec2-ip;  # or your domain if you have

   location / {
      proxy_pass http://127.0.0.1:5000;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header X-Forwarded-Proto $scheme;
   }
}
```









```
    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

  server {
    listen 80;
    server_name 3.84.188.19;  # or your domain if you have

    location / {
        proxy_pass http://127.0.0.1:5000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}


# Settings for a TLS enabled server.
#
#    server {
#        listen       443 ssl;
#        listen       [::]:443 ssl;
#        http2        on;
#        server_name  _;
#        root         /usr/share/nginx/html;
#
#        ssl_certificate "/etc/pki/nginx/server.crt";
```

Ctrl+S

Ctrl+X --? exit

```
Complete!
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ git clone https://github.com/RamishaRaniK/flasktesting.git
Cloning into 'flasktesting'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nano /etc/nginx/nginx.conf
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$
```

sudo nginx -t

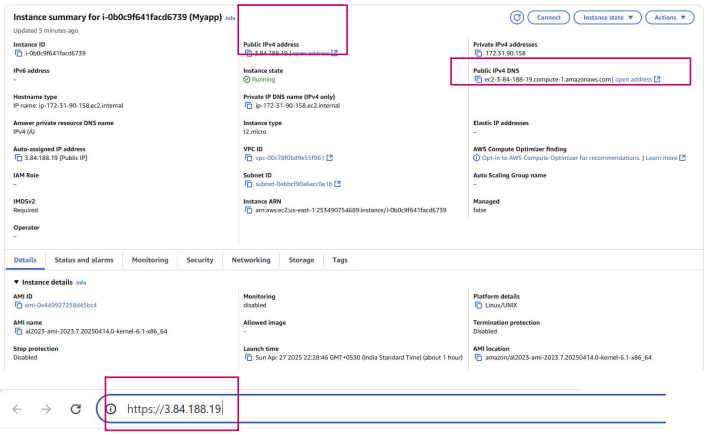
```
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nano /etc/nginx/nginx.conf
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$
```

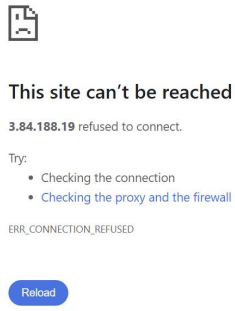
sudo systemctl restart nginx

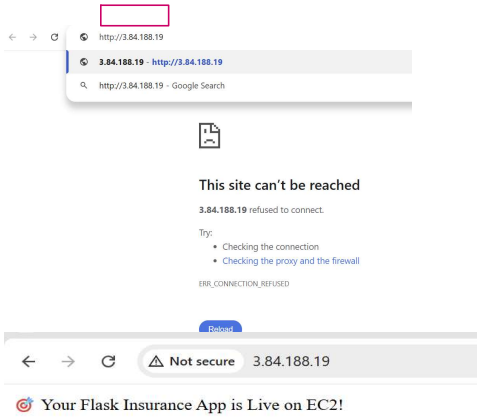
```
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ git clone https://github.com/RamishaRaniK/flasktesting.git
Cloning into 'flasktesting'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nano /etc/nginx/nginx.conf
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo systemctl restart nginx
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$
```

gunicorn -w 3 -b 0.0.0.0:5000 app:app

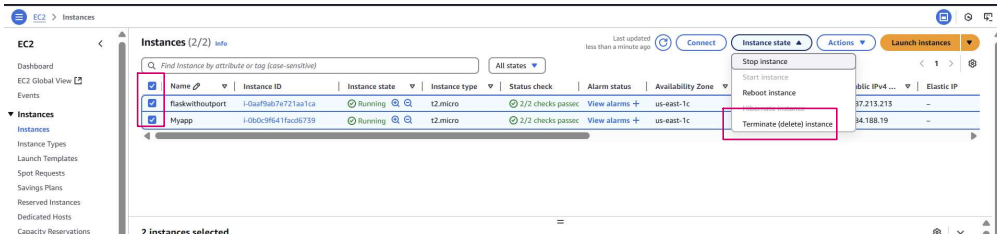
```
nginx: configuration file /etc/nginx/nginx.conf test is successful
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ sudo systemctl restart nginx
(venv) [ec2-user@ip-172-31-90-158 flasktesting]$ gunicorn -w 3 -b 0.0.0.0:5000 app:app
[2025-04-27 18:16:22 +0000] [30105] [INFO] Starting gunicorn 23.0.0
[2025-04-27 18:16:22 +0000] [30105] [INFO] Listening at: http://0.0.0.0:5000 (30105)
[2025-04-27 18:16:22 +0000] [30105] [INFO] Using worker: sync
[2025-04-27 18:16:22 +0000] [30106] [INFO] Booting worker with pid: 30106
[2025-04-27 18:16:22 +0000] [30107] [INFO] Booting worker with pid: 30107
[2025-04-27 18:16:22 +0000] [30108] [INFO] Booting worker with pid: 30108
```

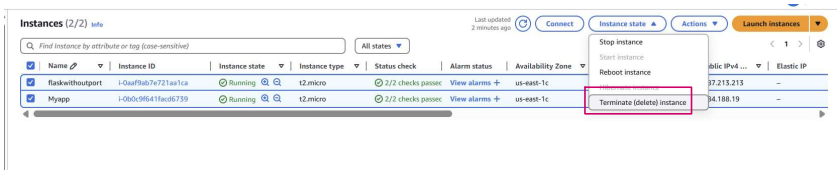


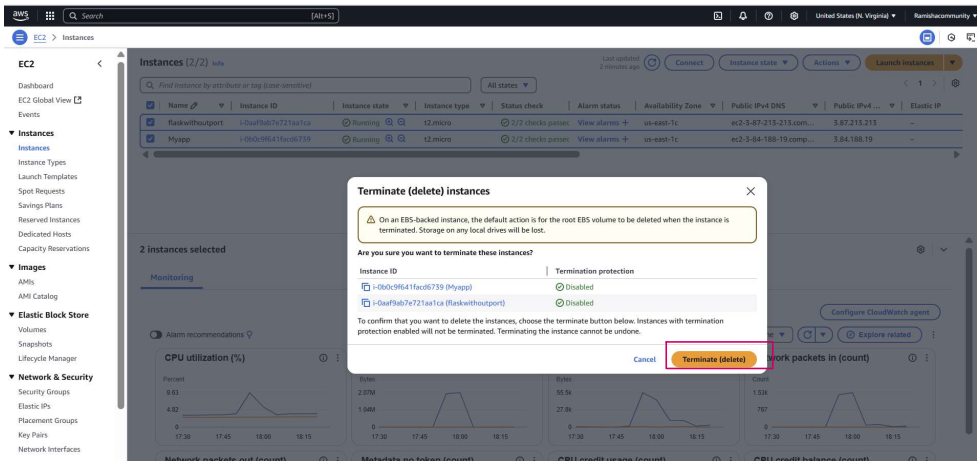


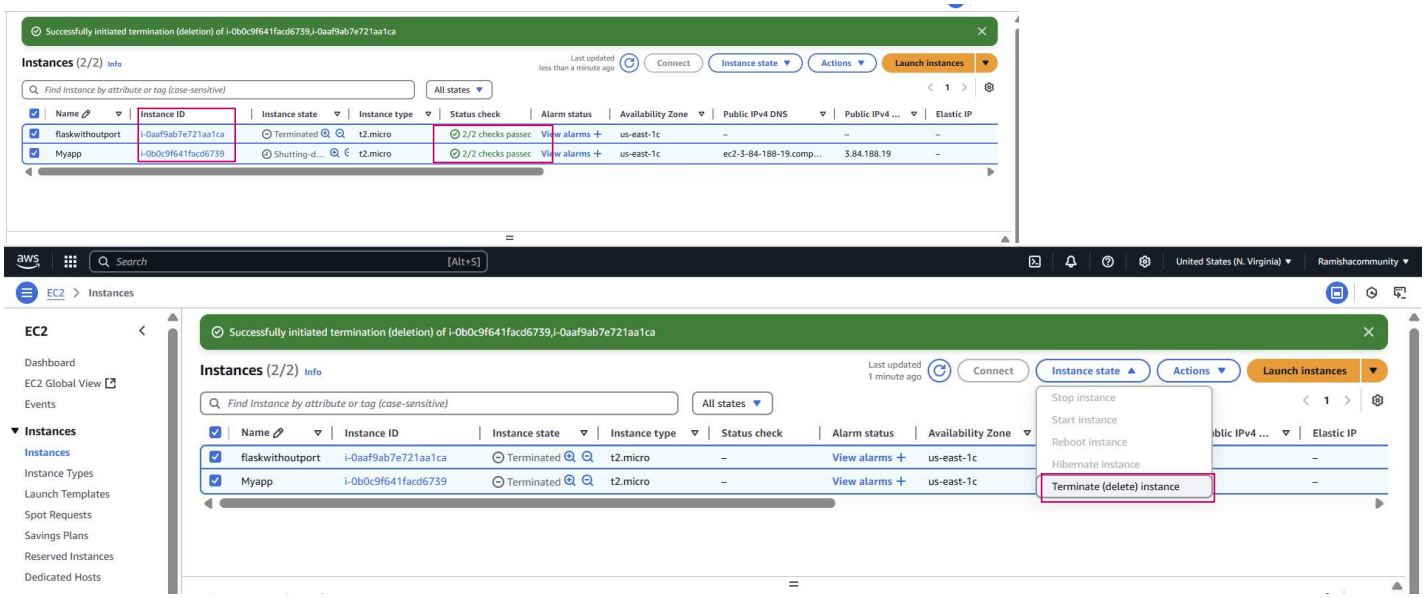


If you are practicing delete it









To get into https:

❗ Important Concept:

Access	HTTPS Possible?	Why?
`3.87.213.213` (only IP)	❌ Not possible	SSL Certificates need domain name
`yourdomain.com`	✅ Possible	SSL certificates are issued to domain names

🎯 Now next important thing:

You have two choices regarding using **hopelearning.net**:

Option	Meaning	Best For
Use a Subdomain (Recommended)	Create a subdomain like `flask.hopelearning.net` and point to your EC2 instance	Safe, no impact on main site
Use Main Domain	Move entire `hopelearning.net` to EC2	Risky, your current WordPress site will break

flask.hopelearning.net
api.hopelearning.net
ec2.hopelearning.net

sudo yum install epel-release -y
sudo yum install certbot python3-certbot-nginx -y
sudo certbot --nginx -d flask.hopelearning.net

https://flask.hopelearning.net/

To run gunicorn in backend service(Production Level)

🚀 2 Powerful Ways to Embed Your Flask App Inside Main Website

✅ 1. Using `<iframe>`

The **simplest way** is to use an **iframe** — it embeds another web page inside your existing page.

Example:

On your WordPress (or custom HTML page), insert:

```html
<iframe src="https://flask.hopelearning.net" width="100%" height="600px" frameborder="0"></iframe
```

gunicorn -w 3 -b 127.0.0.1:5000 app:app &

✅ If you want quick temporary fix:

One-line command:

```bash
gunicorn -w 3 -b 127.0.0.1:5000 app:app &
```

✅ The `&` at the end sends Gunicorn to **background**.

Then you can safely close SSH tab, Gunicorn keeps running.
(But if server reboots, it won't restart.)

✅ If you want **professional full setup (Recommended):**

We create a `systemd` service file.

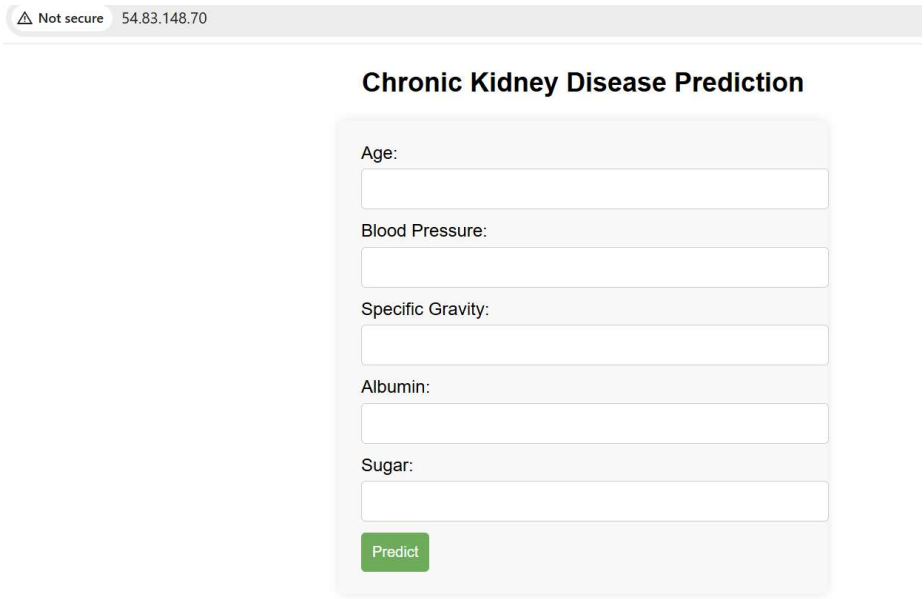
Example service: `/etc/systemd/system/flaskapp.service`

```ini
[Unit]
Description=Gunicorn instance to serve Flask app
After=network.target

[Service]
User=ec2-user
Group=nginx
WorkingDirectory=/home/ec2-user/flasktesting
Environment="PATH=/home/ec2-user/flasktesting/venv/bin"
ExecStart=/home/ec2-user/flasktesting/venv/bin/gunicorn -w 3 -b 127.0.0.1:5000 app:app

[Install]
WantedBy=multi-user.target
```

✅ Then you can control Gunicorn with:

```bash
sudo systemctl start flaskapp
sudo systemctl enable flaskapp
sudo systemctl status flaskapp
```

🔥 Auto-run forever even if EC2 rebooted.


Not secure
54.83.148.70
Chronic Kidney Disease Prediction
Age:
Blood Pressure:
Specific Gravity:
Albumin:
Sugar:
Predict