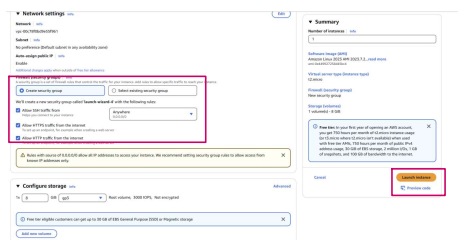
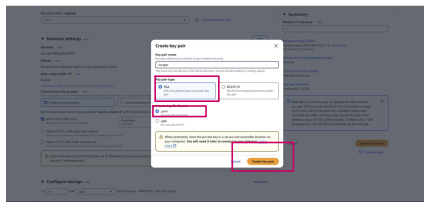
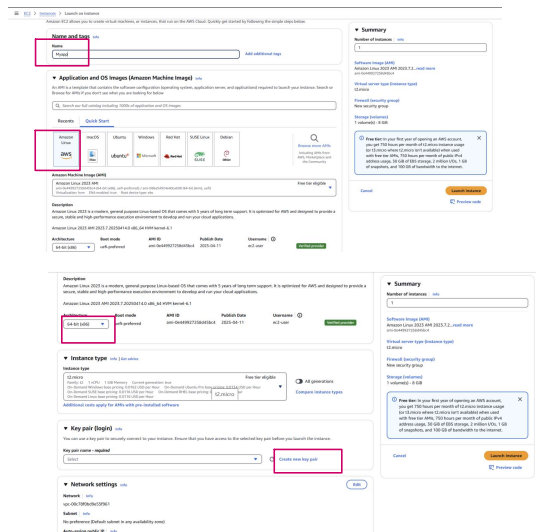
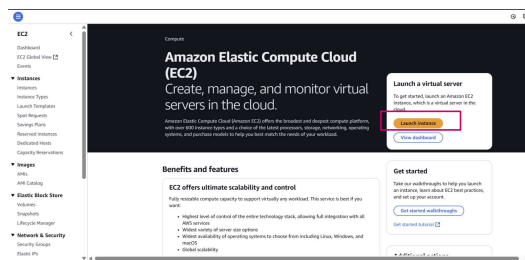


EC2- Hosting without Port

27 April 2025 20:15

Step 1: Go to EC2 Service



Success
Successfully initiated launch of instance i-05b98411ac07350

Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"?

Create billing and free tier usage alerts

Amazon CloudWatch monitors your account for billing and usage. Set up alerts to notify you when your usage exceeds a threshold.

Create billing alerts

Connect to your instance

Open your instance in a terminal, copy logs, or view the instance in the AWS console.

Connect to instance

Connect an RDS database

Configure the connection between an Amazon RDS database and your EC2 instance.

Connect an RDS database

Create EBS snapshot policy

Create a policy that automates the creation, retention, and deletion of EBS snapshots.

Create EBS snapshot policy

Manage detailed monitoring

Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.

Manage detailed monitoring

Create Load Balancer

Create a load balancer, network gateway, or Amazon ElastiCache Load Balancing.

Create Load Balancer

Create AWS budget

AWS Budget allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.

Create AWS budget

Manage CloudWatch alarms

Create or update Amazon CloudWatch alarms for the instance.

Manage CloudWatch alarms

Disaster recovery for your instances

Recover the instances you just launched into a different Availability Zone or a different Region using AWS Elastic Disaster Recovery (EDR).

Disaster recovery for your instances

Monitor for suspicious runtime activities

Amazon GuardDuty enables you to continuously monitor for malicious runtime activity and unauthorized behavior with near real-time visibility into on-host activities occurring across your Amazon EC2 workloads.

Monitor for suspicious runtime

Get instance screenshot

Capture a screenshot from the instance and save it as an image. This is useful for troubleshooting an unresponsive instance.

Get instance screenshot

Get system log

View the instance system log to troubleshoot issues.

Get system log

Connect to instance

Connect to your instance i-05b98411ac07350 (myapp) using any of these options

EC2 Instance Connect

Instance ID: i-05b98411ac07350 (myapp)

Connection Type

Connect using EC2 Instance Connect

Connect using the EC2 Instance Connect browser-based client, with a public IP or IPv4 address.

Public IPv4 address

10.0.0.1

Session Manager

SSH client

EC2 serial console

Connect to instance info

Connect to your instance i-0b0c9f641facd6739 (Myapp) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID
i-0b0c9f641facd6739 (Myapp)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect Session-based client, with a public IP/v4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect Endpoint-based client, with a private IP/v4 address and a VPC endpoint.

☒ Public IPv4 address
5.84.188.19

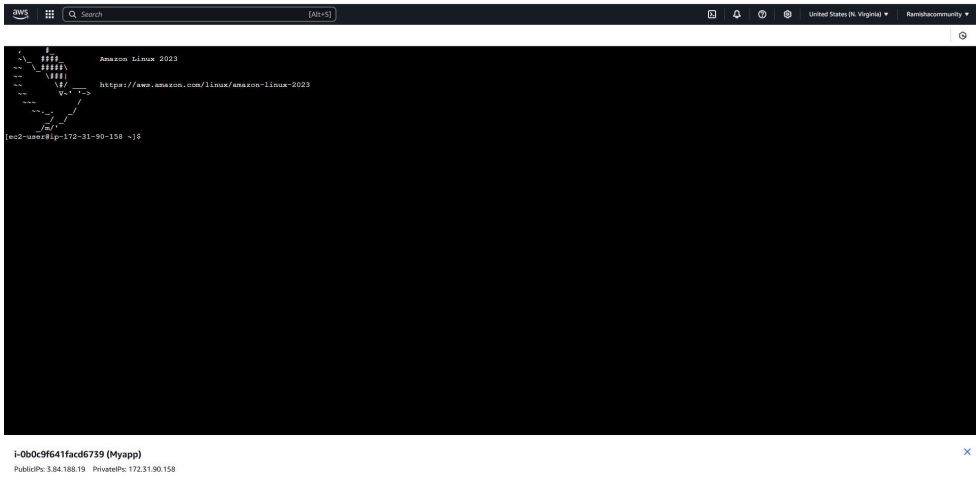
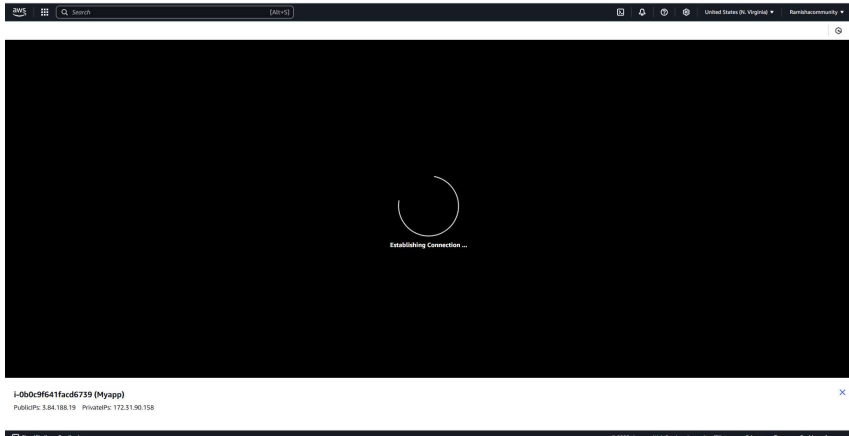
☐ IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

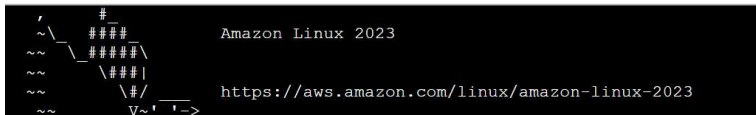
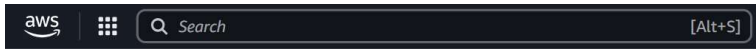
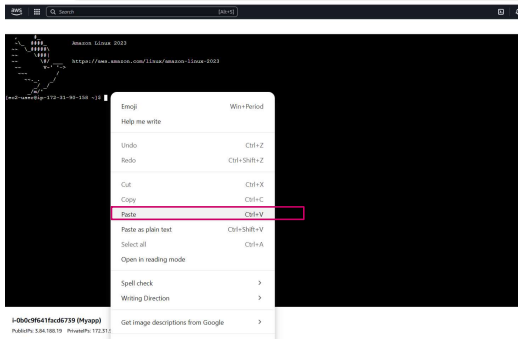
Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect



sudo yum update -y

Copy the command and Right Click it and paste it



```
Verifying      : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
Verifying      : python3-pip-21.3.1-2.amzn2023.0.11.noarch

Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64      libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.26.3-1.amzn2023.0.1.x86_64                  nginx-core-1:1.26.3-1.amzn2023.0.1.x86_64         nginx-filestream-1:1.26.3-1.amzn2023.0.1.x86_64
python3-pip-21.3.1-2.amzn2023.0.11.noarch

Complete!
```

```
Complete!
[ec2-user@ip-172-31-90-158 ~]$ sudo pip3 install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.30.0-py3-none-any.whl (4.3 MB)
    |#####| 4.3 MB 21.5 MB/s
Collecting filelock<4,>=3.12.2
  Downloading filelock-3.18.0-py3-none-any.whl (16 kB)
Collecting distlib<1,>=0.3.7
  Downloading distlib-0.3.9-py2.py3-none-any.whl (468 kB)
    |#####| 468 kB 45.7 MB/s
Collecting platformdirs<5,>=3.9.1
  Downloading platformdirs-4.3.7-py3-none-any.whl (18 kB)
Installing collected packages: platformdirs, filelock, distlib, virtualenv
Successfully installed distlib-0.3.9 filelock-3.18.0 platformdirs-4.3.7 virtualenv-20.30.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the
system's package manager. It is recommended to use pipx instead to manage installations.
[ec2-user@ip-172-31-90-158 ~]$
```

```
Successfully installed distlib-0.3.9 filelock-3.18.0 platformdirs-4.3.7 virtualenv-20.30.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting b
arnings/venv
[ec2-user@ip-172-31-90-158 ~]$ mkdir myflaskapp
[ec2-user@ip-172-31-90-158 ~]$ cd myflaskapp
[ec2-user@ip-172-31-90-158 myflaskapp]$
```

```
WARNING: Running pip as the 'root' user can result in broken permissions a
arnings/venv
[ec2-user@ip-172-31-90-158 ~]$ mkdir myflaskapp
[ec2-user@ip-172-31-90-158 ~]$ cd myflaskapp
[ec2-user@ip-172-31-90-158 myflaskapp]$ python3 -m venv venv
[ec2-user@ip-172-31-90-158 myflaskapp]$ source venv/bin/activate
(venv) [ec2-user@ip-172-31-90-158 myflaskapp]$
```

AWS EC2 Hosting Procedure Page 3

```

Installing virtualenv
Downloading virtualenv-20.30.0-py3-none-any.whl (4.3 MB)
| 4.3 MB 21.5 MB/s
Collecting filelock<4,>=3.12.2
Downloading filelock-3.18.0-py3-none-any.whl (16 kB)
Collecting distlib<4,>=3.7
Downloading distlib-0.3.9-py2.py3-none-any.whl (468 kB)
| 468 kB 45.7 MB/s
Collecting platformdirs<5,>=3.9.1
Downloading platformdirs-4.3.7-py3-none-any.whl (18 kB)
Installing collected packages: platformdirs, filelock, distlib, virtualenv
Successfully installed distlib-0.3.9 filelock-3.18.0 platformdirs-4.3.7 virtualenv-20.30.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
venv/venv
ec2-user@ip-172-31-90-158 ~$ mkdir myflaskapp
ec2-user@ip-172-31-90-158 ~$ cd myflaskapp
ec2-user@ip-172-31-90-158 myflaskapp$ python3 -m venv venv
ec2-user@ip-172-31-90-158 myflaskapp$ source venv/bin/activate
venv) [ec2-user@ip-172-31-90-158 myflaskapp]$ pip install flask gunicorn
Collecting flask
Downloading flask-3.1.0-py3-none-any.whl (102 kB)
| 102 kB 5.2 MB/s
Collecting gunicorn
Downloading gunicorn-23.0.0-py3-none-any.whl (85 kB)
| 85 kB 6.5 MB/s
Collecting blinker<=1.9
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting importlib-metadata<=3.6
Downloading importlib-metadata-8.7.0-py3-none-any.whl (27 kB)
Collecting itdangerous<=2.2
Downloading itdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting Jinja2<=3.1.2
Downloading Jinja2-3.1.6-py3-none-any.whl (134 kB)
| 134 kB 82.1 MB/s
Collecting Werkzeug<=3.1
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
| 224 kB 97.9 MB/s
Collecting click<=8.1.8
Downloading click-8.1.8-py3-none-any.whl (98 kB)
| 98 kB 15.0 MB/s
Collecting packaging
Downloading packaging-25.0-py3-none-any.whl (66 kB)
| 66 kB 8.3 MB/s
Collecting zipp<=3.20
Downloading zipp-3.21.0-py3-none-any.whl (9.6 kB)
Collecting MarkupSafe<=2.0
Downloading MarkupSafe-3.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
Installing collected packages: zipp, MarkupSafe, Werkzeug, Packaging, Jinja2, itdangerous, importlib-metadata, click, blinker, gunicorn, flask
Successfully 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 itdangerous-2.2.0
WARNING: You are using pip version 21.3.1; however, version 25.1 is available.
You should consider upgrading via the '/home/ec2-user/myflaskapp/venv/bin/python3 -m pip install --upgrade pip' command.
venv) [ec2-user@ip-172-31-90-158 myflaskapp]$

```

```

sudo yum install git-y
Installing      : git-2.47.1-1.amzn2023.0.2.x86_64
Running scriptlet: git-2.47.1-1.amzn2023.0.2.x86_64
Verifying      : git-2.47.1-1.amzn2023.0.2.x86_64
Verifying      : git-core-2.47.1-1.amzn2023.0.2.x86_64
Verifying      : git-core-doc-2.47.1-1.amzn2023.0.2.noarch
Verifying      : perl-Error-1:0.17029-5.amzn2023.0.2.noarch
Verifying      : perl-File-Find-1.37-477.amzn2023.0.6.noarch
Verifying      : perl-Git-2.47.1-1.amzn2023.0.2.noarch
Verifying      : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64
Verifying      : perl-lib-0.65-477.amzn2023.0.6.x86_64
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
Completed!

```

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
(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

```

aws  [Q Search] [Alt+S]

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/nginx_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;
    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

```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_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;

    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}

# 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";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROTOCOL=SSLv2;
    ssl_prefer_server_ciphers on;
}
```

```
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;
    }
}
```



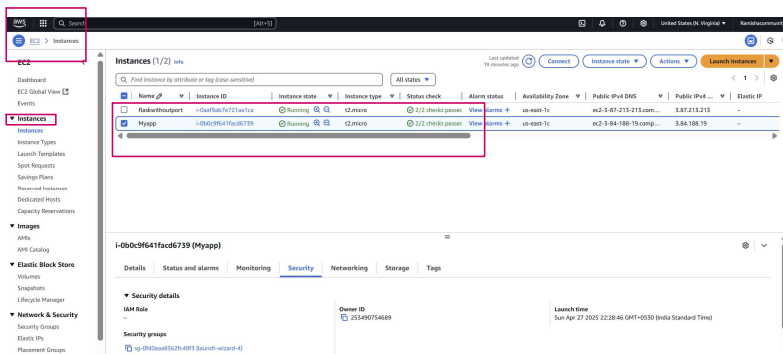
```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

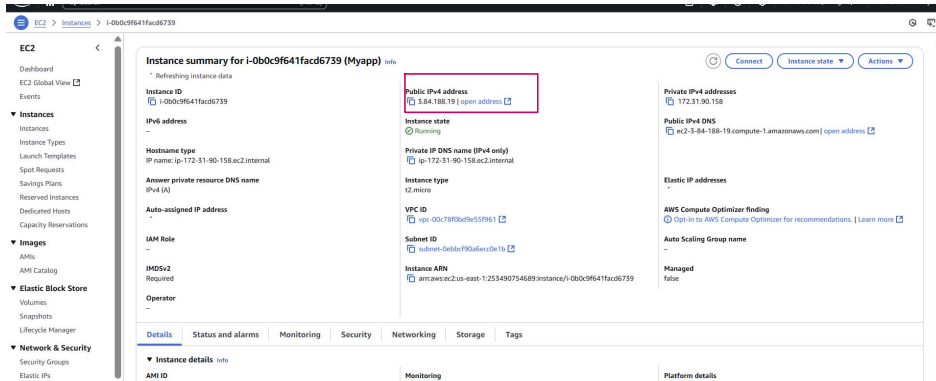
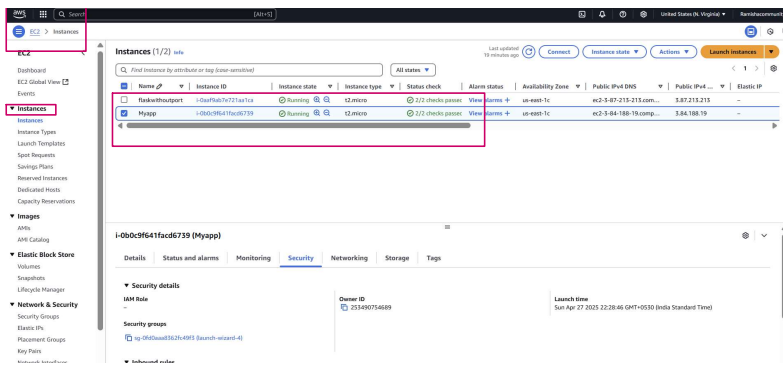
server {
    listen 80;
    server_name your-ec2-ip; # or your domain if you have

    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;
    }
}

# 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";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROTOCOL=SSLv2;
    ssl_prefer_server_ciphers on;
}
```





```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_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

```
(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% (2/2), 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% (2/2), 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
```

Instance summary for i-0b0c9f641facd6739 (Myapp) Updated 5 minutes ago

Instance ID: i-0b0c9f641facd6739

Public IPv4 address: 3.84.188.19

Instance state: Running

Private IP DNS name (IPv4 only): ip-172-31-90-158.ec2.internal

Instance type: t2.micro

VPC ID: vpc-0007803dc3d961

Subnet ID: subnet-bd0b975dus-east-1

Instance ARN: arn:aws:ec2:us-east-1:255480754689:instance/i-0b0c9f641facd6739

Public IPv4 DNS: ec2-3-84-188-19.compute-1.amazonaws.com

Platform details: Linux/UNIX

Termination protection: Disabled

AMI ID: ami-0e4d992725d8d3dc4

AMI name: h2023-ami-2023.7.20250414.0-kernel-6.1-x86_64

Step protection: Disabled

Launch time: Sun Apr 27 2025 22:28:46 GMT+0530 (India Standard Time) (about 1 hour)

AMI location: amazon2023-ami-2023.7.20250414.0-kernel-6.1-x86_64

URL: https://3.84.188.19



This site can't be reached

3.84.188.19 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Reload

http://3.84.188.19

3.84.188.19 - http://3.84.188.19

http://3.84.188.19 - Google Search



This site can't be reached

3.84.188.19 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Reload

Not secure 3.84.188.19

Your Flask Insurance App is Live on EC2!

If you are practicing delete it

EC2 > Instances

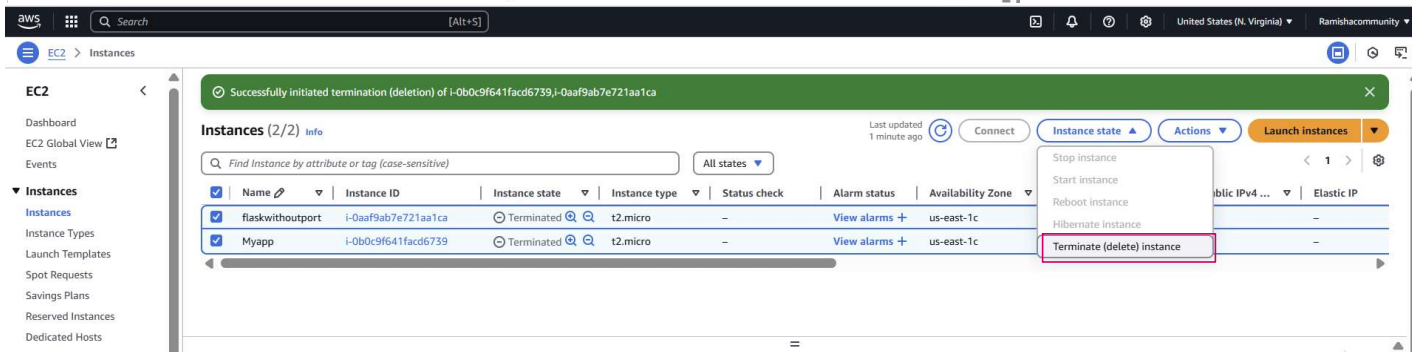
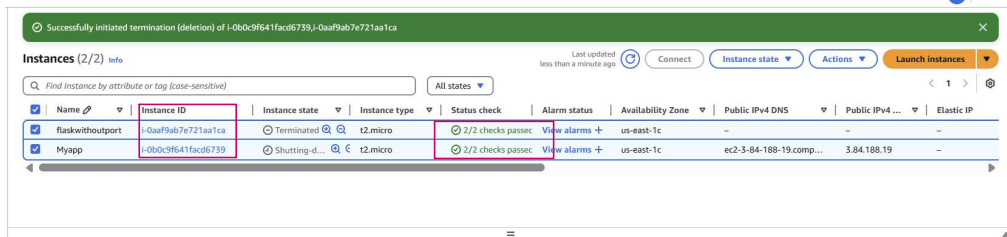
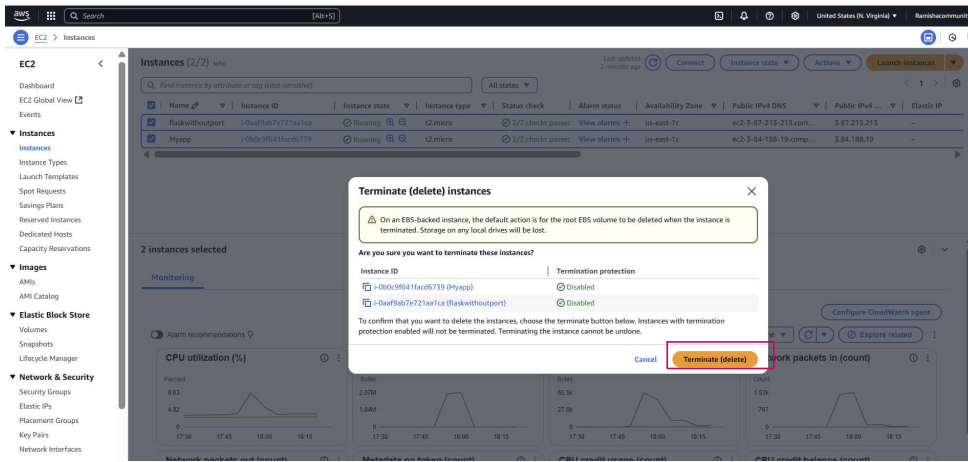
Instances (2/2) info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 ... | Elastic IP |
|------------------|---------------------|----------------|---------------|-----------------|---------------|-------------------|-----------------|------------|
| flaskwiththopurt | i-0aaf9ab7e721aa1ca | Running | t2.micro | 2/2 checks pass | View alarms + | us-east-1c | 37.213.213 | - |
| Myapp | i-0b0c9f641facd6739 | Running | t2.micro | 2/2 checks pass | View alarms + | us-east-1c | 3.84.188.19 | - |

2 instances selected

Instances (2/2) info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 ... | Elastic IP |
|------------------|---------------------|----------------|---------------|-----------------|---------------|-------------------|-----------------|------------|
| flaskwiththopurt | i-0aaf9ab7e721aa1ca | Running | t2.micro | 2/2 checks pass | View alarms + | us-east-1c | 37.213.213 | - |
| Myapp | i-0b0c9f641facd6739 | Running | t2.micro | 2/2 checks pass | View alarms + | us-east-1c | 3.84.188.19 | - |



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.

Chronic Kidney Disease Prediction

Age:

Blood Pressure:

Specific Gravity:

Albumin:

Sugar:

Predict