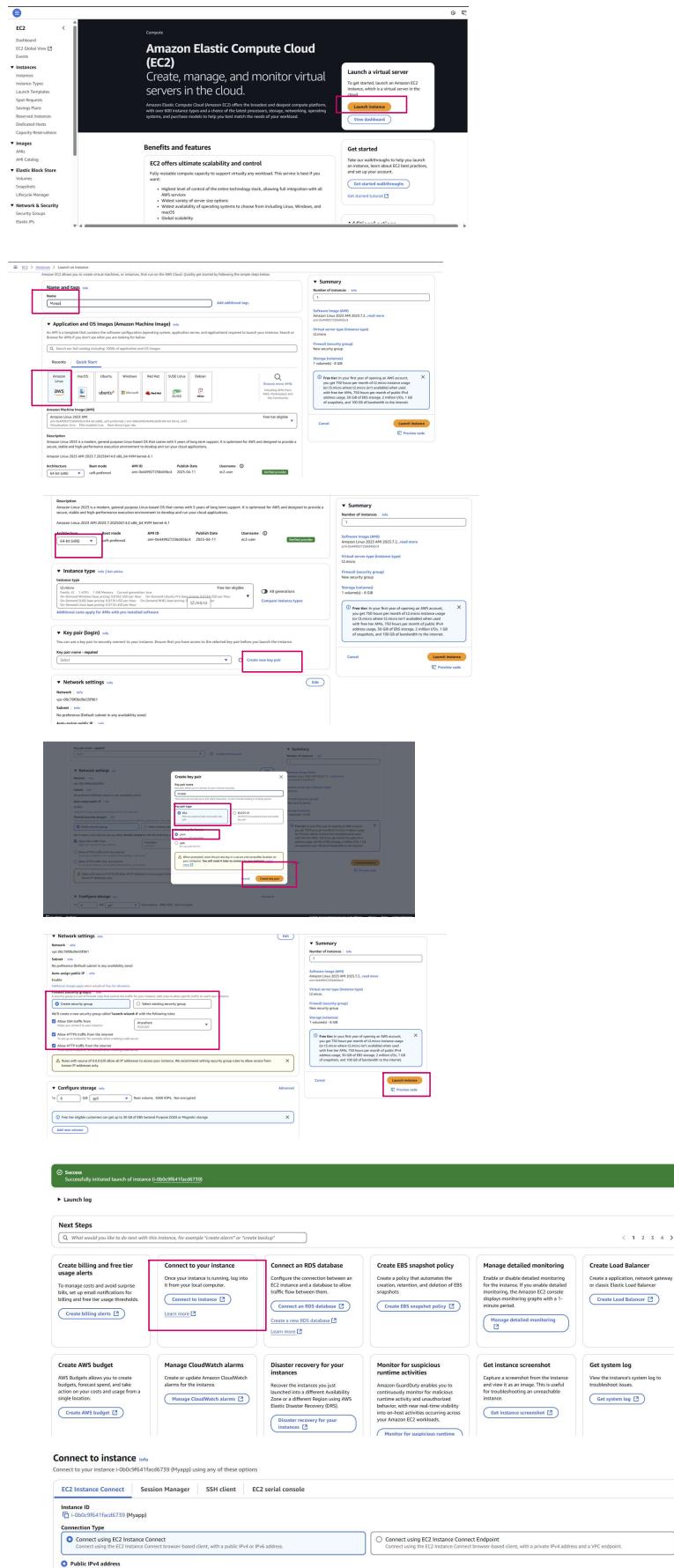
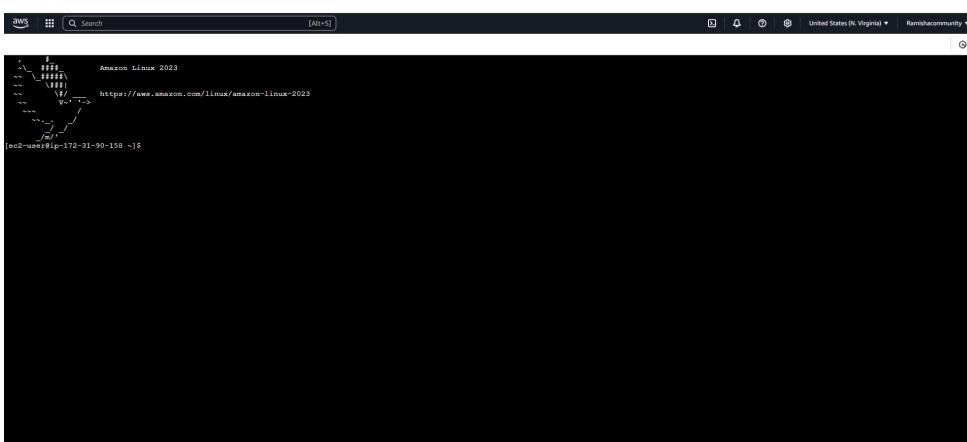
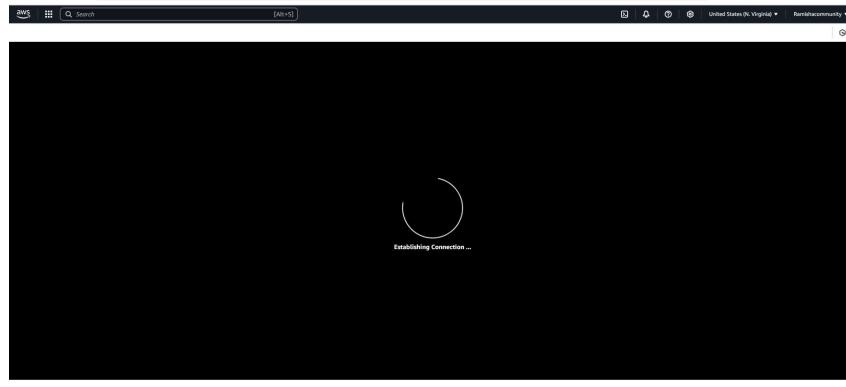
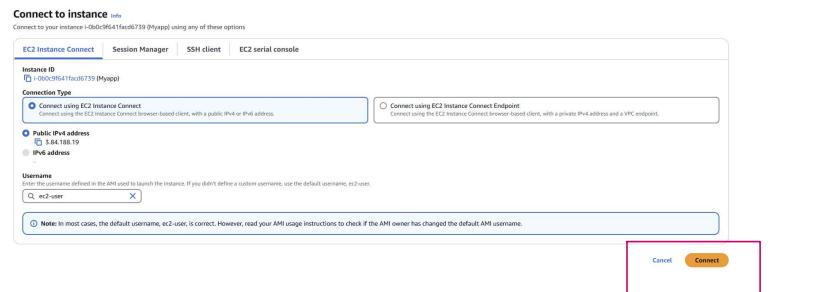


EC2- Hosting without Port

27 April 2025 20:15

Step 1: Go to EC2 Service

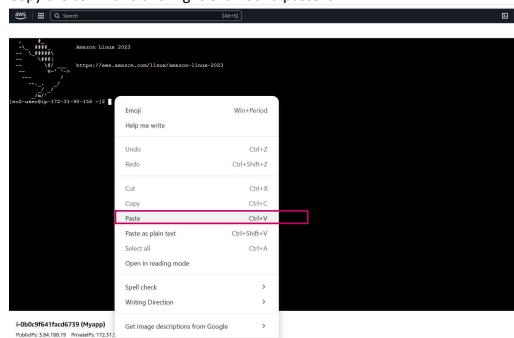




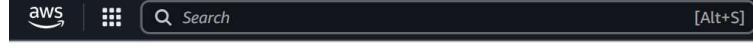
i-0b0c9f641facd6759 (Myapp)
PublicIP: 3.84.188.19 PrivateIP: 172.31.90.158

`sudo yum update -y`

Copy the command and Right Click it and paste it



i-0b0c9f641facd6759 (Myapp)
PublicIP: 3.84.188.19 PrivateIP: 172.31.90.158



```

      #_
      _\ #####
      _\###\ Amazon Linux 2023
      \###|
      \#/ https://aws.amazon.com/linux/amazon-linux-2023
      V_.'__>
      ~~~ /_
      ~~~ /_/
      /m/'_/[ec2-user@ip-172-31-90-158 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-90-158 ~]$ █

```

sudo yum install python3 python3-pip nginx -y

```

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
nginx-1:1.26.3-1.amzn2023.0.1.x86_64                  nginx-core-1:1.26.3-1.amzn2023.0.1.x86_64
python3-pip-21.3.1-2.amzn2023.0.11.noarch

Complete!
[ec2-user@ip-172-31-90-158 ~]$ █

```

sudo pip3 install virtualenv

```

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<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 th
arnings/venv
[ec2-user@ip-172-31-90-158 ~]$ █

```

mkdir myflaskapp

```

cd myflaskapp
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]$ █

```

python3 -m venv venv

source venv/bin/activate

```

WARNING: Running pip as the 'root' user can result in broken permissions at
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]$ █

```

pip install flask unicorn

```

[[root@ip-172-31-90-158 ~]# pip install flask unicorn
  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 package manager. It is recommended to use 'pipenv' instead.
[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 unicorn
Collecting flask
  Downloading flask-3.1.0-py3-none-any.whl (102 kB)
    |██████████| 102 kB 5.2 MB/s
  Collecting unicorn
    Downloading unicorn-23.0.0-py3-none-any.whl (85 kB)
      |██████████| 85 kB 6.5 MB/s
  Collecting blinker>=1.5
    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 itsdangerous>=2.2
    Downloading itsdangerous-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 57.9 MB/s
  Collecting click>=8.1.3
    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-2.0-cp39-cp39-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (20 kB)
  Installing collected packages: zipp, MarkupSafe, Werkzeug, packaging, Jinja2, itsdangerous, importlib-metadata, click, blinker, unicorn, 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 unicorn-23.0.0 importlib-metadata-8.7.0 itsdangerous-2.2.0
WARNING: You are using pip version 21.3.1; however, version 25.1 is available.
You should consider upgrading via the 'python -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.x86_64
  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
  Command: 

```

```

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

```

```

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;
    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/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;

    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;
    ssl_session_timeout 10m;
    ssl_ciphers PROFILE-DEFROST-MEDIUM;
    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/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

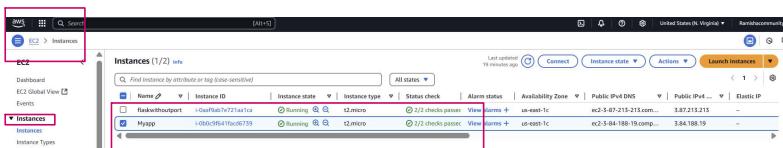
server {
    listen 80;
    server_name your-ec2-ip; # or your domain if you have

    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;
    }
}

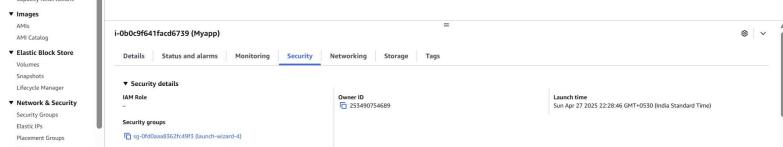
# Settings for a TLS enabled server.

server {
    listen      443 ssl;
    ...
}

```



| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4... | Elastic IP |
|------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|-------------------------|----------------|------------|
| Rdsdb1port | i-0efabfba7c721a1ca | Running | t2.micro | 2/2 checks passed | View alarms | us-east-1c | ec2-3-87-215-213.com... | 3.87.12.215 | - |
| Myapp | i-0b0c9f641fscd6739 | Running | t2.micro | 2/2 checks passed | View alarms | us-east-1c | ec2-3-84-188-19.com... | 3.84.188.19 | - |



| Security details | |
|------------------|--|
| IAM Role | - |
| Security groups | sg-0f0b0e00d522f4991 [Launch wizard-4] |

AWS EC2 Instances

Instances (1/2) list

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4... | Elastic IP |
|---------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|----------------------------|----------------|------------|
| Ranishashah-1 | i-0af9ab7721aa1c1 | Running | t2.micro | 2/2 checks passed | View alarms | us-east-1c | ec2-5-87-113-213.com... | 5.87.113.215 | - |
| Myapp | i-0bd09f641facd6739 | Running | t2.micro | 2/2 checks passed | View alarms | us-east-1c | ec2-5-84-188-19.compute... | 5.84.188.19 | - |

i-0bd09f641facd6739 (Myapp)

- Details
- Status and alarms
- Monitoring
- Security**
- Networking
- Storage
- Tags

Security details

IAM Role

Security groups

Launch time

Instance summary for i-0bd09f641facd6739 (Myapp)

Public IPv4 address

Private IP address

Private IP DNS

Elastic IP addresses

Auto Scaling Group name

AMI ID

Monitoring

Platform details

```
# 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/letsencrypt/live/www.ranishashah.com/fullchain.pem";
    ssl_certificate_key "/etc/letsencrypt/live/www.ranishashah.com/privkey.pem";
}
```

Ctrl+S

Ctrl+X --> exit

```
Complete!
[vnen] [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% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
[vnen] [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nano /etc/nginx/nginx.conf
[vnen] [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
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ sudo systemctl restart nginx
```

sudo nginx -t

```
[vnen] [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
[vnen] [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nano /etc/nginx/nginx.conf
[vnen] [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
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ sudo systemctl restart nginx
```

sudo systemctl restart nginx

```
[vnen] [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.
[vnen] [ec2-user@ip-172-31-90-158 myflaskapp]$ cd flasktesting
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ sudo nano /etc/nginx/nginx.conf
[vnen] [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
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ sudo systemctl restart nginx
```

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

```
unicorn: configuration file /etc/nginx/nginx.conf test is successful
[vnen] [ec2-user@ip-172-31-90-158 flasktesting]$ unicorn -w 3 -b 0.0.0.0:5000 app:app
[2025-04-27 18:16:22 +0000] [30105] [INFO] Starting unicorn 3.0.0
[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: threads
[2025-04-27 18:16:22 +0000] [30106] [INFO] Booting worker with pid: 30107
[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) info

Updated 5 minutes ago

Instance ID i-0b0c9f641facd6739

IPv4 address -

Hostname type IP name: ip-172-31-90-188.ec2.internal

Answer private resource DNS name (ip-172-31-90-188.ec2.internal)

Autogenerated IP address 3.84.188.19 [Public IP]

IAM Role -

IMDv2 Required

Operator -

Details | **Status and alarms** | **Monitoring** | **Security** | **Networking** | **Storage** | **Tags**

Instance details info

AMI ID ami-0e4992725bd45bc4

AMI name a0203-ami-2023.7.20250414.0-kernel-6.1-v86_54

Step protection Disabled

Monitoring disabled

Allowed image -

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

Platform details Linux/UNIX

Termination protection Enabled

AMI location amazon/ami-2023.7.20250414.0-kernel-6.1-v86_54

Public IPv4 address 3.84.188.19

Private IPv4 addresses 172.31.90.188

Public IPv4 DNS ec2-3-84-188-19.compute-1.amazonaws.com [Open address]

Elastic IP addresses

AWS Compute Optimizer Get started with AWS Compute Optimizer for recommendations. [Learn more]

Auto Scaling group name -

Managed false

Connect | **Instance state** | **Actions**

<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](#)

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](#)

3.84.188.19 - <http://3.84.188.19>

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](#)

If you are practicing delete it

EC2 > Instances

Instances (2/2) info

Last updated less than a minute ago

Name **Instance ID** **Instance state** Running Stopped **Instance type** t2.micro **Status check** 2/2 checks passed **Alarm status** View alarms + **Availability Zone** us-east-1c

Stop instance **Start instance** **Reboot instance**

Public IP v4 3.84.188.19 **Elastic IP** -

Terminate (delete) instance

Find Instance by attribute or tag (case-sensitive) **All states**

Instances (2/2) info

Last updated 2 minutes ago

Name **Instance ID** **Instance state** Running Stopped **Instance type** t2.micro **Status check** 2/2 checks passed **Alarm status** View alarms + **Availability Zone** us-east-1c

Stop instance **Start instance** **Reboot instance**

Public IP v4 3.84.188.19 **Elastic IP** -

Terminate (delete) instance

Instances (2/2) info

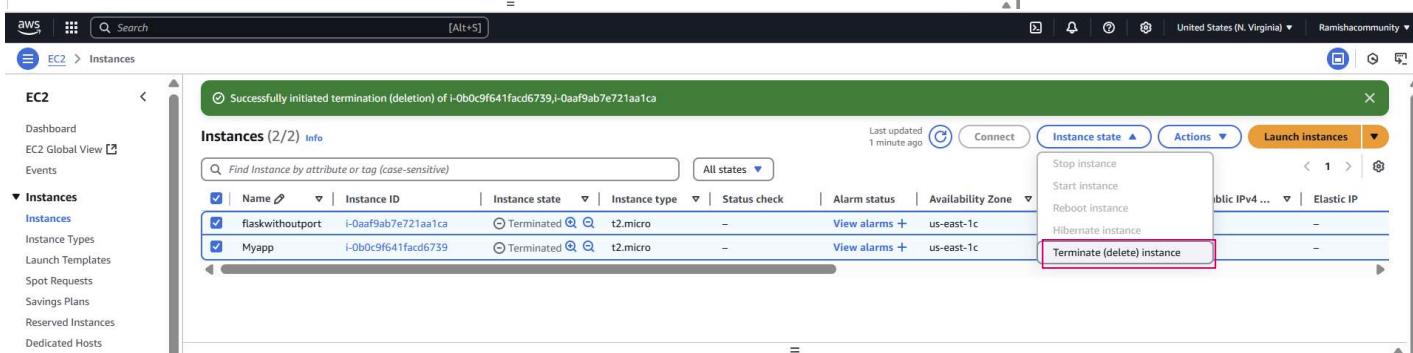
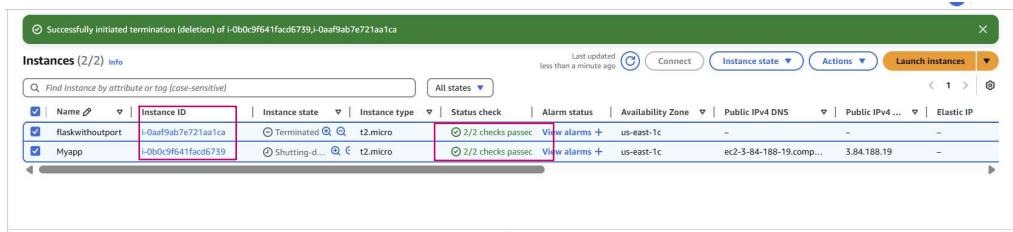
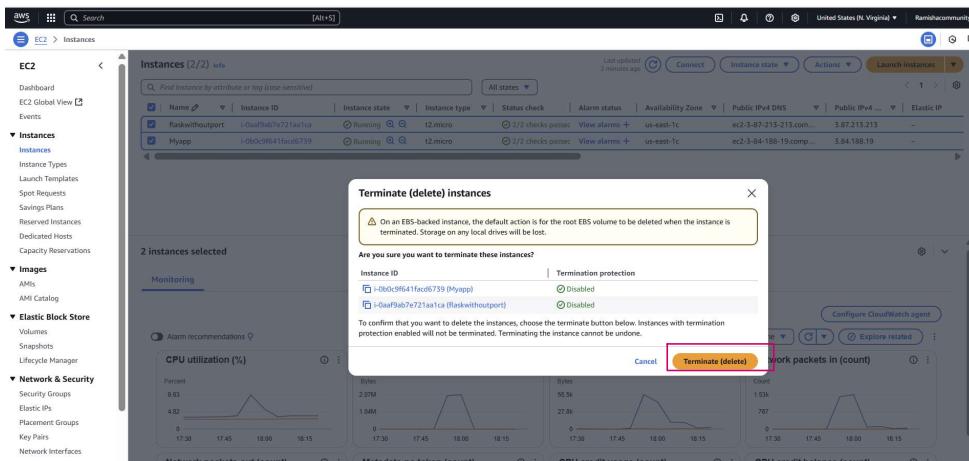
Last updated 2 minutes ago

Name **Instance ID** **Instance state** Running Stopped **Instance type** t2.micro **Status check** 2/2 checks passed **Alarm status** View alarms + **Availability Zone** us-east-1c

Stop instance **Start instance** **Reboot instance**

Public IP v4 3.84.188.19 **Elastic IP** -

Terminate (delete) instance



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 <code>flask.hopelearning.net</code> and point to your EC2 instance | Safe, no impact on main site |
| Use Main Domain | Move entire <code>hopelearning.net</code> 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                                ⌂ Copy ⌂ Edit
<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                                ⌂ Copy ⌂ Edit
[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