**You said:**

create a flask simple appn and then we push code on github after this start deploying appn on aws ec2 - step by step from scratch please - let's start - please step by step from scratch - just a simple flask simple appn

**ChatGPT said:**

Let's go step by step to create a **Flask** app, push the code to **GitHub**, and then deploy it on **AWS EC2**.

**Step 1: Set Up a Flask Application**

We will first create a simple **Flask** app that runs on http://localhost:5000/ and returns "Hello, World!".

**1.1 Install Flask**

Make sure Python is installed. Run:

bash

Copy code

pip install flask

**1.2 Create a Project Directory**

bash

Copy code

mkdir flask-app

cd flask-app

**1.3 Create app.py**

Create a file named app.py inside flask-app directory:

python

Copy code

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return "Hello, World!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

**1.4 Run Flask App Locally**

Run the following command:

bash

Copy code

python app.py

Open http://127.0.0.1:5000/ in your browser. You should see:

Copy code

Hello, World!

**Step 2: Push Code to GitHub**

**2.1 Initialize Git**

bash

Copy code

git init

git add .

git commit -m "Initial commit"

**2.2 Create a GitHub Repository**

1. Go to [GitHub](https://github.com/) and create a new repository (e.g., flask-aws-deploy).
2. Copy the GitHub repository URL.

**2.3 Connect Local Repo to GitHub**

bash

Copy code

git remote add origin <your-repo-url>

git branch -M main

git push -u origin main

✅ **Check GitHub to confirm your code is uploaded.**

**Step 3: Deploy Flask App on AWS EC2**

**3.1 Launch an EC2 Instance**

1. Go to [AWS EC2 Console](https://aws.amazon.com/ec2/).
2. Click **Launch Instance**.
3. Choose **Ubuntu 22.04** as the AMI.
4. Choose **t2.micro** (free tier).
5. Configure Security Group:
   * **Allow HTTP (Port 80)**
   * **Allow SSH (Port 22)**
6. Create and download a key pair (flask-app.pem).

**3.2 Connect to EC2 Instance**

From your local machine, run:

bash

Copy code

ssh -i flask-app.pem ubuntu@<EC2-PUBLIC-IP>

**3.3 Install Dependencies on EC2**

Inside the EC2 instance, run:

bash

Copy code

sudo apt update && sudo apt upgrade -y

sudo apt install python3-pip -y

pip3 install flask gunicorn

**3.4 Clone GitHub Repo**

bash

Copy code

git clone https://github.com/<your-username>/flask-aws-deploy.git

cd flask-aws-deploy

**3.5 Run Flask App using Gunicorn**

bash

Copy code

gunicorn --bind 0.0.0.0:5000 app:app

Your Flask app is now running on EC2's public-ip:5000.

**3.6 Set Up Nginx for Reverse Proxy (Optional)**

If you want to serve the app on port 80:

1. Install **Nginx**:

bash

Copy code

sudo apt install nginx -y

1. Configure **Nginx**:

bash

Copy code

sudo nano /etc/nginx/sites-available/flask

Add the following:

nginx

Copy code

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

1. Enable the configuration:

bash

Copy code

sudo ln -s /etc/nginx/sites-available/flask /etc/nginx/sites-enabled

sudo systemctl restart nginx

Now, visit http://EC2-PUBLIC-IP/, and your Flask app should be live! 🎉

✅ **Next Steps**:  
Would you like to automate this setup using a **systemd service** or a **Docker container**? 🚀

4o

**You said:**

https://github.com/deepanshu1334/flask-aws-deploy

**ChatGPT said:**

I see you've pushed your Flask app to GitHub: [flask-aws-deploy](https://github.com/deepanshu1334/flask-aws-deploy). Now, let's continue deploying it on **AWS EC2** step by step.

**Step 1: Launch an EC2 Instance**

1. Go to [AWS EC2 Console](https://aws.amazon.com/ec2/).
2. Click **Launch Instance** → Name it flask-app.
3. Choose **Ubuntu 22.04** as the AMI.
4. Select **t2.micro** (Free Tier).
5. **Create a Security Group** and allow:
   * **SSH (22)** - Your IP only
   * **HTTP (80)** - Anywhere
   * **Custom TCP (5000)** - Anywhere (For testing Flask before Nginx)
6. **Create a Key Pair** (flask-app.pem) and download it.
7. Click **Launch Instance**.

**Step 2: Connect to EC2 via SSH**

After the instance is running, copy its **Public IP** from the AWS console and run:

bash

Copy code

ssh -i flask-app.pem ubuntu@<EC2-PUBLIC-IP>

**Step 3: Install Dependencies on EC2**

Run the following commands on your EC2 instance:

bash

Copy code

sudo apt update && sudo apt upgrade -y

sudo apt install python3-pip git nginx -y

pip3 install flask gunicorn

**Step 4: Clone Your Flask App from GitHub**

Run:

bash

Copy code

git clone https://github.com/deepanshu1334/flask-aws-deploy.git

cd flask-aws-deploy

**Step 5: Run Flask App Using Gunicorn**

Run:

bash

Copy code

gunicorn --bind 0.0.0.0:5000 app:app

Your Flask app should now be running. Test it in your browser:

cpp

Copy code

http://<EC2-PUBLIC-IP>:5000

**Step 6: Set Up Nginx as a Reverse Proxy**

1. Create an Nginx config file:

bash

Copy code

sudo nano /etc/nginx/sites-available/flask

1. Add the following:

nginx

Copy code

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

1. Save & close (CTRL + X, then Y, then ENTER).
2. Enable the configuration:

bash

Copy code

sudo ln -s /etc/nginx/sites-available/flask /etc/nginx/sites-enabled

sudo systemctl restart nginx

**Step 7: Run Flask App as a Background Service**

1. Create a systemd service:

bash

Copy code

sudo nano /etc/systemd/system/flask-app.service

1. Add:

ini

Copy code

[Unit]

Description=Gunicorn instance to serve Flask app

After=network.target

[Service]

User=ubuntu

WorkingDirectory=/home/ubuntu/flask-aws-deploy

ExecStart=/usr/local/bin/gunicorn --workers 3 --bind unix:flask-app.sock -m 007 app:app

[Install]

WantedBy=multi-user.target

1. Save & close (CTRL + X, then Y, then ENTER).
2. Start the service:

bash

Copy code

sudo systemctl start flask-app

sudo systemctl enable flask-app

**Step 8: Access Your Flask App on EC2**

Now, visit:

cpp

Copy code

http://<EC2-PUBLIC-IP>/

You should see **"Hello, World!"** from your Flask app. 🎉

✅ **Next Steps:** Do you want to add **HTTPS (SSL with Let's Encrypt)** for security? 🚀

4o

**You said:**

User sessions running outdated binaries: ubuntu @ session #5: sshd[3459] ubuntu @ user manager service: systemd[1383] No VM guests are running outdated hypervisor (qemu) binaries on this host. ubuntu@ip-172-31-2-80:~$ pip3 install flask gunicorn error: externally-managed-environment × This environment is externally managed ╰─> To install Python packages system-wide, try apt install python3-xyz, where xyz is the package you are trying to install. If you wish to install a non-Debian-packaged Python package, create a virtual environment using python3 -m venv path/to/venv. Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make sure you have python3-full installed. If you wish to install a non-Debian packaged Python application, it may be easiest to use pipx install xyz, which will manage a virtual environment for you. Make sure you have pipx installed. See /usr/share/doc/python3.12/README.venv for more information. note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages. hint: See PEP 668 for the detailed specification.

**ChatGPT said:**

You're facing the **"externally-managed-environment"** issue because **Ubuntu 22.04+** enforces Python package management best practices to avoid breaking system-wide dependencies.

To fix this, follow **one of the two solutions below**:

**✅ Solution 1: Use a Virtual Environment (Recommended)**

1. **Install Python venv if not installed**:

bash

Copy code

sudo apt install python3-venv -y

1. **Create a virtual environment**:

bash

Copy code

python3 -m venv venv

1. **Activate the virtual environment**:

bash

Copy code

source venv/bin/activate

1. **Install Flask and Gunicorn inside the virtual environment**:

bash

Copy code

pip install flask gunicorn

1. **Run your Flask app**:

bash

Copy code

gunicorn --bind 0.0.0.0:5000 app:app

1. If you close the terminal, reactivate the virtual environment before running the app again:

bash

Copy code

source venv/bin/activate

**✅ Solution 2: Use --break-system-packages (Not Recommended)**

If you don’t want to use a virtual environment, you can **force** the installation system-wide (**not recommended** as it may break dependencies):

bash

Copy code

pip3 install flask gunicorn --break-system-packages

But **Solution 1** (virtual environment) is the best practice.

Let me know if you need further clarification! 🚀

4o

**You said:**

i-08647aaf897aa24b6 (FLASK-APP) PublicIPs: 52.66.240.194 PrivateIPs: 172.31.2.80

**ChatGPT said:**

Your EC2 instance **(FLASK-APP)** is running with:

* **Public IP:** 52.66.240.194
* **Private IP:** 172.31.2.80

Since you've set up the instance, let's continue:

**✅ Step 1: Connect to EC2 via SSH**

Run this on your local machine:

bash

Copy code

ssh -i flask-app.pem ubuntu@52.66.240.194

(Replace flask-app.pem with the path to your key file.)

**✅ Step 2: Set Up the Flask App**

**1. Install Required Packages**

bash

Copy code

sudo apt update && sudo apt upgrade -y

sudo apt install python3-venv python3-pip git nginx -y

**2. Clone Your GitHub Repository**

bash

Copy code

git clone https://github.com/deepanshu1334/flask-aws-deploy.git

cd flask-aws-deploy

**3. Create a Virtual Environment & Install Dependencies**

bash

Copy code

python3 -m venv venv

source venv/bin/activate

pip install flask gunicorn

**4. Test Your Flask App**

Run:

bash

Copy code

gunicorn --bind 0.0.0.0:5000 app:app

Now visit:

cpp

Copy code

http://52.66.240.194:5000

If you see **"Hello, World!"**, your app is working! 🚀

**✅ Step 3: Set Up Nginx as Reverse Proxy**

1. **Create an Nginx config file:**

bash

Copy code

sudo nano /etc/nginx/sites-available/flask

1. **Add this configuration:**

nginx

Copy code

server {

listen 80;

server\_name 52.66.240.194;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

1. **Enable the Nginx configuration:**

bash

Copy code

sudo ln -s /etc/nginx/sites-available/flask /etc/nginx/sites-enabled

sudo systemctl restart nginx

**✅ Step 4: Run Flask as a Background Service**

1. **Create a Systemd Service:**

bash

Copy code

sudo nano /etc/systemd/system/flask-app.service

1. **Add this content:**

ini

Copy code

[Unit]

Description=Gunicorn instance to serve Flask app

After=network.target

[Service]

User=ubuntu

WorkingDirectory=/home/ubuntu/flask-aws-deploy

ExecStart=/home/ubuntu/flask-aws-deploy/venv/bin/gunicorn --workers 3 --bind unix:flask-app.sock -m 007 app:app

[Install]

WantedBy=multi-user.target

1. **Enable and start the service:**

bash

Copy code

sudo systemctl start flask-app

sudo systemctl enable flask-app

**✅ Step 5: Access Your App on the Public IP**

Now, open:

cpp

Copy code

http://52.66.240.194

Your Flask app should be live! 🎉

**🚀 Next Steps**

* Do you want to **add HTTPS (SSL)** with Let’s Encrypt for security?
* Need **automatic deployment from GitHub** (CI/CD)?

Let me know what you need next! 🚀

4o

**You said:**

**ChatGPT said:**

Your **502 Bad Gateway** error likely means that Nginx cannot connect to your Flask app running via Gunicorn. Let's debug step by step.

**✅ Step 1: Check if Gunicorn is Running**

Run this command:

bash

Copy code

sudo systemctl status flask-app

If you see **inactive (dead)** or an error, restart it:

bash

Copy code

sudo systemctl restart flask-app

sudo systemctl enable flask-app

Then check logs:

bash

Copy code

sudo journalctl -u flask-app --no-pager --lines=50

**✅ Step 2: Check Flask/Gunicorn Logs**

Run:

bash

Copy code

cat /var/log/syslog | grep gunicorn

Look for errors like **port binding issues** or **Python-related errors**.

**✅ Step 3: Manually Start Gunicorn (Test)**

Run:

bash

Copy code

cd /home/ubuntu/flask-aws-deploy

source venv/bin/activate

gunicorn --bind 0.0.0.0:5000 app:app

Now, try opening http://52.66.240.194:5000 in your browser.  
If this works, the issue is **with the systemd service**.

**✅ Step 4: Fix Nginx Configuration**

Check your Nginx config:

bash

Copy code

sudo nano /etc/nginx/sites-available/flask

It should look like this:

nginx

Copy code

server {

listen 80;

server\_name 52.66.240.194;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

Save and restart:

bash

Copy code

sudo nginx -t

sudo systemctl restart nginx

**✅ Step 5: Open Required Ports**

Ensure your EC2 **Security Group** allows traffic on ports:

* **80 (HTTP)**
* **5000 (Flask)** Run:

bash

Copy code

sudo ufw allow 80

sudo ufw allow 5000

sudo ufw reload

**🚀 Try Again!**

Now visit http://52.66.240.194  
Let me know if it's still not working! 🚀

4o

**You said:**

(venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$ sudo systemctl status flask-app × flask-app.service - Gunicorn instance to serve Flask app Loaded: loaded (/etc/systemd/system/flask-app.service; enabled; preset: enabled) Active: failed (Result: exit-code) since Thu 2025-03-20 16:46:33 UTC; 1min 17s ago Duration: 2ms Main PID: 22170 (code=exited, status=203/EXEC) CPU: 925us Mar 20 16:46:33 ip-172-31-2-80 systemd[1]: Started flask-app.service - Gunicorn instance to serve Flask app. Mar 20 16:46:33 ip-172-31-2-80 systemd[1]: flask-app.service: Main process exited, code=exited, status=203/EXEC Mar 20 16:46:33 ip-172-31-2-80 systemd[1]: flask-app.service: Failed with result 'exit-code'. (venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$

**ChatGPT said:**

Your **flask-app.service** is failing with **status=203/EXEC**, which usually means:

1. The executable path in your **systemd service file** is incorrect.
2. The virtual environment isn't being used correctly.
3. The file permissions may be wrong.

**✅ Step 1: Check Systemd Service File**

Run:

bash

Copy code

sudo nano /etc/systemd/system/flask-app.service

Ensure it looks like this:

ini

Copy code

[Unit]

Description=Gunicorn instance to serve Flask app

After=network.target

[Service]

User=ubuntu

Group=www-data

WorkingDirectory=/home/ubuntu/flask-aws-deploy

Environment="PATH=/home/ubuntu/flask-aws-deploy/venv/bin"

ExecStart=/home/ubuntu/flask-aws-deploy/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

[Install]

WantedBy=multi-user.target

Save and exit (CTRL + X, then Y, then ENTER).

**✅ Step 2: Reload and Restart**

bash

Copy code

sudo systemctl daemon-reload

sudo systemctl restart flask-app

sudo systemctl status flask-app

If it **fails again**, check logs:

bash

Copy code

journalctl -u flask-app --no-pager --lines=50

**✅ Step 3: Check Gunicorn in Virtual Environment**

Try running it manually:

bash

Copy code

cd /home/ubuntu/flask-aws-deploy

source venv/bin/activate

gunicorn --bind 0.0.0.0:5000 app:app

If this works, the problem is with **systemd**.  
If not, check for missing dependencies:

bash

Copy code

pip install -r requirements.txt

**🚀 Try these steps and let me know what happens!**

4o

**You said:**

Mar 20 16:46:33 ip-172-31-2-80 systemd[1]: Started flask-app.service - Gunicorn instance to serve Flask app. Mar 20 16:46:33 ip-172-31-2-80 systemd[1]: flask-app.service: Main process exited, code=exited, status=203/EXEC Mar 20 16:46:33 ip-172-31-2-80 systemd[1]: flask-app.service: Failed with result 'exit-code'. (venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$ sudo nano /etc/systemd/system/flask-app.service (venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$ sudo nano /etc/systemd/system/flask-app.service (venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$ sudo systemctl daemon-reload sudo systemctl restart flask-app sudo systemctl status flask-app × flask-app.service - Gunicorn instance to serve Flask app Loaded: loaded (/etc/systemd/system/flask-app.service; enabled; preset: enabled) Active: failed (Result: exit-code) since Thu 2025-03-20 16:49:37 UTC; 16ms ago Duration: 7ms Process: 22276 ExecStart=/home/ubuntu/flask-aws-deploy/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app (code=exited, status=203/EXEC) Main PID: 22276 (code=exited, status=203/EXEC) CPU: 2ms Mar 20 16:49:37 ip-172-31-2-80 systemd[1]: Started flask-app.service - Gunicorn instance to serve Flask app. Mar 20 16:49:37 ip-172-31-2-80 systemd[1]: flask-app.service: Main process exited, code=exited, status=203/EXEC Mar 20 16:49:37 ip-172-31-2-80 systemd[1]: flask-app.service: Failed with result 'exit-code'.

**ChatGPT said:**

The **status=203/EXEC** error means **systemd can't execute Gunicorn**. Let's debug step by step.

**✅ Step 1: Verify Gunicorn Path**

Run:

bash

Copy code

which gunicorn

If it returns something like:

swift

Copy code

/home/ubuntu/flask-aws-deploy/venv/bin/gunicorn

Then your **ExecStart** path in the service file is correct. If not, update it.

**✅ Step 2: Check If Gunicorn is Executable**

Run:

bash

Copy code

ls -l /home/ubuntu/flask-aws-deploy/venv/bin/gunicorn

Expected output:

swift

Copy code

-rwxr-xr-x 1 ubuntu ubuntu 234234 Mar 20 12:34 /home/ubuntu/flask-aws-deploy/venv/bin/gunicorn

If you see **"-rw-r--r--"**, it’s not executable. Fix it with:

bash

Copy code

chmod +x /home/ubuntu/flask-aws-deploy/venv/bin/gunicorn

**✅ Step 3: Manually Run Gunicorn**

Try running Gunicorn manually:

bash

Copy code

cd /home/ubuntu/flask-aws-deploy

source venv/bin/activate

gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

* If it **fails**, check the error message.
* If it **works**, the issue is with **systemd**.

**✅ Step 4: Check Systemd Service File**

Run:

bash

Copy code

sudo nano /etc/systemd/system/flask-app.service

Ensure it looks like:

ini

Copy code

[Unit]

Description=Gunicorn instance to serve Flask app

After=network.target

[Service]

User=ubuntu

Group=www-data

WorkingDirectory=/home/ubuntu/flask-aws-deploy

Environment="PATH=/home/ubuntu/flask-aws-deploy/venv/bin"

ExecStart=/home/ubuntu/flask-aws-deploy/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

[Install]

WantedBy=multi-user.target

**Save & Exit (CTRL + X, then Y, then ENTER).**

**✅ Step 5: Reload and Restart**

bash

Copy code

sudo systemctl daemon-reload

sudo systemctl restart flask-app

sudo systemctl status flask-app

**🔍 Still Failing? Check Logs**

Run:

bash

Copy code

journalctl -u flask-app --no-pager --lines=50

This will show **detailed errors**.

Try these steps and let me know what you find! 🚀

4o

**You said:**

(venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$ which gunicorn /home/ubuntu/venv/bin/gunicorn

**ChatGPT said:**

Your **Gunicorn path** is /home/ubuntu/venv/bin/gunicorn, but your **systemd service file** is using:

ruby

Copy code

ExecStart=/home/ubuntu/flask-aws-deploy/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

**✅ Fix the service file**

Run:

bash

Copy code

sudo nano /etc/systemd/system/flask-app.service

Update this line:

bash

Copy code

ExecStart=/home/ubuntu/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

Save & Exit (CTRL + X, then Y, then ENTER).

**✅ Reload and Restart**

Run these commands to apply the changes:

bash

Copy code

sudo systemctl daemon-reload

sudo systemctl restart flask-app

sudo systemctl status flask-app

This should fix the issue! 🚀 Let me know if you still face any errors.

4o

**You said:**

(venv) ubuntu@ip-172-31-2-80:~/flask-aws-deploy$ sudo systemctl daemon-reload sudo systemctl restart flask-app sudo systemctl status flask-app ● flask-app.service - Gunicorn instance to serve Flask app Loaded: loaded (/etc/systemd/system/flask-app.service; enabled; preset: enabled) Active: active (running) since Thu 2025-03-20 16:52:21 UTC; 30ms ago Main PID: 22339 (gunicorn) Tasks: 1 (limit: 1130) Memory: 2.6M (peak: 2.6M) CPU: 11ms CGroup: /system.slice/flask-app.service └─22339 /home/ubuntu/venv/bin/python3 /home/ubuntu/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app Mar 20 16:52:21 ip-172-31-2-80 systemd[1]: Started flask-app.service - Gunicorn instance to serve Flask app.

**ChatGPT said:**

Great! 🎉 Your Flask app is now running successfully with Gunicorn. 🚀

You can verify it's working by running:

bash

Copy code

curl http://127.0.0.1:5000

or

bash

Copy code

curl http://your-server-public-ip:5000

If you're using **AWS EC2**, ensure that:  
✅ Port 5000 is open in the **Security Groups**.  
✅ The Flask app is binding to 0.0.0.0, which it is.

Let me know if you need further help! 🚀🔥

4o

**You said:**

i-08647aaf897aa24b6 (FLASK-APP) PublicIPs: 52.66.240.194 PrivateIPs: 172.31.2.80

**ChatGPT said:**

Your Flask app is running on **AWS EC2** with the public IP **52.66.240.194**.

To access it from your browser, try:  
👉 **http://52.66.240.194:5000**

If it doesn't work, check:  
✅ **Security Group Rules:** Ensure inbound rules allow traffic on **port 5000** (HTTP, source: 0.0.0.0/0).  
✅ **Gunicorn Logs:** Run sudo journalctl -u flask-app --no-pager --lines=50 for error messages.  
✅ **Firewall Rules:** Check if ufw (Uncomplicated Firewall) is blocking the port:

bash

Copy code

sudo ufw allow 5000

sudo ufw status

✅ **Gunicorn Bind Address:** Your Gunicorn command looks correct:

bash

Copy code

/home/ubuntu/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:5000 app:app

If you still face issues, try binding to port **80**:

bash

Copy code

/home/ubuntu/venv/bin/gunicorn --workers 3 --bind 0.0.0.0:80 app:app

(You might need sudo for port 80.)

Let me know what errors you see if it’s still not accessible! 🚀