To **install and run a Flask application** on **Ubuntu**, follow these steps:

**1. Update & Install Dependencies**

First, update your package list and install required dependencies:

sudo apt update && sudo apt upgrade -y

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

**2. Set Up Your Flask Application**

* Navigate to your project directory or create a new one:
  + mkdir flask\_app && cd flask\_app
* Create a virtual environment:
  + python3 -m venv venv
* Activate the virtual environment
  + source venv/bin/activate
* Install Flask and Gunicorn:
  + pip install flask gunicorn

**3. Create a Flask App**

I have app.py with below sample structure:

from flask import Flask

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

@app.route('/')

def home():

return "Hello, Flask is running on Ubuntu!"

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

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

This needs to be added as provided by IT team.

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

**4. Run Flask Locally**

To test if Flask runs on your local Ubuntu machine:

python app.py

Your app will be available at **http://127.0.0.1:5000/**.

**5. Deploy Flask with Gunicorn**

Instead of running with python app.py, use Gunicorn for production:

gunicorn -w 4 -b 0.0.0.0:8000 app:app

* -w 4 → Uses 4 worker processes
* -b 0.0.0.0:8000 → Runs on port 8000

Your app should now be accessible at **http://your-server-ip:8000/**.

**6. Set Up Nginx for Reverse Proxy (Optional, Recommended for Production)**

If you want to serve your Flask app using **Nginx**, install it:

sudo apt install nginx -y

Create a new Nginx configuration:

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

Add the following configuration:

server {

listen 80;

server\_name your\_server\_ip;

location / {

proxy\_pass http://127.0.0.1:8000;

proxy\_set\_header Host $host;

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

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

Enable the configuration:

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

sudo nginx -t # Check for syntax errors

sudo systemctl restart nginx # Restart Nginx

Now, your Flask app should be live at **http://your-server-ip/**

**7. Use Supervisor to Keep Flask Running**

If you want your Flask app to **restart automatically** in case of failure:

sudo apt install supervisor -y

Create a Supervisor configuration file:

sudo nano /etc/supervisor/conf.d/flask\_app.conf

Add:

[program:flask\_app]

command=/home/ubuntu/flask\_app/venv/bin/gunicorn -w 4 -b 0.0.0.0:8000 app:app

directory=/home/ubuntu/flask\_app

autostart=true

autorestart=true

stderr\_logfile=/var/log/flask\_app.err.log

stdout\_logfile=/var/log/flask\_app.out.log

Restart Supervisor:

sudo supervisorctl reread

sudo supervisorctl update

sudo supervisorctl start flask\_app

Now your Flask app **runs automatically**, even after a system reboot.

**8. Check Your Flask App**

Go to **http://your-server-ip/** or **http://your-domain/**, and you should see your Flask app running!