# sudo systemctl restart Sentiment-Analysis

# sudo lsof -i :5001

# sudo kill -9 <PID>

#python /root/Sentiment-Analysis/app.py

# sudo systemctl daemon-reload

# sudo systemctl start Sentiment-Analysis

# sudo systemctl enable sentiment-analysis

Perfect! Since you're using **GitHub for hosting code** and **Hostinger VPS for deployment**, here's the **complete step-by-step guide** to:

* **Clone your Flask project from GitHub to Hostinger VPS**
* **Set up and run it using Nginx + virtual environment**

Let’s go step by step.

**✅ STEP 1: Connect to Your VPS Using PuTTY**

* **Open PuTTY**
* Enter your VPS **Hostname/IP**
* **Port**: 22
* **Username**: root
* **Password**: [enter when prompted]

**✅ STEP 2: Install Git (If Not Installed)**

sudo yum install git -y # For CentOS/RHEL

# OR

sudo apt install git -y # For Ubuntu/Debian

**✅ STEP 3: Clone Your GitHub Repository**

cd /root/

git clone https://github.com/YourUsername/YourRepo.git Final

📝 *This will clone your repo into the /root/Final folder.*

**✅ STEP 4: Create and Activate Virtual Environment**

cd /root/Final

python3 -m venv venv

source venv/bin/activate

📝 *Creates and activates the Python virtual environment.*

**✅ STEP 5: Install Required Python Packages**

pip install -r requirements.txt

📝 *Installs dependencies listed in your project.*

**✅ STEP 6: Test Your Flask App**

python app.py

# or use:

flask run --host=0.0.0.0 --port=5001

🌐 Test on browser: http://your-vps-ip:5001

**✅ STEP 7: Set Up Gunicorn (Recommended with Nginx)**

pip install gunicorn

gunicorn --bind 127.0.0.1:8000 wsgi:app

📝 *Assumes you have a wsgi.py file with app = Flask(\_\_name\_\_). Adjust if needed.*

**✅ STEP 8: Configure NGINX**

**Create config file:**

sudo nano /etc/nginx/conf.d/Final.conf

**Paste this sample config (update domain/IP and paths):**

server {

listen 80;

server\_name yourdomain.com; # Or use your VPS IP

location / {

proxy\_pass http://127.0.0.1:8000;

proxy\_set\_header Host $host;

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

}

error\_log /var/log/nginx/final\_error.log;

access\_log /var/log/nginx/final\_access.log;

}

**✅ STEP 9: Test and Restart NGINX**

sudo nginx -t # Test config

sudo systemctl restart nginx

sudo systemctl status nginx

**✅ STEP 10: Allow Firewall Access (Optional)**

sudo firewall-cmd --permanent --add-service=http

sudo firewall-cmd --reload

**✅ STEP 11: Auto-Start Gunicorn (Optional)**

Create a Gunicorn systemd service:

sudo nano /etc/systemd/system/final.service

Paste this:

[Unit]

Description=Gunicorn instance to serve Final

After=network.target

[Service]

User=root

Group=root

WorkingDirectory=/root/Final

Environment="PATH=/root/Final/venv/bin"

ExecStart=/root/Final/venv/bin/gunicorn --workers 3 --bind 127.0.0.1:8000 wsgi:app

[Install]

WantedBy=multi-user.target

Enable and start:

sudo systemctl start final

sudo systemctl enable final

**✅ STEP 12: Deploy Future Changes from GitHub**

cd /root/Final

git pull origin main # Or your branch

source venv/bin/activate

pip install -r requirements.txt

sudo systemctl restart final

Would you like me to generate a sample wsgi.py file for you too?