**How To Install Flask on Ubuntu 18.04 with uWSGI, Nginx on Google Cloud**

### **Introduction**

How To Install Flask on Ubuntu 18.04 with uWSGI, Nginx on Google Cloud

Flask is an open source micro framework for Python.

In this guide, I will demonstrate how to install and configure Flask, uWSGI and Nginx on Ubuntu 18.04 LTS.

### **Prerequisites**

1. Your Compute Engine Instance running, see the [Setting up Compute Engine Instance](https://www.cloudbooklet.com/setting-up-google-cloud-compute-engine-instance/).
2. Domain name is pointed to your virtual machine.
3. For setting up Cloud DNS, see the [Setting up Google Cloud DNS for your domain](https://www.cloudbooklet.com/setting-up-google-cloud-dns-for-your-domain/).

### **Install required packages**

SSH to your Compute Engine instance and begin typing the following commands to start installing Flask

sudo apt update

sudo apt install python3-pip python3-dev build-essential libssl-dev libffi-dev python3-setuptools nginx curl

### **Creating a Python Virtual Environment for Flask**

sudo -H pip3 install --upgrade pip

sudo -H pip3 install virtualenv

mkdir ~/myprojectdir

cd ~/myprojectdir

virtualenv myprojectenv

Activate the virtual environment by typing

source myprojectenv/bin/activate

Your prompt should change to indicate that you are now operating within a Python virtual environment. It will look something like this:

(myprojectenv)username@host:~/myprojectdir$

### **Setup Flask**

pip install wheel

pip install uwsgi flask

Create a Sample Flask App

sudo nano ~/myprojectdir/myproject.py

Paste the following

from flask import Flask

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

@app.route("/")

def hello():

return "<h1 style='color:blue;text-align:center;margin:200px auto;border:1px solid blue;padding:10px 20px;width:300px;'>Welcome to Flask!</h1>"

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0')

Hit Ctrl+X followed by Y to save the file

deactivate

### Create Socket and Service files for uWSGI

sudo nano ~/myprojectdir/wsgi.py

Now import the Flask instance from your application

from myproject import app

if \_\_name\_\_ == "\_\_main\_\_":

app.run()

Create a uWSGI configuration file for long-term usage

sudo nano ~/myprojectdir/myproject.ini

[uwsgi]

module = wsgi:app

master = true

processes = 5

socket = myproject.sock

chmod-socket = 660

vacuum = true

die-on-term = true

Now create the systemd service unit file to automatically start uWSGI and serve the Flask application whenever the instance boots.

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

[Unit]

Description=uWSGI instance to serve myproject

After=network.target

[Service]

User=username

Group=username

WorkingDirectory=/home/username/myprojectdir

Environment="PATH=/home/username/myprojectdir/myprojectenv/bin"

ExecStart=/home/username/myprojectdir/myprojectenv/bin/uwsgi --ini myproject.ini

[Install]

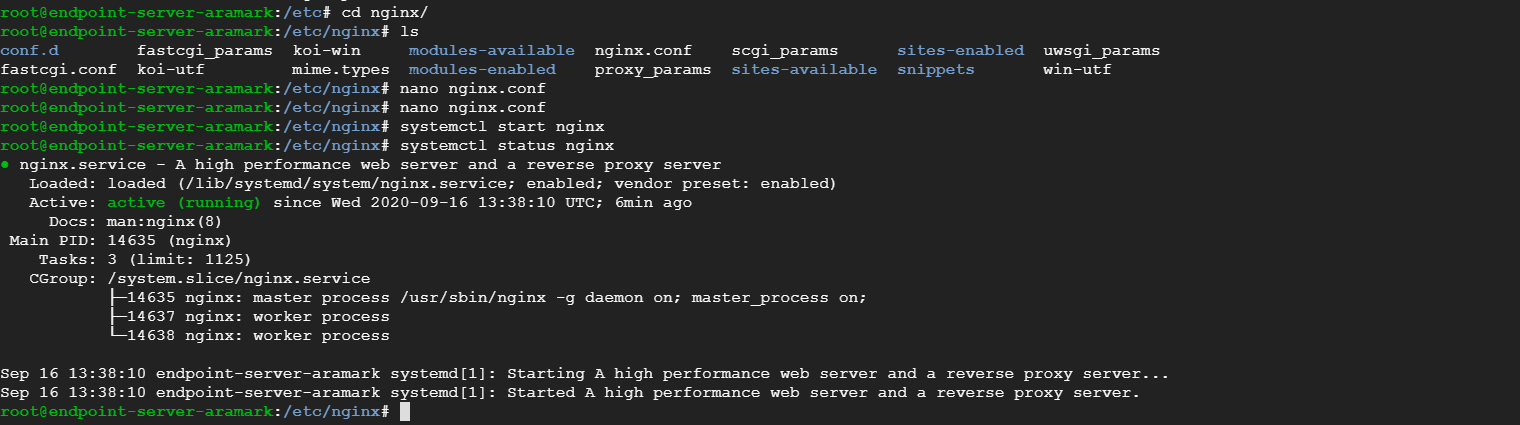
WantedBy=multi-user.target

Start and enable your configuration

sudo systemctl start myproject

sudo systemctl enable myproject

sudo systemctl status myproject



### NGINX Proxy Pass to uWSGI and setup HTTPS

Edit your nginx.conf and replace user www-data with user username

Create a new Nginx configuration for your website in the sites-available directory

sudo nano /etc/nginx/sites-available/yourdomainname.com

Copy and paste the following configuration, ensure that you change the server\_name, error\_log to match your domain name. Hit CTRL+X followed by Y to save the changes.

server {

listen 80;

listen [::]:80;

server\_name yourdomainname.com www.yourdomainname.com;

location = /favicon.ico {

access\_log off;

log\_not\_found off;

}

location / {

include uwsgi\_params;

uwsgi\_pass unix:/home/username/myprojectdir/myproject.sock;

}

}

To enable this newly created website configuration, symlink the file that you just created into the sites-enabled directory.

sudo ln -s /etc/nginx/sites-available/yourdomainname.com /etc/nginx/sites-enabled/yourdomainname.com

Check your configuration and restart Nginx for the changes to take effect

sudo nginx -t

sudo service nginx restart

Now visit your domain name in your web browser, you can view the Flask Sample page you have created.

