The goal of this document is to provide a step-by-step tutorial on linking uWSGI, nginx, and Bottle. Bottle is a web framework written in Python which can be run in production-scale environments through the utilization of nginx and uWSGI.

This tutorial is written from the perspective of a virtual machine running centOS 7, and assumes the user already has an nginx.conf and uwsgi.ini file inside their Git repository. The user should already have Python 2.7 installed. If this is not the case, the user should do so themselves.

It is worth mentioning that, in the case of a VM which has many Python projects running, a Python virtualenv may be useful. This tutorial, however, does not specify how to implement such an environment.

Finally, some steps of this tutorial involve going to the IP of the VM in the web browser. In lieu of a real IP and username, x.x.x.x and %user% will be used, respectively.

# Installation/Preparing VM

To prepare your VM, enter the following commands:

$ sudo yum install epel-release

$ sudo yum install python-pip

$ sudo yum install git

$ git clone https://github.com/Domain-Connect/python-dc-statelesshosting

$ sudo yum install gcc

$ sudo pip install bottle dnspython pycrypto

$ sudo yum erase httpd http-tools apr apr-util[[1]](#footnote-1)

$ sudo yum install nginx

$ sudo yum install uwsgi  
$ sudo yum install uwsgi-plugin-python

# Test Applications

It is important to ensure that Bottle, uWSGI, and nginx all work properly. The repository is in the /home directory.

$ cd python-dc-statelesshosting

$ sudo python statelesshosting.py[[2]](#footnote-2)

Go to x.x.x.x in your web browser and ensure that the site is running. Enter Ctrl + C to kill the Bottle app.

$ uwsgi --ini uwsgi.ini &

This command will start uWSGI, setting up a Unix socket and specifying the app nginx will connect to. Make sure there are no errors. uWSGI will run in the background. Enter the next command.

$ sudo nginx -c ~/python-dc-statelesshosting/nginx.conf

This will launch the nginx server. Go to x.x.x.x and make sure the site is running on nginx!

# Starting nginx and uWSGI on boot.

Although there are multiple ways to launch nginx and uWSGI on boot, creating a service to be managed by systemctl is an effective way to implement an on-boot Bottle app.

$ cd /etc/systemd/system

$ sudo vim statelesshosting.service

In this file, write the following:

[Unit]

Description=Service to start uWSGI and nginx on boot

[Service]

ExecStart=/usr/bin/bash -c 'cd /home/%user%/python-dc-statelesshosting; (uwsgi --ini uwsgi.ini &); nginx -c /home/%user%/python-dc-statelesshosting/nginx.conf'

[Install]

WantedBy=multi-user.target

Save and exit the file. On service start, it will run bash commands to change directory, start uWSGI, and start nginx. It’s just like what we did above, except it’s automated by a service.

Test to make sure the service runs, and if it does, enable it on boot.

$ sudo systemctl start statelesshosting.service

$ sudo systemctl enable statelesshosting.service

Your Bottle app is now running, and will start when the VM boots up.

1. These commands should only be entered if the virtual machine has Apache installed. [↑](#footnote-ref-1)
2. This is just to ensure that the Bottle app works as expected. [↑](#footnote-ref-2)