# Bottle, Gunicorn and NGINX for Production

**Sample App for Deployment**

**Note:** You may want to add a randint function return so that everytime you refresh the page a new random int is shown. This can be used to verify that the page you're seeing wasn't cahed or such.

myapp.py

```
from bottle import route, run, default_app

@route('/')
def index():
        return('Hello World')

if __name__=='__main__':
        run(host='0.0.0.0', port='8080')

myapp = default_app()
```

**In Bottle for run()**

This will only run the run() if the script is called directly. If Gunicorn calls it then the run() will not execute.

```
if __name__ == "__main__":
    run(host='0.0.0.0', port=8080)
```

**Give the App a Name for Gunicorn**

import default_app and then use it to create an app name at the end of the script. This allows Gunicorn to access it. When using Gunicorn you will use script_name:app_name

```
myapp = default_app()
```

**Run Script with Gunicorn**

This runs the script through Gunicorn with 4 workers and binds to any address on the computer at port 8000.

```
gunicorn --workers 4 --bind 0.0.0.0:8000 myapp:app
```

## Tech Stack

**DNS** - Setup DNS so that Domain Name resolves to IP Address

**OS** - Operating System with permissions and firewall settings

**NGINX** - Reverse Proxy Server to handle and redirect traffic

**Gunicorn** - Multithreaded Web Server for Python

**Bottle** - Python Web App Framework

**Python** - Main Programming Language

## Basic Setup

### Prerequisites

You may want to create a specific user account for this app to run under. Such as a user account named **www**

```
sudo adduser www
```

```
sudo apt install nginx
mkdir APPNAME
cd APPNAME
sudo apt install python3-venv
python3 -m venv VENVNAME
source VENVNAME/bin/activate
python3 -m pip install bottle
python3 -m pip install gunicorn
```

**my-app.py**

When everything is setup you will have to restart your app service (not NGINX) for changes to show to users

```
from bottle import route, run, default_app

@route('/')
def index():
    return('hello world')

if __name__ == "__main__":
    run(host='0.0.0.0', port='8080')

#create a name for your app.  wsgi reference will look like my-
app:app_name
app_name = default_app()
```

## Create a Service

```
sudo nano /etc/systemd/system/APPNAME.service
```

```
[Unit]
Description=My Cool App
After=network.target

[Service]
User=USERNAME
Group=www-data
WorkingDirectory=/home/USERNAME/APPFOLDER
Enviornment="PATH=/home/USERNAME/APPNAME/VENVNAME/bin"
ExecStart=/home/USERNAME/APPNAME/VENVNAME/bin/gunicorn --workers 4 --
bind 127.0.0.1:8000 SCRIPTNAME:APPNAME

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl start APPNAME
```

```
sudo systemctl enable APPNAME
```

```
- Go go 127.0.0.1:8000 in web browser and app should be displayed
```

```
sudo nano /etc/nginx/sites-available/default
```

Delete everything in file

```
server {
    listen 80 default_server;
    server_name 127.0.0.1;

    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;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Verify NGINX Configurations Work

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

## Use a SOCKET Instead of IP Address for Service

**Note:** This may be more secure, but requires more knowledge of Linux.

Change Permissions for APP DIRECTORY. Give the User that will access the directroy and the www-data group ownership

```
sudo chown -R /home/www www:www-data
```

```
[Unit]
Description=My cool app
After=network.target

[Service]
User=www
Group=www-data
WorkingDirectory=/home/www/myapp
Environment="PATH=/home/www/myapp/venv/bin"
ExecStart=/home/www/myapp/venv/bin/gunicorn --workers 4 --bind
unix:myapp.sock  myapp:myapp
[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl start APPNAME
```

**Verify Service is Working**

Folder permissions are generally the reason for problems.

```
sudo systemctl status APPNAME
```

```
sudo nano /etc/nginx/sites-available/default
```

```
server {
        listen 80 default_server;
        server_name 127.0.0.1;

        location / {
                proxy_pass http://unix:/home/www/myapp/myapp.sock;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
        }
}
```

Verify NGINX Configurations Work

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```