

2016/01/28

---

# NGINX, DB 설치 과제

# UWSGI 설치 및 CONFIG 설정

---

## UWSGI 설치

### Installing uWSGI

Unlike the guide linked above, in this tutorial, we'll be installing uWSGI globally. This will create less friction in handling multiple Django projects. Before we can install uWSGI, we need the Python development files that the software relies on. We can install this directly from Ubuntu's repositories:

```
sudo apt-get install python-dev
```

Now that the development files are available, we can install uWSGI globally through `pip` by typing:

```
sudo pip install uwsgi
```

## UWSGI 설치 경로로 이동

```
sudo mkdir -p /etc/uwsgi/sites  
cd /etc/uwsgi/sites
```

# UWSGI 설치 및 CONFIG 설정

---

## UWSGI config 파일 수정

```
sudo nano firstsite.ini
```

```
[uwsgi]
project = firstsite
base = /home/user

chdir = %(base)/%(project)
home = %(base)/Env/%(project)
module = %(project).wsgi:application

master = true
processes = 5

socket = %(base)/%(project)/%(project).sock
chmod-socket = 664
vacuum = true
```

# UWSGI 자동실행 스크립트 생성

---

## UWSGI 자동 실행 Script 생성

### Create an Upstart Script for uWSGI

We now have the configuration files we need to serve our Django projects, but we still haven't automated the process. Next, we'll create an Upstart script to automatically start uWSGI at boot.

We will create an Upstart script in the `/etc/init` directory, where these files are checked:

```
sudo nano /etc/init/uwsgi.conf
```

```
description "uWSGI application server in Emperor mode"

start on runlevel [2345]
stop on runlevel [!2345]

setuid user
setgid www-data

exec /usr/local/bin/uwsgi --emperor /etc/uwsgi/sites
```

# NGINX 설치 및 CONFIG 설정

---

## NGINX 설치 및 Config 설정

```
sudo apt-get install nginx
```

Once Nginx is installed, we can go ahead and create a server block configuration file for each of our projects. Start with the first project by creating a server block configuration file:

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

```
server {  
    listen 80;  
    server_name firstsite.com www.firstsite.com;  
  
    location = /favicon.ico { access_log off; log_not_found off; }  
    location /static/ {  
        root /home/user/firstsite;  
    }  
  
    location / {  
        include          uwsgi_params;  
        uwsgi_pass        unix:/home/user/firstsite/firstsite.sock;  
    }  
}
```

# NGINX 실행

---

## 링크 심볼릭 생성

```
sudo ln -s /etc/nginx/sites-available/firstsite /etc/nginx/sites-enabled  
sudo ln -s /etc/nginx/sites-available/secondsite /etc/nginx/sites-enabled
```

## Nginx, uwsgi 실행

```
sudo service nginx configtest
```

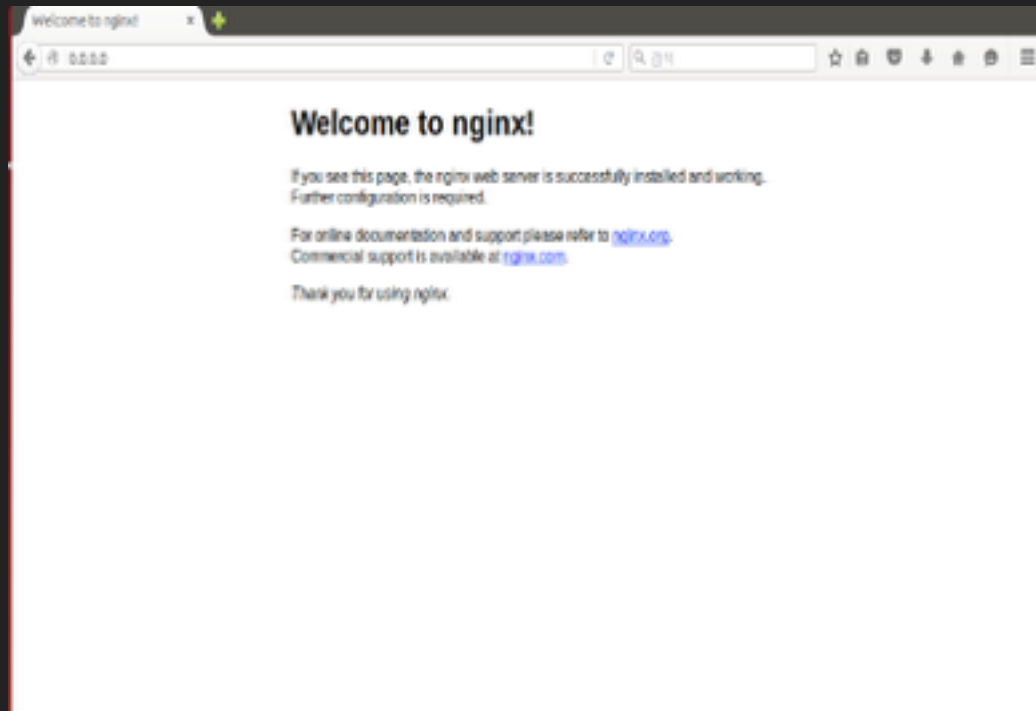
If no syntax errors are detected, you can restart your Nginx service to load the new configuration:

```
sudo service nginx restart
```

If you remember from earlier, we never actually started the uWSGI server. Do that now by typing:

```
sudo service uwsgi start
```

## 최종결과



```
manducku@manducku-VirtualBox: ~/firstsite
(firstsite) manducku@manducku-VirtualBox:~/firstsite$ sudo service nginx start
(firstsite) manducku@manducku-VirtualBox:~/firstsite$ sudo service nginx configtest
* Testing nginx configuration [fail]
(firstsite) manducku@manducku-VirtualBox:~/firstsite$
```

FAIL.....