Deploy Manual (Ubuntu-16.04)

(V-2.0)

Project: ucdgrapenews.com
GitHub: https://github.com/ucd-nlmsc-teamproject/Gungnir-Repo.git
Date: 2017-08-20

Authors: Shen Wenrui
Email: Wenrui.shen@ucdconnect.ie

1. Introduction:

This manual is for the production deployment on Cloud server-side Ubuntu-1604 environment, it supports all functions of this project, such as celery auto scheduler, in this Manual.

2. Environment Requirements:

2.1. Operating system: Ubuntu-16.04;
2.2. Commends environment: Terminal;
2.3. Python 3;
2.4. Django 10 (or above);
2.5. PostgreSQL;
2.6. Install through requirements.txt:

```
pip install -r requirements.txt
```

2.7. Install manually:

3. Server information and settings:

3.1. UCD remote server Vm info:

Version:	Ubuntu 16.04 server VMs
Address:	ucdgrapenews.com
Log User name(root):	student
Log User Password:	gungnir_vmpw
Deploy user:	gungnir_user
Deploy user password:	gungnir_vmpw

3.2. Login to the remote sever:

(please do not use ucd_wireless)

Port:22
Login with SecureCRT(SSH2):
Login with Xshell;
Toturial:http://jingyan.baidu.com/article/ab69b270b0ca3d2ca7189fdc.html

3.3. Ubuntu server basic settings:

3.3.1. Change root’s password:

```
sudo passwd student
```

Old pw: Katoe19!
new password: gungnir_vmpw

3.3.2. Create a new user without root right:

```
sudo adduser gungnir_user
```

pw: gungnir_vmpw

Change to root user: `sudo su`

Exit root user mode: `exit`

Add root right to new user:

```
sudo usermod -a -G sudo gungnir_user
```

Change to new user:

```
su gungnir_user
```

3.3.3. Update Ubuntu settings:

```
sudo apt-get update
sudo apt-get upgrade
```

3.3.4. Install code editor vim:

```
sudo apt-get install vim
```

4. Install Python-3:

```
sudo apt-get install python3.5
vim ~/.bashrc
        add new line:
                alias python=python3
source ~/.bashrc
sudo apt-get install python3-dev
```

5. Install pip3:

```
sudo apt-get install python3-setuptools
sudo apt install python3-pip
pip3 install --upgrade pip3
```

6. Install Django:

```
sudo pip3 install django
```

7. Install and connect PostgreSQL:

7.1. Install PostgreSQL:

```
sudo apt-get install postgresql postgresql-contrib
sudo pip3 install psycopg2
```

7.2. Set PostgreSQL:

```
sudo -i -u postgres
$ psql
        # CREATE USER djangouser01;
        # CREATE DATABASE djangodb01 OWNER djangouser01;
        # \q
        # \password postgres
                *: presqlswr123
$ exit
```

7.3. Check psql’s log:

```
sudo systemctl status postgresql
```

7.4. Migrate Postgresql Database:

7.4.1. Run cmd to create migrations for those changes:

```
python manage.py makemigrations
```

7.4.2. Run cmd to apply those changes to the database:

```
python manage.py migrate
```

7.4.3. If you want to see SQL Statements (Optional):

```
python manage.py sqlmigrate dataCollector 0001
```

7.4.4. Clean Datebase:

```
python manage.py flush
```

7.4.5. Create Super User:

```
python manage.py createsuperuser
        username: gungnir_admin
        email:wenrui.shen@ucdconnect.ie
        ps:gapw2017
```

8. Install Git and downlload code from github:

```
mkdir /home/gungnir_user/gungnir_env
cd /home/gungnir_user/gungnir_env

sudo apt-get install git
git config --global user.email "wenrui.shen@ucdconnect.ie"
git config --global user.name "WenruiShen"
sudo git init Gungnir_mvp
cd Gungnir_mvp
sudo git clone https://github.com/ucd-nlmsc-teamproject/Gungnir-Repo.git
cd Gungnir-Repo/
sudo git fetch
sudo git checkout origin/gungnir_mvp -f
        *:swrgithub123
cd gungnir
sudo chmod -R 777 *
```

9. Install related libraries:

9.1. Install through requirements.txt:

```
sudo pip3 install -r requirements.txt
```

9.2. Install manually:

```
sudo apt-get install libxml2-dev libxslt-dev
sudo apt-get install libjpeg-dev zlib1g-dev libpng12-dev
sudo apt-get install python3-tk
sudo pip3 install djangorestframework
sudo pip3 install newspaper3k
sudo pip3 install arrow
sudo pip3 install simhash
sudo pip3 install djangorestframework-jwt
sudo pip3 install scikit-learn
sudo pip3 install numpy
sudo pip3 install scipy
sudo pip3 install pandas
sudo pip3 install matplotlib
sudo pip3 install sympy
```

9.3. NLTK Downloads:

```
python manage.py shell
>import nltk
        nltk.download('punkt')
        nltk.download('stopwords')
        nltk.download('wordnet')
```

10. Install and set Celery:

10.1. Install Celery:

```
sudo pip3 install django-celery
```

10.2. Install and set Brokers RabbitMQ:

```
sudo apt-get install rabbitmq-server
# Start RabbitMQ
sudo rabbitmq-server -detached
```

10.3. Add foldes and their rights:

```
sudo mkdir /var/run/celery
sudo chmod 777 /var/run/celery
sudo mkdir /var/log/celery
sudo chmod 777 /var/log/celery
```

10.4. Daemonization (Use systemd):

```
sudo chmod 755 Configuration/*
sudo cp Configuration/gungnir_celery_daemon.service /etc/systemd/system/
```

10.5. Restart the Daemon thread:

```
sudo systemctl daemon-reload
sudo systemctl start gungnir_celery_daemon
sudo systemctl enable gungnir_celery_daemon
sudo systemctl restart gungnir_celery_daemon
sudo systemctl stop gungnir_celery_daemon
sudo systemctl status gungnir_celery_daemon
```

10.6. Check log file:

```
sudo journalctl -u gungnir_celery_daemon
tail -f /var/log/celery/worker1.log
```

10.7. Monitor tasks:

```
sudo python manage.py celerycam
celery inspect registere
celery worker --help
celery help
pkill -9 -f 'celery worker'
ps auxww | grep 'celery worker' | awk '{print $2}' | xargs kill -9
```

10.8. Manually beat Periodic Tasks:

```
sudo celery -A gungnir beat -l info
```

11. Build frond-end:

11.1. Initial Install:

```
cd frontend/
sudo apt-get install nodejs
sudo ln -s /usr/bin/nodejs /usr/bin/node
sudo apt install npm
```

11.2. Execute every time updating codes:

```
cd frontend/
npm install
npm run build
cd ../
python manage.py collectstatic
```

12. Initialized data collection:

12.1. Set independent Django running environment:

```
export DJANGO_SETTINGS_MODULE=gungnir.settings
```

12.2. Initialized data collection:

```
Init settings for days:
        data_collector/tasks.py:
                N_days = 30
python Init_data_collector.py
```

13. Install and set Gunicorn:

13.1. Install Gunicorn:

```
sudo pip3 install gunicorn
```

13.2. Set Gunicorn:

```
whereis gunicorn
sudo cp Configuration/grapenews.com.service /etc/systemd/system/
```

13.3. Start Gunicorn thread:

```
sudo systemctl daemon-reload
sudo systemctl start grapenews.com
sudo systemctl enable grapenews.com
sudo systemctl restart grapenews.com
sudo systemctl stop grapenews.com
sudo systemctl status grapenews.com
```

13.4. Check running info:

```
ps -A | grep gunicorn
# Check ports:
netstat -a
lsof -i :8000
# Kill thread:
kill -9 5624
```

13.5. Check Gunicorn log:

```
sudo journalctl -u grapenews.com
sudo journalctl --since "2017-07-15 16:32:00" -u grapenews.com
```

14. Install and set Nginx:

14.1. Install Nginx:

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

14.2. Set Nginx:

```
sudo cp Configuration/gungnir_grapenews /etc/nginx/sites-available/
sudo ln -s /etc/nginx/sites-available/gungnir_grapenews /etc/nginx/sites-enabled/gungnir_grapenews
```

14.3. Start Nginx

(Ensure Gunicorn has been set and under running.)

```
sudo service nginx start
sudo service nginx stop
sudo service nginx restart
sudo service nginx status
```

14.4. Check running log:

```
sudo tail -F /var/log/nginx/error.log
sudo tail -F /var/log/nginx/access.log
```

15. Log and Debug system:

15.1. Default logging files path:

./django_logging_files/

15.2. Default logging settings:

gungnir/settings.py:

'dataCollector': ‘INFO’

Others: >’WARNING’

16. Local DNS Testing:**(Only for local ubuntu debuging)**

```
sudo apt-get install bind9
dpkg -l |grep bind9
# Change /etc/bind/named.conf.local
cd /etc/bind
# 在/etc/bind下新建正向区域文件（复制一份）
sudo cp db.local db.grapenews.com
# 在/etc/bind下新建反向区域文件（复制一份）
sudo cp db.127 db.rev_grapenews.com

sudo vim /etc/bind/named.conf.default-zones
# 添加2个zone,
# ”db.grapenews.com”用来正解,
# “db.rev_grapenews.com”是用来反解
zone "csi6220-2-vm2.ucd.ie" {
        type master;
        file "/etc/bind/db.grapenews.com";
};

zone "168.192.in-addr.arpa"  {
        type master;
        file "/etc/bind/db.rev_grapenews.com";
};

# Set db.grapenews.com
sudo vim /etc/bind/db.grapenews.com
# Set db.rev_grapenews.com
sudo vim /etc/bind/db.rev_grapenews.com

# 在本机(192.168.23.130)上设置DNS,
sudo vim /etc/resolv.conf
~~sudo vim /etc/resolvconf/resolv.conf.d/base~~
# 添加在第一行：
 nameserver 192.168.23.130
~~sudo service network-manager restart~~

sudo service bind9 restart
tail /var/log/syslog

# Test DNS server:
host csi6220-2-vm2.ucd.ie
```

17. Deploying tutorials:

Use ‘Nginx’ and ‘Gunicorn’ deploy Django:

http://zmrenwu.com/post/20/

Auto deploy with Fabric:

http://zmrenwu.com/post/21/

Links about deploying on ‘IIS’, ‘APACH’,’Windows Azure’:

http://www.cnblogs.com/haozi0804/tag/django/

Psql install on ubuntu:

https://help.ubuntu.com/community/PostgreSQL

Daemonization Toturial link:

http://docs.celeryproject.org/en/latest/userguide/daemonizing.html

Others:

http://www.jianshu.com/p/e13417a8bb7f
https://segmentfault.com/a/1190000008507042
http://www.cnblogs.com/fnng/p/5268633.html
http://epicserve-docs.readthedocs.io/en/latest/index.html