NOOSDrift

How to deploy a new node

Table des matières

1. Introduction

   This document is focused on the deployment and setting up of a Node component of the NOOS-Drift network. Deployment of a Central component is not considered here.

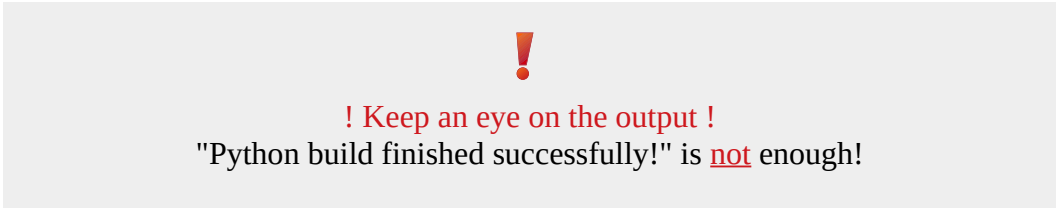
2. Python 3.x Installation

   a) On Ubuntu 14.04 LTS (Trusty Tahr) [with super user rights]

```
sudo add-apt-repository ppa:jonathonf/python-3.6
sudo apt update
sudo apt install python3.6 python3-pip
sudo pip3 install virtualenv
```

   b) On Debian 8 (Jessie) [with super user rights]

```
sudo apt install make build-essential libssl-dev zlib1g-dev
sudo apt install libreadline-dev libsqlite3-dev wget curl llvm
sudo apt install libncurses5-dev libncursesw5-dev xz-utils tk-dev
sudo apt install libbz2-dev liblzma-dev libgdbm-dev checkinstall
wget https://www.python.org/ftp/python/3.6.3/Python-3.6.3.tgz
tar xvf Python-3.6.8.tgz
cd Python-3.6.8/
./configure --enable-optimizations --enable-shared –with-ensurepip=install ↲
            --without-gcc LDFLAGS="-Wl,--rpath=\$$LIB:/usr/local/lib"
make -j8
```




```
sudo checkinstall -D --install=no --fstrans=no --pkgname=python3.6 ↲
                  make altinstall
sudo dpkg -i python3.6_3.6.8-1_amd64.deb
sudo mv python3.6_3.6.8-1_amd64.deb /home/www-data/
cd
sudo rm -r Python-3.6.8/
rm Python-3.6.8.tgz
```

   c) On Debian 10 (Buster) [with super user rights]

   Python 3 used by Debian Buster is Python 3.7

```
apt install python3-venv
```

d) On CentOS 7 [with super user rights]

```
sudo yum install https://centos7.iuscommunity.org/ius-release.rpm
sudo yum update
sudo yum install python36u python36u-libs python36u-devel python36u-pip
```

3. Create virtual environments

The configuration path and virtual environments have been put in slightly different locations for different OSes.

```
/var/opt/noosdrift/ on Debian/Ubuntu type systems
/opt/noosdrift/ on CentOS systems
```

Both pathes will be refered to as /pathconfdir

a) On Ubuntu < 18,04 / Debian <= 8,0 [with super user rights]

```
cd /pathconfdir
 mkdir -p Django
mkdir -p Django_env_vars
sudo apt install virtualenv
virtualenv -p /usr/bin/python3.6 ~/venv
```

b) On Debian 10, Ubuntu >= 18,04 and CentOS 7 [with super user rights]

```
cd /pathconfdir
mkdir -p Django
mkdir -p Django_env_vars
sudo python3 -m venv ./venv
```

4. Install mod_wsgi

   Debian and Ubuntu distributions now have well structured packages which make these procedures not necessary for Debian > 8,0 and Ubuntu > 14,04

   a) On Ubuntu 14.04 LTS (Trusty Tahr) [with super user rights]

```
apt install python3.6-dev apache2-dev
su - www-data
source /pathconfdir/venv/Apache_WSGI_MOD_Env/bin/activate
pip install --no-cache-dir mod_wsgi
deactivate
exit
vim /etc/apache2/mods-available/wsgi.load
```

```
LoadModule wsgi_module
"/pathconfdir/venv/Apache_WSGI_MOD_Env/lib/python3.6/↲
site-packages/mod_wsgi/server/mod_wsgi-py36.cpython-36m-x86_64-linux-gnu.so"
WSGIPythonHome "/pathconfdir/venv/Apache_WSGI_MOD_Env/"
```

```
chown root:root /etc/apache2/mods-available/wsgi.load
chmod 644 /etc/apache2/mods-available/wsgi.load
a2enmod wsgi
service apache2 restart
service apache2 status                                  (check)
apachectl -M                                            (check)
```

   b) On Debian 8 (Jessie) [with super user rights]

```
apt install apache2-dev
su - www-data
source /pathconfdir/venv/Apache_WSGI_MOD_Env/bin/activate
pip install --no-cache-dir mod_wsgi
deactivate
exit
vim /etc/apache2/mods-available/wsgi.load
```

```
LoadModule wsgi_module
"/pathconfdir/venv/Apache_WSGI_MOD_Env/lib/python3.6/↲
site-packages/mod_wsgi/server/mod_wsgi-py36.cpython-36m-x86_64-linux-gnu.so"
WSGIPythonHome "/pathconfdir/venv/Apache_WSGI_MOD_Env/"
```

```
chown root:root /etc/apache2/mods-available/wsgi.load
chmod 644 /etc/apache2/mods-available/wsgi.load
a2enmod wsgi
service apache2 restart
service apache2 status                                  (check)
apachectl -M                                            (check)
```

   c) On Debian 10 (Buster) [with super user rights]

```
apt install libapache2-mod-wsgi-py3
```

   d) On CentOS 7 [with super user rights]

```
yum install python36u-mod_wsgi
httpd -M                                                (check)
```

5. Set environment variables

Setting environment variables requires to update the ‘activate’ file which is present in the virtual environment directory. [with super user rights]

```
su - www-data
vim /pathconfdir/venv/bin/activate
```

The variables to be set depend on the role of the Node. Two roles are possible, “Node” or “Central”.
In this case we will use the variables necessary to set up a node with role “Node”.

a) Python virtual environment variables

Add towards the end of "deactivate" function:

```
#Unset NOOS_Drift variables
unset NOOS_BROKER_URL
unset NOOS_CENTRAL_ID
unset NOOS_ENV
unset NOOS_MME_ID
unset NOOS_NODE_ID
unset NOOS_ROLE
unset NOOS_SECRET_KEY
unset NOOS_SFTP_HOST
unset NOOS_SFTP_PWD
unset NOOS_SFTP_USER
unset NOOS_USERNAME
unset NOOS_USERPWD
```

Add at the end of file:
Replace # with correct value

```
#NOOS_Drift variables
NOOS_BROKER_URL="ampq://rabbitMQUser:rabbitMQPwd@localhost:5672/
rabbitMq_vhost"
NOOS_CENTRAL_ID="1"
NOOS_ENV="PROC"
NOOS_MME_ID="1"
NOOS_NODE_ID="#"
NOOS_ROLE="Node"
NOOS_SFTP_HOST="#"
NOOS_SFTP_PWD="#"
NOOS_SFTP_USER="#"
NOOS_SECRET_KEY="#"
NOOS_USERNAME="#"
NOOS_USERPWD="#"

export NOOS_BROKER_URL NOOS_CENTRAL_ID NOOS_ENV NOOS_MME_ID NOOS_NODE_ID
NOOS_ROLE NOOS_SECRET_KEY NOOS_SFTP_HOST NOOS_SFTP_PWD NOOS_SFTP_USER
NOOS_USERNAME NOOS_USERPWD
```

b) Application environment variables [with super user rights]

This part makes sure that the variables set up in the OS environment are read by the Django application environment

```
mkdir /pathconfdir/Django_env_vars
chmod 755 /pathconfdir/Django_env_vars
vi /pathconfdir/Django_env_vars/noosdrift.py
```

Replace # with correct value

```
"""
Environment variables for noosDrift project.
Only strings are permitted.

"""

import os

os.environ["NOOS_BROKER_URL"] =
"ampq://rabbitMQUser:rabbitMQPwd@localhost:5672/rabbitMq_vhost"
os.environ["NOOS_CENTRAL_ID"] = "1"
os.environ["NOOS_ENV"] = "PROD"
os.environ["NOOS_MME_ID"] = "1"
os.environ["NOOS_NODE_ID"] = "#"
os.environ["NOOS_ROLE"] = "Node"
os.environ["NOOS_SFTP_HOST"] = "#"
os.environ["NOOS_SFTP_PWD"] = "#"
os.environ["NOOS_SFTP_USER"] = "#"
os.environ["NOOS_SECRET_KEY"] = "#"
os.environ["NOOS_USERNAME"] = "#"
os.environ["NOOS_USERPWD"] = "#"
```

```
chown root:root /pathconfdir/Django_env_vars/noosdrift.py
chmod 644 /pathconfdir/Django_env_vars/noosdrift.py
```

6. Deploy application code

If the git software is not yet installed. Install it using apt (Debian/Ubuntu) or yum (CentOS)

a) Clone the code from a git repository

```
cd  /pathtconfdir
source venv/bin/activate
git clone --branch nodeBranch https://gitUrl Django/noosDrift
cd Django/noosDrift
pip install --no-cache-dir --upgrade --force-reinstall -r requirements.txt
python manage.py collectstatic --settings=deploy_settings
vim noosDrift/settings.py
```

Check or edit "ALLOWED_HOST"

```
deactivate
cd ../..
```

b) Set the appropriate user rights for this directory (and under)

On Ubuntu 14.04 LTS (Trusty Tahr) , Debian 8 (Jessie), 10 (Buster)

```
chown -R www-data.www-data Django
```

On CentOS 7

```
chown -R apache.apache Django
```

7. Install RabbitMQ

a) On Ubuntu 14.04 LTS (Trusty Tahr), Debian 8 (Jessie), Debian 10 (Buster) [with super user rights]

```
apt install rabbitmq-server
rabbitmqctl add_user noosdrift #           (Replace # with a password)
rabbitmqctl add_vhost noosdrift_vhost
rabbitmqctl set_permissions -p noosdrift_vhost noosdrift ".*" ".*" ".*"
rabbitmqctl delete_user guest
```

Change value of NOOS_BROKER_URL parameter to

```
ampq://noosdrift:theRabbitmqPwd@localhost:5672:noosdrift_vhost
```

in files

```
/pathconfdir/venv/bin/activate
/pathconfdir/Django_env_vars/noosdrift.py
```

Modify credentials for Celery to match what was made at step 2 & 3

b) On CentOS 7 [super user]

```
sudo yum install rabbitmq-server
sudo systemctl enable rabbitmq-server.service
sudo systemctl start rabbitmq-server.service
sudo rabbitmqctl add_user noosdrift #             (Replace # with a password)
sudo rabbitmqctl add_vhost noosdrift_vhost
sudo rabbitmqctl set_premissions -p noosdrift_vhost noosdrift ".*" ".*" ".*"
sudo rabbitmqctl delete_user guest
```

Change value of NOOS_BROKER_URL parameter to

```
ampq://noosdrift:theRabbitmqPwd@localhost:5672:noosdrift_vhost
```

in files

```
/pathconfdir/venv/bin/activate
/pathconfdir/Django_env_vars/noosdrift.py
```

Modify credentials for Celery to match what was made at step 4 & 5

8. Check Celery

a) On Ubuntu 14.04 LTS (Trusty Tahr), Debian 8 (Jessie), Debian 10 (Buster) [www-data]

```
cd /pathconfdir/
source venv/bin/activate
cd Django/noosDrift
clear && celery -A noosDrift worker -E --loglevel=INFO
[Ctrl + C]
[Ctrl + C]
deactivate
```

b) On CentOS 7 [with super user rights]

```
cd /pathconfdir/
source venv/bin/activate
cd Django/noosDrift
clear && celery -A noosDrift worker -E --loglevel=INFO
[Ctrl + C]
[Ctrl + C]
deactivate
```

9. Daemonizing Celery using Supervisord (system package version) [with super user rights]

```
apt install supervisor
vi /etc/supervisor/supervisord.conf
```

Add into [supervisord] section

```
user=root
```

Add into both [unix_http_server] and [supervisorctl] sections

```
username=noos#dummy_user
password=noos#dummy_password
```

```
mkdir /var/log/celeryd
touch /var/log/celeryd/noosdrift.log
touch /var/log/celeryd/noosdrift_error.log
mkdir /etc/supervisor/conf.d
vi /etc/supervisor/conf.d/noosdrift-celery.conf
```

Replace # with correct value

```
[program:noosdrift-celery]
command=/pathconfdir/venv/bin/celery -A noosDrift worker –loglevel=info
environment=PYTHONPATH="/pathconfdir/Django/noosDrift",↲
            PATH="/pathconfdir/venv/bin:%(ENV_PATH)s",↲
NOOS_BROKER_URL=”ampq://
            noosdrift:theRabbitmqPwd@localhost:5672:noosdrift_vhost”,↲
            NOOS_CENTRAL_ID="#",NOOS_ENV="#",NOOS_MME_ID="#",NOOS_NODE_ID="#",↲
            NOOS_ROLE="#",NOOS_SFTP_HOST="#",NOOS_SFTP_PWD="#",NOOS_SFTP_USER="#",↲
            NOOS_SECRET_KEY="#",NOOS_USERNAME="#",NOOS_USERPWD="#"
directory=/pathconfdir/Django/noosDrift
user=www-data
numprocs=1
stdout_logfile=/var/log/celeryd/noosdrift.log
stderr_logfile=/var/log/celeryd/noosdrift_error.log
autostart=true
autorestart=true
startsecs=10
stopwaitsecs = 600
priority=998
```

```
chmod 640 /etc/supervisor/conf.d/noosdrift-celery.conf
systemctl restart supervisor
```

Check those logging files to see if any error occurs :

```
/var/log/celeryd/noosdrift.log
/var/log/celeryd/noosdrift_error.log
/var/log/supervisor/supervisord.log
```

10. Daemonizing Django using Apache2 [with super user rights]

    Add to the appropriate config file (eg. /etc/apache2/sites-available/default-ssl.conf)

```
##########
# Django #
##########

#NOOS-Drift
Alias /noosdrift/api/static "/pathconfdir/Django/noosDrift/static"
Alias /noosdrift/api/media "/pathconfdir/Django/noosDrift/media"
Alias /noosdrift/api/templates "/pathconfdir/Django/noosDrift/templates"

<Directory "/pathconfdir/Django/noosDrift/static">
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
<Directory "/pathconfdir/Django/noosDrift/media">
        Options -Indexes
        AllowOverride None
        Require all granted
</Directory>
<Directory "/pathconfdir/Django/noosDrift/templates">
        Options -Indexes
        AllowOverride None
        Require all granted
</Directory>
<Directory "/pathconfdir/Django/noosDrift/noosDrift">
        <Files "wsgi.py">
                Require all granted
        </Files>
</Directory>

WSGIScriptAlias /noosdrift/api /pathconfdir/Django/noosDrift/noosDrift/wsgi.py
WSGIDaemonProcess noosDriftTest user=www-data group=www-data ↲
                  python-home=/pathconfdir/venv ↲
                  python-path=/pathconfdir/venv/lib/python3.X/site-packages/: ↲
                  /pathconfdir/Django/noosDrift
WSGIPassAuthorization On
WSGIProcessGroup noosDriftTest

WSGIApplicationGroup %{GLOBAL}
```

```
a2ensite default-ssl                    (if site not enabled yet)
systemctl restart apache2         (if site not enabled yet)
systemctl reload apache2          (if site already enabled)
```

Check if url `noosdrift/api` is available.

11. Test Site

    `wget http://localhost/noosdrift/api/admin`

12. Node component finer settings

Besides adding values to the environment parameters in files « activate », « noosdrift.py » « noosdrift-celery.conf », some values need to be set in the « settings.py » file in the Django app.

a) ALLOWED HOSTS (Already mentionned)

b) Settings linked to security certificates. This varies from one system to another. Some systems have the certificate on the app machine, some not.

c) schema_dict

13. Node system directories

a) Logging

```
/var/log/celery/noosdrift.log
/var/log/celery/noosdrift_error.log
/var/log/supervisor/supervisor.log
/var/opt/noosdrift/Django/noosdrift/logs/
```

b) Configuration

```
/pathconfdir/venv/bin/activate
/pathconfdir/Django/noosDrift/noosDrift/settings.py
/pathconfdir/Django_env_vars/noosdrift.py
/etc/supervisor/conf.d/noosdrift-celery.conf
```

c) Demand and result files

On a Node system, the files containing the demands are stored here
`/var/opt/noosdrift/Django/noosdrift/results/`
those are the paramaters of the demand with a name of the form
`noosdrift_NN_NN_NN.json`
and the resulting netCDF file with a name of the form
`noosdrift_NN_NN_NN.nc`

Those two files are copied into an archive file named
`noosdrift_NN_NN_NN.tgz`
which is sent to the central system to perform the MME analysis

14. Important remark about the Model software components.

The point of the Node is to execute a piece (or several pieces) of software referred to as ‘the Model’. This software will be executed by the Node using www-data / apache user rights. So make sure that these software components are set with the appropriate Unix access rights.