

Настраиваем среду разработки под Linux

- Лучший вариант получить среду разработки в 2016 году использовать [branch feature/docker](#).
- Список ПО
- Зачем?
- Что получилось?
- Ок, что надо делать?
- Грабли

! Лучший вариант получить среду разработки в 2016 году использовать [branch feature/docker](#).

Список ПО

- pycharm <http://www.jetbrains.com/pycharm/>
- openssh
- keepass <http://keepass.info/>
- keeagent <http://lechnology.com/KeeAgent+Beta>

Зачем?

- Хотелось позаниматься написанием скриптов для SA в удобной IDE.
- не хотелось подымать еще одну инсталляцию дома для разработки

Что получилось?

- Локальный репозиторий

• Возможность работать с удаленной базой установленного нока

• Возможность запускать скрипты через debug-scripts непосредственно из под среды разработки

- Debugger

Ок, что надо делать?

1. настраиваем репозиторий например так [Pull request на bitbucket](#)
2. Слегка улучшаем

```
% cd <repo_path>
% sudo aptitude install python-virtualenv
% virtualenv .
% ./bin/pip install -r etc/requirements/noc.txt
```

```
. python-dev postgresql-server-dev-9.1. .
```

3. Сильно не хотелось подымать локальный mongodb и postgres. Будем использовать те что уже есть на dev сервере. для этого сделаем так вот <http://www.debian-administration.org/articles/539> . с небольшим дополнением. проброс портов у меня получилось вот так вот:

/etc/network/interfaces

```
iface tunl inet static
    pre-up ssh -S /var/run/ssh-myvpn-tunnel-control -M -f -L 63334:127.0.0.1:27017 -L 63333:
127.0.0.1:5432 -w 1:1 root@<NOC-DEV-SERVER-IP> start
    pre-up sleep 5
    address 10.0.0.2
    pointopoint 10.0.0.1
    netmask 255.255.255.252
    up route add -net <TEST-HW-NETWORK> gw 10.0.0.1 tunl
    post-down ssh -S /var/run/ssh-myvpn-tunnel-control -O exit <NOC-DEV-SERVER-IP>
```

этот кусок который отличается от статьи. остальное строго по ней.

Оказывается ssh может пробрасывать порты и прикидываться интерфейсом и все это делать в рамках одного соединения.

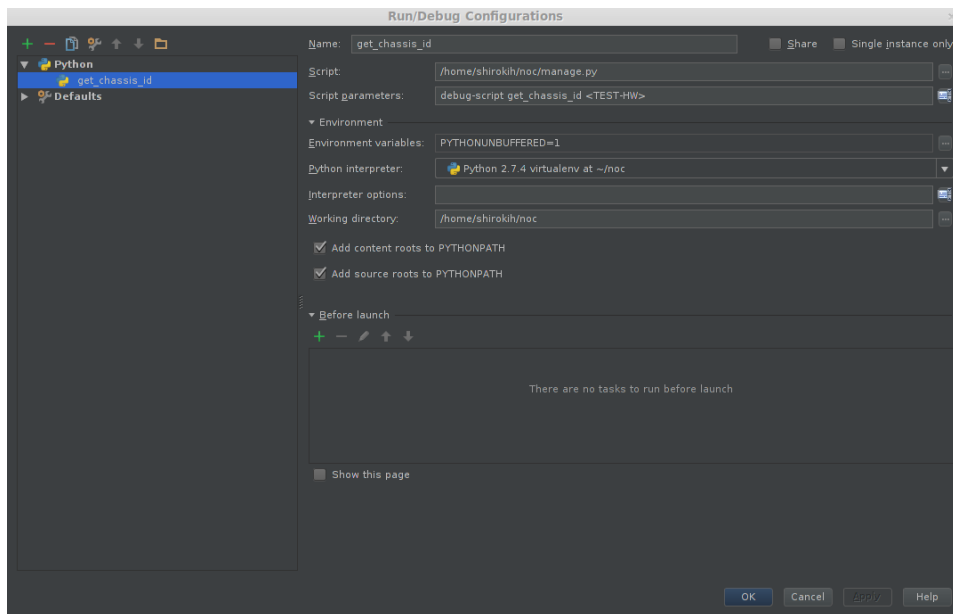
4. Доступ к базам надо прописать в `etc/noc.conf` в соответствующих секциях.

```
[database]
# postgresql_psyncpg2
engine = postgresql_psyncpg2
# Database name
name = noc
# Database user
user = noc
# User password
password = noc
# Set to empty string for localhost
host = 127.0.0.1
# Set to empty string for default
port = 63333

[nosql_database]
# MongoDB database name
name = noc
# MongoDB database user
user = noc
# MongoDB database password
password = noc
#
host = 127.0.0.1
#
port = 63334
```

за сим настройка нока завершена. других изменений не делал. что бы проверит как это работает пришлось поставить клиенты mongo и

5. ставим `ruscharm`
6. На созданную директорию настраиваем `ruscharm`. он радостно все хакает.
7. Делаем вот такие настройки запуска скриптов



8. Радуетесь. Pycharm не только запускает скрипты но и дает их отлаживать!

Габри

- Pycharm не смотря на всю крутость не умеет mercurial branch. так что их по прежнему надо делать через консоль.
- что бы заниматься разработкой надо переключаться в linux.
- Надо бы описать как во всей этой схеме помогает keepass, но влом.