

NOC Tower & NOC

впиши в конф_преф:

```
legacy:///yam::///opt/noc/etc/settings.yml,env:///NOC
```

и в models/environment.py добавь

```
alerta_url = CharField(default="")
```

```
alerta_token = CharField(default="")
```

```
(после metrics_collector = CharField(default=""))
```

```
/etc/selinux/config
```

```
cd /etc/yum.repos.d/
```

```
wget http://download.opensuse.org/repositories/home:Alexx2000/CentOS\_7/home:Alexx2000.repo
```

```
yum install doublecmd-gtk
```

```
service mongod restart
```

```
systemctl restart noc
```

```
subscription-manager repos --enable rhel-7-server-optional-rpms
```

Если у вас есть сервер под Linux Centos и вам необходимо установить рутовый сертификат, используйте:

```
yum install ca-certificates
```

Включаем динамическое обновление сертификатов:

```
update-ca-trust enable
```

Далее копируем наш сертификат в доверенные сертификаты:

```
cp foo.crt /etc/pki/ca-trust/source/anchors/
```

И обновляем доверенные сертификаты:

```
update-ca-trust extract
```

Если у вас на сервере Linux Debian, создаем директорию:

```
dir /usr/local/share/ca-certificates/
```

Копируем сертификат:

```
sudo cp foo.crt /usr/local/share/ca-certificates/foo.crt
```

Обновляем его:

```
sudo update-ca-certificates
```

ребут и нжинкс в мониторинге позеленеет

Некоторое время назад в круг моих интересов вошли сетевые технологии в целом и централизованное управление сетевым оборудованием в частности. В подкасте LinkMeUp я услышал о проекте NOC Project, и очень им заинтересовался. На тот момент возможности "пощупать его" на реальной сети не оказалось. Недавно такая возможность появилась, но зайдя на сайт я увидел что там лежит инструкция к установке версии от мая 15-го года. Я немного огорчился, предположив что проект заглох. Но что есть, то есть, и я решил попробовать использовать эту версию. Установил на виртуалке с Debian 8.8. Импортировал туда железки и был приятно удивлен функционалом. Продолжив изучать сайт я наткнулся на несколько статей, намекающих на то что проект не брошен. Вчитавшись, я понял, что для продолжения разработки проекта разработчики изменили архитектуру и продолжают активно развивать проект. На данном (10-6-17) этапе проект представляет из себя систему развертывания NOC (она называется NOC Tower) и собственно сам NOC. И теперь чтобы получить текущую версию NOC, надо разобраться с установкой NOC Tower (системой развертывания) и с тем как собственно развернуть NOC используя NOC Tower. Для эксперимента я создал две виртуальные машины в VMWare Workstation. На одной я поднял NOC Tower, а на другой развернул сам NOC. В процессе эксперимента я часто обращался к разработчикам в их канал в Telegram, за подсказками и разъяснениями.

По адресу <https://code.getnoc.com/noc/tower/> есть файл <https://code.getnoc.com/noc/tower/blob/master/Readme.md> в котором описан процесс установки NOC Tower.

По адресу https://code.getnoc.com/noc/tower/blob/master/FAQ_rus.md есть несколько подсказок

По адресу [Install NOC Tower on Debian 8](#) тоже можно кое что почерпнуть

Для Debian 8.8 (netinst с GNOME3 и системными утилитами, ничего другого ставить не рекомендую, на этапе установки hostname(ом) указал NOC-Tower):

К этим командам добавить нечего кроме того что на пользователя tower пароль не ставим:

```
# apt-get install python-virtualenv libffi6 libffi-dev python-dev gcc libssl-dev
```

```
# groupadd tower
```

```
# useradd -d /home/tower -g tower -s /bin/bash -m tower
```

Если вы ставили Debian через netinst то dbus уже установился, но "чтоб точно":

```
# apt-get install dbus
```

Следующие команды подготавливают площадку и ставят NOC Tower

```
# mkdir /opt/tower
```

```
# cd /opt/tower
```

Важный аспект "точка не опечатка", она там должна быть:

```
/opt/tower# virtualenv .
```

```
/opt/tower# ./bin/pip install --upgrade pip
```

```
/opt/tower# ./bin/pip install https://cdn.getnoc.com/tower/noc-tower-latest.zip или ./bin/pip install https://cdn.getnoc.com/tower/noc-tower-master.zip
```

```
/opt/tower# chown -R tower var/
```

Учесть --proxy http://user:password@proxyserver:port

```
./bin/pip install --proxy http://user:password@proxyserver:port --upgrade pip
```

```
./bin/pip install --proxy http://user:password@proxyserver:port https://cdn.getnoc.com/tower/noc-tower-latest.zip
```

На этапе генерации SSH ключей предлагается либо ввести парольную фразу, либо ее не вводить. Рекомендация: не вводить.

```
/opt/tower# su - tower -c "ssh-keygen -t rsa -b 4096"
```

Эта команда запускает собственно NOC Tower. Она не пишется в автозагрузку и в случае перезагрузки хоста NOC Tower надо будет запускать заново этой командой:

```
/opt/tower# su - tower -c "cd /opt/tower && ./bin/tower-web"
```

После запуска этой команды, можно открыть web-браузер и введя ip-адрес вашей виртуалки с указанием порта 8888 войти в web-интерфейс NOC-Tower (<http://1.1.1.1:8888>)

логин admin пароль admin

На виртуалке, предназначенной для развертывания NOC, создаем пользователя и пароль:

```
useradd -d /home/ansible -s /bin/bash -m ansible
```

passwd ansible

устанавливаем пакет sudo:

```
apt-get install sudo
```

настраиваем безпарольное использование sudo пользователем ansible:

```
gedit /etc/sudoers
```

В раздел # User privilege specification

добавляем строчку

```
ansible ALL=(ALL) NOPASSWD:ALL
```

Теперь чуть более сложный момент:

сгенерированные ранее ключи надо скопировать с NOC-Tower на NOC-Node

ip-адрес NOC-node 192.168.40.139

на NOC-Node ставим ssh

```
apt-get install ssh
```

Вводим команду из под пользователя tower так как именно из под него мы и делали ключи

(так как на tower мы пароль не ставили, то сначала команда su и пароль root(a) а потом команда su tower)

```
ssh-copy-id -i /home/tower/.ssh/id_rsa.pub ansible@192.168.40.139
```

Вот такой вывод должен появиться

```
tower@NOC-Tower:/home/gav$ ssh-copy-id -i /home/tower/.ssh/id_rsa.pub ansible@192.168.40.139
```

```
The authenticity of host '192.168.40.139 (192.168.40.139)' can't be established.
```

```
ECDSA key fingerprint is a2:4a:c4:23:97:9c:7c:1e:28:a7:96:b3:7a:33:72:97.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
```

```
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
```

```
ansible@192.168.40.139's password:
```

```
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh 'ansible@192.168.40.139'"
```

```
and check to make sure that only the key(s) you wanted were added.
```

```
tower@NOC-Tower:/home/gav$
```

Количество добавленных ключей

Теперь надо зайти на Node по SSH

из-под пользователя tower используя для подключения логин ansible

```
root@noc-tower:~# su tower
```

```
tower@noc-tower:/root$ ssh ansible@192.168.40.139
```

Фишка в том что после этих манипуляций с виртуалки NOC-Tower на виртуалку NOC-Node по SSH из под пользователя tower используя логин ansible можно подключаться не вводя пароля

далее проверяем применилась ли наша настройка безпарольного использования sudo на NOC-Node из под пользователя ansible

```
ansible@noc-node1:~/$ sudo -s
```

```
ansible@noc-node1:~/$ python
```

первая команда должна примениться без ввода пароля

вторая проверяет запуск командной строки Python

ctrl-d выход из консоли python

Далее будет еще сложнее

разработчик NOC-tower рекомендует следующее:

лучше если там будет имя. совсем хорошо если имя будет регнуто в днсе компании. идеально если к этому имени будет сертификат. ну и божественно если к имени будет валидный https сертификат

так как мы используем виртуалки то для достижения нужного эффекта воспользуемся файлом hosts

gedit /etc/hosts

и добавим строчку связывающую ip и имя

192.168.40.139 NOC-Nod2

про развертывание чуть позже