

Установка NOC Tower на FreeBSD

Перед установкой

Требования.

Для установки желательно 2 сервера с установленной ОС FreeBSD 11.X. Первый сервер будет выполнять роль NOC Tower, с него будет разворачиваться NOC.

Назовём их, условно, nos-tower и nos-main01

 В случае, если установка ведётся на единственный сервер, то имена будут совпадать. Также, возможен вариант развёртывания Tower на хост с Linux, а развёртывание NOCа на FreeBSD

 В случае, если установка ведётся на компьютер без доступа в Интернет - необходимо указать HTTP прокси. Это делается установкой переменных окружения HTTP_PROXY и HTTPS_PROXY

 Если установка ведётся на версию FreeBSD **Release**, необходимо заменить в в /etc/pkg/FreeBSD.conf url с "pkg+[http://pkg.FreeBSD.org/\\${ABI}/quarterly](http://pkg.FreeBSD.org/${ABI}/quarterly)" на "pkg+[http://pkg.FreeBSD.org/\\${ABI}/latest](http://pkg.FreeBSD.org/${ABI}/latest)", иначе будет недоступен Telegraf и Nginx 1.10, что приведёт к ошибкам во время установки.

Sudo

Используется ansible для развёртывания окружения NOC

```
# pkg install sudo
```

 Если на целевой системе не установить sudo, то при попытке сделать Deploy Нока выведется ошибка:
FAILED! => {"changed": false, "failed": true, "module_stderr": "sudo: not found\n", "module_stdout": "", "msg": "MODULE FAILURE", "parsed": false}

Python

Python 2.7

необходим для работоспособности NOC'а и Ansible

```
# pkg install python
```

 Если на целевой системе не будет установлен Python, то при попытке произвести деплой (Deploy) NOC'а из башни выведется ошибка вида:
"stderr": "Traceback (most recent call last):\n File \"/opt/noc/scripts/deploy/install-packages"

Остальные необходимые приложения будут установлены Ansible при развёртывания НОКа.

Пользователи

На всех машинах, на которые будет производиться развёртывание NOC'а через башню, необходимо добавить пользователя ansible, задать ему пароль и установить в качестве оболочки (shell'а) csh

```
# pw groupadd -n ansible  
  
# pw useradd -g ansible -s /bin/csh -d /home/ansible -n ansible -m  
  
# passwd ansible
```

 В Ansible 2.0.X присутствует баг, если для пользователя ansible использовать shell sh (/bin/sh), то при установке не будет работать подсчёт контрольной суммы конфигов файлов:
FAILED! => {"changed": false, "failed": true, "msg": "failed to checksum remote file. Checksum error code: 0"}

Добавляем пользователя в sudoers

```
# echo 'ansible ALL=(ALL) NOPASSWD:ALL' >> /usr/local/etc/sudoers
```

Установка Башни

Установка башни возможна в 2 вариантах: установка в контейнере docker или установка напрямую на хост. Docker доступен для FreeBSD начиная с версии 10.3 Release.

Установка докер-контейнера

Установка описана в [NOC Tower FAQ](#). Приведём команды для FreeBSD.

 На данном этапе (в версии FreeBSD 10.3) поддержка docker находится в экспериментальном режиме. Поэтому рекомендуется установка напрямую на хост.

Устанавливаем и запускаем docker

```
# pkg install docker  
# service docker start
```

Скачиваем образ NOC Tower

```
# docker pull registry.gitlab.com/nocproject/tower:latest
```

Запускаем

```
# docker run -t registry.gitlab.com/nocproject/tower
```

Установка напрямую на хост

Для установки башни можно воспользоваться скриптом на shell.

Скрипт для автоматического развёртывания Tower

```
#!/bin/sh

info ( ) {
    printf "\033[1;32m$1\033[0m\n" 1>&2
}
OSNAME=`uname -s`
rc_arg=$1
if [ -z $rc_arg ]; then
    towerdir="tower"
else
    towerdir="$rc_arg"
fi
info "Installing packages..."
if [ $OSNAME = "FreeBSD" ]; then
    PREFIX="/usr/local"
    pkg install -y python2 libffi py27-setuptools py27-pip py27-virtualenv ca_root_nss py27-sqlite3
    rehash

    if [ ! -d "/home/tower" ]; then
        info "Creating user..."
        pw groupadd -n tower
        pw useradd -g tower -s /bin/csh -d /home/tower -n tower -m
    fi
else
    PREFIX="/opt"
    apt-get install python-virtualenv libffi6 libffi-dev python-dev gcc libssl-dev
    if [ ! -d "/home/tower" ]; then
        info "Creating user..."
        groupadd tower
        useradd -d /home/tower -g tower -s /bin/bash -m tower
    fi
fi
if [ ! -f "/home/tower/.ssh/id_rsa.pub" ]; then
    info "Generating SSH key..."
    su - tower -c "ssh-keygen -t rsa -b 4096" && exit
fi

if [ ! -d $PREFIX/$towerdir ]; then
    mkdir $PREFIX/$towerdir
else
    rm -rf $PREFIX/$towerdir/*
fi
cd $PREFIX/$towerdir
info "Installing virtual enviropment..."
virtualenv .
info "Installing tower..."
./bin/pip install --upgrade pip
./bin/pip install https://cdn.getnoc.com/tower/noc-tower-latest.zip
chown -R tower var/
info "Running tower..."
su - tower -c "cd $PREFIX/$towerdir && ./bin/tower-web"
```

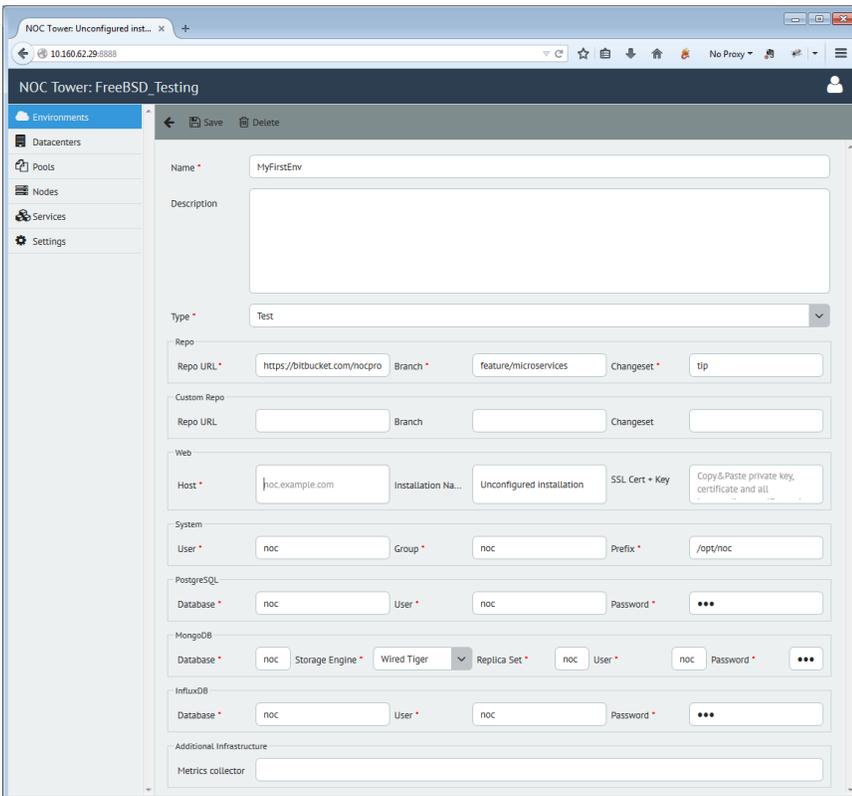
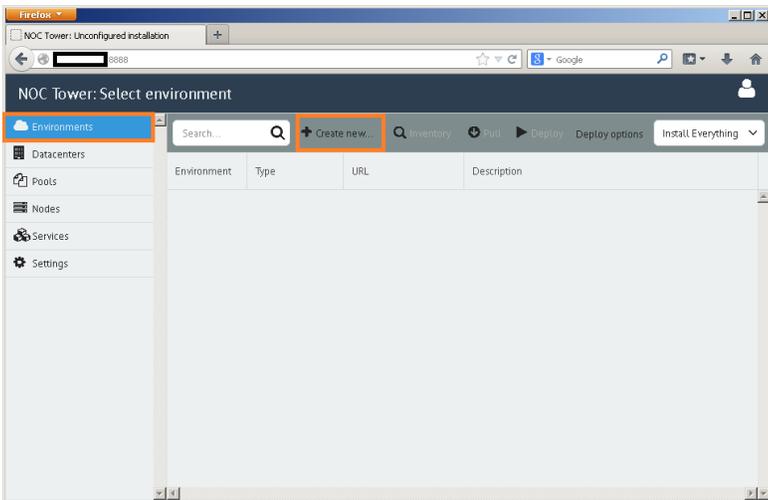
Инструкция по ручной установке пос-tower: <https://bitbucket.org/nocproject/noc-tower/overview>

Если при попытке запуска башни появилось сообщение: "tower-web reeewe.OperationalError: unable to open database file" необходимо проверить права на папку var/ - владельцем должен быть пользователь tower (или из-под которого запускается башня).

Настройка окружения (Environments)

Environments

Заходим в веб-интерфейс Башни и оказываемся в пункте "Environments". Переходим по клавише "Create new" в форму создания окружения.



В случае первой установки можно оставить по умолчанию (или поменять только пароли к базам). В поле host необходимо добавить IP адрес машины, на которой будет устанавливаться веб-интерфейс НОКа (IP адрес в формате XX.XX.XX.XX)

! Названия окружения допустимы только буквенно-цифровые символы. Наличие специсмволов приведёт к невозможности развёртывания NOC'a.

Тип на данный момент можно выбрать любой, единственное отличие - при типе Productive папка с NOC'ом очищается при каждом развёртывании.

После выполнения всех настроек нажимаем клавишу Save.

Pull

Следующий шаг - это скачивание файлов, необходимых для развёртывания NOC'a (это playbook'и для ансибля). Для этого: выделяем в пункте меню "Environments" созданное окружение, и нажимаем кнопку "Pull". Запустится процесс загрузки файлов дистрибутива НОКа и правил развёртывания.

Datacenters

Далее выделяем окружение и переходим на пункт меню "Datacenters". Нажимаем кнопку "Create new" сверху и создаём "Datacenters" (при необходимости прописываем http прокси в формате IP:port)

 Datacenters образно говоря это множество машин на которые будут разворачиваться компоненты нока. На данный момент там доступна только одна опция - http_proxy.

Pools

Далее переходим в пункт меню "Pools". Здесь проверяем что уже создан пул default.

 Pools или пулы это сущность NOC'a, в пределах пула NOC раздаёт задания для выполнения (по сборке конфигураций, опросу оборудования и пр.).

Nodes

Далее переходим в пункт меню "Nodes". В нём добавляем наши ноды (хосты), на которые будут разворачиваться компоненты NOC'a (в нашем случае нода одна - пос-main01).

 Name ноды должен совпадать с hostname сервера, который вы добавляете. Иначе возникнут проблемы при запуске NGINX.
При развёртывании НОКа на тот же самый сервер, необходимо указывать его имя и внешний IP адрес!

В "Node Type" выбираем "FreeBSD" (мы же пишем руководство для FreeBSD:), адрес заполняем IP адрес ноды пос-main01, в поле login As должен находиться ansible (менять его необходимо только в случае если для развёртывания NOC'a применяется другой пользователь). После заполнения всех обязательных полей нажимаем "Save".

Services

Переходим в самое главное меню - Services. В нём определяется на какой хост, какие службы будут установлены. Поскольку, в нашем случае, хост у нас 1 - это облегчает нам задачу - всё будем ставить на него.

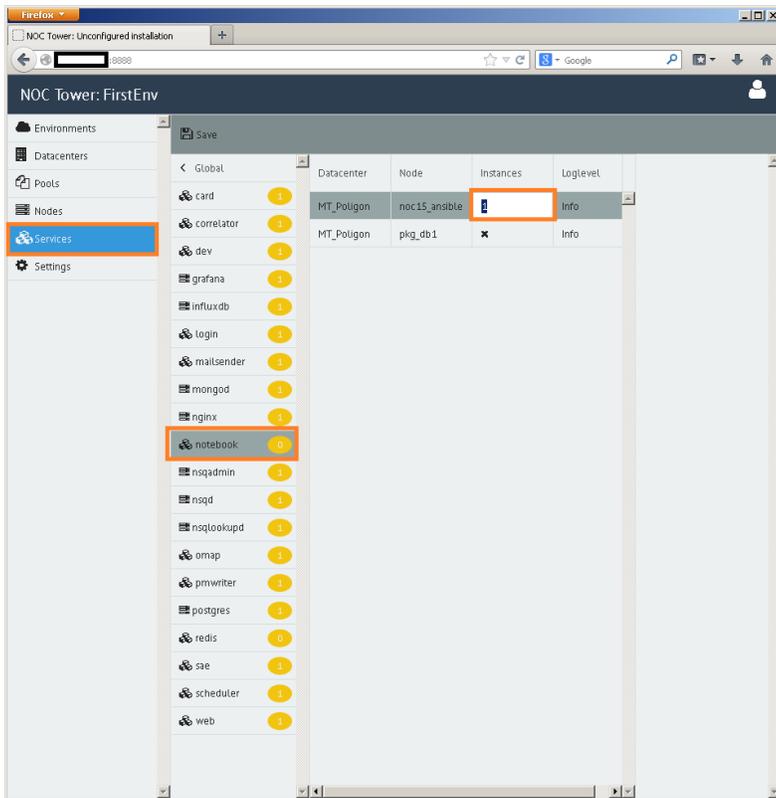
 Если в подпунктах меню "Global" и "default" пусто, значит на первой странице - Environments - не был выполнен Pull. Либо он завершился неудачно - необходимо выяснить причину неудачного выполнения по логам и повторно его запустить

Кликаем на подменю "Global", проходим по всем пунктам и везде напротив нашего хоста проставляем галочку:

 Будьте очень аккуратны при простановке галочек. Фиксация галочки происходит при событии потери фокуса с поля. Поэтому делаем так:

1. Выбираем пункт - щёлкаем на него мышкой.
2. Выбираем ноду, на которую будет сервис
3. Щёлкаем по полю Instances. Вписываем цифру 1
4. После этого щёлкаем мышкой на любой свободной области в пределах окна браузера.
5. Повторяем действия для всех необходимых пунктов.

Если так случилось - что произошло переключение на следующий пункт без фиксации значения Instances - необходимо обновить окно браузера (F5) и начать заново.



После распределения инстансов - нажимаем кнопку Save сверху и переходим к следующему пункту. Проверить правильность установки можно, обновив окно браузера, после нажатия кнопки Save: напротив сервисов (в жёлтом кружке) должна появиться цифра 1.

Те же действия повторяем с Пулом default (именно в сервисах можно определить на какие hosts будут установлены различные сервисы для пулов, и если были созданы пулы кроме default - они покажутся здесь). После внесения всех изменений нажимаем Save.

Осталась последняя страница - Settings. На ней находятся настройки местонахождения Башни. Необходимо посмотреть что тут всё верно и нажать Save.



Если не нажать Save на странице Settings, то при развёртывании вы получите ошибку на этапе "TASK [node : Pull NOC] "

Проверки

После выполнения всех настроек имеет смысл посмотреть, а нормально ли отработает Ansible. Для этого, на хосте, на котором работает башня необходимо:

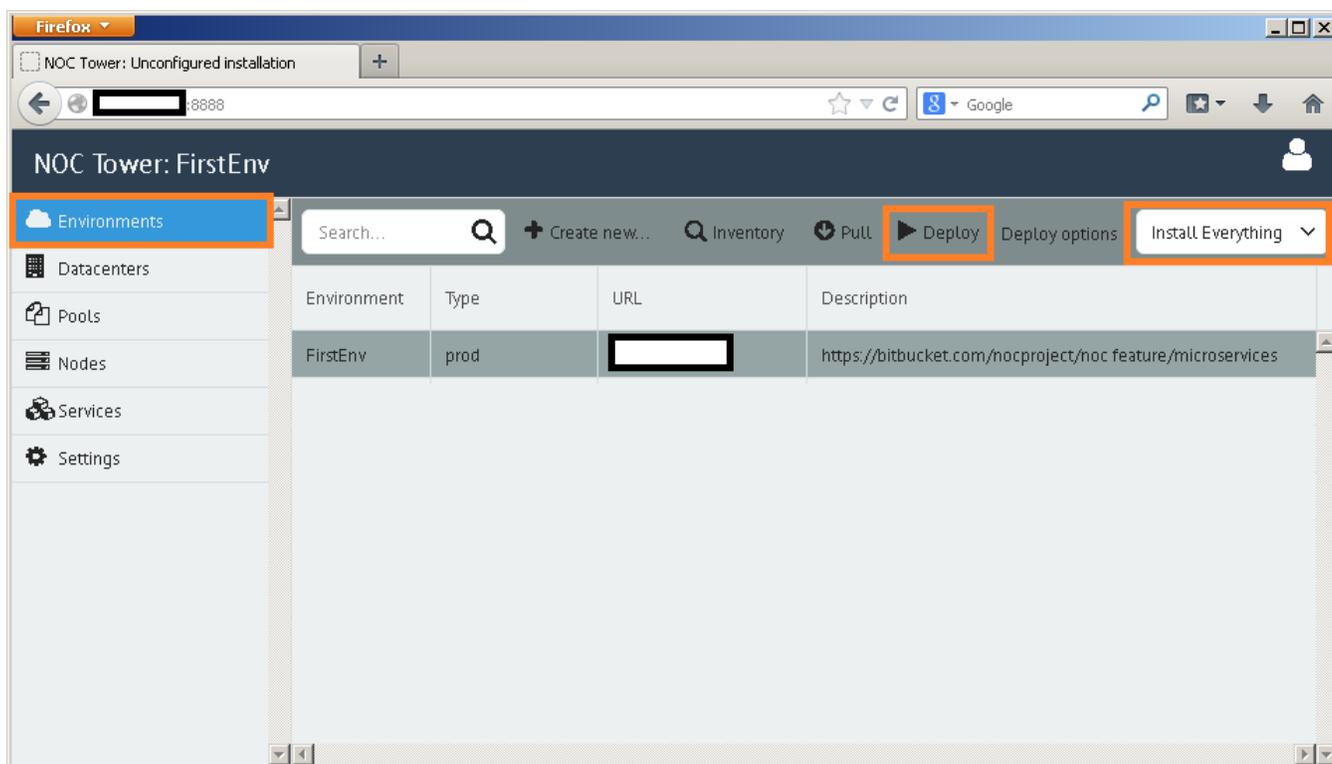
```
# su - tower
# cd /usr/local/tower
# setenv NOC_ENV MyFirstEnv
# ./bin/ansible -vvvv -m ping -c ssh -u ansible -i bin/tower-inv noc-main01
```

После выполнения мы должны получить:

```
noc-main01 | SUCCESS => {
  "changed": false,
  "invocation": {
    "module_args": {
      "data": null
    },
    "module_name": "ping"
  },
  "ping": "pong"
}
```

Развёртывание (Deploy) NOC'a

После того как все настройки сделаны и выполнены проверки. Теперь можно запустить деплой. Для этого переходим в пункт меню "Environments", выбираем созданное нами окружение и нажимаем клавишу "Deploy".



Всё. Можно идти отдыхать минут на 10-15. Ansible будет делать работу за вас.

После установки

Материалы

- [Описание установки башни.](#)
- [FAQ](#)
- [Глоссарий NOC-Tower](#)

Список ошибок

```
TASK [migrate : Synchronize collections] 04:06
fatal: [Node1]: FAILED! => {"changed": true, "cmd": ["/noc", "collection", "sync"], "delta": "0:00:01.862260",
"end": "2016-07-24 17:20:51.126712"}
```

Часто встречающаяся ошибка при синхронизации коллекций. Необходимо пройти на НОДУ с ноком и из папки НОКа выполнить (возможно, придётся повторить команду несколько раз):

```
# ./noc collection sync
```

```
TASK [post : check if noc running] 00:43
fatal: [Node1]: FAILED! => {"changed": true, "cmd": "/opt/noc/noc ctl status", "delta": "0:00:00.007324",
"end": "2016-07-24 17:13:09.872542", "failed": true, "rc": 127, "start": "2016-07-24 17:13:09.865218",
"stderr": "./commands/ctl.sh: 2: exec: ./bin/supervisorctl: not found", "stdout": "", "stdout_lines": [],
"warnings": []}
```

В списке опций развёртывания - "Deploy Options" (список справа от Deploy) не выбрана опция "Install Everything".