# Install NOC Tower on Debian 8



Конфигурации NOC разрослась настолько сильно, что сконфигурировать NOC на крупной инсталляции руками стало очень сложно. В новой версии весь NOC разбили на отдельные сервисы – микросервисы ( детальнее о микросервисах ), которыми управляет служба supervisord. В связи со сменой архитектуры системы NOC, изменился и способ установки. Теперь установка стала на много удобнее, нагляднее и ее легче контролировать. Нужно установить Tower и создании node-ов.

Tower позволяет сделать реестр сервисов (service registry).  Работа башни основана на системе менеджмента конфигурации Ansible. Ansible оркестрирует, конфигурирует, администрирует или разворачивает  системы. Частично playbook названы по спортивной аналогии (схема игры), и подразумевается, что использовать их будет весело и что это не рабочая тетрадь (workbook) (Перевод с  сайта проекта).  (Статья по Ansible)

Т.е. башня и ноды отдельные друг от друга конструкции. Мы рекомендуем разделить командную ноду и саму систему, хотя при желании можно поставить и на один хост.

Подготовим, по одной машине (образу) – для NOC Tower  и  NOC-node1. Количество нод выбирается в зависимости от сложности инсталляции. Обратите внимание на предложения по сборке там описаны типичные сценарии и рекомендации.

> В терминологии системы, нода - это машина на которой будут работать сервисы NOC или часть сервисов системы при распределенной системе. (Например: DB или web сервер)

> Обратите внимание, что башня считает, что она единственный пакет в системе и довольно фривольно меняет дефолты системы в том числе sysctl.

## Подготовка Ноды

Я думаю Debian вы устанавливали не раз, так что детальную установку системы мы опустим. В случае если возникнут затруднения можно воспользоваться  хорошей статьей.

Для установки Debian лучше всего использовать минимальный образ (netintstall), а при выборе пакетов отменить установку всех дополнительных пакетов (X-сы, web-server, стандартные утилиты…), оставив только ssh – нужные пакеты, в дальнейшем, установит башня сами.

При установке системы выбираем нужную локаль, создаем стандартного пользователя, размечаем диск, создаем файловую систему. По большому счету при простой инсталяции, разметку диска, можно доверить инсталлятору дистрибутива или использовать все пространство диска, но можно и позаботится о проблемах переполнения корневого раздела или ограничения количества inode-ов и подумать о iops.

Если все-таки решили разделить дисковое пространство можно разделить по своему усмотрению, учитывая:

- · что при mbr всего четыре основных раздела;
- · в NOC используются три базы ( mongo, postgres, clickhouse ) для каждой можно выделить раздел под данные;
- · также, есть еще две базы, которые могут записывать данные на диск, хоть и небольшое количество, но при пиковой нагрузке, достаточно интенсивно, ( nsqd );
- · можно выделить, также раздел под журналы и индексы.

Стандартные пути:

/opt/noc

Базы:

/var/lib/clickhouse

/var/lib/mongo ( https://docs.mongodb.org/manual/administration/production-notes )

/var/lib/postgresql

/var/spool/nsqd

Все остальные разделы по вашему вкусу и в соответствии с потребностями.

При разбиение дискового пространства, особенно в простых инсталяциях, создатели NOC советуют, что: «нет смысла грести все инсталяции под одну гребенку. В большинстве случаев даже отдельная точка монтирования под базы данных - оверкил. Ведь за ними придется приглядывать.»

«Вообще этот релиз очень удобен для децентрализованной системы - базы данных в одном месте, воркеры в другом, активаторы и веб еще где-то…» и при сложных инсталяциях это ощутимо увеличит производительность.

Если вы все же не совсем уверены, какой объем дискового пространства выделить под нужды системы, можно использовать менеджер логических дисков LVM, чтобы в дальнейшем расширить объем (Неплохая инструкция для Debian:). Также неплохо использовать XFS и можно обратить внимание на технологию для SSD, которая для Postgress называется WAL и journal в mongo

## Теперь позаботимся о нужных пакетах

```
root@noc-node1:/# apt-get install python python-dev sudo dbus
```

пакет ssh - метапакет, с ним поставится openssh-client и openssh-server, другие пакеты нужны для сборки.

Кроме локального пользователя на noc-node1, и пользователя root, нужно будет создать дополнительного пользователя ( Например: ansible ) - под этим пользователем noc-tower будет настраивать данную Ноду.

```
root@noc-node1:/# useradd -d /home/ansible -s /bin/bash -m ansible
```

Устанавливаем и запоминаем пароль - в дальнейшем пригодится при копировании сертификата с NOC-tower.

Чтобы башня смогла настраивать Ноды - ansible должен иметь возможность сделать `sudo -s` *без пароля* –для этого в файле /etc/sudoers добавляем строку:

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

Пользователь noc будет создан автоматически башней.

На этом настройка Ноды закончена!

Т.е. ***главная задача*** при настройки ноды это настроить доступ башни к ноде, а дальше башня сделает все в автоматическом режиме.

# Подготовка NOC-tower

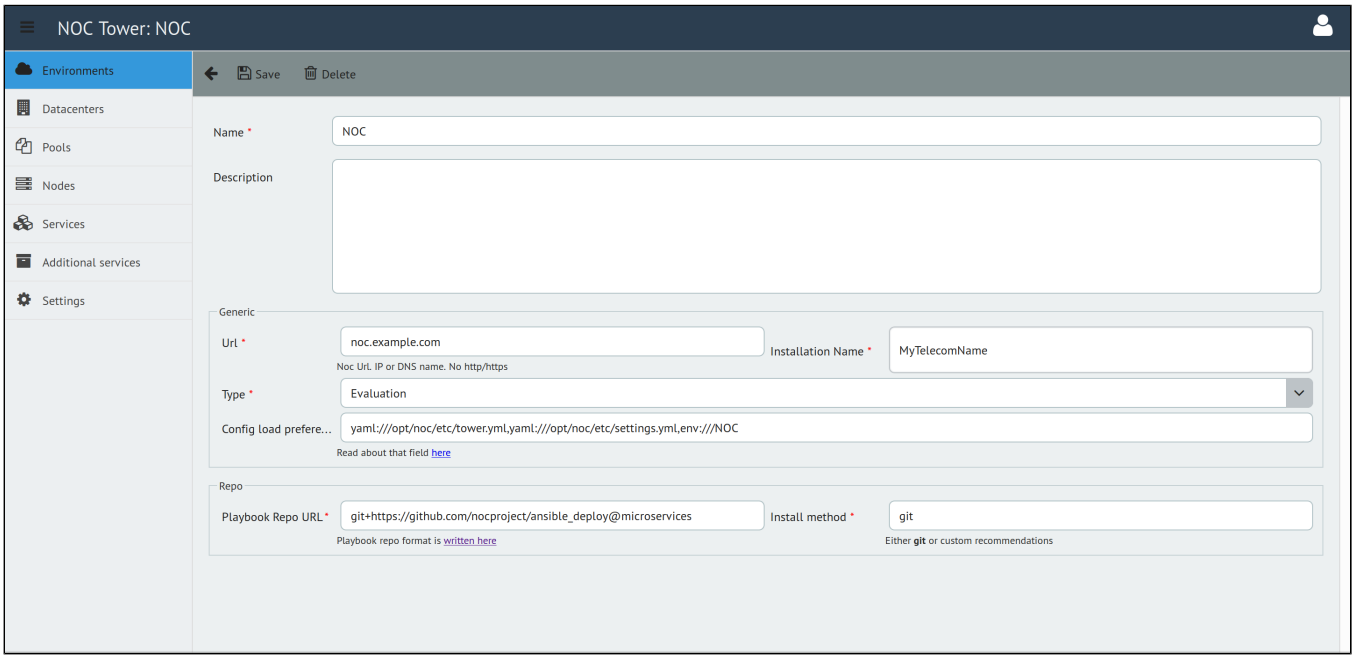
Во многом начальная настройка башни похожа на настройку НОД - чистая netinstall без X и дополнительных утилит.

Установка башни описана в сопроводительной документации к башне.  https://code.getnoc.com/noc/tower/blob/master/Readme.md

## Теперь разворачиваем NOC

`http://noc-tower:8888/`

Создаем  Environment




Environments:

- заводим новую запись

В имени Environment нельзя использовать спецсимволы - только буквы и цифры.

Типы `Environment`

Evaluation если будешь пилить и добавлять скрипты локально. (Если не будешь, то печально...)

Productive - на текущий момент отличие только одно - в этом режиме Башня заботиться о поддержании в чистоте каталога `/opt/noc`. *Все файлы в этом каталоге, которые не внесены в репозитарий - удаляются* ( hg revert - команда  ansible)! Нельзя, чтобы в этом режиме были измененные файлы.

- **Url - задается dns-имя или ip-адрес ноды, которая будет обслуживать web-сервис.**

Выбираем нужное окружение и нажимаем Pull

При нажатии кнопки Pull скачивается PlayBook для ansible.

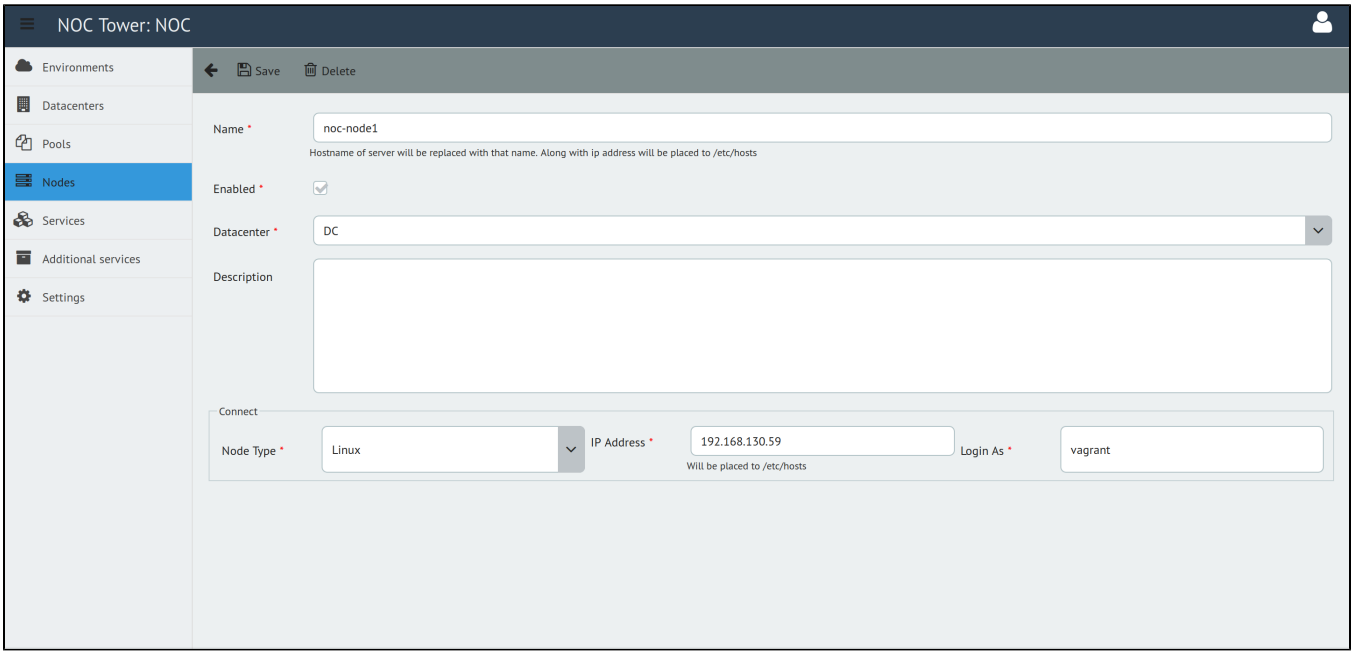
Если браузер заглючило и пришлось перезагрузить страницу, то не забудьте снова выбрать нужное окружение и только после этого производить настройки в других пунктах меню.

Datacenters:

- заводим новую запись

Nodes:

- заводим новую запись;
- выбираем нужный Datacenter (созданный ранее);
- прописываем IP и login от серверов noc-node;
- в вебе башни прописываем  пользователя, с помощью которого башня будет настраивать НОДы, в нашем случае значение по-умолчанию– ansible.

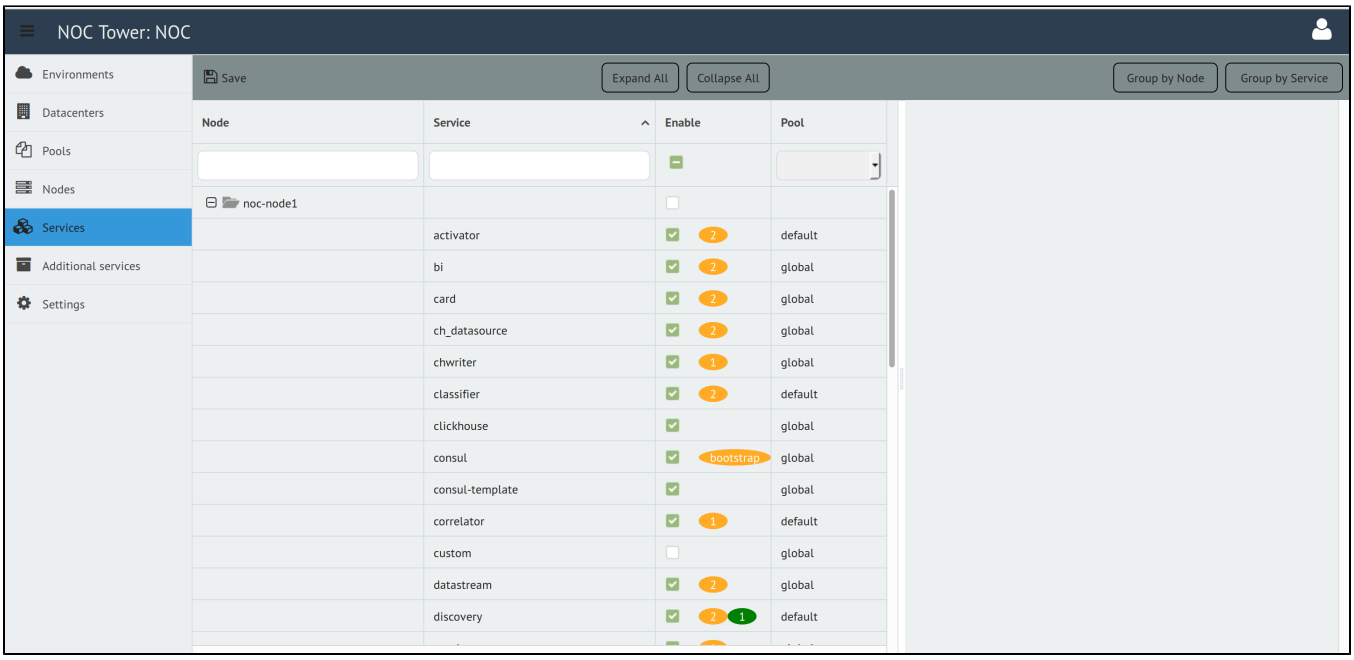



Ноды строго относятся к выбранному в данный момент Environment! Поэтому, если, допустим, есть желание использовать 1 сервер Баз данных для 2 Environment, то ноду необходимо добавить 2 раза (для первого и второго Environment). При этом все параметры могут совпадать (она будет отличаться только ID в базе).

***На вкладке Services:***

Выбираем сервисы которые будут работать на нужной ноде. В инсталляции по умолчанию стоит просто выбрать все.

***В глосарии можно найти описание сервисов. Так же есть некоторое описание для каждого при его выборе***




Если захотелось разнести базы и другие сервисы по разным нодам-серверам, то в нужном окружении необходимо добавить разные ноды и, затем, в сервисах поставить галочку напротив ноды

## Запускаем установку

Environments:

- выбираем запись
- нажимаем Deploy
- ждем окончания и радуемся!

*Если ставим первый раз - Deploy может быть долгим. около 20 минут и сильно зависит от качества интернета*

Deploy нужно делать каждый раз после внесения изменений в вебе Башни.