

Manual Installation

Installation directory

NOC installed in `/opt/noc/` directory by default.

Required packages

Python

Ensure Python 2.6 or later installed. Check your installed python version by:

```
# python --version
```

Python 2.7 is recommended

 NOC is not compatible with Python 3

 Common installation troubles are caused by several different versions of python installed in the system. If your system have several versions of Python installed ensure you calling right version when installing required packages, NOC and performing NOC maintenance. Please note, *noc* and *root* users and daemon environment may have different **\$PATH**. Ensure you are calling proper Python version each case.

PostgreSQL

[PostgreSQL](#) 8.4 or later required. Please install PostgreSQL according to your operation system requirements. PostgreSQL 9.0 is recommended.

 Common installation troubles are caused by two versions of PostgreSQL installed in the system. If your system have several versions of PostgreSQL installed ensure you calling and linking with right version when installing required packages, NOC and performing NOC maintenance. Please note, *noc* and *root* users and daemon environment may have different **\$PATH** and **\$!** **NCLUDE**. Ensure you are calling proper PostgreSQL version each case.

MongoDB

[MongoDB](#) 2.4 or later required. Please install MongoDB according to your operation system requirements. MongoDB 2.6 is recommended

 MongoDB uses mmap() method to access database. Ensure you have 64-bit OS in order to use databases larger than 2Gb

pip

[pip](#) is the tool for installing and managing python packages.

Install pip from your system package.

virtualenv

[virtualenv](#) is the tool to create isolated python environment. Install virtualenv from your system package or via pip

```
# pip install virtualenv
```

HTTP Server

Though the NOC is bundled with built-in [HTTP Server](#), additional front-end HTTP-server with reverse proxy is strongly recommended. Commonly used choices are:

- [nginx](#)
- [lighttpd](#)
- [apache](#)

Install HTTP server according your system's requirements. Refer to [Webserver Setup](#) part for configuration details

libsmi

[libsmi](#) is a library and collection of tools to access SMI/SNMP MIB Information. Please install libsmi according to your operation system's requirements.

Mercurial

[Mercurial](#) is the Distributed Version Control System ([DVCS](#)). Required to fetch updates from repository. Mercurial also is a default format of Configuration Management repository. Mercurial 1.3 or later required. Please install *mercurial* from your system's package or via pip

```
# pip install mercurial
```

Checkout from mercurial repo

Checkout from mercurial repo is a best way to stay on bleeding edge of fresh updates. You need *mercurial* to perform checkout and further update.

To fetch latest updates available:

```
# hg clone http://bitbucket.org/nocproject/noc noc
```

To fetch particular release (0.7(4) in example):

```
# hg clone -r "0.7(4)" http://bitbucket.org/nocproject/noc noc
```

Prepare virtualenv

Initialize virtualenv

```
# virtualenv /opt/noc
```

System Users and Groups

All noc files except */opt/noc/local* directory must be owned by *root*.

All noc daemons are running from *noc* user. Create *noc* user and group before continuing installation:

For Linux

```
# groupadd noc
# useradd -g noc -s /bin/sh -d /home/noc noc
```

For FreeBSD

```
# pw groupadd -n noc
# pw useradd -g noc -s /bin/sh -d /home/noc -n noc
```

Create PostgreSQL database

Create database user *noc* from PostgreSQL superuser account (*postgres* in example), then create database *noc*

```
# su - postgres
postgres@/$ psql
postgres=# CREATE USER noc SUPERUSER ENCRYPTED PASSWORD '<password>';
postgres=# CREATE DATABASE noc ENCODING 'UTF8' OWNER noc;
postgres=# \q
```

Set MongoDB password

```
# mongo
MongoDB shell version: 2.0.1
connecting to: test
> use noc
switched to db noc
> db.addUser("noc", "noc")
...
```

Configure upgrade system

```
# cp /opt/noc/etc/upgrade.defaults /opt/noc/etc/upgrade.conf
```

Edit `/opt/noc/etc/upgrade.conf` to your local requirements

Run upgrade

Run upgrade for the rest of your system installation

```
# /opt/noc/scripts/upgrade
```

Revoke PostgreSQL superuser permissions

Revoke superuser permissions from PostgreSQL's *noc* user

```
# su - postgres
postgres@/$ psql
postgres=# ALTER USER noc NOSUPERUSER;
postgres=# \q
```