

FreeBSD

- [Installing requirements from ports](#)
- [Installing python modules](#)
- [Changing /etc/rc.conf](#)
- [Configuring PostgreSQL:](#)
 - [mongo](#)
 - [Создаем системного пользователя для NOC](#)
 - [Стягиваем с репозитория NOC:](#)
 - [Ставим NOC](#)
 - [Настраиваем конфиги:](#)
 - [Завершающие штрихи:](#)
 - [Configuring nginx:](#)
 - [Запускаем nginx](#)
 - [запускаем NOC](#)
 - [go to <http://noc.xxx.ru>](#)

Installing requirements from ports

```
# cd /usr/ports/databases/postgresql92-server
/usr/ports/databases/postgresql92-server# make install clean
# cd /usr/ports/databases/postgresql92-contrib
/usr/ports/databases/postgresql92-contrib# make install clean
# cd /usr/ports/databases/postgis
/usr/ports/databases/postgis# make install clean
# cd /usr/ports/databases/mongodb
/usr/ports/databases/mongodb# make install clean
# cd /usr/ports/lang/python27/
/usr/ports/lang/python27/# make install clean
# cd /usr/ports/devel/py-setuptools/
/usr/ports/devel/py-setuptools/# make install clean
# cd /usr/ports/math/py-gmpy
/usr/ports/math/py-gmpy# make install clean
# cd /usr/ports/devel/py-cjson
/usr/ports/devel/py-cjson# make install clean
# cd /usr/ports/devel/mercurial/
/usr/ports/devel/mercurial/# make install clean
# cd /usr/ports/net/fping
/usr/ports/net/fping# make install clean
# cd /usr/ports/net-mgmt/libsmi
/usr/ports/net-mgmt/libsmi# make install clean
# cd /usr/ports/security/gnupg
/usr/ports/security/gnupg# make install clean
# cd /usr/ports/net/rsync
/usr/ports/net/rsync# make install clean
# cd /usr/ports/www/nginx/
/usr/ports/www/nginx/# make install clean
```

Installing python modules

```
# easy_install pycrypto pycopg2 netifaces gmpy cjson
```

Changing /etc/rc.conf

```
postgresql_enable="YES"
postgresql_class="postgres"
postgresql_data="/usr/local/pgsql/data"
mongodb_enable="YES"
nginx_enable="YES"
noc_enable="YES"
```

Configuring PostgreSQL:

```
# /usr/local/etc/rc.d/postgresql initdb
```

 **Hint:** Tune `/usr/local/pgsql/data/postgresql.conf` if required

```
# /usr/local/etc/rc.d/postgresql start
```

```
# su postgres
$ createuser noc
Shall the new role be a superuser? (y/n) y
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
$ createdb -EUTF8 -Onoc noc
psql -d templatel -c "ALTER USER noc WITH PASSWORD 'noc_password';"
psql -f /usr/local/share/postgis/contrib/postgis-1.5/postgis.sql -d noc
```

настраиваем `pg_hba.conf` как нам нужно для безопасности.

mongo

запускаем mongo

```
# /usr/local/etc/rc.d/mongod start
# mongo
> use noc
> db.addUser('noc', '12345');
> db.auth('noc', '12345');
> exit
```

Создаем системного пользователя для NOC

```
# pw add group noc
# pw add user noc -g noc -s /bin/csh -d /home/noc
# passwd noc
# mkdir /home/noc
# chown noc:noc /home/noc
```

Стягиваем с репозитория NOC:

```
# cd /usr/local/
/usr/local# hg clone http://hg.nocproject.org/noc noc
```

Ставим NOC

По дефолту ставится в /opt/noc/, мне кажется для FreeBSD идеологически вернее заменить на /usr/local/noc.

Данные хранятся в /var/repo и /var/backup. Меняем их соответственно на /var/db/noc/repo и /var/db/noc/backup.

Pid файлы хранятся по умолчанию в каталоге /var/log/noc. Меняем его на /var/run/noc.

```
# cd ./noc
/usr/local/noc# ./scripts/post-install -p /usr/local/noc -r /var/db/noc/repo -b /var/db/noc/backup -P /var/run/noc
```

Настраиваем конфиги:

```
#cd /usr/local/noc/etc
```

в noc.conf изменяем пути к необходимым бинарникам

```
ssh = /usr/bin/ssh
rsync = /usr/local/bin/rsync
pg_dump = /usr/local/bin/pg_dump
tar = /usr/bin/tar
gzip = /usr/bin/gzip
smidump = /usr/local/bin/smidump
smilint = /usr/local/bin/smilint
dig = /usr/bin/dig
pgpg = /usr/local/bin/pgpg
mongodump = /usr/local/bin/mongodump
```

прописываем параметры БД:

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

[nosql_database]
# MongoDB database name
name = noc
# MongoDB database user
user = noc
# MongoDB database password
password = 12345
#
host = 127.0.0.1
#
port =
```

в noc-activator.conf

```
[activator]
name = default
listen_instance = 0
listen_traps = <listen_ip>
listen_syslog = <listen_ip>
listen_pm_data = 127.0.0.1
secret = secret
software_update = true
max_scripts = 10
pm_data_secret = secret
```

В noc-probe.conf

```
[path]
fping = /usr/local/bin/fping

[activator]
host = 127.0.0.1
port = 19704
local_address = 127.0.0.1
secret = secret
```

В noc-launcher.conf

```
[noc-activator]
enabled = true
user = root
group =
config.0 = etc/noc-activator.conf
config.1 = etc/noc-activator.conf
config.2 = etc/noc-activator.conf
config.3 = etc/noc-activator.conf
config.4 = etc/noc-activator.conf
```

```
5 .
user = root ( /snmp, - noc net.inet.ip.portrange.reservedhigh=160)
```

Завершающие штрихи:

```
/usr/local/noc# ./scripts/sync-contrib
/usr/local/noc# chown noc:noc etc/ssh
/usr/local/noc# su noc
$ cd /usr/local/noc
/usr/local/noc$ ./scripts/post-update
```

Configuring nginx:

In */usr/local/etc/nginx/nginx.conf*

```

worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name <yourdomain>;
        charset utf-8;
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root /usr/local/www/nginx-dist;
        }
        location /media/ {
            alias /usr/local/noc/contrib/lib/django/contrib/admin/static/;
            gzip on;
            gzip_types text/css text/x-js;
        }
        location ~ ^/static/ {
            root /usr/local/noc;
            gzip on;
            gzip_types text/css text/x-js;
        }
        location ~ ^([^\/]+)/([^\/]+)/([js|css|img])/(\.+) $ {
            root /opt/noc;
            rewrite ^([^\/]+)/([^\/]+)/([js|css|img])/(\.+) $
                /$1/apps/$2/$3/$4 break;
            gzip on;
            gzip_types text/css text/x-js;
        }
        location / {
            proxy_pass http://127.0.0.1:8000/;
            gzip on;
            gzip_types text/css text/x-js;
        }
    }
}

```

Запускаем nginx

```
# /usr/local/etc/rc.d/nginx start
```

запускаем NOC

```
# cd /usr/local/noc
root@usr/local/noc# ./scripts/noc-launcher.py start
```

Если имеется файл запуска в `/usr/local/etc/rc.d` (берется из `/usr/local/noc/share/FreeBSD/rc.d`), то запускать так:

```
# /usr/local/etc/rc.d/noc start
```

go to <http://noc.xxx.ru>

Учетная запись по умолчанию noc/noc

Заходим в меню:

Service Activation › Activators ›

И создаем активатор, аналогично тому, который прописан в `/usr/local/noc/etc/noc-activator.conf`



Примечание: Для реальных применений на сервере с MongoDB должна быть 64-битная платформа.

(By Alexander Maksimkin aka `_4ePTeHok` on <irc://freenode.net> @nocproject.org)