

# Nextcloud @Nginx @Oracle Linux v9.5 with self-signed cert (for virtualization)

## preparations

disable IPv6

```
TODO: sysctl ipv6...
```

```
dnf update
dnf install \
    mariadb \
    wget \
    unzip
```

## Create database

Connect to database server, create database and user for application from wherever it is possible. When safe post-script installation been executed, most probably remote root access is not permitted. Login locally to create new database and user.

```
ssh anton@lt58ncp1dbn1
sudo su
mariadb -u maxscale -p -h 10.120.12.xxx
```

```
CREATE DATABASE ncp1 CHARACTER SET utf8mb4;
CREATE USER 'ncp1rw'@ '%' IDENTIFIED BY 'superpass';
```

```
GRANT ALL ON ncp1.* TO 'ncp1rw'@'%';  
FLUSH PRIVILEGES;  
SHOW GRANTS FOR ncp1rw;
```

```
+-----+  
| Grants for ncp1rw@% |  
+-----+  
| GRANT USAGE ON *.* TO `ncp1rw`@`%` IDENTIFIED BY PASSWORD  
'*BD9925F1D4C650B93F105762F0FC7F494AD66AC8' |  
| GRANT ALL PRIVILEGES ON `ncp1`.* TO `ncp1rw`@`%` |  
+-----+  
2 rows in set (0.009 sec)
```

Check connectivity to database server from application server

```
[root@lt58ncp1app1 anton]#  
mariadb -h lt58ncp1dbn1 -u ncp1rw -p
```

# Generate certificates (self-signed)

Prepare storage

```
export fqdn="host"  
export dir="/data/certs/"  
mkdir -p ${dir}/CA  
mkdir -p ${dir}/${fqdn}  
cd ${dir}/CA  
pwd
```

Create CSR and certificate

TODO: add echoing pass to file and from \${fqdn}.pass

create key for CA, give passphrase minimum of four(4) symbols (you want complicated for stronger setups)

```
openssl genrsa -out ca.key 2048
```

create request for CA certificate

```
openssl req -new -sha256 -key ca.key -out ca.csr
```

create a CA certificate

```
openssl x509 -req -days 3600 -in ca.csr -out ca.crt -signkey ca.key
```

generate a key for new certificate for server (modern systems accept key size minimum 2048)

```
openssl genrsa -out server.key 2048
```

create request for the new certificate

```
openssl req -new -sha256 -key server.key -out server.csr
```

sign request for new certificate with the CA

```
openssl x509 -req -sha256 -days 3600 -in server.csr -signkey server.key -out server.crt
```

Now we have pair of certificate and key, we can rename them

```
ls -la

mv ${dir}/CA/server.* ${dir}/${fqdn}/
cd ${dir}/${fqdn}/
mv server.csr ${fqdn}.csr
mv server.crt ${fqdn}.crt
mv server.key ${fqdn}.key
```

Fix SELinux contexts for certificates

# Install Nginx webserver

```
dnf install ngi
```

```
ss -ntap | grep nginx
```

```
LISTEN 0      511          0.0.0.0:80    0.0.0.0:*    users:(("nginx",pid=3459,fd=6),("nginx",pid=3457,fd=6))
LISTEN 0      511          [::]:80      [::]:*      users:(("nginx",pid=3459,fd=7),("nginx",pid=3457,fd=7))
```

## Create firewall rules for webserver

```
firewall-cmd --add-service=http --permanent
firewall-cmd --add-service=https --permanent
systemctl restart firewalld
firewall-cmd --list-all
```

## Prepare local storage for application (not for data)

```
sudo su
export dir="/var/www/"
mkdir -p ${dir}
cd $dir
pwd
```

Download nextcloud and check integrity. Decide which version is going to be deployed. Rule of thumb is to go current major version minus one.

```
export v=29
wget https://download.nextcloud.com/server/releases/latest-$v.zip
curl https://download.nextcloud.com/server/releases/latest-$v.zip.sha256
sha256sum latest-$v.zip
unzip latest-$v.zip
```

## Change permission to nginx's configuration and applications directories

```
cat /etc/nginx/nginx.conf | grep user
mkdir -p ${dir}/nextcloud/data/
chown -R nginx:nginx ${dir}/nextcloud/data/
chown -R nginx:nginx ${dir}/nextcloud/config/
chown -R nginx:nginx ${dir}/nextcloud/apps/
namei -mo ${dir}/nextcloud/config
```

## Create nginx configuration file

```
nano /etc/nginx/conf.d/server.conf
```

[https://docs.nextcloud.com/server/latest/admin\\_manual/installation/nginx.html](https://docs.nextcloud.com/server/latest/admin_manual/installation/nginx.html)

## Install prerequisites

```
dnf install \  
  php \  
  php-fpm \  
  php-mysqlnd \  
  php-zip \  
  php-xml \  
  php-mbstring \  
  php-curl \  
  php-gd
```

## Enable php-fpm (v8.0)

```
systemctl enable php-fpm  
systemctl start php-fpm  
systemctl status php-fpm
```

Modify webserver config to forward requests to correct socket. Config file comes with php-fpm package, but check and adjust configs:

```
cat /etc/nginx/conf.d/php-fpm.conf
```

## Determine where PHP socket is listening

```
fgrep -irn www.sock /etc/php-fpm.d/
```

```
/etc/php-fpm.d/www.conf:38:listen = /run/php-fpm/www.sock
```

## Reconfigure config

```
vi /etc/nginx/conf.d/host.conf
```

```
upstream php-handler {  
  # server 127.0.0.1:9000;  
  # server unix:/run/php/php8.2-fpm.sock;  
  server unix:/run/php-fpm/www.sock;  
}
```

Enable logging: check log file is created after reloading nextcloud page and it can be tail'ed

```
mkdir -p /var/www/nextcloud/log  
chown -R www-data:www-data /var/www/nextcloud/log/  
nano /var/www/nextcloud/config/config.php  
ls -la /var/www/nextcloud/log/  
tail -f /var/www/nextcloud/log/nextcloud.log
```

---

Revision #4

Created 20 February 2025 04:59:42 by Anton

Updated 20 February 2025 12:14:26 by Anton