Nextcloud

- Nextcloud 15 on Ubuntu 18.04 LTS Server
- Spreed Standalone Signaling Server for Nextcloud

Nextcloud 15 on Ubuntu 18.04 LTS Server

This guide assumes you have a working Ubuntu 18.04 LTS server installation and you have sudo provileges.

Credits:

Carsten Reiger

Become root

It's much easier to type in commands without having to worry about typing "sudo" in front of every command and having to authenticate each time. The command below will allow you to become root and only authenticate once:

```
sudo su
```

Type in the password of your user account with root prileges:

```
[sudo] password for SOME_USER:
```

Install MySQL

Install MySQL with the following command:

```
sudo apt install mysql-server
```

Check if MySQL server is running:

```
sudo systemctl status mysql
```

You should get an output similar to below:

```
● mysql.service - MySQL Community Server
Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
Active: active (running) since Wed 2019-01-02 11:31:30 UTC; 1min 18s ago
Main PID: 2578 (mysqld)
Tasks: 27 (limit: 4915)
CGroup: /system.slice/mysql.service
└─2578 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
Jan 02 11:31:30 cloud systemd[1]: Starting MySQL Community Server...
Jan 02 11:31:30 cloud systemd[1]: Started MySQL Community Server.
```

Secure MySQL

MySQL server package comes with a script called mysql_secure_installation that can perform several security related operations.

Run the script by typing:

```
sudo mysql_secure_installation
```

You will be asked to setup VALIDATE PASSWORD plugin. You can simply press ENTER to skip or "Y" to setup:

```
Securing the MySQL server deployment.
Connecting to MySQL using a blank password.
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
Press y|Y for Yes, any other key for No
```

On the next prompt you will be asked to set up the password for the MySQL root user. Specify and confirm the password:

```
New password:
Re-enter new password:
```

On the next prompt you will be asked to remove the MySQL anonymous user. You should select "Y" in this prompt:

```
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
```

```
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No) :
```

On the next prompt you will be asked to Disallow root login remotely. You should select "Y" in this prompt:

```
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.
Disallow root login remotely? (Press y|Y for Yes, any other key for No) :
```

On the next prompt you will be asked to remove the test database and the access to it. You should select "Y" in this prompt:

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No) :
```

On the next prompt you will be asked to reload the privilege tables. You should select "Y" in this prompt:

```
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for No) :
```

Create the MySQL Database

Login to the MySQL shell:

```
sudo mysql
```

Run the following SQL statements to create a database named nextcloud, user named nextclouduser and to grant all necessary permissions to the user where SOME_PASSWORD is a strong password you specify:

```
CREATE DATABASE nextcloud CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
GRANT ALL ON nextcloud.* TO 'nextclouduser'@'localhost' IDENTIFIED BY 'SOME_PASSWORD';
FLUSH PRIVILEGES;
EXIT;
```

Install PHP and Apache

Install PHP and Apache2 packages:

```
sudo apt install apache2 php php-gd php-json php-mysql php-curl php-mbstring php-intl php-imagick php-xml php-zip php-ldap php-smbclient php-soap libapache2-mod-php
```

Download Nextcloud

As of writing this article, the latest Nextcloud version was 15.0.0. Visit the Nextcloud Downloads page to get the link the latest version:

Once you have the link, download using wget to your system:

```
wget https://download.nextcloud.com/server/releases/nextcloud-15.0.0.zip
```

Install unzip if you don't already have it:

```
apt install unzip
```

Unzip the Nextcloud archive you downloaded previously:

```
unzip nextcloud-15.0.0.zip
```

Move the nextcloud directory that got created when you unzipped the Nextcloud archive to the /var/www/html/ directory:

```
mv nextcloud /var/www/html
```

Setup Apache user to have full access to the Nextcloud directory:

```
sudo chown -R www-data: /var/www/html/nextcloud/
```

Configure Apache

Ensure you have obtained a 3rd party PEM certificate, key and PEM certificate chain file and placed them in the corresponding directories before proceeding below.

Create a Nextcloud website configuration:

```
vi /etc/apache2/sites-available/nextcloud.conf
```

Paste the following ensuring you replace the **bold** fields with your information:

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin someone@domain.tld
DocumentRoot /var/www/html/nextcloud
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
SSLEngine on
SSLCertificateFile /etc/ssl/certs/cert.pem
SSLCertificateKeyFile /etc/ssl/private/key.key
SSLCertificateChainFile /etc/ssl/certs/cert_chain.pem
<FilesMatch "\.(cgi|shtml|phtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>
<Directory /var/www/html/nextcloud>
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Require all granted
</Directory>
</VirtualHost>
</IfModule>
```

Enable required modules:

```
sudo a2enmod ssl
sudo a2enmod rewrite
sudo a2enmod headers
sudo a2enmod env
sudo a2enmod dir
sudo a2enmod mime
```

Enable the nextcloud website:

```
cd /etc/apache2/sites-available/
sudo a2ensite nextcloud
```

Disable the Default website:

```
sudo a2dissite 000-default
```

Restart Apache to activate changes:

```
systemctl reload apache2
```

Install Nextcloud

With a browser navigate to the host or IP address of your server using https:

```
https://IP_ADDRESS
```

You will be presented with the Nextcloud setup page:

Enter a desired Username and Password to create an admin account, and enter the Database user, Database password, Database name you created earlier and click on the **Finish setup** button.

Configure Nextcloud Memory Caching

We will be configuring memory caching in order to improve the Nextcloud server performance using Redis for distributed caching as well as local cache for Transactional File Locking.

Install Redis Server and php-redis:

```
apt install redis-server php-redis
```

The installer will automatically launch redis-server and configure it to launch at startup. Verify redis-server is running:

```
ps ax | grep redis
```

should output similar to below (note the port number 6379):

```
17056 ? Ssl 0:00 /usr/bin/redis-server 127.0.0.1:6379
17211 pts/0 S+ 0:00 grep --color=auto redis
```

Edit the Nextcloud config.php file:

```
vi /var/www/html/nextcloud/config.php
```

Add the following entries right above the bottom); entry so it looks like below:

```
'installed' => true,
'memcache.local' => '\OC\Memcache\Redis',
'redis' => array(
'host' => 'localhost',
'port' => 6379,
),
'memcache.locking' => '\OC\Memcache\Redis',
);
```

Reload Apache:

```
systemctl reload apache2
```

Configure PHP OPcache

Find the php.ini file used by Apache:

Create a phpinfo.php file in the /var/www/html/nextcloud directory:

```
vi /var/www/html/nextcloud/phpinfo.php
```

Paste the following in the file and save:

```
<?php
// Show all information, defaults to INFO_ALL
phpinfo();
?>
```

Load the page in your browser:

```
https://IP_ADDRESS/phpinfo.php
```

Should be presented with the following screen:

Look for the Loaded Configuration File line, in this example it's **/etc/php/7.2/apache2/php.ini**:

Edit the Configuration File:

```
vi /etc/php/7.2/apache2/php.ini
```

Look the the [opcache] section of the file and remove the **;** (uncomment) and set the following variables as shown below:

```
opcache.enable=1
opcache.enable_cli=1
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=10000
opcache.memory_consumption=128
opcache.save_comments=1
opcache.revalidate_freq=1
```

Reload Apache:

```
systemctl reload apache2
```

Delete the phpinfo.php file you created earlier:

```
rm -rf /var/www/html/nextcloud/phpinfo.php
```

Convert Database Columns to big int

Take Nextcloud instance offline by editing the /var/www/html/nextcloud/config/config.php file:

```
vi /var/www/html/nextcloud/config/config.php
```

Insert the following entry if it doesn't exist right below the 'maintenance' => true, entry, so it looks like below and save the file:

```
'installed' => true,
'maintenance' => true,
```

Ensure instance is offline by navigating with a browser to the IP or host name of your Nextcloud server:

```
https://IP_ADDRESS/
```

Should present the following screen:

Run the following commands to convert the tables to big int:

```
cd /var/www/html/nextcloud/
chmod +x occ
sudo -u www-data ./occ db:convert-filecache-bigint
```

In the following prompt, click "Y' to proceed:

```
Nextcloud is in maintenance mode - no apps have been loaded
Following columns will be updated:
* filecache.mtime
* filecache.storage_mtime
This can take up to hours, depending on the number of files in your instance!
Continue with the conversion (y/n)? [n]
```

Put the Nextcloud instance back online by editing the /var/www/html/nextcloud/config/config.php file:

```
vi /var/www/html/nextcloud/config/config.php
```

Edit the **'maintenance' => true,** entry and set it to **'maintenance' => false**, so it looks like below and save the file:

```
'installed' => true,
'maintenance' => false,
```

Ensure instance is online by navigating with a browser to the IP or host name of your Nextcloud server:

```
https://IP_ADDRESS/
```

Review any additional setup warning under **Settings --> Administration --> Overview**.

Enable External SMB Storage Support

In order to be able to mount external SMB shares into the Nextcloud instance you must you must install smbclient package:

```
apt install smbclient
```

Change the Nextcloud Data Directory to SMB Storage

If you don't wish to use local storage for the Nextcloud Data directory, you can mount a SMB share via fstab and point the "datadirectory" field in the /var/www/html/nextcloud/config.php to that share.

Create a mount point to mount your SMB share:

```
mkdir /mnt/nextclouddata
```

Create .smbcredentials file to save the credentials to mount the SMB share where "username" is the name of the user you are logged in as:

```
vi /home/username/.smbcredentials
```

Paste the following:

```
(For non-domain based Share)
#username=MyUsername
#password=MyPassword
# OR: (for Windows 2008 and above Domain based Share)
#username=MyUsername
#password=MyPassword
#domain=MyDomain
# OR: (for cifs on Windows Server 2003 Domain Based Share)
# username=MyDomain/MyUsername
# password=MyPassword
```

Uncomment (remove the #) from the username,password and if applicable domain from the section you wish to use and replace the MyUsername, MyPassword and MyDomain if applicable with your information.

Change permissions to the .smbcredentials file to prevent unwanted access to your credentials:

```
chmod 600 /home/username/.smbcredentials
```

Get the www-data user UID in order to mount the SMB share with the www-data user as the owner :

```
id -u www-data
```

Should output similar to below. Take note of the UID (Don't use the UID from below, your system might differ):

```
33
```

Edit /etc/fstab:

```
vi /etc/fstab
```

Paste the following in a new line under all the existing entries where "SERVER" is your SMB server name/IP, "SHARE" is the SMB share, "33" is the www-data UID from above, and "home/username" is the location of the ".smbcredentials" file you created earlier:

```
#MOUNT NEXTCLOUD DATA SHARE
//SERVER/SHARE /mnt/nextclouddata cifs
uid=33,file_mode=0770,dir_mode=0770,credentials=/home/username/.smbcredentials
```

mount the share:

```
mount -a
```

Ensure you can access the files/directories in that share

Edit /var/www/html/nextcloud/config/config.php

```
vi /var/www/html/nextcloud/config/config.php
```

Locate the "datadirectory" field and change it to the /mnt/nextclouddata so it looks like below and save the file:

```
'datadirectory' => '/mnt/nextclouddata',
```

Restart Apache:

```
systemctl restart apache2
```

Nextcloud Command Line Commands Reference

All commands below must be run from the /var/www/html/nextcloud directory or wherver your Nextcloud installation directory is:

```
cd /var/www/html/nextcloud
```

Enable Maintenance Mode

```
sudo -u www-data php occ maintenance:mode --on
```

Disable Maintenance Mode

```
sudo -u www-data php occ maintenance:mode --off
```

Re-Scan Nextcloud Data for ALL Users

```
sudo -u www-data php occ files:scan --all -v
```

Re-Scan Nextcloud Data for specific user

```
sudo -u www-data php occ files:scan <username> -v
```

Cleanup Nextcloud Filecache

```
sudo -u www-data php occ files:cleanup
```

Remove Deleted Files for ALL Users

```
sudo -u www-data php occ trashbin:cleanup --all-users
```

Remove Deleted Files for Specific User

```
sudo -u www-data php occ trashbin:cleanup <username>
```

Set Nextcloud deleted file policy

Edit /var/www/html/nextcloud/config/config.php

```
vi /var/www/html/nextcloud/config/config.php
```

Edit the **trashbin_retention_obligation** line as follows:

```
auto – standard behaviour --> 'trashbin_retention_obligation' => 'auto'
```

```
D, auto – change the minimum days (30 Days) a file is kept with standard behaviour -->
'trashbin_retention_obligation' => '30, auto'
```

```
auto, D – delete after a number of days (30 Days), but earlier if space is required -->
'trashbin_retention_obligation' => 'auto, 30'
```

```
D1, D2 – do not delete before (30 Days), but definitely delete after a certain number of days
(35 Days) --> 'trashbin_retention_obligation' => '30, 35'
```

List ALL Apps

```
sudo -u www-data php occ app:list
```

Disable App for a Specific User

```
sudo -u www-data php occ twofactorauth:disable <username>
```

Enable App for a Specific User

```
sudo -u www-data php occ twofactorauth:enable <username>
```

Disable App for ALL Users

```
sudo -u www-data php occ app:disable gallery
```

Enable app for ALL Users

```
sudo -u www-data php occ app:enable gallery
```

List all Netxloud Configuration Parameters and Remove Sensitive Data

```
sudo -u www-data php occ config:list
```

List All Nextcloud occ Commands

```
sudo -u www-data php occ
```

Enable User

```
sudo -u www-data php occ user:enable username
```

Disble User

```
sudo -u www-data php occ user:disable username
```

Reset User Password

```
sudo -u www-data php occ user:resetpassword user
```

Add User to a Group

```
sudo -u www-data php occ user:add username -g groupname
```

Repair Nextcloud Installation

```
sudo -u www-data php /var/www/html/nextcloud/occ maintenance:repair
```

Spreed Standalone Signaling Server for Nextcloud

About

Spreed standalone signaling server for Nextcloud.

For more information please visit https://github.com/strukturag/nextcloud-spreed-signaling.

Original Credit and Inspiration: MARKUS WEINGÄRTNER

General Requirements

Spreed standalone signaling server requires that you have a fully updated Ubuntu 18.04 (also tested successfully on Ubuntu 20.04) machine with Docker and Docker Compose as well as a fully functioning Nextcloud installation.

Prerequisites

The installation script will prompt you for the following information. Ensure you have the following information available **BEFORE** running the script:

- **Nextcloud-Signal Hostname** This is the hostname you wish to use for Nextcloud-Signal **WITHOUT** the domain (Example: signal). This is used in combination with the Nextcloud-Signal Domain to form the URL of your Nextcloud-Signal instance.
- **Nextcloud-Signal Domain** This is the domain you wish to use for Nextcloud-Signal (Example: domain.tld). This is used in combination with the Nextcloud-Signal Hostname to form the URL of your Nextcloud-Signal instance (Example: Nextcloud-Signal.domain.tld).
- **Nextcloud Instance URL** This is the FQDN URL of a Nextcloud server instance that will be connecting to the Nextcloud-Signal server. You **MUST** include http:// or https://

  (Example: https://cloud.domain.tld).

Installation

Clone the deeztek-docker repository with git:

```
sudo git clone https://github.com/deeztek/deeztek-docker.git
```

This will clone the repository and create a docker directory in the directory you ran the git clone command from.

Change to the nextcloud-spreed-signaling directory:

```
cd deeztek-docker/Linux/nextcloud-spreed-signaling
```

Run the following script as root:

```
sudo bash ubuntu_install_nextcloud_signal.sh
```

The script will create a **/opt/nextcloud-spreed-signaling/** directory and configure all necessary directories and files under that directory.

It will automatically generate the following random variables and insert those variables in the appropriate configuration files (server.conf, .env, default):

- Static Secret
- Hash Key
- Block Key
- Shared Secret
- Api Key

It will output the following variables which you will need in order to configure your Nextcloud instance, so ensure you take note of them:

- Static Secret
- Shared Secret

It will build and deploy the following six containers:

- nextcloud_spreed_nginx
- nextcloud_spreed_backend
- nextcloud_spreed_janus
- nextcloud_spreed_coturn
- nextcloud_spreed_nats
- nextcloud_spreed_certbot

Let's Encrypt Certificates

The installation script automatically configures a scheduled cron job to renew Lets Encrypt Certificates on a daily basis.

By default, the script will launch Nginx reverse proxy with a self-signed certificate. If you wish to leverage a Lets Encrypt certificate, ensure that BOTH ports 80/TCP and 443/TCP are open to the Internet and the FQDN of your Nextcloud-Signal server which is derived from the **Nextcloud-**

Signal Hostname and **Nextcloud-Signal Domain** you specified during the installation of the script point to the public IP of your Nextcloud-Signal instance.

Before attempting to get a production Lets Encrypt certificate, you should test with the Lets Encrypt Staging environment first by running the following script:

```
sudo bash certbot_certificate_staging.sh
```

The script will ask you for an e-mail address and it will ask you to agree to the Lets Ecnrypt Terms of Service. If it all goes well, it will output the following indicating success:

```
...
The dry run was successful.
...
```

If succesful, you can now safely run the following script to get a production Lets Encrypt Certificate:

```
sudo bash certbot_certificate_production.sh
```

The script will ask you for an e-mail address, it will ask you to agree to the Lets Ecnrypt Terms of Service and ask you to agree to shared your e-mail address with the Electronic Frontier Foundation. If it all goes well, it will output the following indicating success:

```
....
Successfully received certificate.
....
```

Connect Nextcloud to Nextcloud-Signal

Login to Nextcloud as an Administrator and add and enable the **Nextcloud Talk** app.

Next navigate to **Settings --> Administration --> Talk**.

In the **STUN servers** section, enter the FQDN of your Nextcloud-Signal instance on port 3478 (Example: signal.domain.tld:3478), in the **TURN severs** section, select **turn: and turns:** on the drop-down and enter enter the FQDN of your Nextcloud-Signal instance on port 3478 (Example: signal.domain.tld:3478) and enter the **Static Secret** variable that was output earlier in the **secret** field.

In the High-performance backend section, click the + button and enter https://signal.domain.tld/standalone-signaling/ where **signal.domain.tld** is tha actual FQDN of your Nextcloud-Signal instance and in the **Shared secret** field enter the **Shared Secret** variable that was output ealier (**Figure 1**):

Figure 1

Screenshot

Image not found or type unknown