Development environment

We have a tutorial available on setting up your development environment using docker. You can find the tutorial here. We recommend you to follow that tutorial.

This page describes how to set up your development environment without docker.

Please follow the steps on this page to set up your development environment manually.

Set up Web server and database

First set up your Web server and database (**Section**: Manual Installation - Prerequisites).

Get the source

There are two ways to obtain Nextcloud sources:

- Using the stable version
- Using the development version from GitHub which will be explained below.

To check out the source from GitHub you will need to install Git (see Setting up Git from the GitHub help)

Gather information about server setup

To get started the basic Git repositories need to be cloned into the Web server’s directory. Depending on the distribution this will either be

- **/var/www**
- **/var/www/html**
- **/srv/http**

Then identify the user and group the Web server is running as and the Apache user and group for the **chown** command will either be

- **http**
- **www-data**
- **apache**
- **wwwrun**

Check out the code

The following commands are using **/var/www** as the Web server’s directory and **www-data** as user name and group.

Make the directory writable so you can install the code as your regular user, and don’t need root privileges:

```
sudo chmod o+rw /var/www
```

Then install Nextcloud at the root of your site from Git:

```
git clone https://github.com/nextcloud/server.git /var/www/
cd /var/www
git submodule update --init
```

If you prefer to install Nextcloud in a sub-folder, replace */var/www* with */var/www/<folder>*.

Create the data folder:

```
cd /var/www
mkdir data
```

Adjust permissions:

```
cd /var/www
sudo chown -R www-data:www-data config data apps
sudo chmod o-rw /var/www
```

Finally, restart the Web server (this might vary depending on your distribution):

```
sudo systemctl restart httpd.service
```

or:

```
sudo systemctl restart apache2.service
```

or:

```
sudo /etc/init.d/apache2 restart
```

Now access the installation at http://localhost/ (or the corresponding URL) in your web browser to set up your instance.

Check out external shipped apps

This optional step is especially necessary if you want to test upgrading, as the following apps are required to be present during an upgrade.

Install the viewer app:

```
cd /var/www/apps
git clone https://github.com/nextcloud/viewer.git
```

Make sure to use a version compatible with the server by checking out the matching tag. You can check the `appinfo/info.xml` of the app to see if its `min-version` field is compatible with the current server.

When upgrading the server code you might need to upgrade the app code as well before running `occ upgrade`.

Note

The same applies to all the apps listed under `alwaysEnabled` in shipped.json but most are already present in the server repository.

Enabling debug mode

Note

Do not enable this for production! This can create security problems and is only meant for debugging and development!

To disable JavaScript and CSS caching, debugging has to be enabled by setting `debug` to `true` in `config/config.php`:

```
<?php
$CONFIG = array (
    'debug' => true,
    ... configuration goes here ...
);
```

Keep the code up-to-date

If you have more than one repository cloned, it can be time consuming to do the same the action to all repositories one by one. To solve this, you can use the following command template:

```
find . -maxdepth <DEPTH> -type d -name .git -exec sh -c 'cd "{}"/../
&& pwd && <GIT COMMAND>' \;
```

then, e.g. to pull all changes in all repositories, you only need this:

```
find . -maxdepth 3 -type d -name .git -exec sh -c 'cd "{}"/../ && pwd
&& git pull --rebase' \;
```

or to prune all merged branches, you would execute this:

```
find . -maxdepth 3 -type d -name .git -exec sh -c 'cd "{}"/../ && pwd
&& git remote prune origin' \;
```

It is even easier if you create alias from these commands in case you want to avoid retyping those each time you need them.