Manual ownCloud Upgrade

Table of Contents

Introduction

This document describes how to manually upgrade your ownCloud installation. Because installations can vary, this guide can only give an overview of methods and examples. These examples need to be adapted according your to needs and environment.

> This guide assumes that you have basic knowledge about Unix terminology, commands and concepts. In case you are not familiar with these, ownCloud highly recommends that you create a playground first to try the steps and/or get in touch with ownCloud support to avoid breaking your system or losing data.
> This guide covers the upgrade of only the ownCloud instance.
> When planning to update/upgrade your server environment or server packages, read the Detailed Installation Guide first to ensure that the prerequisites are in place.

After preparing the upgrade, you can decide between two ways of upgrading your instance:

Script-Guided Upgrade
This upgrade automates most of the tasks to be done, including setting the correct ownership and permissions.

Manual Step-by-Step Upgrade
When using this type of upgrade, you have to do all the steps manually but you can also handle special setups.

> In this description, we assume that your ownCloud installation was located in the default directory: `/var/www/owncloud` and the new release will reside there as well. The path might differ, depending on your installation.

General Preparation

You will need to perform several steps before you can start with upgrading your ownCloud instance.

Enable Maintenance Mode

Put your server in maintenance mode and disable Cron jobs. Doing so prevents new logins, locks the sessions of logged-in users, and displays a status screen so that users know what is happening.

In a clustered environment, check that all nodes are in maintenance mode.

Prevent Browser Access

Next, stop your webserver to prevent users trying to access ownCloud via the web. As an alternative, you can stop serving the virtual host for ownCloud. Stop the web server:

```
sudo service apache2 stop
```

Backup the Database

First, back up ownCloud and the server database as described in section Backing up ownCloud. This is independent of the next upgrade steps but important in case something goes wrong.

Review Third-Party Apps

Review any installed third-party apps for compatibility with the new ownCloud release. Ensure that they are all disabled before beginning the upgrade. Third-party apps are all apps that are not distributed by ownCloud or not listed in Supported Apps in ownCloud.

1. Disable Apps via Command Line
   This command lists all apps by <app-id> and app version:
   ```
   sudo -u www-data occ app:list
   ```
   This command disables the app with the given <app-id>:
   ```
   sudo -u www-data occ app:disable <app-id>
   ```
2. Disable via Browser
   Go to `Settings` > Admin > Apps and disable all third-party apps.

Backup Manual Changes in `.htaccess`

If you have made changes in `.htaccess` located at the webroot of ownCloud, you must backup these changes. Back up only the changes made and not the complete file as this file will be recreated on upgrades and may contain different settings provided by ownCloud. Manual changes in `.htaccess` can be necessary when you, for example Integrate ownCloud into Microsoft Teams or need to fix WebDav connection errors as The network name cannot be found on Windows.

Backup Manual Changes in `.user.ini`

If you have made changes in .user.ini located at the webroot of ownCloud, you must backup these changes.

Download the Latest Release

Download the latest ownCloud server release to the same location where your previous instance is located; in this example the default directory /var/www/.

```
cd /var/www/
sudo wget
https://download.owncloud.com/server/stable/owncloud-complete-latest.tar.bz2
```

Script-Guided Upgrade

When using the script-guided upgrade, the script from the Script-Guided Installation is used. This script can not only install a new instance of ownCloud, but can also upgrade an existing one or manage ownership and permissions. When using the script for upgrading, the script renames the current instance and creates a new instance, copies `config.php` set ownership and permissions etc.
The script asks questions and beside other parameters, the upgrade an existing installation option is selected.
Follow the script documentation for details on how to install and use it.

> The script is most convenient if you use links for your `apps-external` and your data directory, as it takes care of recreating the links. You will be asked about this when you run the script. If you are using regular directories, these are created, but content must be moved or copied manually before finalizing the upgrade. If you are not using the `apps-external directory`, you must manually take care of copying only those apps that are not part of the new source.

When the script has finished, continue with the Finalize the Upgrade step described below.

> When using the script, any manual changes in `.htaccess` or `.user.ini` must be manually transferred from the backup directory the script created to the actual instance directory.

After the upgrade is finished, you can re-run this script to secure the `.htaccess` files.

Manual Step-by-Step Upgrade

Note that this procedure is not an in-place upgrade but an upgrade to a new directory identically named as originally set. This ensures that no code or settings can get orphaned or overwritten.

Move Current ownCloud Directory

Although you have already made a backup, move your current ownCloud directory to a different location for easy access later:
This example assumes Ubuntu Linux and MariaDB. Rename the ownCloud directory:

```
sudo mv /var/www/owncloud /var/www/backup_owncloud
```

Extract the New Source

Extract the new server release in the location where your previous ownCloud installation used to be.

```
sudo tar -xf owncloud-complete-latest.tar.bz2
```

Copy the data/ Directory

If you have kept your `data/` directory *inside* your `owncloud/` directory and you have not linked it, move it from your old version of ownCloud to your new version:

```
sudo mv /var/www/backup_owncloud/data /var/www/owncloud/data
```

If you have linked the data directory, for ownCloud it is still inside the `owncloud` directory and you have to re-link it.
If you keep your data outside of your `owncloud` directory, then you don’t have to do anything with it, because its location is configured in your original `config.php`, and none of the upgrade steps affect it.

Copy Relevant config.php Content

With the new source files now in place of where the old ones used to be, copy all `*config.php` and `*.json` files (if any exist) from your old ownCloud directory to your new ownCloud directory:

```
sudo cp /var/www/backup_owncloud/config/*config.php \
        /var/www/owncloud/config/


sudo cp /var/www/backup_owncloud/config/*.json \
        /var/www/owncloud/config/
```

Market and Marketplace App Upgrades

Before getting too far into the upgrade process, consider how the Market app and its configuration options affect the upgrade process. The Market app — and other apps from the Marketplace will not be updated when you upgrade ownCloud if `upgrade.automatic-app-update` is set to `false` in `config.php`.

In addition, if there are installed apps (whether compatible or incompatible with the next version, or missing source code) and the Market app is enabled but there is no internet connection available, these apps will need to be manually updated when the upgrade is finished.

Copy Old Apps

If you are using third party or enterprise applications, look in your new `/var/www/owncloud/apps`/ or `/var/www/owncloud/apps-external/` directory to see if they are present. If not, copy them from your old instance to your new one.

Make sure that all app directories that are defined in the apps_paths section of your config.php file do exist in your new /var/www/owncloud/ directory.
Also, make sure, that all app directories listed in apps_path actually exist.
If occ complains about missing apps-external then try

```
sudo mkdir /var/www/owncloud/apps-external
```

Permissions

To finalize the preparation of the upgrade, you need to set the correct ownership and permissions of the new ownCloud files and folders.

Set correct ownership

Set the ownership for all files and folders to `root:www-data` except the config and data directory:

```
sudo find -L /var/www/owncloud \
    \( -path ./data -o -path ./config \) -prune -o \
    -type d -print0 | xargs -0 chown root:www-data
```

```
sudo find -L /var/www/owncloud \
    \( -path ./data -o -path ./config \) -prune -o \
    -type f -print0 | xargs -0 chown root:www-data
```

Set the ownership for all files and folders to `www-data:www-data` for the config, data and apps directories. Note that it is not mandatory to set the ownership of the `data/` directory as it should already have the correct ownership and it can take a while to finish, depending on the size:

```
sudo chown -R www-data:www-data /var/www/owncloud/config
sudo chown -R www-data:www-data /var/www/owncloud/data
sudo chown -R www-data:www-data /var/www/owncloud/apps
```

Set correct permissions

Use `chmod` on files and directories with different permissions:

- For all files use `0640`
  ```
  sudo find -L /var/www/owncloud -type f -print0 | xargs -0
  chmod 640
  ```

- For all directories use `0750`
  ```
  sudo find -L /var/www/owncloud -type d -print0 | xargs -0
  chmod 750
  ```

- Set the `occ` command to executable:
  ```
  sudo chmod +x /var/www/owncloud/occ
  ```

If you have configured a script for guided installations, you can use it for this step as well as it automates it.

Finalize the Upgrade

Start the Upgrade

With the apps disabled and ownCloud in maintenance mode, start the upgrade process from the command line. Note that the example is based on Ubuntu Linux. Execute this within the ownCloud root folder.

```
sudo -u www-data occ upgrade
```

The upgrade operation can take anywhere from a few minutes to a few hours, depending on the size of your installation. When it is finished, you will see either a success message or an error message that indicates why the process did not complete successfully.

Reapply Manual Changes

Reapply any manual changes made to `.htaccess` files and the `.user.ini` file located in the owncloud webroot.
The following example command eases to find the differences of two files, which is helpful for reapplying manually added changes to `.htaccess` and `.user.ini`. Replace that paths, directories and files accordingly.

```
diff -y -W 70 --suppress-common-lines owncloud/.user.ini
owncloud_2022-02-15-09.18.48/.user.ini
post_max_size=513M                |     post_max_size=1G
```

Strong Permissions

- Check that chmod with `0640` for `.htaccess` and `.user.ini` files has been applied.
  If you have configured a script for guided installations, you can use it for this step as well as it automates it.

Disable Maintenance Mode

Assuming your upgrade succeeded, disable maintenance mode by using the `occ` command.

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

Enable Browser Access

When the above steps are successfully implemented, restart your web server, or alternatively re-enable the virtual host serving ownCloud:

```
sudo service apache2 start
```

Check the Upgrade

With maintenance mode disabled and the web server running, log in via the web interface and perform the following steps:

1. Check that the version number reflects the new installation.
   It can be reviewed at the bottom of Settings > Admin > General.
2. Check that your other settings are correct.
3. Go to the Settings > Admin > Apps page and review the core apps to make sure the right ones are enabled.
4. After the upgrade is complete, re-enable any third-party apps that are compatible with the new release. Use `occ app:enable <app-id>` or go to Settings > Admin > Apps > "Show disabled apps" and enable all compatible third-party apps.

Install or enable unsupported apps at your own risk.

Rollback

If you need to rollback your upgrade, see the Restoring ownCloud documentation.

Troubleshooting

When you are upgrading ownCloud while running MySQL or MariaDB with binary logging enabled, your upgrade may fail with these errors in your MySQL/MariaDB log:

```
An unhandled exception has been thrown:
exception 'PDOException' with the message 'SQLSTATE[HY000]:
General error: 1665
Cannot execute statement: impossible to write to binary log
since
BINLOG_FORMAT = STATEMENT and at least one table uses a
storage engine limited to
row-based logging. InnoDB is limited to row-logging when
transaction isolation level
is READ COMMITTED or READ UNCOMMITTED.'
```

Please refer to MySQL / MariaDB on how to correctly configure your environment.
In the unlikely case that files do not show up in the web-ui after the upgrade, use the files:scan command to make them visible again. Here is an example of how to do so:

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

See the Docs & Guides page for further resources for both home and enterprise users.
Sometimes, ownCloud can get *stuck in an upgrade*. This is usually due to the process taking too long and running into a PHP time-out. Stop the upgrade process this way:

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

Then start the manual process:

```
sudo -u www-data occ upgrade
```

If this does not work properly, try the repair function:

```
sudo -u www-data occ maintenance:repair
```