

Upgrading Clowder

This page describes how to upgrade the Clowder software. The steps described will do an in-place upgrade of clowder. The biggest advantage of this upgrade is that it is fast and requires the least amount of changes to the current system.

1. Before you start

- **Read about the new version** - Review the release notes for the version of Clowder. If you skip a few versions, we strongly recommend that you read the release notes of the versions you have skipped.
- **Check for known issues** - Use the [JIRA](#) to search for any issues in the new version that will affect you.
- **Check for compatibility:**
 - Confirm that your operating system, database, and other software installed still comply with the [requirements for Clowder](#).
 - If you have installed Clowder extractors, verify that they will be compatible with the version of Clowder you are upgrading to. If not you will need to update the extractors as well.
- **Prestaging and testing your new version of Clowder:**
 - We strongly recommend performing your upgrade in a test environment first. Do not upgrade your production Clowder server until you are satisfied that your test environment upgrade has been successful.
 - If you have any problems with your test environment upgrade which you cannot resolve, create an issue at [JIRA](#) so that we can assist you.

2. Backing up your database

Before you begin the upgrade process, make you have upgraded your database. During the upgrade process your database will be updated to match with the new version of the software. If you ever want to rollback to a previous version of the software you will have to rollback the database as well. Following are command to backup your database, as well as the commands needed to restore the specific database

2.1 Backing up MongoDB

This will describe how to backup the mongo database. If you have the files stored in the mongo database (default) this can take a long time and take up a significant amount of space since it will also dump the actual files. This assumes you are using the default database name (clowder) on the local host. If your database is stored somewhere else or has a different name you will need to modify the commands below. To backup the mongo database use:

Backing up MongoDB

```
mongodump --db clowder --out clowder-upgrade
```

2.2 Restoring MongoDB

This will describe how to backup the mongo database. If you have the files stored in the mongo database (default) this can take a long time and take up a significant amount of space since it will also restore the actual files. There are two ways to restore the mongo database, the first one will drop the database first, and thus will also remove any additional collections you added. The second way will only drop those collections that are imported, this can leave some additional collections that could create trouble in future updates.

Restoring MongoDB 1

```
echo "db.dropDatabase();" | mongo --db clowder  
mongorestore --db clowder clowder-upgrade/clowder
```

Restoring MongoDB 2

```
mongorestore --drop --db clowder clowder-upgrade/clowder
```

2.3 Backing up PostgreSQL

If you leverage of the geostreams capabilities in Clowder you will be using a PostgreSQL database. Again this assumes you will be using the default database (geostream) on localhost. This will dump the database in a directory. Each of the tables will be a separate file that is compressed.

Backing up PostgreSQL

```
pg_dump -F d -Z 9 -d bety -f geostream
```

2.4 Restoring PostgreSQL

To restore the database geostream database you can use the following command.

Restoring PostgreSQL

```
pg_restore -d geostream geostream
```

3. Performing the upgrade

The actual update consists of a few steps. After these steps are completed you will have an updated version of clowder.

1. Make sure you have backed up your database.
2. Download the version you want to install, some common versions are:
 - a. [Latest stable version](#) - This version is more tested, but is not as up to date as the development version.
 - b. [Latest development version](#) - This version contains the latest code and has been lightly tested.
3. Stop the current version of clowder you have running
4. Move the folder of the current version
5. Unzip the downloaded version of clowder
6. Move the custom folder of the original clowder to the custom folder of the new clowder
7. Start clowder. Make sure your startup script uses the flag `-DMONGOUPDATE=1` and `-DPOSTGRESUPDATE=1` to update the databases. If the database is not updated the application might not run correctly and/or you might not be able to login.

To make this process easier we have a [script](#) "update-clowder.sh" that will perform all these tasks for you (except for the backup, your are still responsible for the backup). The script does assume you have in the startup script that will have the UPDATE flags enabled.

- To upgrade to the latest **development** version, as root, do:

```
CLOWDER_BRANCH=CATS-CORE0 ./update-clowder.sh
```

- To upgrade to the latest **stable** version, as root, do:

```
./update-clowder.sh
```

For both, if this does not update it, add `--force` after "update-clowder.sh".

4. Post upgrade checks and tasks

Once you have confirmed the availability of compatible versions of the extractors, you should upgrade your extractors after successfully upgrading Clowder.

Congratulations! You have completed your Clowder upgrade.