

Installing Clowder

Before installing clowder make sure you have looked at the [list of requirements](#) and have setup all the software that is needed.

The first step is for you to figure out which one to use, there are two major versions to choose from:

- [Latest stable version](#) - This version is more tested, but is not as up to date as the development version.
- [Latest development version](#) - This version contains the latest code and has been lightly tested.

After downloading the version you have selected you will unzip that version.

The final step in the installation is to customize your installation of clowder. The customizations of the clowder software are stored in the custom folder inside the clowder installation. The main two files you will need to configure are custom.conf and play.plugins. Adding changes made to the files in the conf folder will **NOT** be used by clowder.

The play.plugins describes all the additional plugins that should be enabled. This file can only add additional plugins, and is not capable of turning of any of the [default plugins](#). For example the following play.plugins file will add some additional plugins:

play.plugins

```
9992:services.RabbitmqPlugin
10002:securesocial.core.providers.GoogleProvider
11002:services.ElasticsearchPlugin
```

The file custom.conf file is used to overwrite any of the [defaults](#) values for Clowder. Some common examples that are modified are:

custom.conf

```
# mongodb
mongodb.default = "mongodb://mongoserver:27017/mongodatabase"

# where to store the blobs (highly recommended)
service.byteStorage=services.filesystem.DiskByteStorageService
medici2.diskStorage.path="/home/clowder/data"

# rabbitmq
clowder.rabbitmq.uri="amqp://guest:guest@server/virtualhost"
clowder.rabbitmq.exchange=exchange

initialAdmins="youremail@address"

# elasticsearch
elasticsearchSettings.clusterName="name"
elasticsearchSettings.serverAddress="server"
elasticsearchSettings.serverPort=9300

# securesocial customization
# set this to true if using https
securesocial.ssl=true
# this will make the default timeout be 8 hours
securesocial.cookie.idleTimeoutInMinutes=480

# google setup
securesocial.google.authorizationUrl="https://accounts.google.com/o/oauth2/auth"
securesocial.google.accessTokenUrl="https://accounts.google.com/o/oauth2/token"
securesocial.google.clientId="magic"
securesocial.google.clientSecret="magic"
securesocial.google.scope="https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email"

# security options
application.secret="some magic string"
commKey=magickey
```