Installation

Installing and configuring OpenCGA consist of different steps, as you will see in this page you must first make sure that the server(s) have all dependencies installed, then you can configure and complete installation.

Table of Contents:

- Installation and Configuration
  - Step 1 - Configuring the Server
  - Step 2 - Getting OpenCGA
    - Building from Sources
    - Download Binaries
  - Step 3 - Install OpenCGA Binaries
  - Step 4 - Configure OpenCGA
  - Step 5 - Deploy WAR file
    - Deploying in a Web Server
    - Running Embedded REST Server

Installation and Configuration

Step 1 - Configuring the Server

OpenCGA requires Java 8, Tomcat or MongoDB. We try to keep dependencies to the minimum to ease development, installation and administration, and divide dependencies into *required* and *optional.* You can learn about all the dependencies and how to install them at **Installation Guide > Server Configuration**.

Step 2 - Getting OpenCGA

There are two main ways to get OpenCGA for installation:

- You can download the source code from GitHub and use **Apache Maven** to compile and build it.
- Or you can download a *prebuilt* binary from the **OpenCGA GitHub Releases** web page, notice that only *stable* and *pre-releases* are tagged and prebuilt.

Here you can learn more about these two options.

Building from Sources

Although most users will use stable *prebuilt* binaries (see below) there is still the need for different users to compile and build OpenCGA, for instance to test a development version. You can learn how to build from the source code at **Installation Guide > Building from Source Code**.

Download Binaries

You can download *stable* and *pre-release* (beta and release candidate) versions from **OpenCGA GitHub Releases** web page. You will find a *tar.gz* file with the name of *opencga* and the *version,* for instance to download OpenCGA 1.0.0-rc2.1 you can go to the GitHub Release at:

https://github.com/opencb/opencga/releases/tag/v1.0.0-rc2.1

and download the file *opencga-1.0.0-rc2.1.tar.gz* from the *Downloads* section.

Step 3 - Install OpenCGA Binaries

These instructions assume that you have already downloaded or built openCGA binaries as described on Using Binaries and Building from Source Code.

Create an installation directory called */opt/opencga* and copy the contents of opencga into this :

Note: In case of reinstallation, you must clean the installation directory (*/opt/opencga*)

```
mkdir /opt/opencga
cp -r build/* /opt/opencga
```

Step 4 - Configure OpenCGA

Execute the following command line to install and initialise Catalog database:

```
./opencga-admin.sh catalog install --secret-key
any_string_you_want <<< admin_P@ssword
```

Step 5 - Deploy WAR file

This is main interface to perform any action with OpenCGA. User have two different options to start web services:

- You can deploy OpenCGA in a Web Server.
- Or you can use the OpenCGA admin command line.

Next you can learn more about these two options.

Deploying in a Web Server

Install Apache Tomcat and copy the deploy `opencga.war`. To do this, just copy it from the compilation directory (where you downloaded the OpenCGA repository) into the Tomcat `webapps` directory:

```
cp /opt/opencga/opencga.war $(path_to_tomcat)/webapps
```

`path_to_tomcat` is where you downloaded it, or probably `/var/lib/tomcat8` if you installed via apt-get. Then, you should be able to see the swagger page at `http://localhost:8080/opencga/`. See Using RESTful web services for a tutorial.

Tomcat server will look for the configuration files in the installation directory, that can be changed in compilation time changing the property OPENCGA.INSTALLATION.DIR.

If the installation directory is empty at compilation time, the web services will search for the environment variable `OPENCGA_HOME`. If none of this is properly set, the webservices will not work.

Running Embedded REST Server

Serving the webservices using the admin command line. These webservices will be served with Jetty as follows:

```
/opencga-admin.sh server rest --start -p
```

Warning: This method is still under development.