Docker

New **OpenCGA 2.0.0** Feature. Work in progress

Table of Contents:

Overview

OpenCGA produces **four** different Docker images for different use cases, these are:

- *opencga-base*: base image for building the other images, this image contains the OpenCGA basic installation folder
- *opencga-rest*: image running REST web services
- *opencga-master*: image running *master* service
- *opencga*: all-in-one image with all OpenCGA components running

You can use *opencga* Docker image to run a complete OpenCGA platform locally. You can use the other images to deploy a whole cluster solution in a cloud environment using Kubernetes. Docker Images are deployed in Docker Hub OpenCB organisation.

Design and Implementation

Images

All images run with the user ***opencga*** and run **Java 8**. Docker images are deployed in Docker Hub.

opencga-base

This image contains the basic installation in directory /opt

More info at https://hub.docker.com/repository/docker/opencb/opencga-base

opencga-rest

This image is based in *opencga-base* and runs REST web services using the REST server command-line.

opencga-master

This image runs the master service

opencga

all-in-one image with all OpenCGA components running

Implementation

OpenCGA publishes a number of images into DockerHub for user ease. These images are based on *Alpine JRE* images to keep sizes as small as possible, contains OpenCGA binaries, for complete contents of image, please have a look at Dockerfile in github. A typical image name will follow the following structure :

- opencga:{*OPENCGA_VERSION_NUMBER*}-{*VARIANT_STORAGE_FLAVOUR*}
  - OPENCGA_VERSION_NUMBER will be like 1.4.0, 1.4.2, 2.0 etc
  - VARIANT_STORAGE_FLAVOUR can be mongoDB 4.0, hdinshigh, emr etc
  - e.g. **opencga:1.4.0-mongo4.0**

OpenCGA has published a docker image for quick testing and playing without going through hassle to learn, wait and install each and every OpenCGA components. The docker image is available at the public docker registry under the repository ***opencb/opencga-demo****.*

The OpenCGA demo docker image contains the following components:

- OpenCGA binaries
- MongoDB 4.0
- Solr 6.6 (*default*)
- ***init.sh,*** a bash script to :
  1. Install OpenCGA catalog
  2. Populate data (optional)

To download OpenCGA demo image, use the command *docker pull* with the OpenCB enterprise, opencga-demo repository and tag. For example :

Download the OpenCGA demo image

```
$ docker pull opencb/opencga-demo:1.4.2
```

To start the OpenCGA demo container, use the command *docker run*. For example: (This command will pull image as well if not present in local repository)

Start the OpenCGA container (without loading demo data)

```
$ docker run --name demo -d opencb/opencga-demo:1.4.2
```

The first time you run the OpenCGA demo container, it installs the OpenCGA catalog in mongoDB, starts the OpenCGA REST server and finally, it starts the OpenCGA daemon. In this case, no data is loaded. If you want to load some demo data, set the option l*oad* to true that instructs OpenCGA demo container to download data from *Corpasome* project and then, to run an ETL pipeline that loads, annotates, computes statistics and indexes data into the Solr search engine. This process roughly takes 35-40 minutes depending on the internet speed and the machine specifications. Following command starts the OpenCGA container loading demo data:

Start the OpenCGA container with loading demo data

```
$ docker run --name demo -e load=true -d opencb/opencga-demo:1.4.2
```

User can also mount customise directories to store **MongoDB** and **Solr** data for ease. The following command will mount the ~/data/mongodb and ~/data/solr directories from host machine to inside container at "/data/opencga/mongodb", "/data/opencga/mongodb" respectively.

Start the OpenCGA container loading demo data And mount user directories

```
$ docker run --name demo -e load=true -v ~/data/mongodb:/data/opencga
/mongodb -v ~/data/solr:/data/opencga/solr -d opencb/opencga-demo:1.4.2
```

In case, user has previously used mongodb and solr data outside of docker, mounted on host machine and now want to use NEW docker image and force docker to NOT install catalog, use the following command:

Start the OpenCGA container loading demo data And mount user directories Without Installing Catalog

```
$ docker run --name demo -e installCatalog=false -v ~/data/mongodb:/data
/opencga/mongodb -v ~/data/solr:/data/opencga/solr -d opencb/opencga-demo:
1.4.2
```

You can connect to the OpenCGA demo container using the following command:

Connect to the OpenCGA demo container

```
$ docker exec -it demo bash
```

To fetch the logs of the OpenCGA demo container, use the following command:

Fetch the logs of the OpenCGA demo container

```
$ docker logs demo -f
```

In order to access to the *Solr* dashboard, use the following URL on your internet browser:

Solr Dashboard

```
http://OpenCGA-Demo-Container-IP:8983/solr
```

And to access to the OpenCGA REST webservices, use the following URL:

OpenCGA REST Server

```
http://OpenCGA-Demo-Container-IP:9090/opencga/webservices/rest/v1
```

How To Create OpenCGA Demo Docker Image From Source Code

Distributed systems with multiple dependencies require a lot of work to setup, install, configure system and developers would like to have a quick way to test their changes locally and quickly with a system mimicking as close as possible to production systems. Following are the steps to create your own opencga demo image from modified oepncga source code :

Compile OpenCGA

```
~/appl/opencga[develop*]$ mvn clean install -DskipTests -Dopencga.war.
name=opencga
```

Build OpenCGA Next image :

Build OpenCGA Next Image

```
~/appl/opencga[develop*]$ docker build -t opencga-next -f opencga-app/app
/cloud/docker/opencga-next/Dockerfile .
```

Next Build OpenCGA Demo Image :

Build OpenCGA Demo Image

```
~/appl/opencga[develop*]$ docker build -t opencga-demo -f opencga-app/app
/cloud/docker/opencga-demo/Dockerfile .
```

Catalog installation and steps to create project, study etc are in listed in "*opencga-app/app/scripts/docker/opencga-demo/**init.sh***" script. User can modify as per own needs for example, change the variant file etc.

Following command will run local OpenCGA Demo :

Run OpenCGA Demo Container

```
~/appl/opencga[develop*]$ docker run --name demo -d opencga-demo
```