Orthanc and Nginx Documentation

Orthanc

The orthanc server on cloud is configured using docker compose. Docker compose requires a file called **docker-compose.yml** in which all images are to be configured. In the EC2 instance containing Orthanc, a docker compose file exists in the home directory called **docker-compose.yml**. All Orthanc and nginx configurations are done here under the keyword **services**.

In order to start the Orthanc server, the command is: **docker compose up -d**. The command **-d** means that the services will run in the background and you can continue to do other things on the EC2 instance.

**image: orthancteam/orthanc**

This is the name of the official docker image maintained by the Orthanc team.

**restart: unless-stopped**

This ensures that unless the server is manually stopped, it will continue to run.

**ports: - 8042:8042**

Orthanc server always runs on port 8042 in the EC2 instance. The 8042 on the right means that orthanc is running on 8042 in the EC2 instance. The 8042 on the left is the port that anybody outside the server will use. For example, if it was 80:8042, then the server will run on 8042 in the instance, but if we want to access the orthanc server on our browser or in our code, we will use 80. This is called **port forwarding**.

**volumes: - /orthanc-logs:/logs**

This setting is for mapping the orthanc logs folder in the docker container to a folder on our EC2 instance called **orthanc-logs**. This way, all logs will be stored in **orthanc-logs** in our instance and we won’t have to go inside the container to see the logs.

**environment:**

All orthanc configuration settings that are usually stored in **orthanc.json** will be defined here. Syntax for environment variables is –

**ORTHANC\_\_PLUGIN\_NAME\_\_VARIABLE\_NAME=value**

For example, ORTHANC\_\_AWS\_S3\_STORAGE\_\_BUCKET\_NAME

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Function** | **Type** |
| AWS\_S3\_STORAGE\_\_BUCKET\_NAME | Used to define the name of the S3 bucket | String |
| AWS\_S3\_STORAGE\_\_REGION | Used to define the region of the S3 bucket | String |
| AWS\_S3\_STORAGE\_\_ACCESS\_KEY | Used to define the access key of the S3 bucket | String |
| AWS\_S3\_STORAGE\_\_SECRET\_KEY | Used to define the secret key of the S3 bucket. **Should always be stored as a docker secret.** | String |
| AWS\_S3\_STORAGE\_\_ENABLE\_AWS\_SDK\_LOGS | Enables the logs of the AWS client used by the plugin | Boolean |
| AWS\_S3\_STORAGE\_\_REQUEST\_TIMEOUT | Used to define the timeout when sending a request to the bucket | Number |
| STABLE\_AGE | If no more new data comes for a patient/study/series in this much time, it is considered ‘stable’ | Number (in seconds) |
| AUTHENTICATION\_ENABLED | If set to true, then user will be prompted to enter Orthanc username and password stored in Registered\_Users when accessing the server from browser | Boolean |
| DICOM\_SERVER\_ENABLED | Makes the Orthanc server is a complete DICOM server | Boolean |
| VERBOSE\_ENABLED | Enables verbose logging in the Orthanc logs. **Should always be set to true** | Boolean |
| REGISTERED\_USERS | Used to define the username and password for the orthanc server. Can define multiple. Defined using docker secrets. | JSON |
| LUA\_SCRIPTS | Used to define all the Lua scripts that the server will use. Define using docker secrets. It will be an array of the paths of the Lua scripts | Array |
| HOUSEKEEPER\_ENABLED | Enables the housekeeper plugin that optimizes the server. Just enabling it is enough | Boolean |
| HOUSEKEEPER\_SCHEDULE | Defines the schedule during which the housekeeper plugin will run. Stored as a JSON object and defined as a docker secret. | JSON |
| DELAYED\_DELETION | Enables the delayed deletion plugin which handles file deletion asynchronously. | Boolean |
| HTTP\_TIMEOUT | Used to define how much time should pass before HTTP timeout occurs | Number (in seconds) |
| DICOM\_WEB\_\_ENABLE | Enables the DICOM web client plugin. Important as cornerstone.js fetches images from Orthanc using this client | Boolean |
| DICOM\_WEB\_\_ROOT | Used to define the root URL which will be used to access the DICOM web client. It will be IP:PORT/dicom-web/ | String |
| DICOM\_WEB\_\_ENABLE\_WADO | Enables web access to DICOM objects through HTTP | Boolean |
| DICOM\_WEB\_\_WADO\_ROOT | Used to define the WADO Root URL. Currently set to /wado | String |
| DICOM\_WEB\_\_ENABLE\_\_SSL | Used to define whether to use SSL when access the DICOM web client. Set this to true in the future when deploying. | Boolean |
| DICOM\_WEB\_\_STUDIES\_METADATA | Used to define how much of the metadata of a dicom study is retrieved through the client. **Should always be to set to Full** | String |
| DICOM\_WEB\_\_SERIES\_METADATA | Used to define how much of the metadata of a DICOM series is retrieved through the client. **Should always be set to Full** | String |
| DICOM\_WEB\_\_ENABLE\_METADATA\_CACHE | Enables caching of metadata of recently stored studies in the server | Boolean |
| DICOM\_WEB\_\_METADATA\_WORKER\_THREADS | Used to define how many threads will be used in any metadata related requests | Number |
| TRANSFERS\_PLUGIN\_\_ENABLED | Enables the transfers plugin which can be used to sent studies to peers in a fast manner. Has not been tested yet. Feel free to do so. | Boolean |
| TRANSFERS\_\_MAX\_HTTP\_RETRIES | Used to define how many retries the transfers plugin will use when transferring data | Number |
| LOGDIR | Defines the directory name in the docker container that will store the orthanc logs | String |
| DE\_IDENTIFY\_LOGS | Defines whether log will contain any sensitive information or not. True means the logs will not contain any sensitive information. | Boolean |
| POSTGRESQL\_\_ENABLE\_INDEX | Used to define whether the Postgres index will be used or not. Always set to true | Boolean |
| POSTGRESQL\_\_ENABLE\_STORAGE | Used to define whether the Postgres storage will be used or not. Always set to false as we are using AWS S3 for storage | Boolean |
| POSTGRESQL\_\_PORT | Used to define which port the PostgreSQL server is running on. | Number |
| POSTGRESQL\_\_USERNAME | Used to define the username of the PostgreSQL server | String |
| POSTGRESQL\_\_PASSWORD | Used the define the password of the PostgreSQL server. **Should always be stored as a docker secret.** | String |
| POSTGRESQL\_\_DATABASE | Used to define the name of the database being user in the PostgreSQL server | String |
| POSTGRESQL\_\_HOST | Used to define the IP address of the VM on which the PostgreSQL server is running. **Should always be stored as a docker secret.** | String |
| POSTGRESQL\_\_ENABLE\_SSL | Used to define whether SSL is to be used when accessing the database. Currently set to false, Has not been tested yet | Boolean |
| POSTGRESQL\_\_ENABLE\_VERBOSE\_LOGS | Used to define whether logs related to database connection should be verbose. **Always set to true** | Boolean |
| POSTGRESQL\_\_TRANSACTION\_MODE | **DO NOT CHANGE** | String |

**secrets:**

Docker secrets is a concept that is used for defining sensitive information such as credentials, API tokens, IP addresses etc. Certain environment variables that hold private information (for example, S3 Storage secret key or PostgreSQL Server Host) must be stored in text files or JSON files (as required). The variables can then be defined by providing the path to the file under the **secrets:** section. Currently, all such files are stored in the **Credentials** folder in the home directory of the EC2 instance.

TO DO:

* Enable SSL for the AWS S3 and PostgreSQL plugins and test.
* Define Lua Scripts variable for the script that will write the patient information to the reporting-bot database
* Place the Lua script writeToDatabase.lua and the python file reportingbot.py in the Scripts folder in the home directory and test whether it is working
* Reach out to Orthanc team on the Orthanc forum and tell them about our project. Clarify any commercial uses and make sure any licensing issue is taken care of.
* Test under load

References

* Orthanc Documentation: <https://orthanc.uclouvain.be/book/index.html>
* Orthanc Forum: <https://discourse.orthanc-server.org/>
* Docker Compose setup samples: <https://github.com/orthanc-server/orthanc-setup-samples>

NGINX

Nginx is a server that is being used as a reverse proxy. It is required for us as it enables CORS so that cornerstone.js is able to properly display the dicom files.

**image: nginx**

The official docker image is called nginx.

**ports: -2002:80**

Recall port forwarding. The nginx service runs on port 80 on the EC2 instance but for outside users, we access it on port 2002.

**depends\_on: orthanc**

This means that when using **docker compose up**, the nginx service will always start after the orthanc server has properly started.

**restart: unless-stopped**

Same as for Orthanc

**volumes: - ~/Credentials/default.conf:/etc/nginx/conf.d/default.conf**

This is very important. This maps the default.conf file stored in the Credentials directory to the default.conf file in the docker container for nginx. Any reverse proxy configuration is done in Credentials/default.conf and this line allows for the docker container to use our own configuration that we have set. **Please do not make changes to Configuration/default.conf unless you know what you are doing**.