**Set Up for Admin Backend**

**Prerequisite**

* MongoDB - Ensure MongoDB is set up on the server.
* Docker - Ensure Docker and Docker compose is installed on the server.

**Extract the tar files**

Load the .tar file provided in the shared drive by using the command below.

docker load -i image\_name.tar

**Backend Configuration Details**

**backend\_config.yaml** has all the necessary configuration for Apollo Admin Backend.

* **Flask Config:**

Configured values:

flask:

secret\_key: 'apollobackend'

debug: True

host: '0.0.0.0'

port: 5002

* **MongoDB Config:** Provide MongoDB connection details here

Configured values:

mongo:

host: mongodb

port: 27017

username: ''

password: ''

db\_name: user-system

**NOTE:** Provide the host name as the container name of your mongodb container.

* **allowed\_sequences\_types:** Provide the sequence type you want to allow for excel sheet

Configured values:

allowed\_sequence\_types:

- DWI

- FLAIR

- SWI

- GRE

* **series\_file\_path:** Set the path of the series excel sheets inside the docker.
* **scanner\_file\_path:** Set the path of the scanner sheets inside the docker.
* **series\_description\_character\_limit:** Set the character limit according to requirement.

* **scanner\_description\_character\_limit:** Set the character limit according to requirement.

* **series\_image\_name:** Set the image name which will have updated series excel files.
* **series\_image\_tag:** Set the image tag which will have updated series excel files.
* **scanner\_image\_name:** Set the image name which will have updated scanner excel files.
* **scanner\_image\_tag:** Set the image tag which will have updated scanner excel files.
* **series\_container\_name:** Set the container name from where you want to copy the series\_mapping\_config.xlsx.
* **series\_container\_path:** container path where series\_mapping file is present.
* **scanner\_container\_name:** Set the container name from where you want to copy the scanner\_list.xlsx.
* **scanner\_container\_path:** container path where scanner\_list file is present.
* **log\_file\_path:** set the file path where log file is present.
* **log\_file:** set the log file name in which logs will be present.

**Run docker compose**

* **docker-compose up** will start both the container(admin backend and mongodb).
* **MongoDB Configuration:**

Ensure that you have a directory which will be used to store MongoDB data.

Update the absolute paths in the volumes section of the mongodb service of **docker-compose.yml** file.

Sample values which were used for testing on the Aikenist server are provided below.

- /home/ubuntu/cerebriu\_project/mongodb\_data:/data/db

* **Apollo Backend Configuration:**

Update the absolute paths to the configuration files (backend\_config.yaml, series\_mapping\_config.xlsx, scanner\_list.xlsx) in the volumes section of the apollo\_backend service of **docker-compose.yml** file.

Modify the configuration files according to your requirements before running the services.

Sample values which were used for testing on the Aikenist server are provided below. Please update to relevant paths before testing in the Apollo VM.

* /home/ubuntu/cerebriu\_project/apollo\_interface/backend\_config.yaml:/app/backend\_config.yaml
* /home/ubuntu/cerebriu\_project/apollo\_interface/series\_mapping\_config.xlsx:/app/series\_mapping\_config.xlsx
* /home/ubuntu/cerebriu\_project/apollo\_interface/scanner\_list.xlsx:/app/scanner\_list.xlsx
* Please verify both containers are running on the network you prefer before running the docker compose.
* If you need to do any changes in any of the mounted files, then bring down the container using **docker-compose down** and again up the container using **docker-compose up**.
* **MongoDB Setup for Admin Collection:**

In order to manage administrative tasks such as user creation, modification, and deletion in the Apollo Admin backend application, it is essential to set up the admin collection in MongoDB.

This collection stores admin account information, including username and password (bcrypt hashed).

Below are the steps to achieve this setup:

**MongoDB Shell Commands:**

* Go inside the mongodb docker container and add the admin details:

docker exec -it <container\_name> mongo

* Switch to the appropriate database:

use user-system

* Create the admin collection:

db.createCollection("admin")

* Insert admin user data into the admin collection:

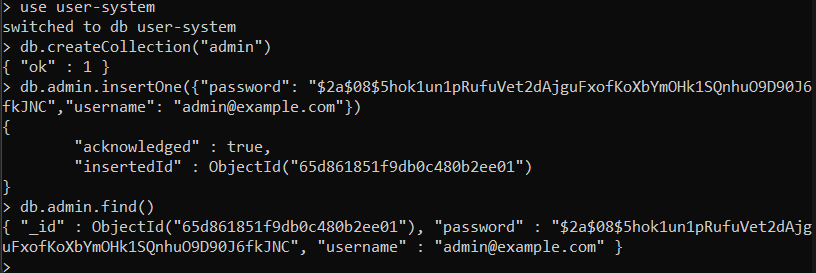
db.admin.insertOne({"username":"admin1","password":"hashed\_pwd" })

**Note**:Admin username and password must be between 6 and 20 characters.

* After inserting the admin user data, verify that the documents are correctly added to the admin collection

db.admin.find()

* Below is the snapshot of command used:



**NOTE:-** We need to add admin data once and thereafter it will be stored in the location you have given while setting the mongodb configuration.