

Steps to deploy OMOPonFHIR-main-v54-r4 – Docker Deployment

Author: Juan Diego González Serrano (juandigs2002@gmail.com) – v.10.05.2024

Introduction

This guide provides detailed help to configure OMOPonFHIR on Linux/Ubuntu using FHIR version R4 and OMOP CDM v5.4 as a base. It focuses on setting up an OMOP CDM v5.4 database and focuses on deploying the solution with Docker. The objective is to guide you step by step, avoiding common mistakes and facilitating the understanding of each step to solve problems that may arise during the process.

Step 1. Install the main repository and its subrepositories

First of all, we will clone the following repository:

```
sudo git clonehttps://github.com/omoponfhir/omoponfhir-main-v54-r4.git
```

This is the main repository along with its pom.xml and its server.xml, which in turn includes 3 subrepositories: the FHIR server module, the mapping module and the interaction module with the OMOP DB. To clone these 3 subrepositories correctly, we will have to modify the default file “updateSubmodules.sh” and then execute it.

The default file is the following: **sudo nano updateSubmodules.sh**

```
git submodule update --remote -- omoponfhir-omopv5-sql
git submodule update --remote -- omoponfhir-omopv5-r4-mapping
git submodule update --remote -- omoponfhir-r4-server
```

Only “--remote” is specified in the file, which is used to keep the submodules synchronized with the source repository. However, these submodules have not yet been initialized, so the “--init” command will have to be added to the file so that it remains:

```
git submodule update --init -- omoponfhir-omopv5-sql
git submodule update --remote -- omoponfhir-omopv5-sql
git submodule update --init -- omoponfhir-omopv5-r4-mapping
git submodule update --remote -- omoponfhir-omopv5-r4-mapping
git submodule update --init -- omoponfhir-r4-server
git submodule update --remote -- omoponfhir-r4-server
```

Next we will execute the file located in the main repository with:

```
./updateSubmodules.sh
```

If this gives permission errors, the following command will be run: **sudo chown -R \$(whoami) /omoponfhir-main-v54-r4** and we will run again **./updateSubmodules.sh**

Finally, we verify that each subrepository has its corresponding folders and files.

Step 2. Deploy the database with its corresponding tables and schemas

* The first step that we should follow is the “Running CPT4 Script” that is explained in the following document <https://github.com/omoponfhir/omoponfhir-site-n->

[docs/blob/main/documentation/](#)

[OMOP%20on%20FHIR%20R4%20to%20V6%20Deployment%20v1.0.1.pdf](#) .This is necessary for certain vocabularies to function correctly in the system. However, this guide has only tested the “Students” demo vocabularies mentioned in “Step 3”.

1.We will download a postgres `docker pull postgres` **yes** and we will deploy the container image:

```
docker run --name omopv54 -d -p 5432:5432 -e POSTGRES_PASSWORD=secret --restart unless-stopped postgres:latest
```

2.For any possible errors while executing the commands from now on, we will install Postgresql from the terminal with the commands:

```
sudo apt update
```

```
sudo apt install postgresql
```

Next, we will create the database and a “results” schema that “will be used for some additional tables other than those that will be included in the default public schema”:

```
docker exec -it omopv54 psql -U postgres -c "CREATE DATABASE omop;" -c "lc omop" -c "CREATE SCHEMA results; "
```

3.Later, we will configure the database for the OMOP project using a modified version of the official OMOP scripts.<https://github.com/OHDSI/CommonDataModel> .

Thus, we will clone the following repository:

```
sudo git clone https://github.com/omoponfhir/omopv5_4_setup.git
```

(The following steps have been carried out with this repository, although I later found a more updated one<https://github.com/SmartChartSuite/Registry-Database.git> where inside it has the /omoponfhir_v5_4_setup folder with slight modifications to the previous repository. However, I have run the ones from the repository that we have already cloned.)

We will go to the following folder `cd /CommonDataModel-5.4.0/inst/ddl/5.4/postgresql/` and execute, in order:

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_ddl.sql
```

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_primary_keys.sql
```

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_indices.sql
```

```
- Only if data has been uploaded: psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_constraints.sql
```

Now in the main folder of the cloned repository, in /omopv5_4_setup we will follow the following order:

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f omoponfhir_f_person_ddl.txt
```

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f omoponfhir_v5.4_f_observation_ddl.txt
```

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f omoponfhir_v5.2_f_immunization_view_ddl.txt
```

```
- psql -h localhost -p 5432 -U postgres -W -d omop -f omoponfhir_f_cache_ddl.txt
```

```
- (*Only if it is BigData: psql -h localhost -p 5432 -U postgres -W -d omop -f observation_column_size_change_ddl.txt )
```

Step 3. Load the vocabularies into our database

It is recommended to download the following vocabularies <https://athena.ohdsi.org>:

1. SNOMED
2. ICD9CM
3. ICD9Proc
4. CPT4
5. HCPCS
6. LOINC
7. RxNorm
8. NDC
9. Gender (OMOP Gender)
10. Race (Race and Ethnicity Code Set (USBC))
11. CMS Place of Service (Place of Service Codes for Professional Claims (CMS))
12. ATC
13. Revenue Code (UB04/CMS1450 Revenue Codes (CMS))
14. Ethnicity (OMOP Ethnicity)
15. NUCC (National Uniform Claim Committee Health Care Provider Taxonomy Code Set (NUCC))
16. Medicare Specialty (Medicare provider/supplier specialty code (CMS))
17. SPL (Structured Product Labeling (FDA))
18. Currency (International Classification of Diseases, Tenth Revision, Clinical Modification (NCHS))
19. ICD10CM
20. ABMS (Provider Specialty (American Board of Medical Specialties))
21. RxNorm Extension (RxNorm Extension OHDSI)
22. OMOP Extension (OMOP Extension (OHDSI))

To do this, within athena we will go to **"Download"**. A confirmation window will appear and, at the top, you must give your set of vocabularies a name (you can choose whatever you want). To the right of the name, there will be a drop-down menu that will allow you to select your version of OMOP. Even if you are deploying version 6.0, here you must select **"5.x"** from the drop-down list. When the process is finished, we will download and wait for the confirmation email to be able to obtain the vocabularies.

This deployment has been tested with the following vocabularies, which are recommended in the repository of the demo for students : <https://github.com/omoponfhir/OMOP-on-FHIR-Student-Demo>

DEMO OMOPONFHIR			
01:03 AM 16-APR-2024			
VS.0 29-FEB-24			
ID	CDM	CODE (CDM V5)	NAME
13	CDM 5	Race	Race and Ethnicity Code Set (USBC)
12	CDM 5	Gender	OMOP Gender
8	CDM 5	RxNorm	RxNorm (NLM)
6	CDM 5	LOINC	Logical Observation Identifiers Names and Codes (Regenstrief Institute)
1	CDM 5	SNOMED	Systematic Nomenclature of Medicine - Clinical Terms (IHTSDO)

When it let's download in a zip, it we will extract in the file /omopv5_4_setup/VocabImport/. Once located in this folder, we must modify the file `sudo nano OMOP\ CDM\ vocabulary\ load\ -l PostgreSQL.sql` to mark the route correctness of each csv of the vocabularies extraction. A brief example:

```
lcopy DRUG_STRENGTH FROM '...path/DRUG_STRENGTH.csv' WITH DELIMITER E'\t' CSV
HEADER QUOTE E'\b';
```

After having changed the routes, we will proceed to copy the vocabularies into our database by executing:

```
psql -U postgres -h localhost -p 5432 -d omop -f "OMOP CDM vocabulary load - PostgreSQL.sql"
```

After:

- `psql -U postgres -h localhost -p 5432 -d omop -c "CREATE SCHEMA vocab;"`
- We will clone the following repository out of our repository with `sudo git clone https://github.com/SmartChartSuite/Registry-Database.git` and we will head to /omoponfhir_v5_4_setup/ to run `psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_vocabulary_ONLY_ddl.sql`
- We execute `sudo nano ~/.bashrc` and we will add:
 - either `export JDBC_DATA_SCHEMA="omopv54"`
 - either `export JDBC_VOCABS_SCHEMA="vocab"`
- We execute `source ~/.bashrc`

Finally, to construct randomly selected names in our database, we will execute, in order, the following commands:

1. We will be redirected to the folder `cd /omopv5_4_setup/fhir_names`
2. `docker cp names.dmp omopv54:/`
3. `docker exec -it omopv54 /bin/bash`
4. `your postgres`
5. `psql omop < names.dmp`

This may take a few minutes. Once executed, if we want to build names in our database, we will execute:

```
psql -U postgres -h localhost -p 5432 -d omop -f "insert_names_to_f_person.sql"
```

Step 4. Compilation of the project and final deployment of the system in Docker.

We will be located at `cd /omoponfhir-main-v54-r4` and we will create a file `sudo nano env.list` with the variables:

```
JDBC_URL=jdbc:postgresql://omopv54:5432/omop
JDBC_USERNAME=postgres
JDBC_PASSWORD=secret
JDBC_DATASOURCENAME=org.postgresql.ds.PGSimpleDataSource
JDBC_POOLSIZE=5
JDBC_DATA_SCHEMA=public
JDBC_VOCABS_SCHEMA=public
SERVERBASE_URL=http://localhost:8080/fhir/
# AUTH_BASIC=<username_you_want>:<password_you_want>
```

```
#AUTH_BEARER="<any value>"
```

```
FHIR_READONLY=False
```

```
TARGETDATABASE=postgresql
```

```
OMOPONFHIR_NAME=OMOP v5.4 on FHIR R4
```

* If when creating a patient in the already deployed system or trying anything else it does not work, the non-configuration of the variables could be failing. **# AUTH_BASIC** • Yes that's how it is, Try configuring this variable with a username and password.

To finally deploy the system we will execute the following commands:

```
- docker network create omop
- docker network connect omop omopv54
- docker build -t omoponfhir .
- docker run --env-file env.list --name omoponfhir --network=omop -p 8080:8080 -d
  omoponfhir:latest
```

If everything went correctly, the OMOPonFHIR-v54 system should be able to be deployed.

To access the OMOPonFHIR user interface you must access the address **http://localhost:8080/**. If data has been previously imported, you can view related resources listed by resource type in the left panel. Even without having imported data, there will be a large amount of resources due to the load of vocabularies.