

Pasos para desplegar OMOPonFHIR-main-v54-r4 – Docker Deployment

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

Introducción

En esta guía se proporciona una ayuda detallada para configurar OMOPonFHIR en Linux/Ubuntu utilizando la versión R4 de FHIR y OMOP CDM v5.4 como base. Se centra en la configuración de una base de datos OMOP CDM v5.4 y se enfoca en desplegar la solución con Docker. El objetivo es guiar paso a paso, evitando errores comunes y facilitando el entendimiento de cada paso para solucionar problemas que puedan surgir durante el proceso.

Paso 1. Instalar el repositorio principal y sus subrepositorios

En primer lugar, clonaremos el siguiente repositorio:

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

Este es el repositorio principal junto a su pom.xml y su server.xml, el cuál engloba a su vez 3 subrepositorios: el módulo servidor FHIR, el módulo de mapeo y el módulo de interacción con la BD OMOP. Para clonar estos 3 subrepositorios de manera correcta, tendremos que modificar el archivo que nos viene por defecto “updateSubmodules.sh” y después ejecutarlo.

El archivo por defecto es el siguiente: `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
```

En el archivo solo se especifica “--remote” que sirve para mantener los submódulos sincronizados con el repositorio de origen. No obstante, estos submódulos todavía no se han inicializado por lo que habrá que añadir al archivo el comando “--init” de forma que quede:

```
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
```

A continuación ejecutaremos el archivo localizado en el repositorio principal con: `./updateSubmodules.sh`

Si esto da errores de permisos, se ejecutará el siguiente comando: `sudo chown -R $(whoami) /omoponfhir-main-v54-r4` y volveremos a ejecutar `./updateSubmodules.sh`

Por último, verificamos que cada subrepositorio cuenta con sus carpetas y archivos correspondientes.

Paso 2. Desplegar la base de datos con sus correspondientes tablas y esquemas

*El primer paso que deberíamos seguir es el de “Running CPT4 Script” que se explica en el siguiente documento <https://github.com/omoponfhir/omoponfhir-site-n->

[docs/blob/main/documentation/OMOP%20on%20FHIR%20R4%20to%20V6%20Deployment%20v1.0.1.pdf](https://docs.broadinstitute.org/omop-on-fhir-to-v6-deployment-v1.0.1.pdf) . Esto es necesario para que ciertos vocabularios funcionen de manera correcta en el sistema. No obstante, esta guía únicamente se ha probado los vocabularios de la demo de “Students” mencionados en el “Paso 3”.

1. Descargaremos una imagen de postgres **`docker pull postgres`** y pondremos en despliegue el contenedor:

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

2. Para cualquier posible error al ejecutar los comandos de ahora en adelante, instalaremos Postgresql desde la terminal con los comandos:

```
sudo apt update
```

```
sudo apt install postgresql
```

A continuación, crearemos la base de datos y un esquema “results” que “se usará para algunas tablas adicionales que no sean las que se incluirán en el esquema público predeterminado”:

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

3. Posteriormente, configuraremos la base de datos para el proyecto OMOP utilizando una versión modificada de los scripts oficiales de <https://github.com/OHDSI/CommonDataModel>.

Así, clonaremos el siguiente repositorio:

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

(Los siguientes pasos han sido realizados con este repositorio, aunque posteriormente encontré uno más actualizado <https://github.com/SmartChartSuite/Registry-Database.git> donde dentro tiene la carpeta /omoponfhir_v5_4_setup con ligeras modificaciones al repositorio anterior. No obstante, he ejecutado las del repositorio que ya hemos clonado.)

Nos dirigiremos a la siguiente carpeta **`cd /CommonDataModel-5.4.0/inst/ddl/5.4/postgresql/`** y ejecutaremos, por orden:

```
- 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
- Solo si se han subido datos: psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_constraints.sql
```

Ahora en la carpeta principal del repositorio clonado, en /omopv5_4_setup seguiremos el siguiente orden:

```
- 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
```

- (*Solo si se trata de BigData: `psql -h localhost -p 5432 -U postgres -W -d omop -f observation_column_size_change_ddl.txt`)

Paso 3. Cargar los vocabularios en nuestra base de datos

Se recomienda descargar los siguientes vocabularios de <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))

Para ello, dentro de athena nos dirigiremos a “**Download**”. Aparecerá una ventana de confirmación y, en la parte superior deberás darle un nombre a tu conjunto de vocabularios (puedes elegir el que quieras). A la derecha del nombre, habrá un menú desplegable que te permitirá seleccionar tu versión de OMOP. Aunque estés implementando la versión 6.0, aquí deberás seleccionar “**5.x**” de la lista desplegable. Cuando se haya terminado el proceso, daremos a descargar y esperaremos el correo de confirmación para poder obtener los vocabularios.

Este despliegue ha sido probado con los siguientes vocabularios, los que se recomiendan en el repositorio de la demo para estudiantes: <https://github.com/omoponfhir/OMOP-on-FHIR-Student-Demo>

DEMO OMOPONFHIR			
01:03 AM 16-APR-2024			
V5.0 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)

Cuando lo descarguemos en un zip, lo extraeremos en la carpeta /omopv5_4_setup/VocablImport/. Una vez localizados en esta carpeta, deberemos modificar el archivo `sudo nano OMOP\ CDM\ vocabulary\ load\ -\ PostgreSQL.sql` para señalar la ruta correcta de cada csv de la extracción de vocabularios. Un ejemplo breve:

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

Tras haber cambiado las rutas, procederemos a copiar los vocabularios en nuestra base de datos con la ejecución de:

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

Después:

- `psql -U postgres -h localhost -p 5432 -d omop -c "CREATE SCHEMA vocab;"`
- Clonaremos fuera de nuestro repositorio el siguiente repositorio con `sudo git clone https://github.com/SmartChartSuite/Registry-Database.git` y nos dirigiremos a /omoponfhir_v5_4_setup/ para ejecutar `psql -h localhost -p 5432 -U postgres -W -d omop -f OMOPCDM_postgresql_5.4_vocabulary_ONLY_ddl.sql`
- Ejecutamos `sudo nano ~/.bashrc` y añadiremos :
 - o `export JDBC_DATA_SCHEMA="omopv54"`
 - o `export JDBC_VOCABS_SCHEMA="vocab"`
- Ejecutamos `source ~/.bashrc`

Por último, para construir nombres seleccionados aleatoriamente en nuestra base de datos, ejecutaremos, en orden, los siguientes comandos:

1. Nos redigiremos a la carpeta `cd /omopv5_4_setup/fhir_names`
2. `docker cp names.dmp omopv54:/`
3. `docker exec -it omopv54 /bin/bash`
4. `su postgres`
5. `psql omop < names.dmp`

Esto puede durar unos minutos. Una vez ejecutado, si queremos construir nombres en nuestra base de datos, ejecutaremos:

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

Paso 4. Compilación del proyecto y despliegue final del sistema en Docker.

Nos localizaremos en `cd /omoponfhir-main-v54-r4` y crearemos un archivo `sudo nano env.list` con las 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
```

*Si a la hora de crear un paciente en el sistema ya desplegado o probar cualquier otra cosa no funciona, podría estar fallando la no configuración de las variables de **#AUTH_BASIC**. Si es así, probar a configurar esta variable con un usuario y contraseña.

Para finalmente desplegar el sistema ejecutaremos los siguientes comandos:

```
- 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
```

Si todo ha ido correctamente, debería poder desplegarse el sistema de OMOPonFHIR-v54.

Para acceder a la interfaz de usuario OMOPonFHIR hay que acceder a la dirección **http://localhost:8080/**. Si se han importado datos previamente, se podrá visualizar los recursos relacionados listados por tipo de recurso en el panel izquierdo. Aún sin haber importado datos, se contará con una gran cantidad de recursos debido a la carga de vocabularios.