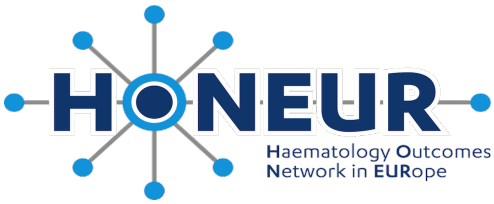

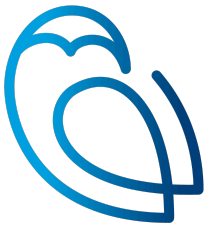

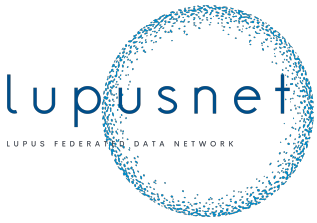

Feder8 local installation instructions

21 December 2021

CONTEXT

Feder8 aims to establish a collaborative, federated network of health care centres with the objective of using Real-World Data to address key research questions.

As it's a federated data network, the granular or patient-level data itself are not shared, only the analysis results. To accommodate this objective, every data source is converted to the same data model (aka a Common Data Model - CDM). This is done locally at the participating centres.

Since all centres are using the same model, a data analysis script can be defined centrally and sent to each of the participating centres for performing the analysis against their local data. There is no remote automated execution of the scripts – it requires explicit action from the local site responsible to execute the scripts. The summary results from the individual centres can then be shared with participants through the central database, and summary analysis and visualisations can be done in the central workbench.

In addition, the technology stack that we provide locally through Feder8 will not only help you to participate in network studies, but also provides you with a state-of-the-art infrastructure for advanced analytics on your own data.

This document provides an overview of the software components that are included in the local Feder8 solution, what infrastructure is required, what level of involvement typically is required from IT departments, how we ensure data privacy protection and how to install the software.

REQUIREMENTS

Hardware

Modern 64 bit (Intel/AMD) dual core processor (or better)
8 GB RAM, 16 GB RAM recommended
120 GB free disk space (or more)

Operating system

Windows 10, MacOS or Linux (Ubuntu, CentOS, Debian, …)

Linux is recommended

Docker

- Windows: https://docs.docker.com/docker-for-windows/install/
- MacOS: https://docs.docker.com/docker-for-mac/install/
- Linux: https://docs.docker.com/install/linux/docker-ce/ubuntu/

Assign 2 or more CPU's, 8 GB of RAM and 120 GB of disk space to Docker in Docker Desktop.

On Linux Docker compose (v1.24 or higher) should be installed separately.

Docker images

All Docker images are stored on a Feder8 image repository. A Feder8 (HONEUR, PHederation, ESFURN or Athena) account is required to be able to pull the images.

Installation Components

The software components that are to be installed locally include the following:

PostgreSQL database

The Postgres image for Feder8 contains the OHDSI database with 5 predefined schema's:

omopcdm
: OMOP CDM (v5.3.1) schema containing the vocabulary and custom concepts for Feder8.

webapi
: Schema used by the backend of Atlas (WebAPI) to store application data

results
: Schema used by the backend of Atlas (WebAPI) to store result data

public
: Schema used by the Docker runner service to store a log of all Docker runs

scratch
: Empty schema that can be freely used

See https://www.ohdsi.org/data-standardization/the-common-data-model/ for more info about OMOP CDM

User Management

Optional component that can be used to enable authentication based on a local user database (as an alternative for integrating with an LDAP server).

User Management is a simple web application to create users, roles and permissions and assign permissions to roles and roles to users.

More information can be found on the FEDER8 portal

Local configuration server

The local configuration server serves the local configuration for the other components. The configuration can be modified via a user interface in the local portal (see below).

Local Portal

All installed Feder8 components can be accessed via the local portal page. The portal page also shows the status (green: ok, red: not ok) of the connection to the local database and the connection to the Feder8 central services. The local configuration can be modified via the "Configuration" link.

The local portal application also provides a service and a UI to run analyses as Docker containers. This capability can be enabled at the time of installation.

Altas / WebAPI

Atlas is used to explore data in the OMOP CDM database. It can also be used to create (or import) cohort definitions and run them on the local data.

More information can be found on the FEDER8 portal and on the OHDSI website

Zeppelin

Zeppelin is a notebook server that is used to import and run notebooks. Notebooks will be shared on the central Feder8 platform. Results that might be created by running a notebook can be uploaded to the central platform (manually or automatically). Zeppelin can also be used to prepare data for other analysis scripts.

More information can be found on the FEDER8 portal and on the Zeppelin website https://zeppelin.apache.org/

Feder8 Studio

Feder8 Studio is an optional component that provides Visual Studio Code server, R Studio server and Shiny server. It is possible to import and run Python and R code from within a browser. When authentication is enabled during the installation, each (local) user will have a personal web space.

More information can be found on the FEDER8 portal

Distributed Analytics Remote

Distributed Analytics Remote is an optional component that is required to participate in studies that demand analysis on virtually pooled datasets. This local component will poll the central Distributed Analytics central service for incoming distributed requests. Requests are propagated to the Distributed Analytics R-Server component (see below) or to the Docker runner service (if enabled). Results of this computation are sent back to the central server where an analyst can process the results.

Distributed Analytics R-Server

Distributed Analytics R Server is an optional component that is part of the setup for Distributed Analytics. The R server can run R functions of the (pre-installed) distributedUtils package. The R functions depend on input data (CSV files) that are prepared in context of a distributed request. The results returned by the R functions are sent back (as JSON) to the distributed analytics remote component.

Feder8 NGINX

Feder8 NGINX acts as a HTTP proxy to the different Feder8 components. The proxy makes it possible to access all Feder8 components on HTTP port 80 (or HTTPS port 443 if a certificate is available). Only port 80 (or port 443 for HTTPS) should be opened on the host machine to make all Feder8 components accessible on the local network.

FEDER8 AND DATA PRIVACY

Patient-level data stays local at all times. In addition, there is no access for 3rd parties needed to develop the ETL. The Feder8 data model doesn't contain direct identifiers, but due to its granularity can be considered to be pseudonymised data. Where needed, the ETL can be setup in such a way to take additional measures for ensuring near-anonymisation. An example Data Privacy Impact Assessment is available.

INSTALLATION OPTIONS

The Feder8 local components are provided as a set of Docker images and can be installed with authentication enabled or disabled.
When authentication is disabled, Atlas, Feder8 Studio and Zeppelin can be accessed without login.
When authentication is required, the installer can choose to integrate with a local LDAP server (if available) or use the security database that comes out of the box with the setup.

INSTALLATION EFFORT

Installing the local HONEUR components takes +/- 30 minutes once a (physical or virtual) host machine - with Docker installed on it - is available.

INSTALLATION STEPS

Full installation instructions are documented on the official Github page.

A script to install all or a selection of the local components can be found here for each of the therapeutic area's (HONEUR, PHederation, ESFURN, ATHENA, LupusNet, …)

It is also possible to install each component separately. All individual installation scripts can be found here.

Installation Notes

Downloading the Docker images can take some time.

The following ports are used on the host machine:

- NGINX proxy: 80 (HTTP) or 443 (HTTPS)
- Postgres: 5444 (optional)

Please contact the HONEUR support team if these ports are not available or should be changed.

The local installation should only be accessible within the local network. Inbound internet access should be blocked! Outbound internet access is only needed for the applicable domain of the central services: honeur.org, phederation.org, esfurn.org or athenafederation.org.

The database data will be stored in a Docker volume named “pgdata”

HOW TO ACCESS THE FEDER8 INSTALLATION

After the installation is completed. All Feder8 components are available and can be accessed as follows:

Postgres DB

1. Open a SQL Client (e.g.: https://dbeaver.io/)
2. Connect to the Postgres database running on:
   Host: hostname or IP address of the host machine
   Port: **5444**
   Database: **OHDSI**
   Username: honeur, Password: <entered during installation> (to select, update and delete data) or
   Username: honeur_admin, Password: <entered during installation> (to make DB changes)
   Note: *The password of the honeur and honeur_admin DB accounts can be changed at any time. Changing the password will not have impact on the correct working of the installation.*
3. The “omopcdm”, “results”, “webapi” and scratch schemas can be accessed in the database.

Local Portal

1. Open a modern browser (e.g. Google Chrome)
2. Navigate to **http://[[hostname or IP of host machine]**
3. The Feder8 local portal page will be displayed. Atlas, Zeppelin, User Management (if applicable), Visual Studio Code (if applicable), R Studio server (if applicable) and the local configuration page can be opened via this portal page.

Atlas

1. Open a modern browser (e.g. Google Chrome)
2. Navigate to **http://[hostname or IP of host machine]/atlas**
3. The Atlas home page will be displayed.

User Management (optional)

1. Open a modern browser (e.g. Google Chrome)
2. Navigate to **http://[hostname or IP of host machine]/user-mgmt**
3. The user management home page will be displayed.

Zeppelin

1. Open a modern browser (e.g. Google Chrome)

2. Navigate to **http://[hostname or IP of host machine]/zeppelin**
3. The Zeppelin home page will be displayed.

HONEUR Studio – VSCode (optional)

4. Open a modern browser (e.g. Google Chrome)
5. Navigate to **http://[hostname or IP of host machine]**/honeur-studio/app/vscode
6. VSCode IDE will be displayed.

HONEUR Studio – Rstudio (optional)

7. Open a modern browser (e.g. Google Chrome)
8. Navigate to **http://[hostname or IP of host machine]**/honeur-studio/app/rstudio
9. RStudio IDE will be displayed.

HONEUR Studio - Local Shiny Apps (optional)

10. Open a modern browser (e.g. Google Chrome)
11. Navigate to **http://[hostname or IP of host machine]**/honeur-studio/app/reports
12. A list of locally deployed shiny apps will be displayed.

HONEUR Studio - Local Documents (optional)

13. Open a modern browser (e.g. Google Chrome)
14. Navigate to **http://[hostname or IP of host machine]**/honeur-studio/app/documents
15. A list of local documents will be displayed.

HONEUR Studio - personal (optional)

16. Open a modern browser (e.g. Google Chrome)
17. Navigate to **http://[hostname or IP of host machine]**/honeur-studio/app/personal
18. A list of files in the personal folder will be displayed.

RUNNING THE ETL

The ETL code to map the source data to the OMOP Common Data Model (CDM) will be custom build for each data center. The script to execute the ETL will be provided as soon as it is available.

POST ETL INSTALLATION STEPS

Add constraints and indexes to the OMOP CDM tables

After the ETL is successfully executed, it's recommended to add the constraints and indexes to the OMOP CDM tables. It will improve the performance and reduce the risk of corrupt data in the database.

The instructions can be found here.

Update custom concepts in the OMOP CDM DB

When new custom concepts are available, they can be easily loaded in the OMOP CDM database.

The instructions can be found here.

QA database

QA database can be used as a test database. It's an exact replica of the full database installed with the script start-postgres.cmd (on windows) or start-postgres.sh (on Linux or Mac). It is primarily used for testing scripts on data in the OMOP CDM schema.

The instructions for installation and removal can be found here.

MAINTENANCE

Starting and stopping the Docker Containers

The Docker containers for HONEUR will automatically restart when the host machine is restarted. To manually start and stop the containers, open a terminal window and run one of the following commands.

Postgres
- **to start**: `docker start postgres`
- **to stop**: `docker stop postgres`

Atlas
- **to start**: `docker start atlas`
- **to stop**: `docker stop atlas`

WebAPI *(Atlas backend)*
- **to start**: `docker start webapi`
- **to stop**: `docker stop webapi`

Zeppelin
- **to start**: `docker start zeppelin`
- **to stop**: `docker stop zeppelin`

User Management
- **to start**: `docker start user-mgmt`
- **to stop**: `docker stop user-mgmt`

NGINX proxy
- **to start**: `docker start nginx`
- **to stop**: `docker stop nginx`

Local Portal proxy
- **to start**: `docker start local-portal`
- **to stop**: `docker stop local-portal`

Configuration server

- **to start**: `docker start config-server`
- **to stop**: `docker stop config-server`

Feder8 Studio

- **to start**: `docker start honeur-studio`
- **to stop**: `docker stop honeur-studio`

Distributed Analytics Remote

- **to start**: `docker start distributed-analytics-remote`
- **to stop**: `docker stop distributed-analytics-remote`

Distributed Analytics R-Server

- **to start**: `docker start distributed-analytics-r-server`
- **to stop**: `docker stop distributed-analytics-r-server`

Backup and restore of the database

Backup

The database backup instructions can be found here.

The backup script will create a tar file name '<db_name>_<date_time>.tar.bz2' in the current directory.
Creating the backup file can take a long time depending on the size of the database.
Copy the backup file to a save location for long term storage.

Restore

The database restore instructions can be found here.

Hot snapshot of the database

The instructions to take a snapshot of the database Docker volume can be found here.

Accessing log files

Console logs

Docker provides real time access to the applications logs.

- Atlas: `docker logs atlas -f`
- WebAPI: `docker logs webapi -f`
- Zeppelin: `docker logs zeppelin -f`
- Postgres: `docker logs postgres -f`
- NGINX Proxy: `docker logs nginx -f`
- Feder8 Studio: `docker logs honeur-studio -f`
- Distributed Analytics: `docker logs distributed-analytics-remote -f`
- Local portal: `docker logs local-portal -f`

Logs files

The log files of Zeppelin are available on the host machine where the Docker containers are running. The location of the logs files can be configured during the setup. The default location is /zeppelin/logs

The logs files of Atlas are available within the Docker container and can be copied to the host machine if needed. The logs files are available in the /usr/local/tomcat/logs directory.

Execute the following commands to inspect the log files:

- `docker exec -it webapi /bin/bash`
- `cd logs`
- `tail -200  webapi.log` or `tail catalina.<yyyy-mm-dd>.log`

Execute the following command to copy all log files to a logs directory below the current directory on host machine:

- `docker cp webapi:/usr/local/tomcat/logs .`

FURTHER READING

More documentation is available in the user documentation on the FEDER8 portal:

- HONEUR: https://portal.honeur.org/
- PHederation: https://portal.phederation.org
- ESFURN: https://portal.esfurn.org
- ATHENA: https://portal-uat.athenafederation.org
- LupusNet: https://portal-uat.lupusnet.org

SECURITY SCAN OF DOCKER IMAGES

All Feder8 Docker images are scanned for vulnerabilities. Scan reports can be provided on request.
The report of a security assessment that was performed by EY can be found here.