Test Automation for DQE

Fork project

Fork the repository into your own repository space.

Containers

docker-compose.yml creates containers for PostgreSQL and Jenkins, all connected to the same network.

Compose using terminal in folder with docker-compsoe.yml file.

```
podman-compose up -d
```

If podman-compose is not installed:

```
pip install podman-compose
```

Check that the containers are running and connected to the same network:

```
podman ps
podman network inspect tafordqenetwork
```

Access Jenkins

Open your browser and navigate to: http://localhost:8080

Retrieve Initial Admin Password

From a file:

```
jenkins_home\secrets\initialAdminPassword
```

From logs:

```
podman logs jenkins | grep "Please use the following password"
```

First initialization

1. Login using password from previous step to start initializing and click on “Install suggested plugins”.
2. Wait until all plugins are installed.
3. Create first admin user (remember credentials).

Access PostgreSQL

PostgreSQL can be accessed:

- from local machine, port 5434;
- at postgres:5432 from other containers in the Podman network.

Credentials can be found in docker-compose.yml file: services->postgres->environment.

Run data pipeline

1. Create a New Pipeline Job:

- In Jenkins, click New Item.
- Select Pipeline and give it a name.
- Click OK.

2. Configure the Pipeline:

- Under Pipeline, select Pipeline script from SCM.
- Choose Git as the SCM.
- Enter the URL of your GitHub repository (e.g., https://github.com/your-username/your-repo.git).
- If the repository is private, add your GitHub credentials under Credentials.
- Specify the branch to use (e.g., main).
- Specify Script Path to Jenkins file: data_dev/Jenkinsfile.

3. Build the Pipeline.

4. Verify result:

- Verify that the pipeline runs successfully without errors (Console output - logs).

- Enter Jenkins container.

```
podman exec -it jenkins /bin/bash
```

- Check that 3 folders are created and filled with parquet files inside:

Folders with files:

```
parquet_data/
├── patient_sum_treatment_cost_per_facility_type/
│  └── ... (partitioned subdirectories with parquet files)
├── facility_name_min_time_spent_per_visit_date/
│  └── ... (partitioned subdirectories with parquet files)
```

```
    └── facility_type_avg_time_spent_per_visit_date/
    └── ... (partitioned subdirectories with parquet files)
```

- Check that report.html file is created:

```
generated_report/
├── report.html
```