

CI/CD Pipeline Documentation

Setting up GitHub Actions

1. Create a GitHub Repository

- Initialize your project repository on GitHub.

2. Add Workflow Configuration

- In your repository, create the directory structure: `.github/workflows/`.
- Add a workflow YAML file (e.g., `ci_cd_pipeline.yml`) inside this directory.

3. Configure the Workflow

- The workflow should specify the trigger (e.g., `push` to the `main` branch).
- Set up the required services (e.g., MySQL) and environment (e.g., Python, dependencies).
- Add steps to check out the code, install dependencies, and run your SQL scripts.

4. Commit and Push

- Commit your workflow file and push it to the `main` branch.
- GitHub Actions will automatically detect and run the workflow on every push to `main`.

Workflow Testing and Results

Workflow Overview

github.com/Appigle/PROG8850Mysql-A2/actions/runs/15501002602

Appigle / PROG8850Mysql-A2

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

← MySQL CI/CD Pipeline

✓ **A2 commit 02 #3**

Summary

Jobs

✓ setup_and_run_sql

Run details

Usage

Workflow file

Triggered via push 3 minutes ago

Appigle pushed 015ad3b main

Status: **Success**

Total duration: **42s**

Artifacts: —

ci_cd_pipeline.yml

on: push

✓ setup_and_run_sql 36s

- The workflow is triggered on every push to the `main` branch.
- It sets up a MySQL 8.0 service and creates a test database.
- Installs Python and the `mysql-connector-python` library.

- Executes all SQL scripts found in the `sql` directory against the test database.

The screenshot shows the GitHub Actions interface for a workflow named `setup_and_run_sql`. The left sidebar contains a 'Summary' section with a 'Jobs' list showing `setup_and_run_sql` as the active job. Below this are links for 'Run details', 'Usage', and 'Workflow file'. The main panel displays the job's execution log, which is divided into two steps: 'Install Python and mysql-connector-python' and 'Run SQL scripts'. The first step shows the installation of Python 3.12.3 and the mysql-connector-python package. The second step, highlighted with a red box, shows the execution of SQL scripts. It first runs `./sql/project.sql`, which creates a table with columns `project_id`, `project_name`, `start_date`, `end_date`, and `budget`. The output shows three rows of data. Then, it runs `./sql/add_departments.sql`, which creates a table with columns `department_id`, `department_name`, and `location`. The output shows three rows of data.

```
setup_and_run_sql
succeeded now in 36s

▼ ✓ Install Python and mysql-connector-python
47 Reading state information...
48 python3 is already the newest version (3.12.3-0ubuntu2).
49 python3-pip is already the newest version (24.0+dfsg-1ubuntu1.1).
50 0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
51 Defaulting to user installation because normal site-packages is not writeable
52 Collecting mysql-connector-python
53   Downloading mysql_connector_python-9.3.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (
54   Downloading mysql_connector_python-9.3.0-cp312-cp312-manylinux_2_28_x86_64.whl (33.9 MB)
55   _____ 33.9/33.9 MB 113.2 MB/s eta 0:00:00
56 Installing collected packages: mysql-connector-python
57 Successfully installed mysql-connector-python-9.3.0

▼ ✓ Run SQL scripts

1 ► Run for sqlfile in $(find ./sql -name '*.sql'); do
9 Running ./sql/project.sql
10 project_id  project_name  start_date  end_date    budget
11 1      Project 1      2024-01-01  2024-01-31  100000.00
12 2      Project 2      2024-02-01  2024-02-28  200000.00
13 3      Project 3      2024-03-01  2024-03-31  300000.00
14 Running ./sql/add_departments.sql
15 department_id  department_name  location
16 1      HR      New York
17 2      IT      San Francisco
18 3      Finance  London
```

Testing Steps

1. Push Code to Main

- Make changes to your SQL scripts or workflow file.
- Push the changes to the `main` branch.

2. Workflow Execution

- GitHub Actions will start the workflow automatically.
- You can monitor the progress in the "Actions" tab of your repository.

3. Step-by-Step Actions

- **Checkout code:** Retrieves the latest code from the repository.
- **Wait for MySQL:** Ensures the MySQL service is ready before running scripts.

- **Install Python & Dependencies:** Installs Python 3 and `mysql-connector-python`.
- **Run SQL Scripts:** Executes each `.sql` file in the `sql` directory against the test database.

Results

- Each step's output and status are visible in the GitHub Actions UI.
- If all SQL scripts execute successfully, the workflow will complete with a green checkmark.
- If any script fails, the workflow will stop and display error logs for troubleshooting.

Output

- **Success:**
 - All steps complete without errors.
 - SQL scripts are applied to the test database.
 - **Failure:**
 - The workflow stops at the failed step.
 - Error messages and logs are available for debugging.
-