Continuous Integration with GitHub Actions

For our Java-based application, we've set up a workflow using GitHub Actions. This workflow is triggered every time a developer pushes code to the main branch or raises a pull request against it. The workflow ensures that:

1. **Code is Checked Out**: The latest code from the repository is fetched.
2. **Environment is Set Up**: A specific version of Java is set up to ensure consistency.
3. **Build is Executed**: Our application is built using Maven, ensuring that all dependencies are fetched, and tests are run.

By using GitHub Actions as our CI tool, we've created an automated and consistent process to ensure the health of our codebase. This not only gives the development team confidence in the code they're writing but also allows us to deliver a robust software product to our stakeholders.

Currently the build is failing because the MySQL database is not in the repository.

Let's delve deeper into the YAML configuration file we created for CI using GitHub Actions:

1. **Name of the Workflow**:

```
name: Java CI
```

This simply gives a name to our CI workflow. This name will appear on the GitHub Actions page of your repository.

2. **Event Triggers**:

```
on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]
```

The **on** section defines the events that should trigger the CI workflow. In this case, it's configured to trigger on:

- A **push** event to the **master** branch.
- A **pull_request** event targeting the **master** branch.

3. **Job Definitions**:

```
jobs:
  build:
```

runs-on: ubuntu-latest.

Here, we're defining a job named **build**. This job will be executed on the latest version of the Ubuntu operating system provided by GitHub.

4. **Steps to Run**:

steps:

- uses: actions/checkout@v2

- name: Set up JDK 11

uses: actions/setup-java@v2

with:

java-version: '11'

distribution: 'adopt'.

- name: Build with Maven

run: mvn -B package --file pom.xml.

Each job can have multiple steps. These are the individual tasks that will be run in sequence:

- The **actions/checkout@v2** action checks out the code from the current repository into the runner.
- The **actions/setup-java@v2** action sets up a specific version of Java on the runner. Here, we're specifying JDK 11 from the AdoptOpenJDK distribution.
- The last step builds the project using Maven. We're specifically packaging the project using the provided **pom.xml**.