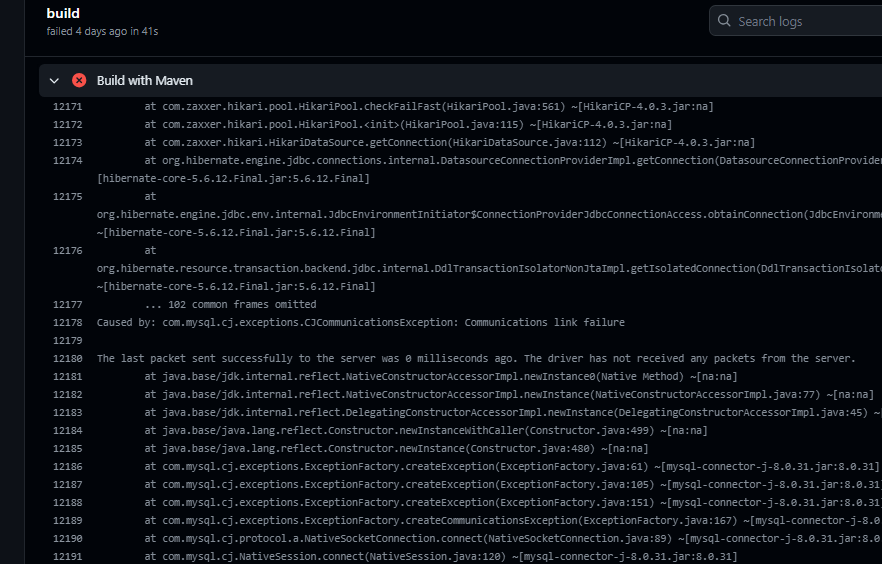
**COMMUNICATION LINK FAILURE ERROR**



**Network Accessibility:**

**Localhost:** When our application is running with the database host set to localhost, it expects the database to be running on the same machine where the application is deployed. In the context of GitHub Actions, our workflow is running on GitHub's servers, which is a different environment than our local machine. Therefore, the database on our local machine is not accessible from GitHub Actions.

**Database Endpoint:** When we provide the actual database endpoint, GitHub Actions can connect to the database server over the network. This is possible if your database server allows external connections and is reachable from the GitHub Actions environment.

**ERROR : Maven project detected. You should run the goal 'org.sonarsource.scanner.maven:sonar' during build rather than using this GitHub Action**

**mvn -B verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=${{ secrets.SONAR\_PROJECTKEY }} -Dsonar.host.url=${{ secrets.SONAR\_HOST\_URL }} -Dsonar.login=${{ secrets.SONAR\_TOKEN }}**

The Maven command is used for integrating SonarQube analysis into your Maven-based project.

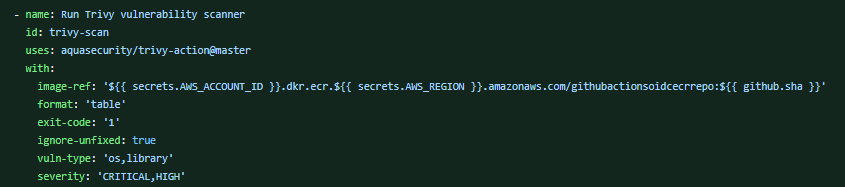


**TRIVY VULNERABILITY SCANNER ACTION ERROR**

Run Trivy vulnerability scanner action was giving error as it has the exit-code:'1' parameter in the Trivy step. This means that if the Trivy scan reports any vulnerabilities with a severity of 'CRITICAL' or 'HIGH', it will exit with a non-zero status code, causing the workflow to fail.

To continue the workflow to run add : continue-on-error=true

If this property is set to true , the job or workflow will not be interrupted even if the corresponding step fails. The remaining steps will continue to be executed.



**SONARQUBE QUALITY GATE ERROR**

Quality gate was passed even after the code coverage was less than 80% ( which was set)

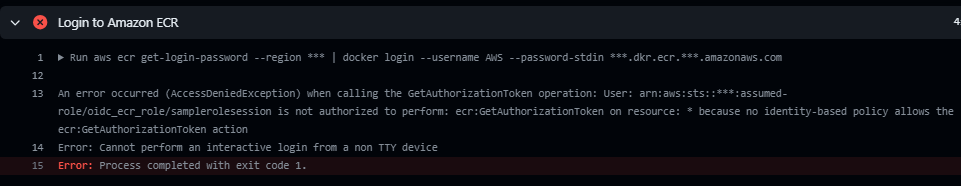
The condition was set for new project. Applying the quality gate condition after sonarqube scan will not consider the overall code, it will only consider the recent commits . So we have to add the condition for the overall code.

In SonarQube, quality gates are used to define and enforce quality criteria for your project. Quality gates can be set based on various conditions, and you have the flexibility to configure them according to your project requirements.

When setting quality gate conditions in SonarQube, we typically have the option to set conditions based on both the overall project and the new code. The choice between overall project and new code conditions depends on your specific goals and the way you want to manage code quality.

1. **Overall Project Conditions:**
   * Conditions set for the overall project are applied to the entire codebase, including both existing and new code.
   * This approach ensures that your entire codebase meets certain quality standards, regardless of when the code was written.
   * It helps maintain a consistent level of code quality across the entire project.
2. **New Code Conditions:**
   * Conditions set for new code are specifically applied to changes made in the most recent code commits or pull requests.
   * This approach allows you to focus on maintaining or improving code quality as you add new features or make changes to the existing code.
   * It helps prevent the introduction of new issues and ensures that recent code additions meet the defined quality criteria.

**WHILE LOGIN TO ECR AND PUSHING IMAGE TO ECR  WAS FACING PROCESS COMPLETED WITH EXIT  CODE1 ERROR**



Attach the above policies to role :-

"ecr:CompleteLayerUpload",

"ecr:GetAuthorizationToken",

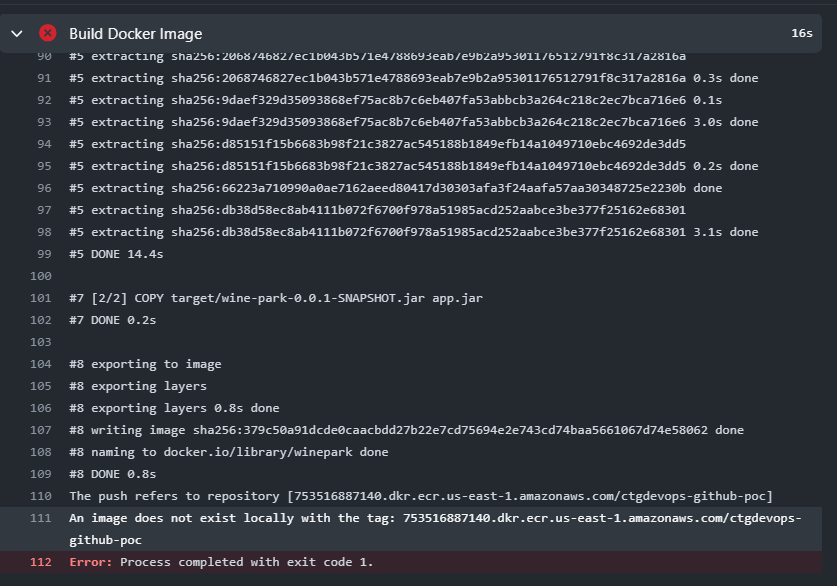
"ecr:UploadLayerPart",

"ecr:InitiateLayerUpload",

"ecr:BatchCheckLayerAvailability",

"ecr:PutImage",

"ecr:BatchGetImage"



Docker image was not present in the amazon ECR