**Statement 1:-**

A dockerized “Hello World” with database application:

* Hello\_world.py 🡺 This file contains the logic of the hello world which will connect to sqlite3 database and insert into sqlite3 database and display message to the user
* Dockerfile 🡺 Contains the code dependency to build the docker image

**Statement 2**

Least privilege access concepts are represented.

* For this we have used dockerfile to create container with normal user not the root user

**Statement 3**

Implement a continuous integration/continuous deployment (CI/CD) pipeline for your

code using your solution of choice (cloud solutions are acceptable). What you provide

should support a peer code review process and seamless application or artifact

* We used Gitlab as CI/CD, where we created testing-feature branch and raised a PR request to merge the code in main branch. Snapshot is attached in below pages.

**Statement 4**

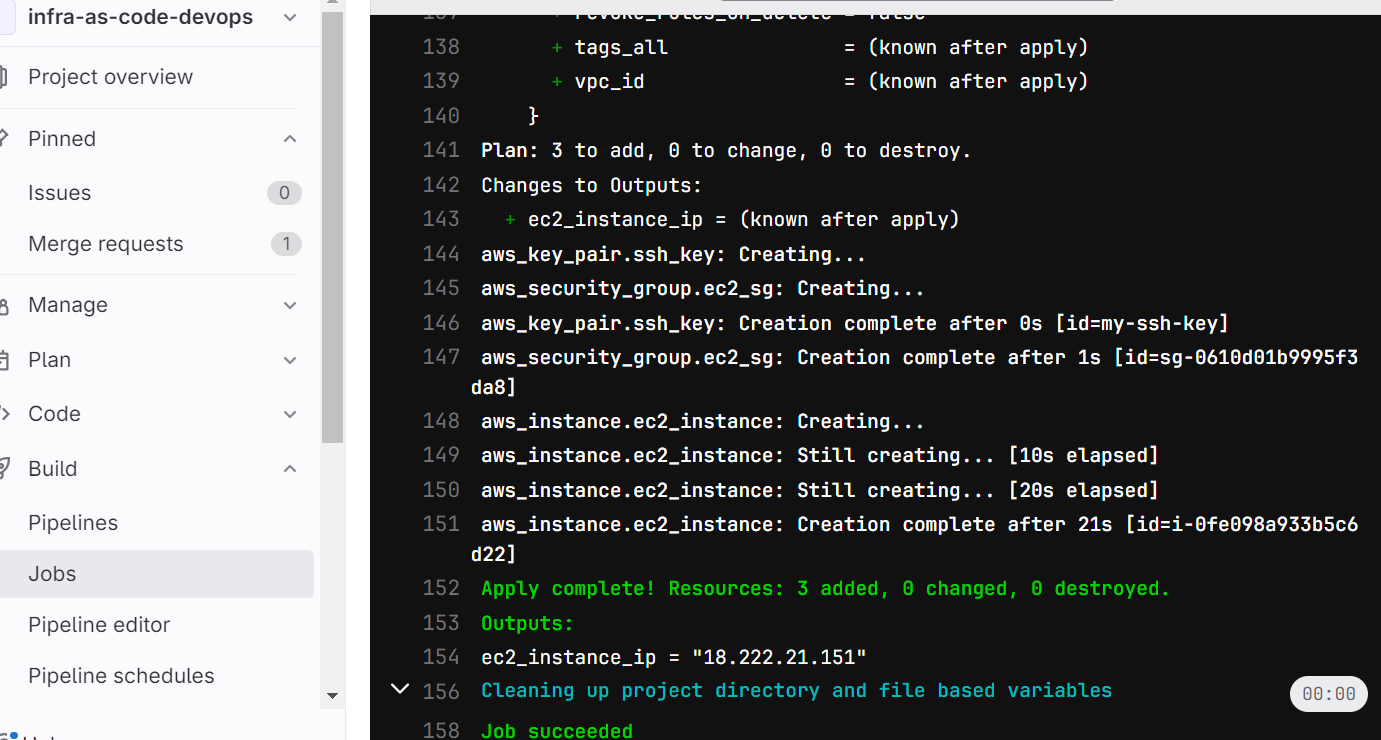
Demonstrable code change through Pull Request and environment implementation

* PR raise and merge in below snapshots

**Statement 4**

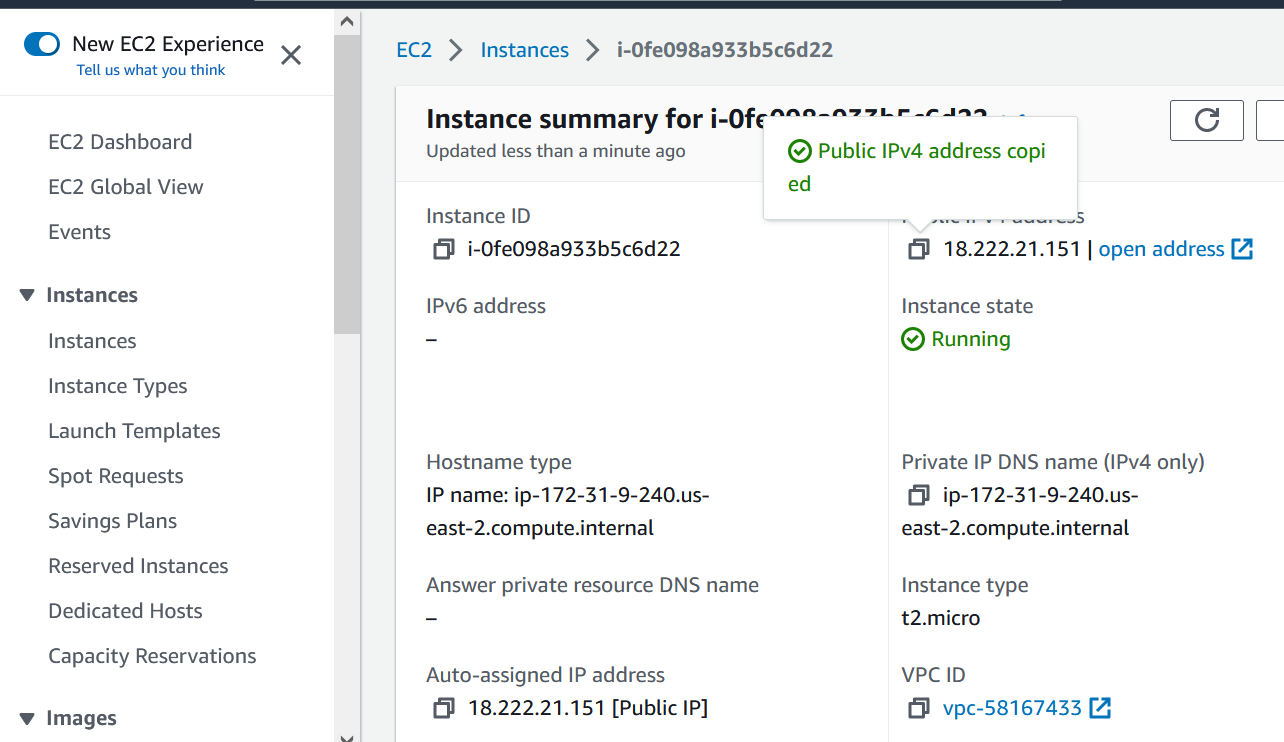
Infrastructure is implemented/configured using scripts or IaC

* For this we are using terraform which created the ec2 and deploy the infra in AWS



**Statement 5**

Infra created on AWS



To push the Docker image to Amazon Elastic Container Registry (ECR) and deploy it on an Amazon EC2 instance using Terraform, you can follow these steps:

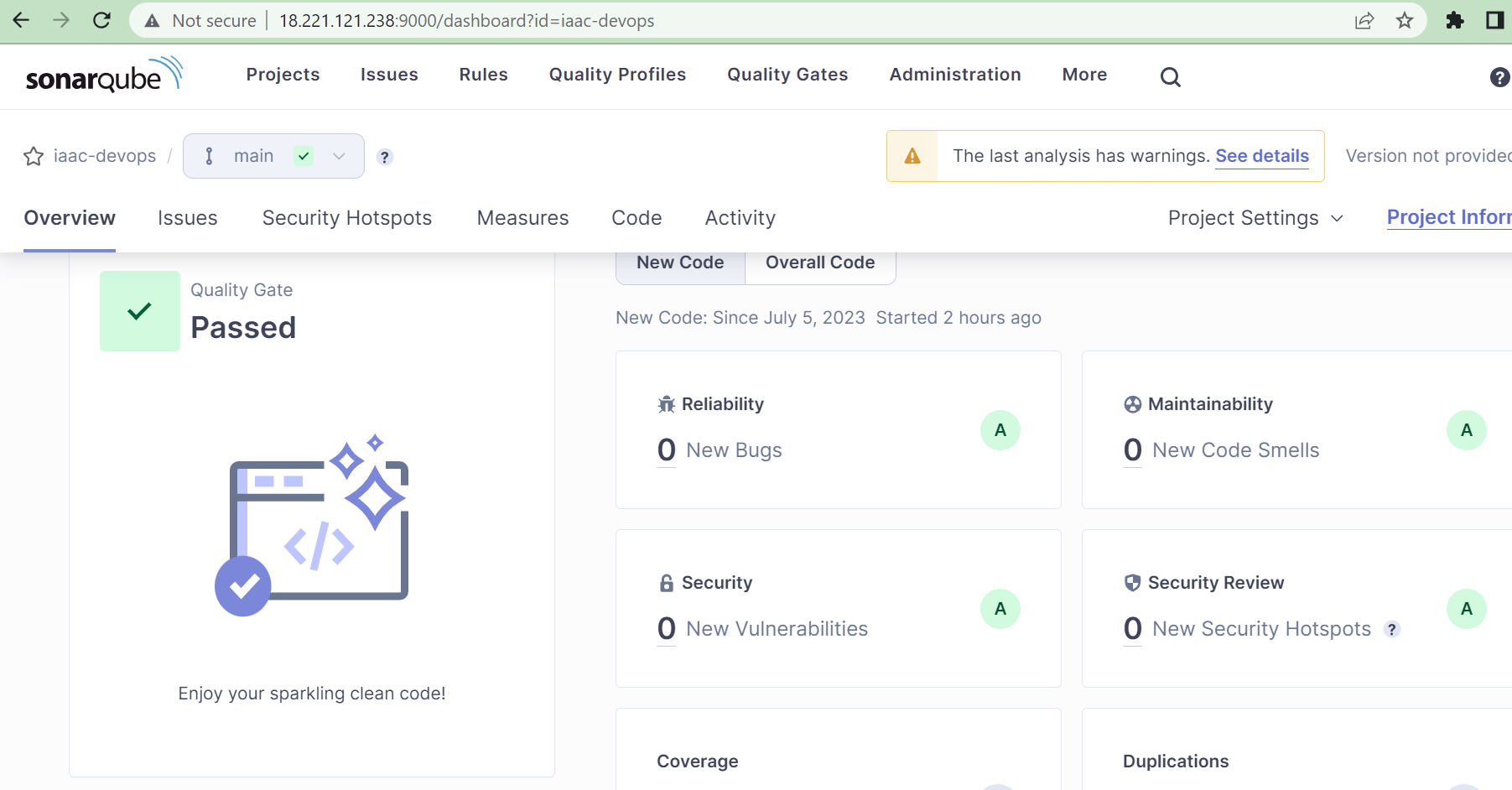
1. Set up AWS credentials: Make sure you have AWS credentials configured on your local machine. You can set them up using the AWS CLI or by exporting the AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY environment variables.
2. Tag the Docker image: Before pushing the image to ECR, you need to tag it with the ECR repository URL. Assuming your ECR repository is named my-ecr-repo in the us-east-1 region, you can tag the image with the following command:

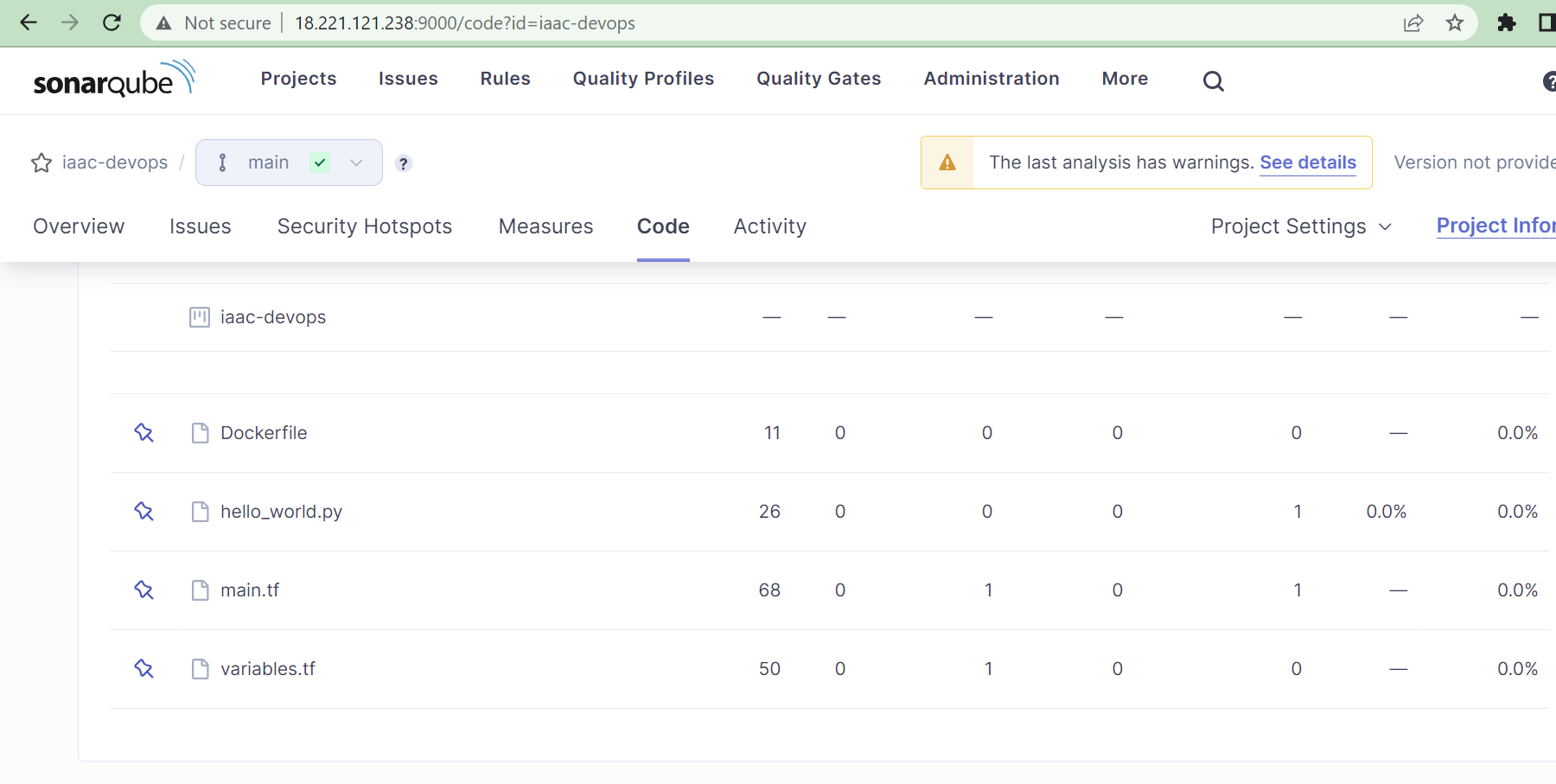
docker tag hello-world-app:latest <AWS\_ACCOUNT\_ID>.dkr.ecr.us-east-1.amazonaws.com/my-ecr-repo:latest

1. Push the Docker image to ECR: Use the following command to push the tagged image to ECR:

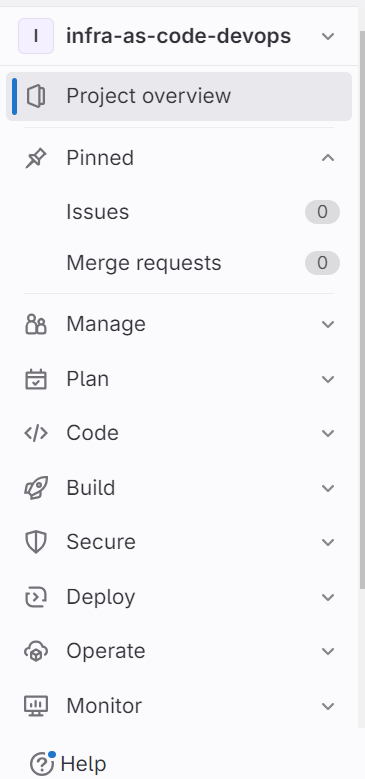
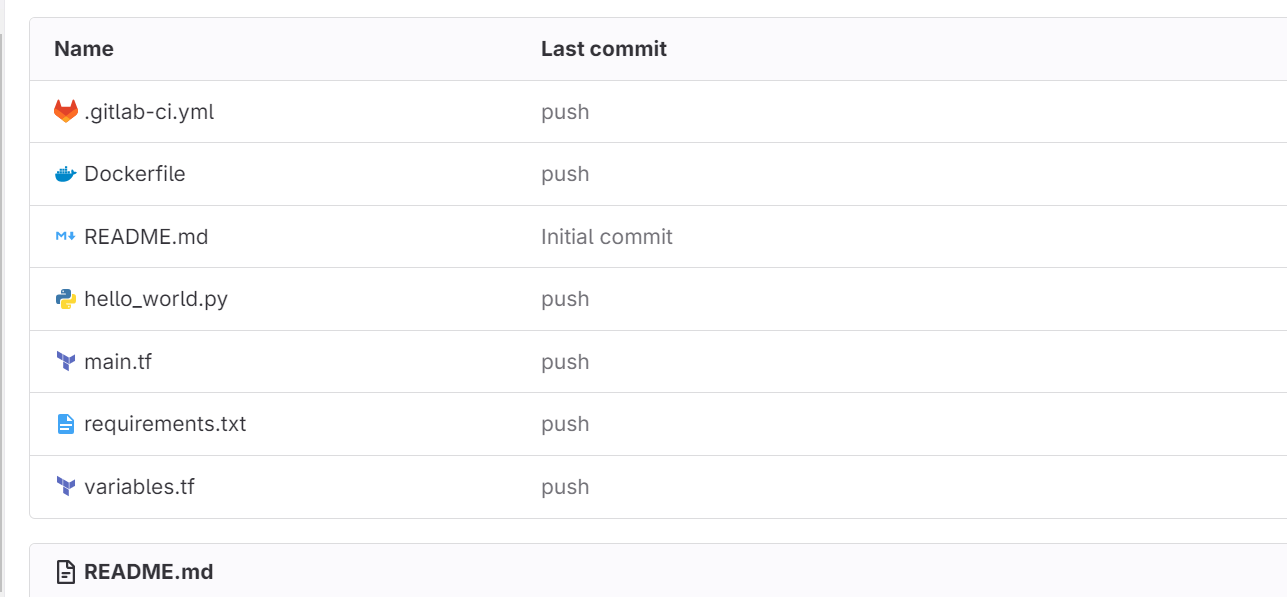
docker push <AWS\_ACCOUNT\_ID>.dkr.ecr.us-east-1.amazonaws.com/my-ecr-repo:latest

**Code Quality using Sonarqube:-**

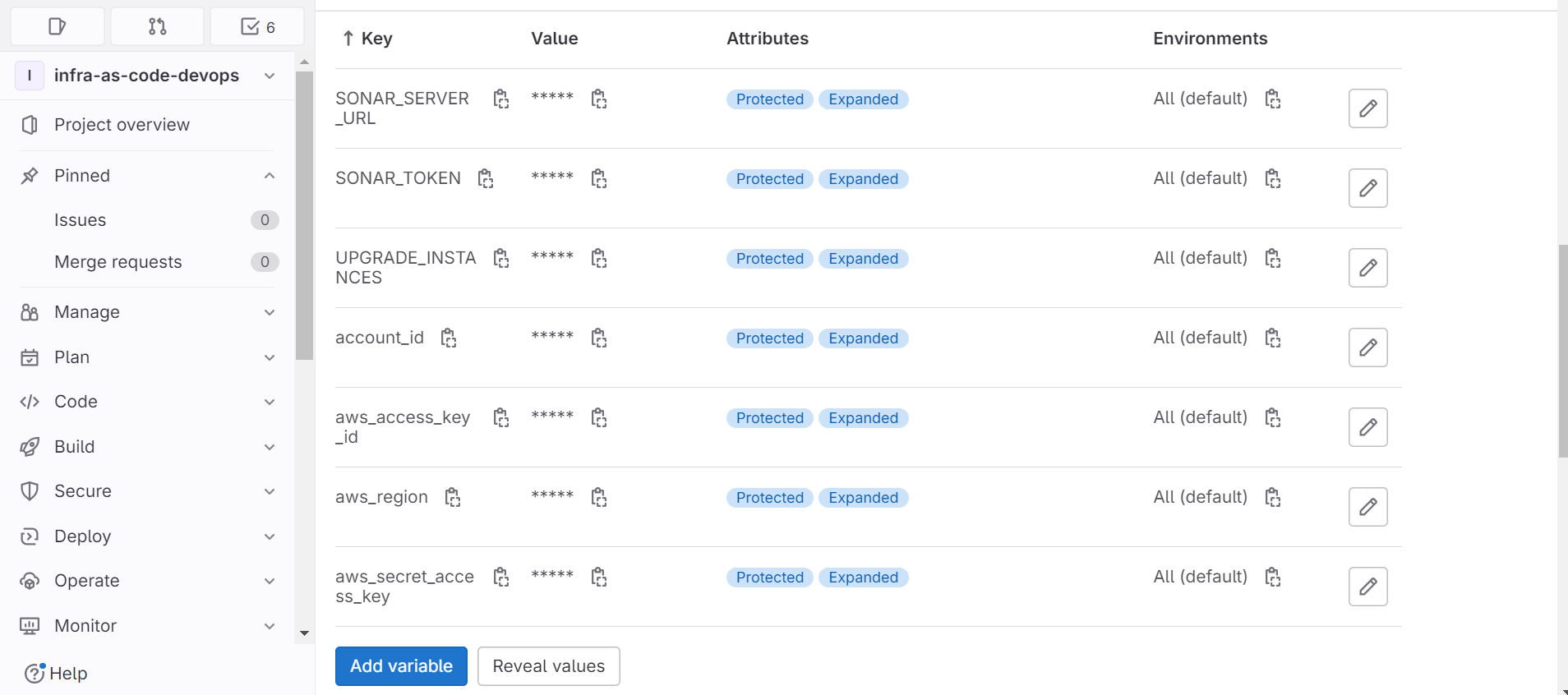




**For CI/CD used Gitlab:**

**Define environment variables in settings🡺 CI/CD**

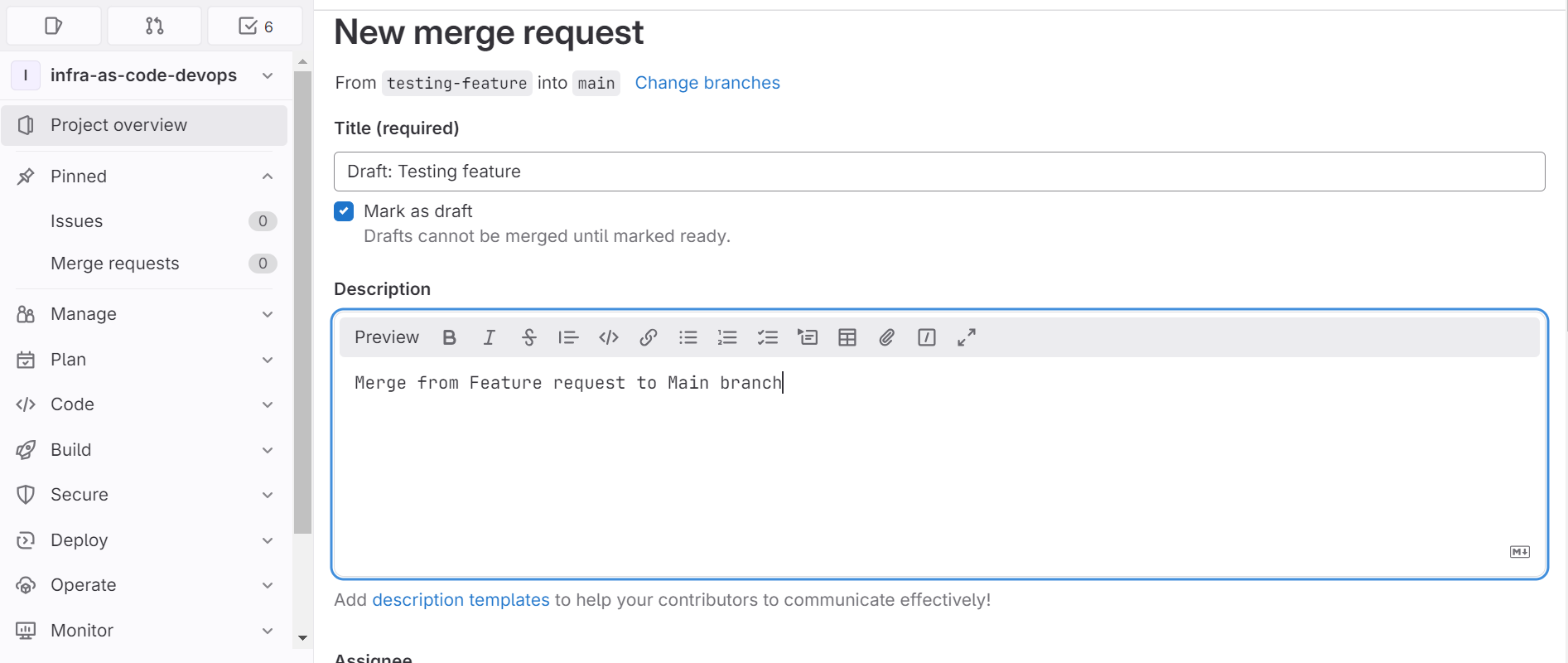


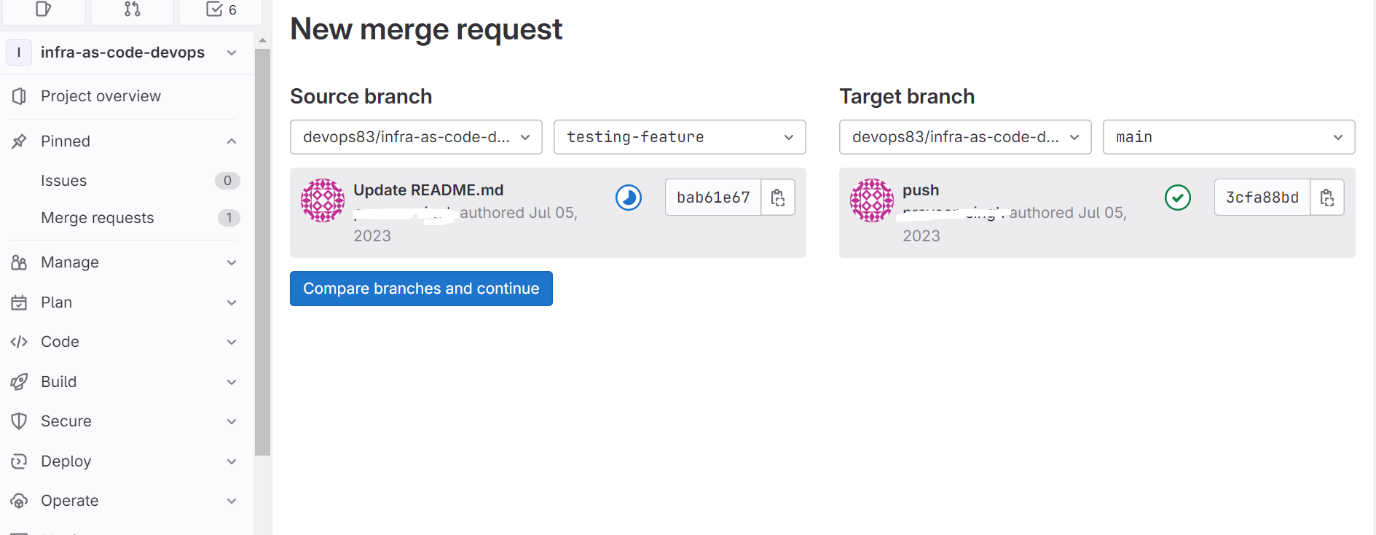
How to run the pipeline:-

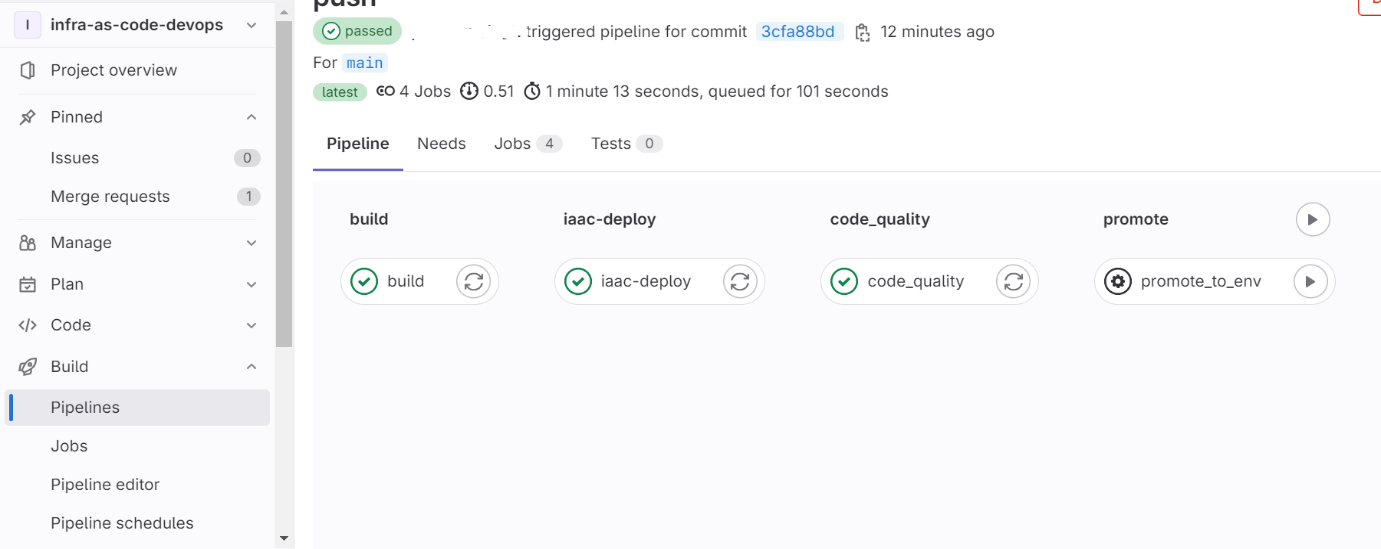
Login to Gitlab-> go to the repo-> Click on the build

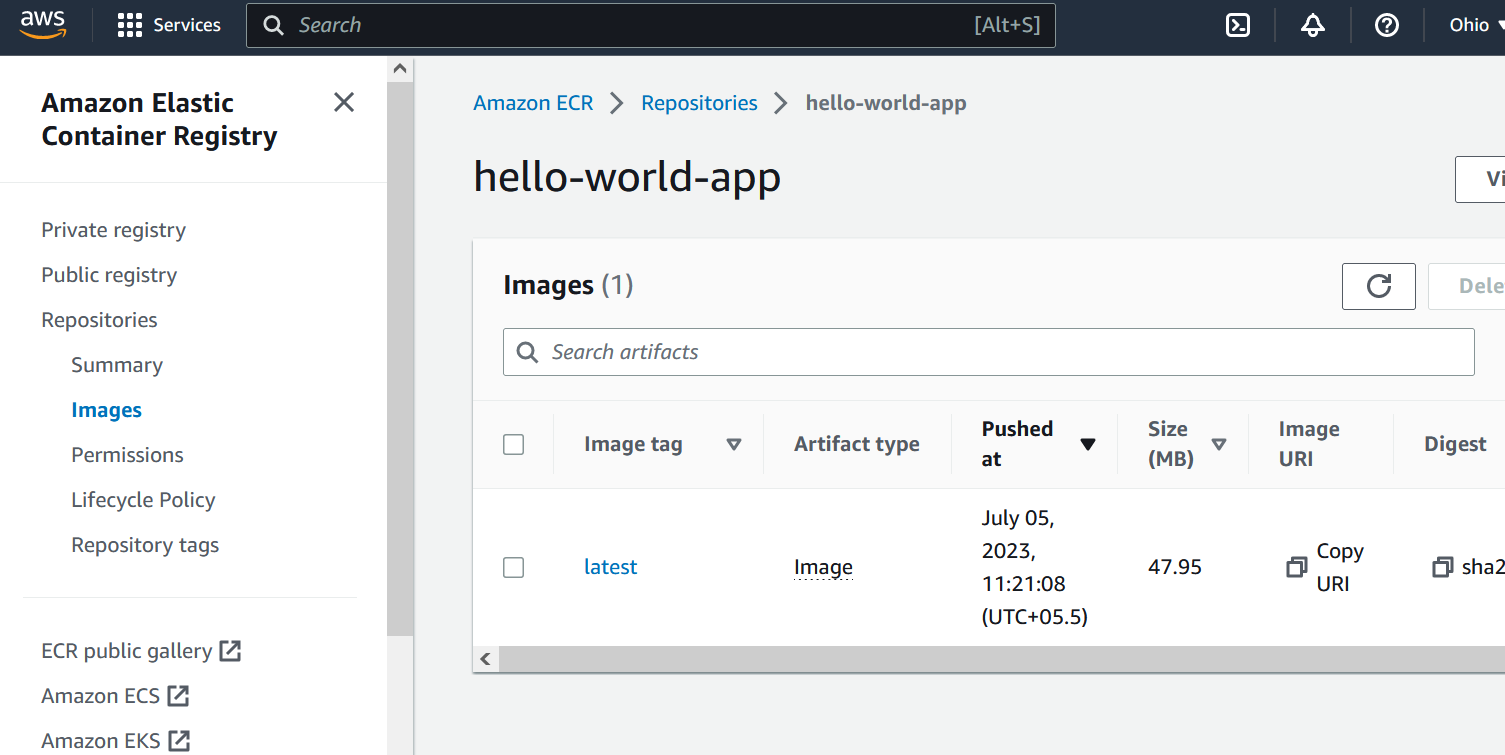


PR request to merge from feature to Main branch









The app is deployed on newly created machine using terraform.

