Pipeline Project: EKS

Pipeline Project: Jenkins pipeline, docker, sonarCloud, ECR, and EKS:

Problem Statement:

-A client created a software project used by doctors and patients for geolocation. The doctors can see their patients through the app and the patients can seek medical assistance through the app too. The application is built using Java, HTML, and JavaScript for the source code and maven as the build tool. The client wants to set up a code analysis for the project and build a pipeline to deploy the application using Amazon EKS.

Solution Statement:

-I am going to create a Jenkins server to manage the pipeline and implement the tools to set up the code analysis process. Finally, integrate the Jenkins server with Amazon EKS to deploy the artifact from Amazon ECR to the EKS cluster.

Key components:

-Linux Centos7 server: For the Jenkins server

-GitHub: To store the source code

-Git: To access and manage the Github and the Jenkins server

-AWS: For providing cloud services

-Amazon ECR: To store the final artifact

-AWS IAM user: To provide access between the Jenkins server and the ECR repository

-Docker: To build the dockerfile

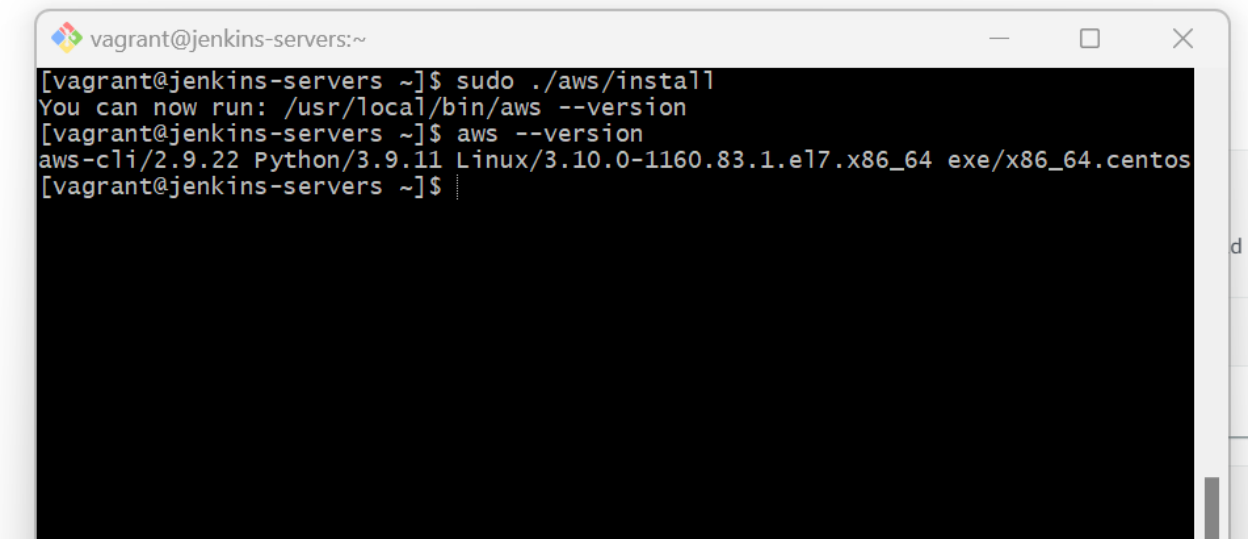
-Maven Pipeline: To automate the build of the project

-Jenkinsfile: For the maven pipeline

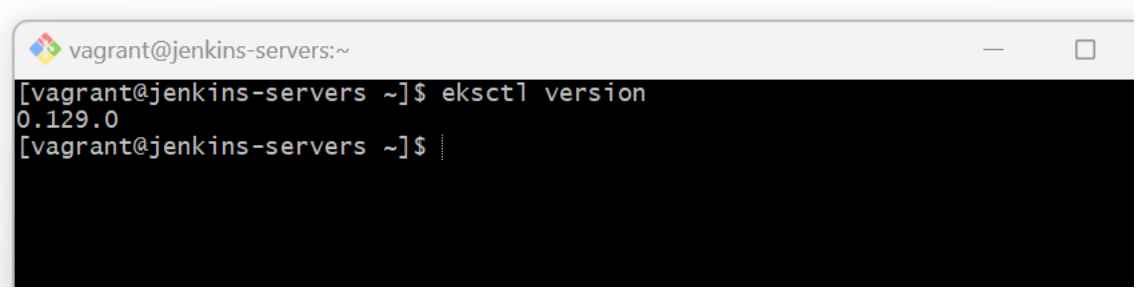
-Amazon EKS cluster: Running the cluster

-more

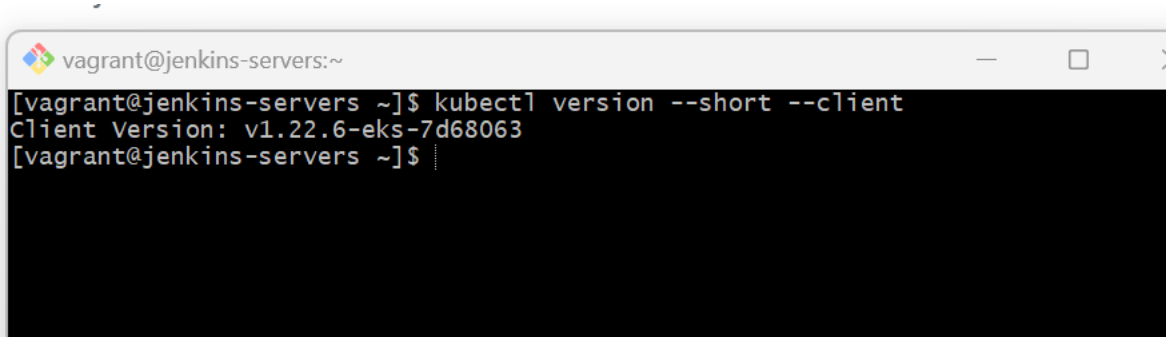
Installing AWS CLI to Jenkins server



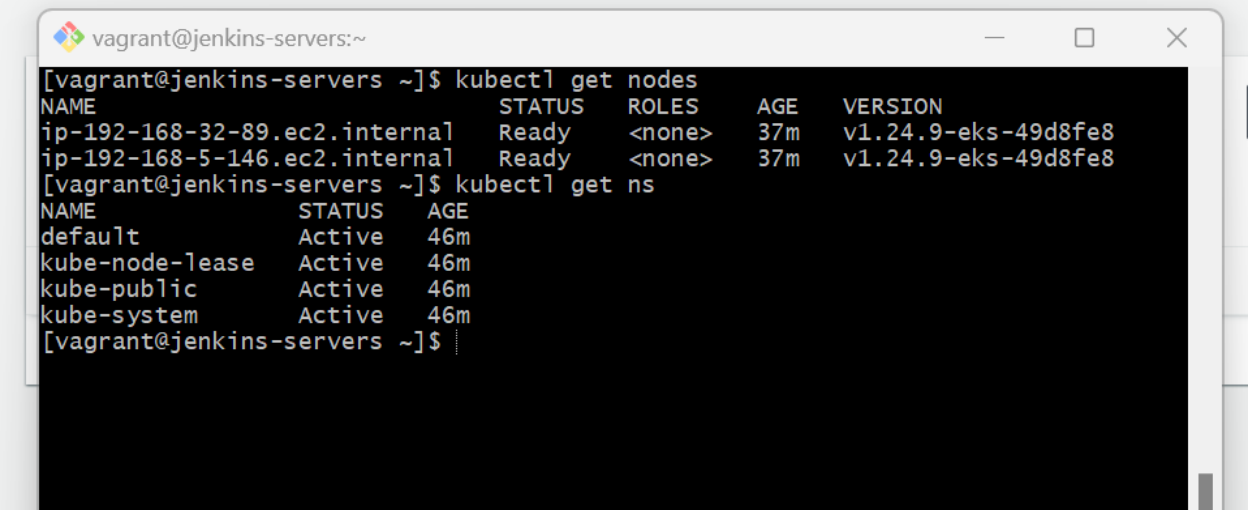
Checking to see if I have eksctl service on the Jenkins server



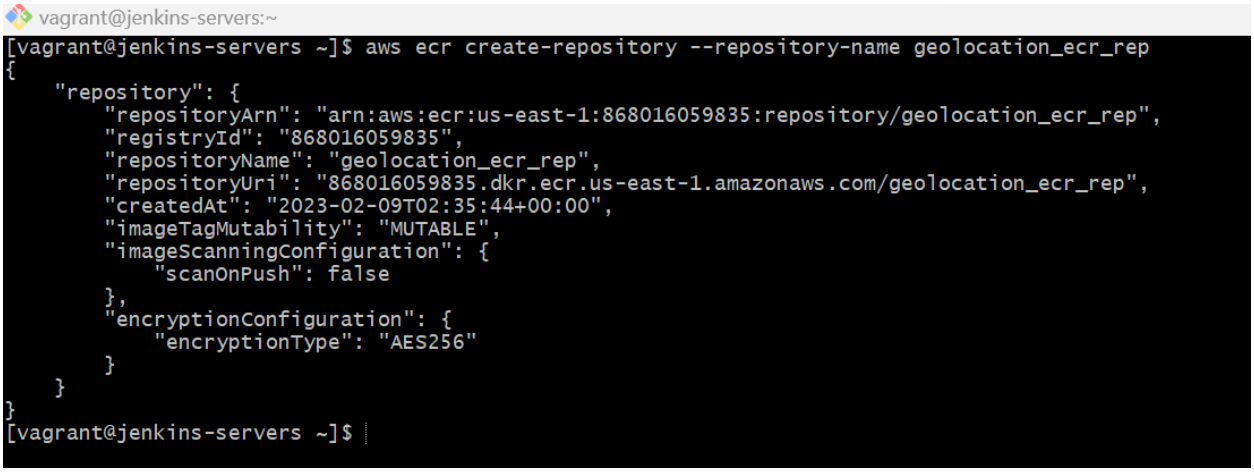
Make sure that I have kubectl installed on the Jenkins server



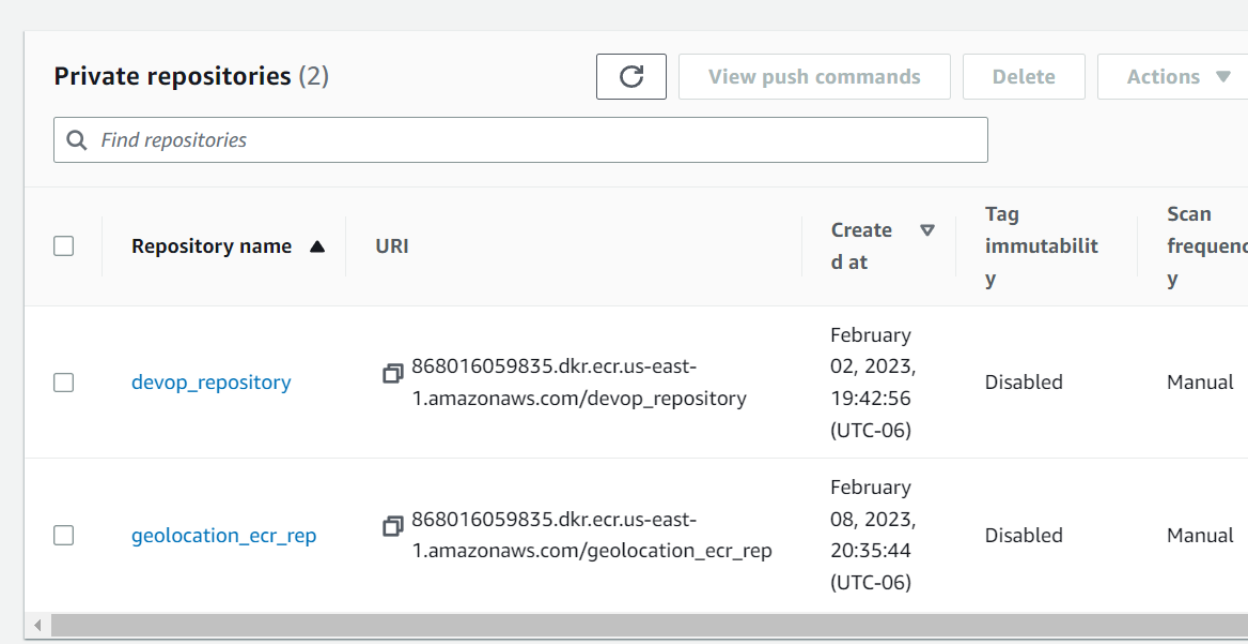
Cluster nodes



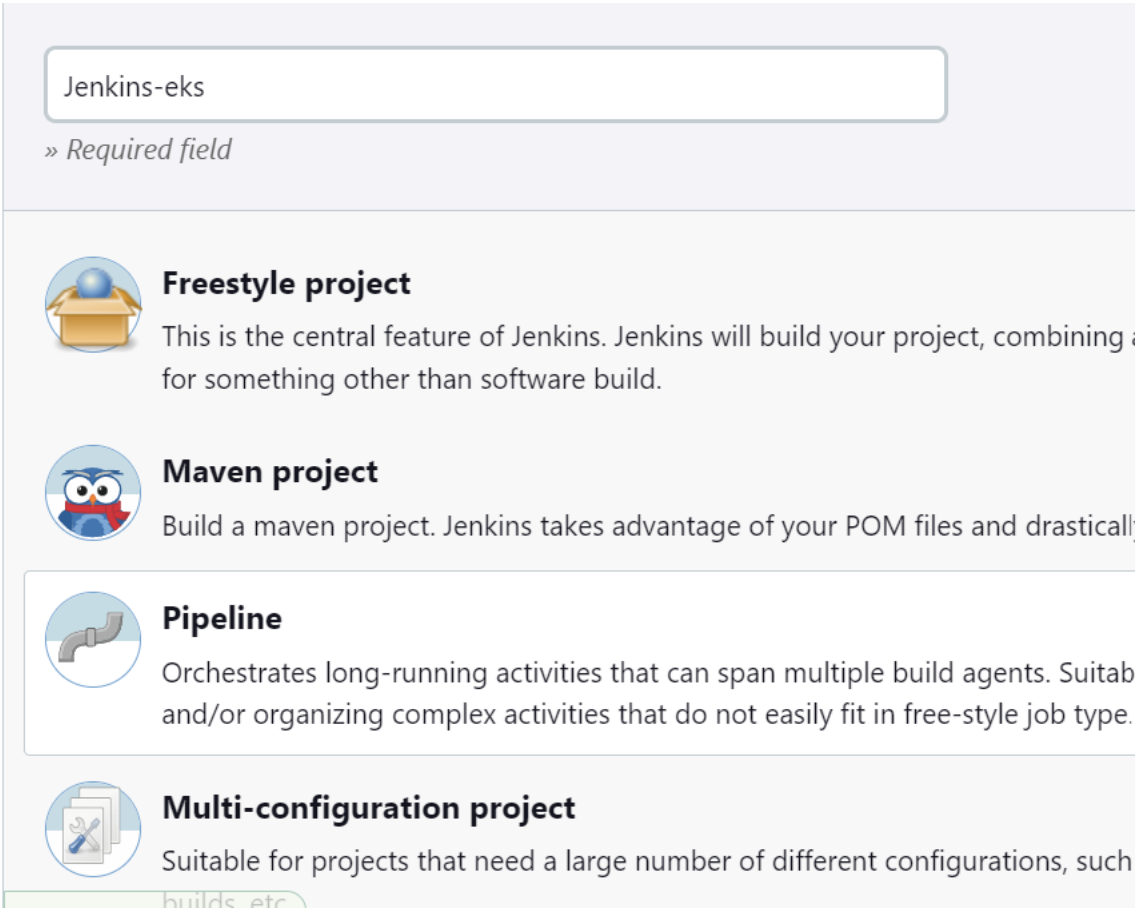
Creating an ECR repository to store the Docker image



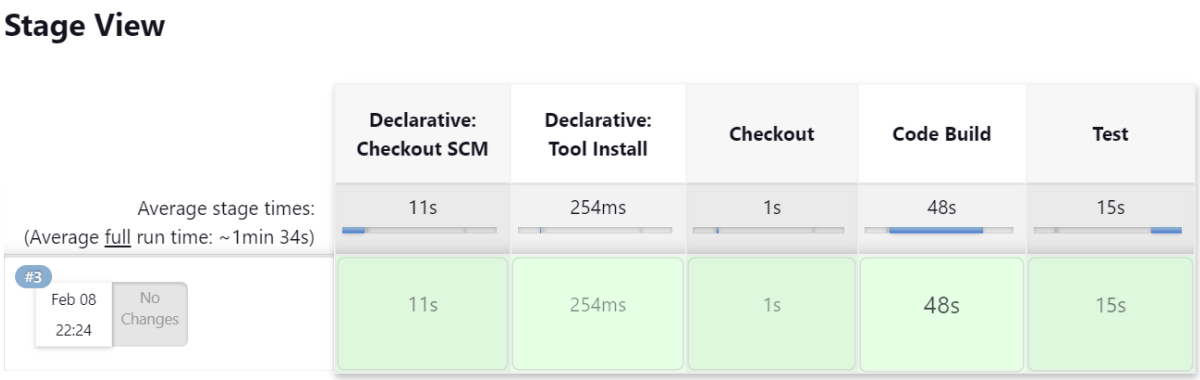
Amazon ECR console



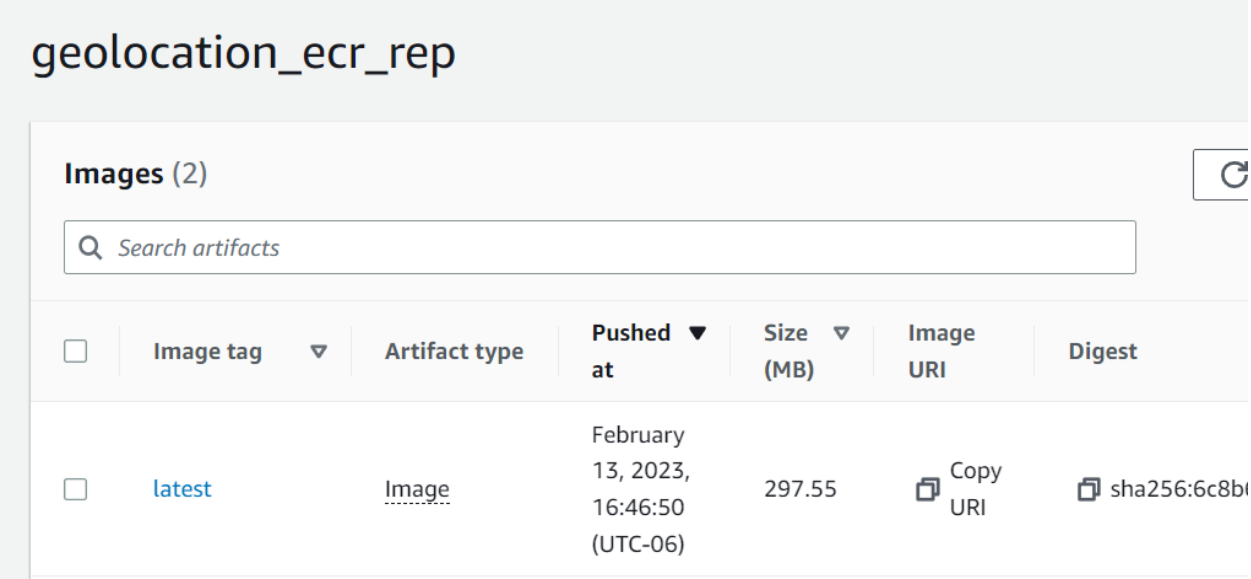
Creating the project in the Jenkins console



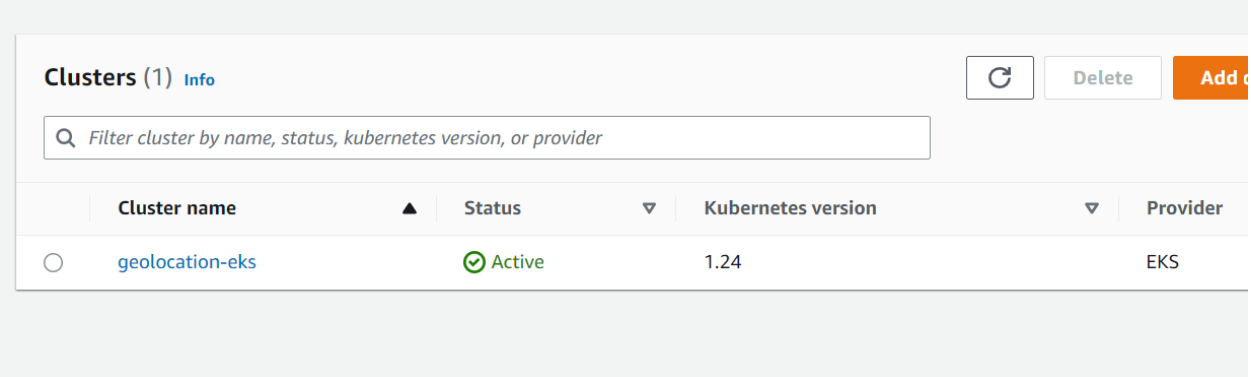
Testing the pipeline through stages



Deploying the application through Jenkins



Active cluster in the AWS account



The application is now running in the cluster.