**Continuous Integration for Docker**

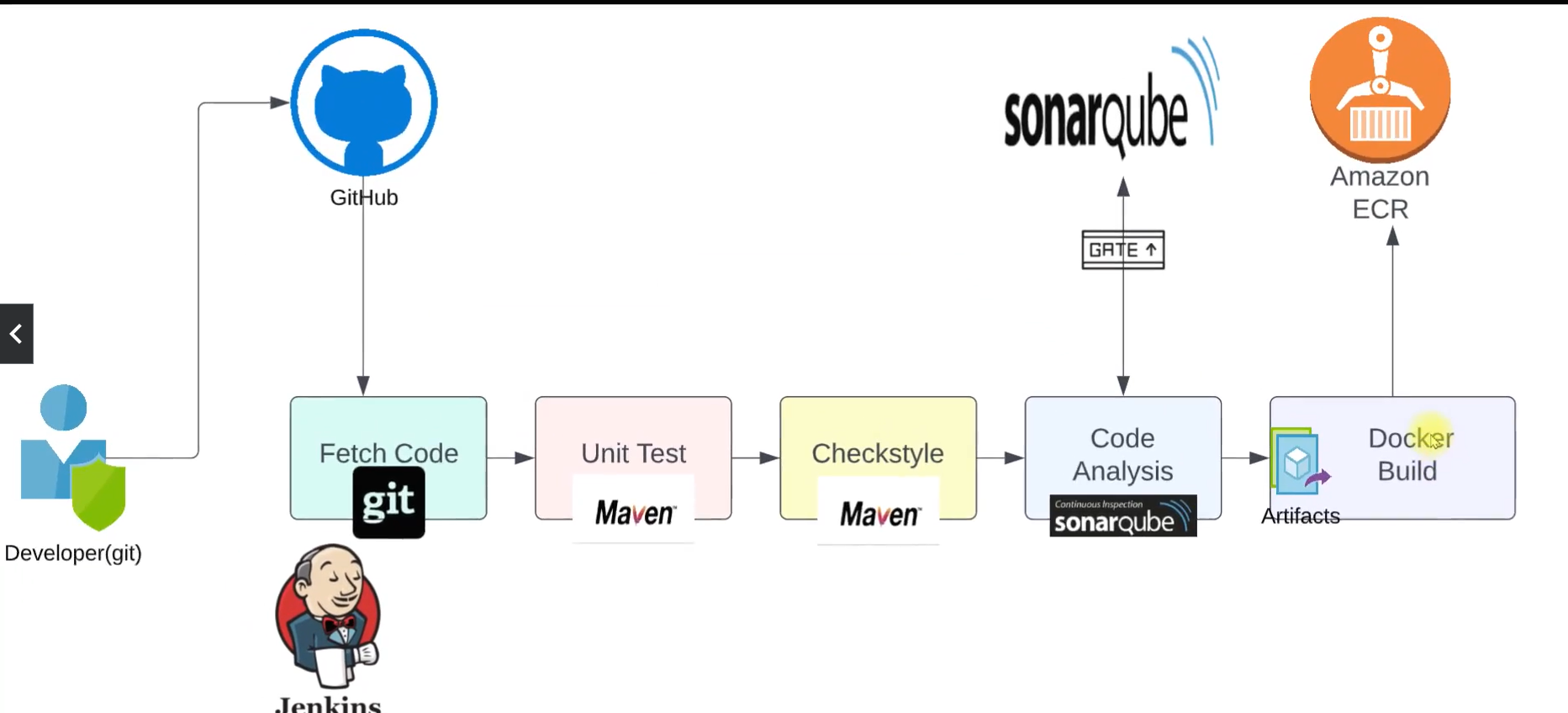
**Problem Statement**

Whenever developers write code and merge it into the central repository, it is detected by Jenkins hence code is fetched and run the test unit test, then will do code analysis with checkstyle and then again code analysis with SonarQube and upload the result on SonarQube server, waits for quality gate when everything is good we are going to build docker images. The Docker image will contain artifacts.

**What we will be using for docker?**

We will be using Amazon ECR, elastic container registry.

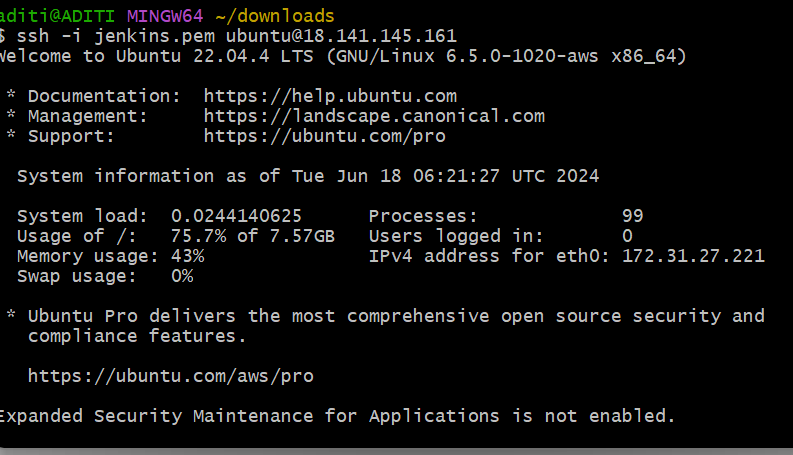
**Related Image**



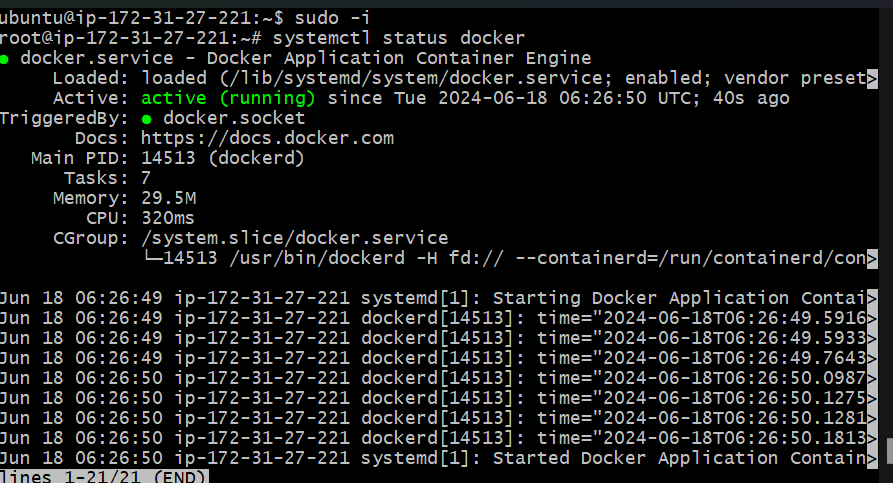
**Steps of execution**

* Install the docker engine in Jenkins
* Add Jenkins user to docker group & reboot
* Install AWS CLI
* Create IAM User
* Create ECR repo
* Plugins
* ECR, docker pipeline, AWS SDK for credentials
* Store AWS credentials in Jenkins
* Run the pipeline

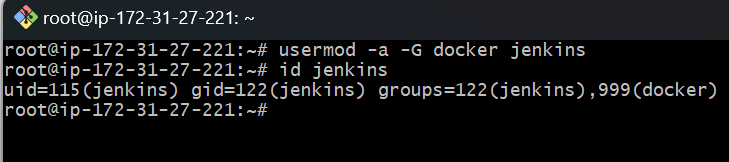
1. SSh into Jenkins Server

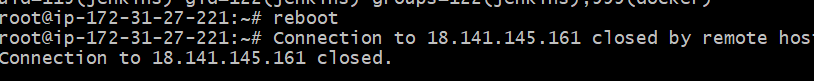


1. Search for Docker engine in google and install it for Ubuntu

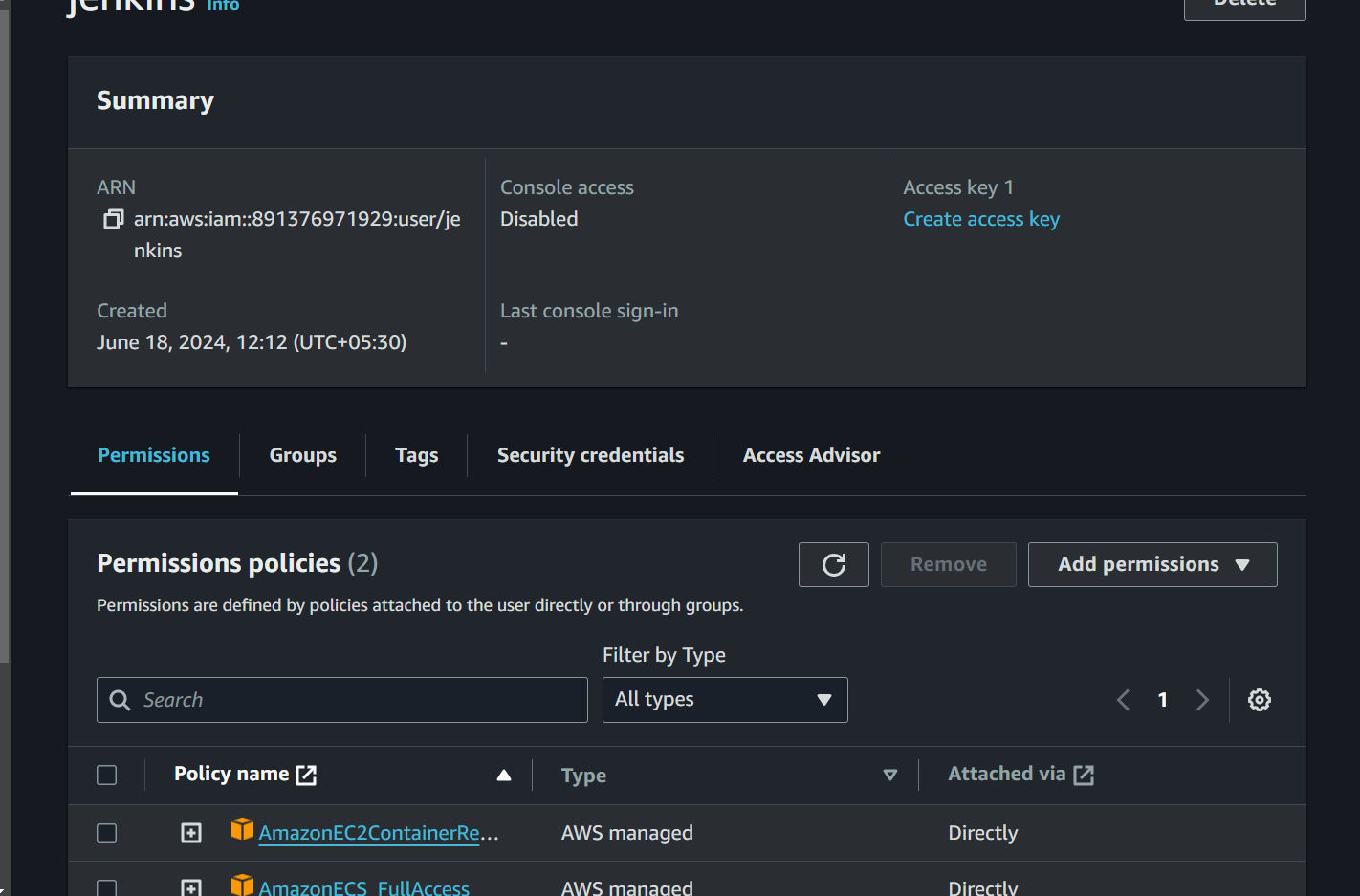


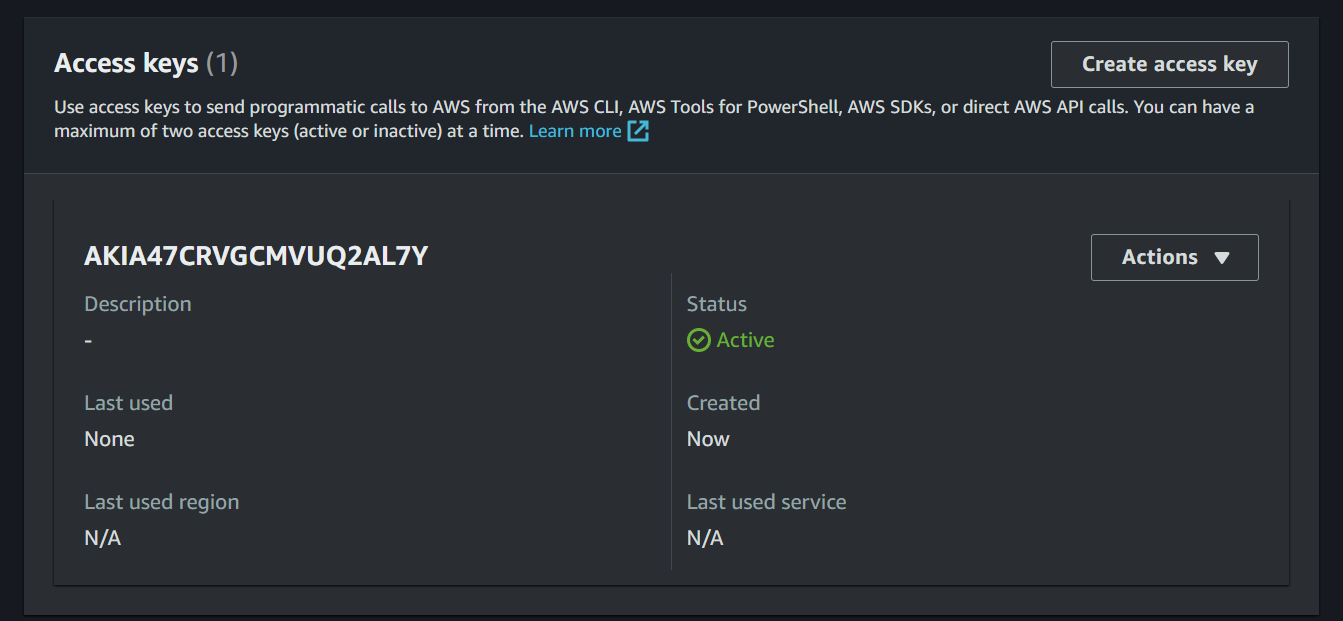
1. Add Jenkins user to the docker group



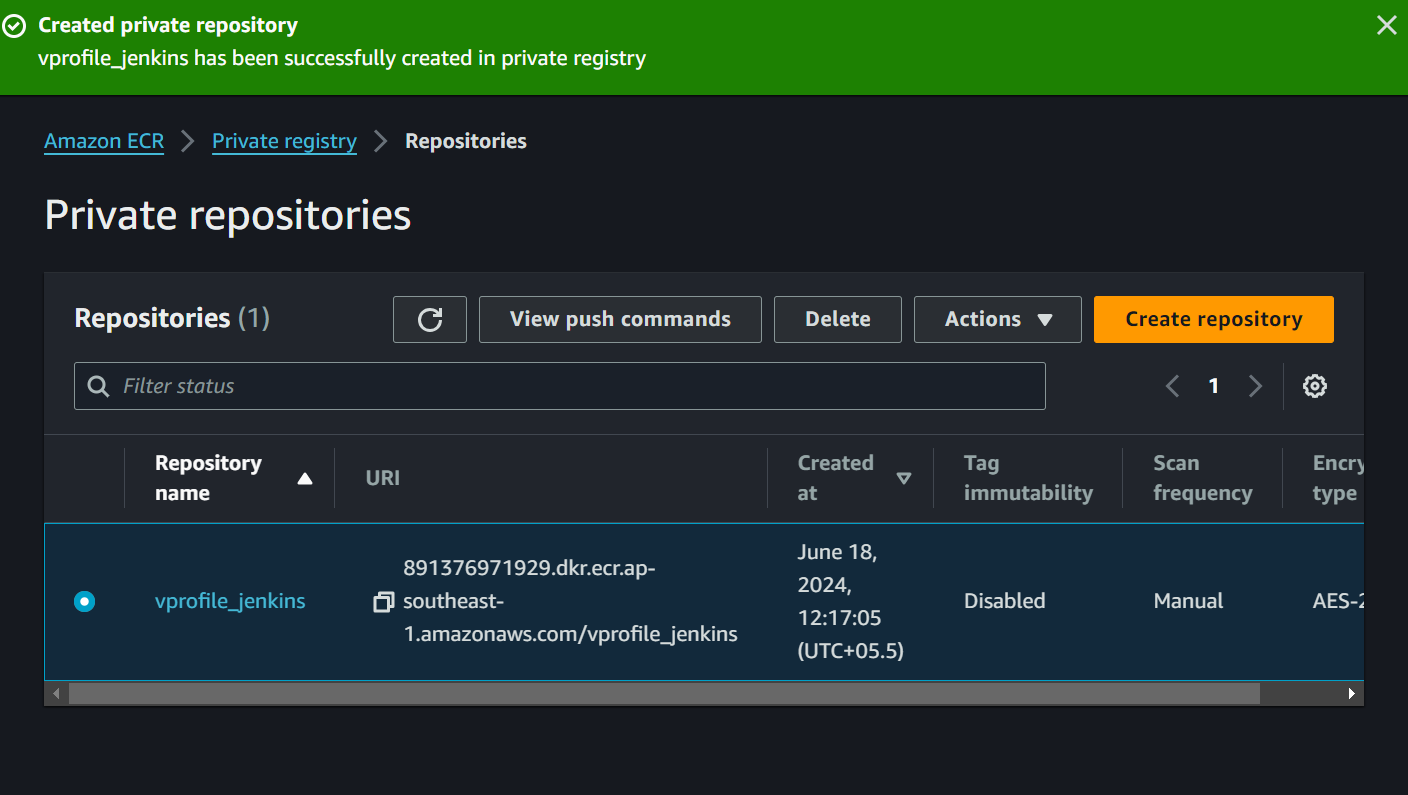


1. Creating an IAM user

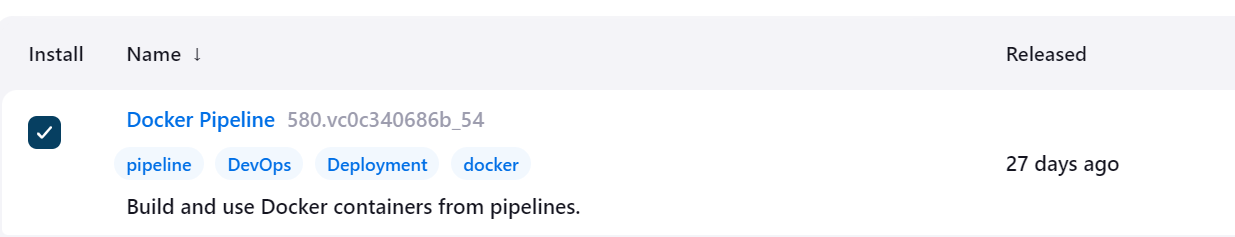


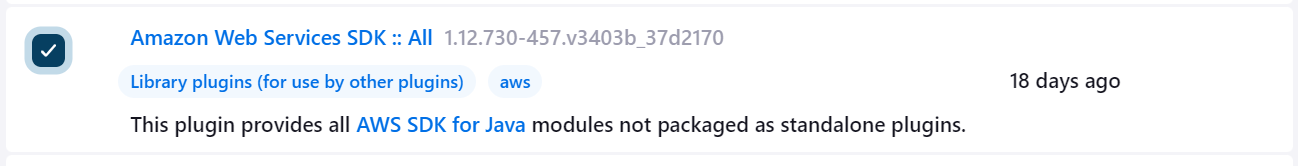


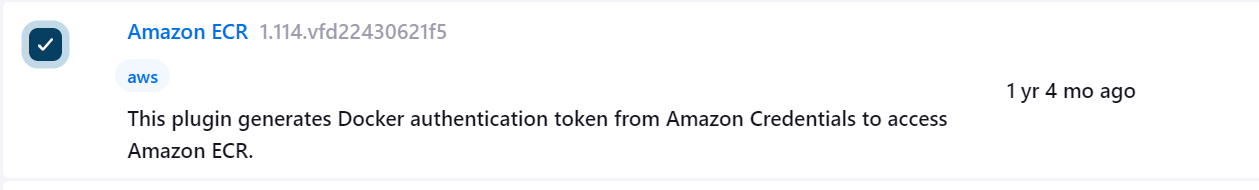
1. Elastic Container Registry

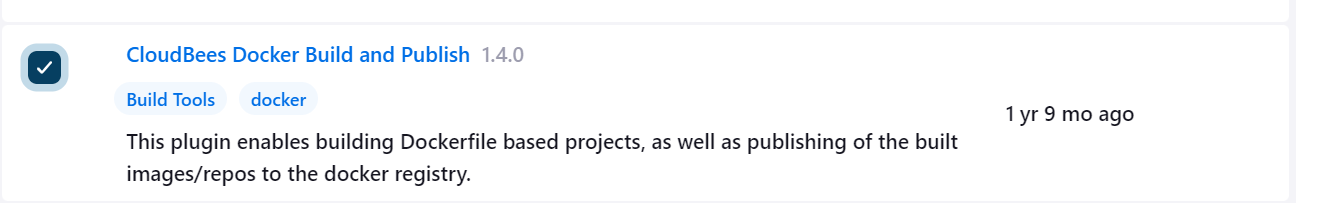


1. Installing Plugins

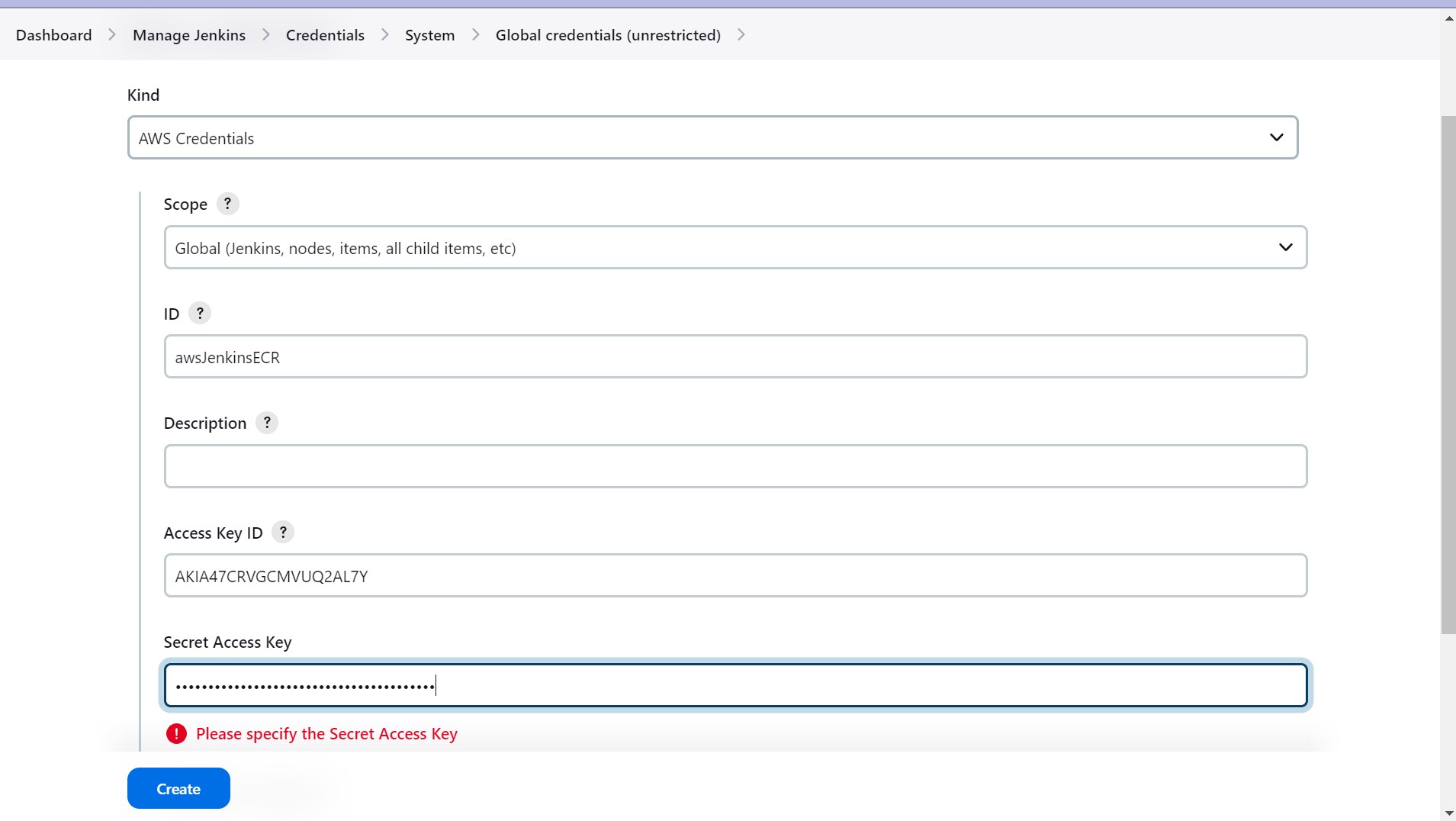




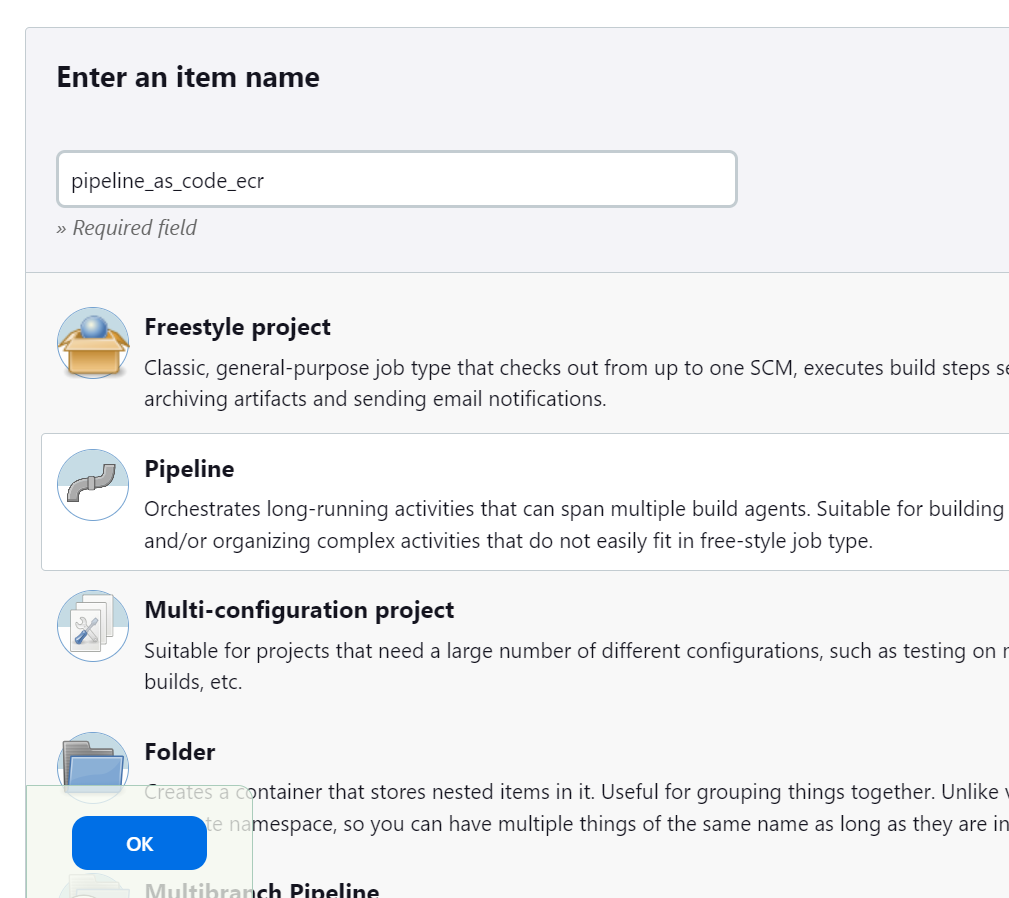


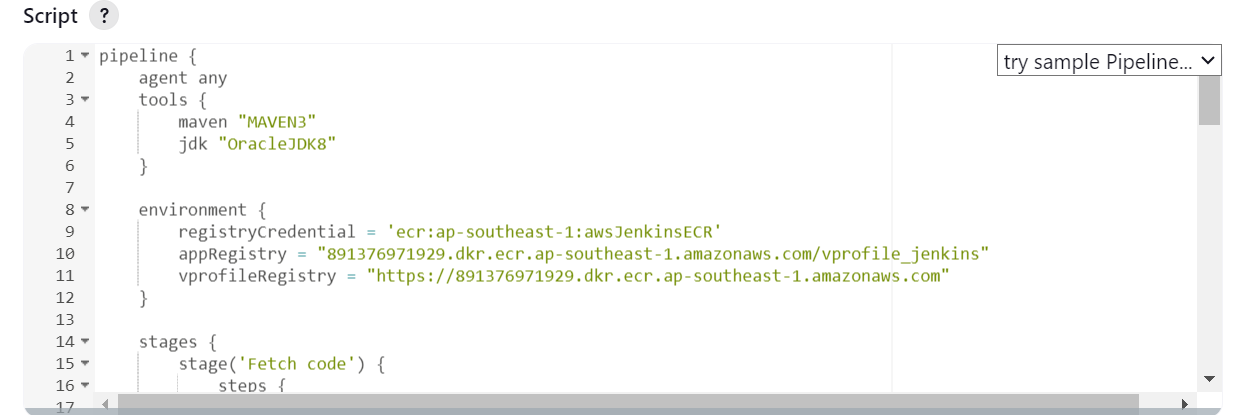


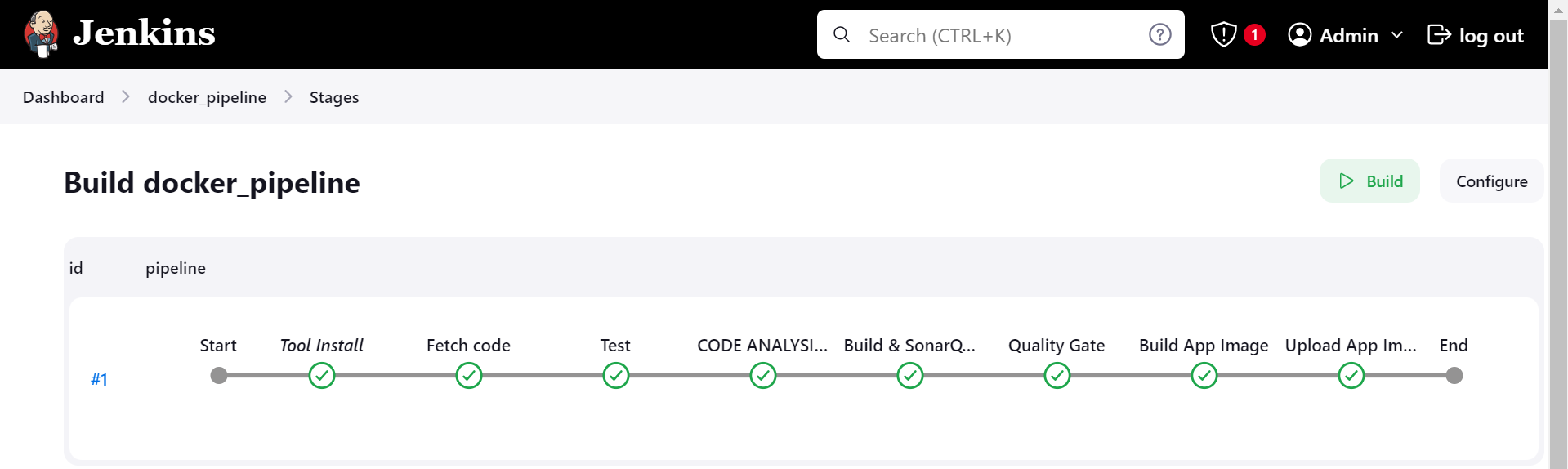
1. Go to Manage credentials and store the credentials for AWS

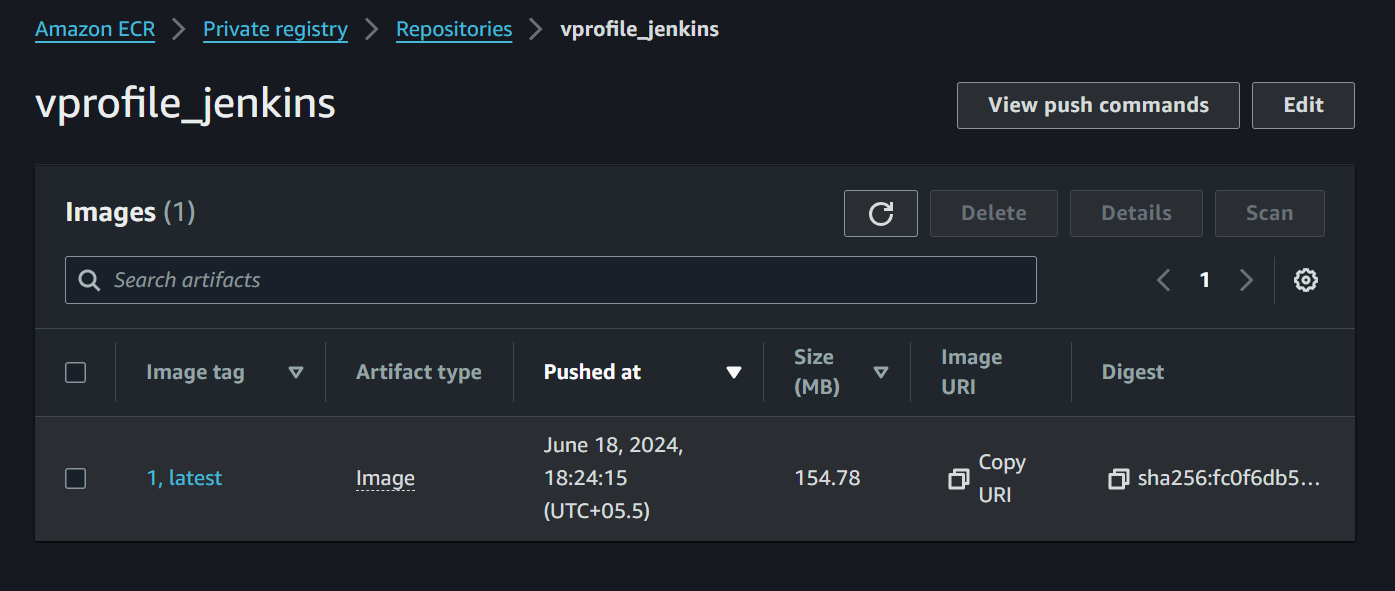


1. Run and build this on Jenkins



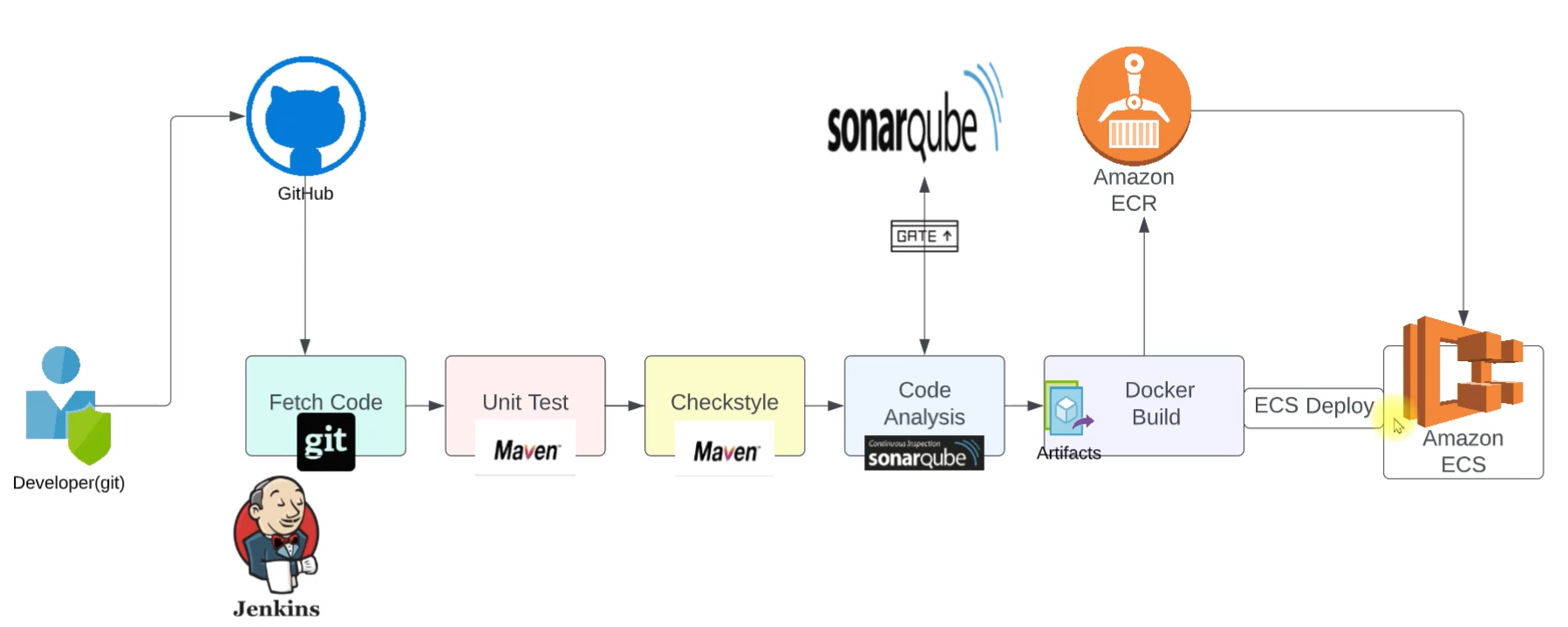






**Hosting**

* Host this docker image on docker solution like ECS Deploy



* Amazon ECS is going to fetch the image from Amazon ECR and host it on ECS

**Container hosting platforms**

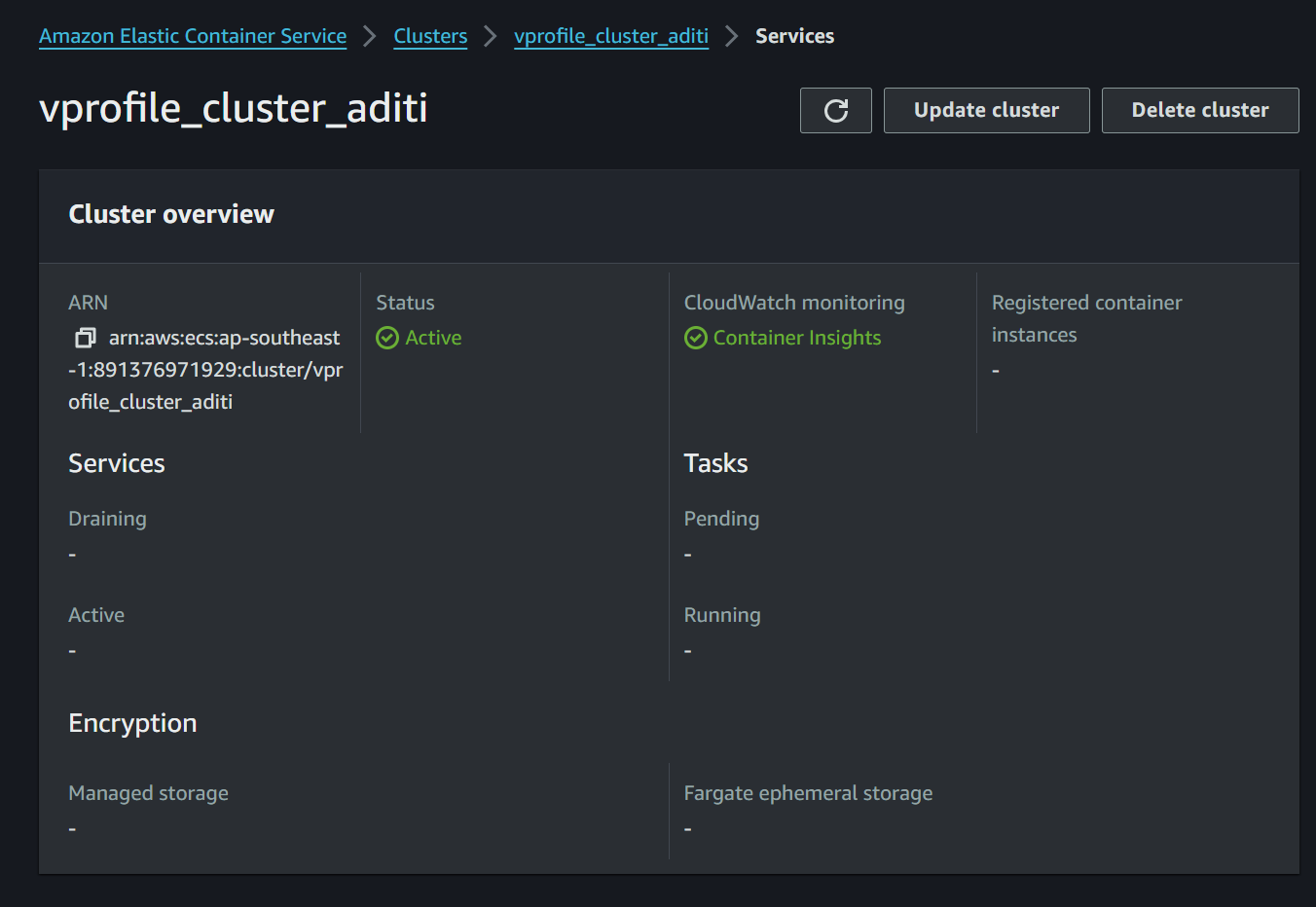
* Host the image on docker which is a local method using the docker engine which uses the command Docker Run
* This is very much for testing and the local development environment.
* Do not give production vibes and related features like High availability, self-healing etc.
* Hence, for the production environment we will go with Kubernetes.
* For now we will use docker.

**Amazon Elastic Container Service (Amazon ECS):** is a highly scalable and fast container management service that makes it easy to run, stop, and manage containers on a cluster.

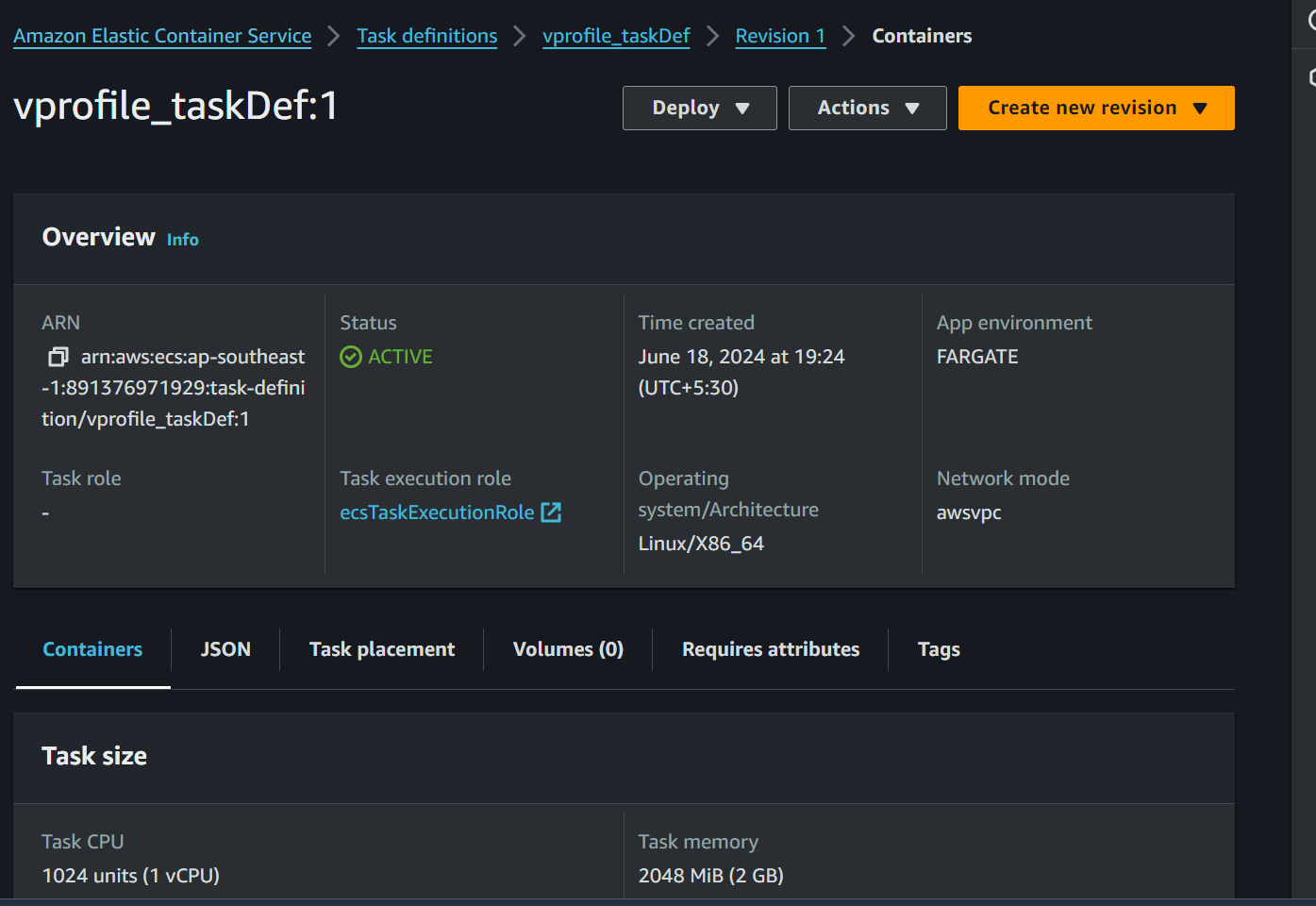
**Deploying image on ECS**

**Steps of execution**

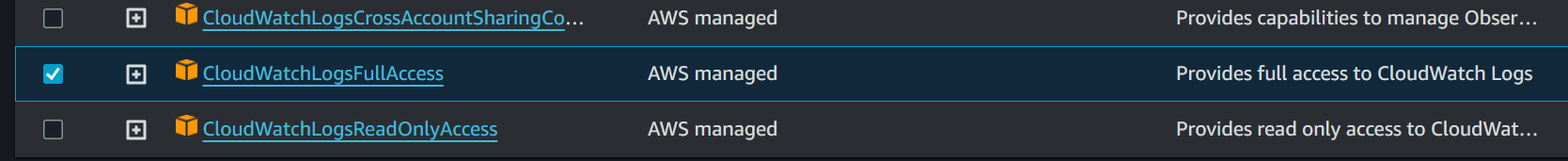
1. Setup ECS cluster and run container on that.



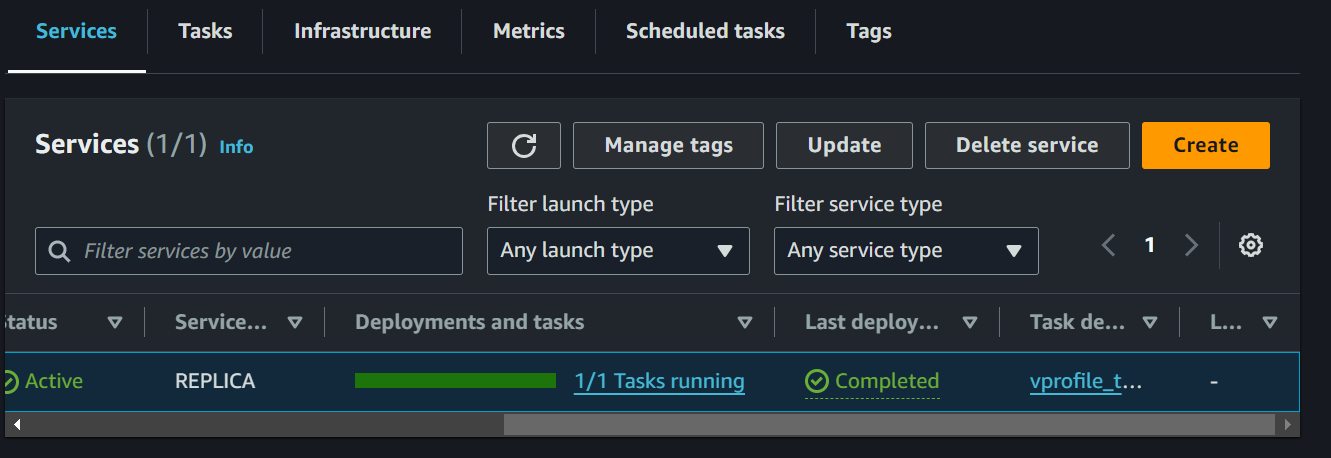
1. Go to task definition: Information about docker. Where to pull docker image, how much RAM CPU etc.



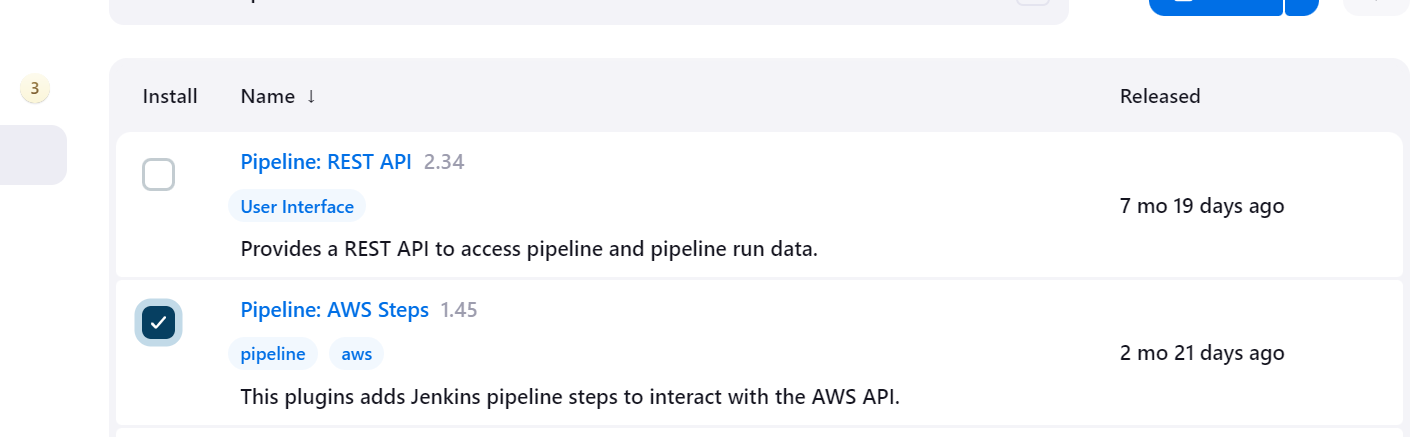
1. Update IAM role ecsTaskExecutionRole



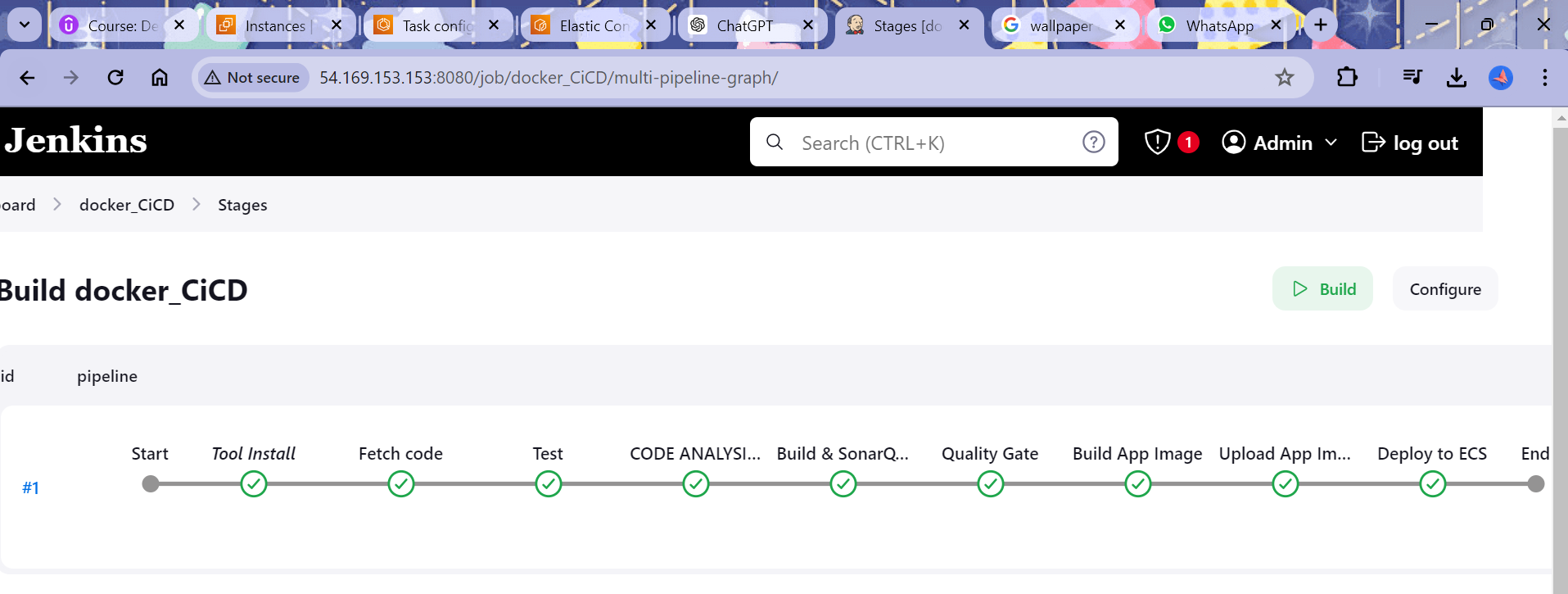
1. Create service which will automate the task of container

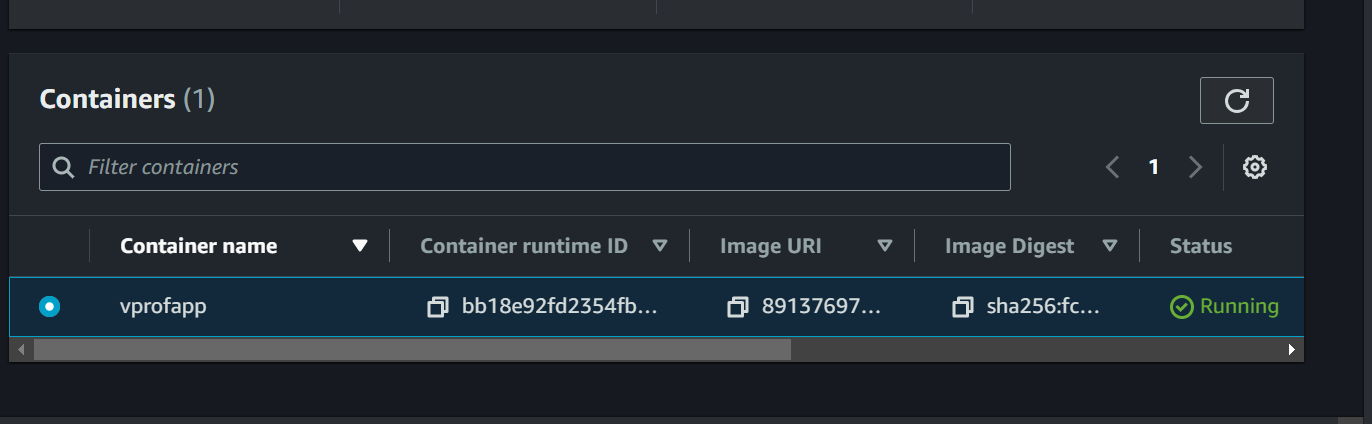


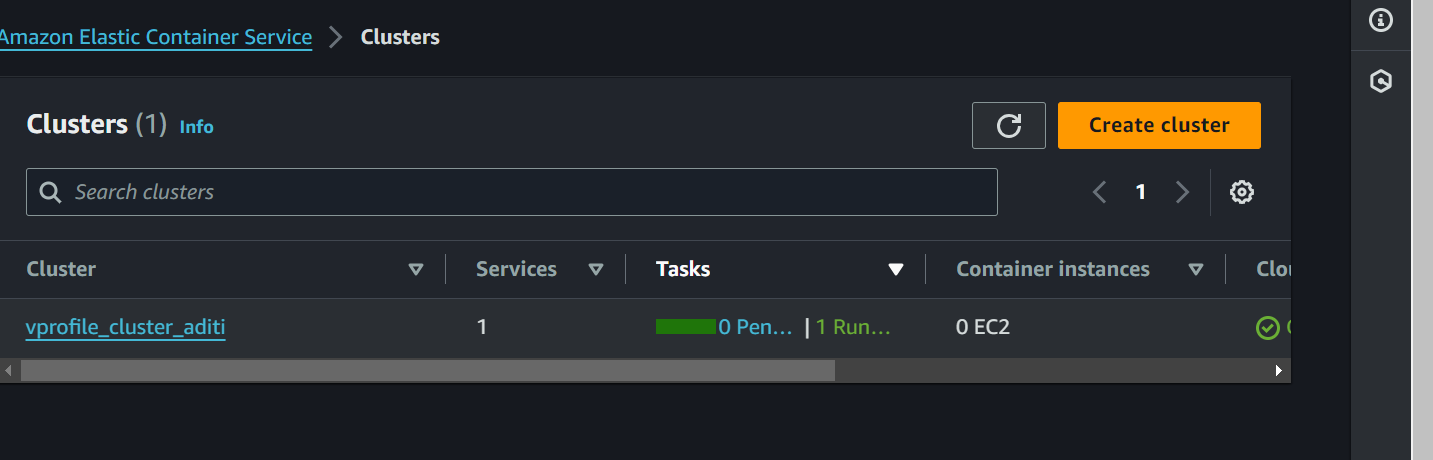
1. Update your Jenkins file
2. Install Plugin



1. Build the Pipeline



****

****

**All DONE!**