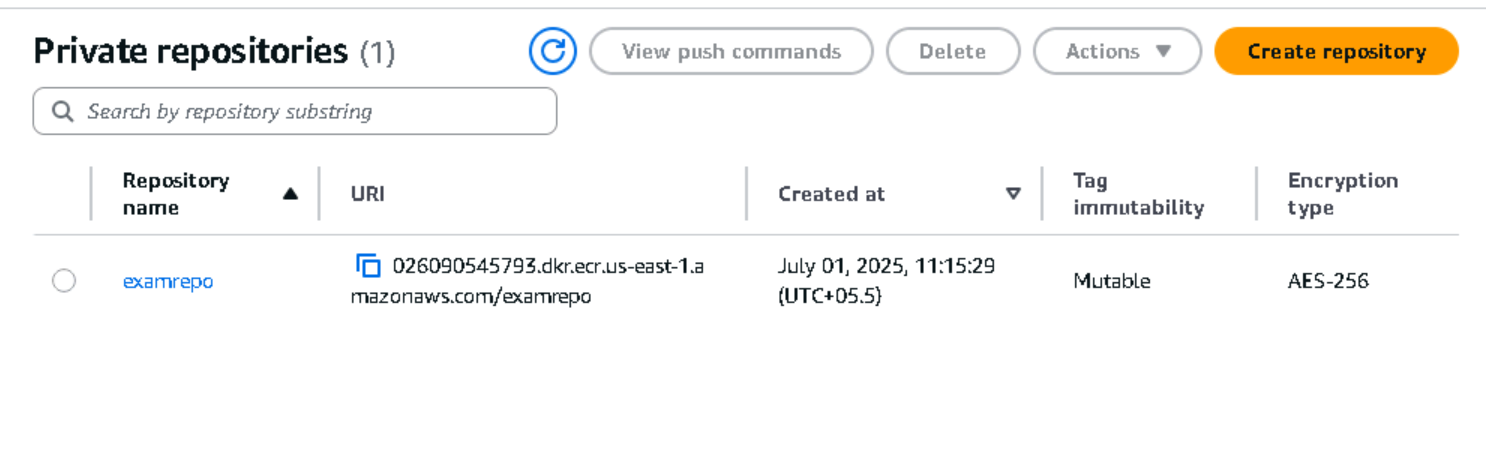
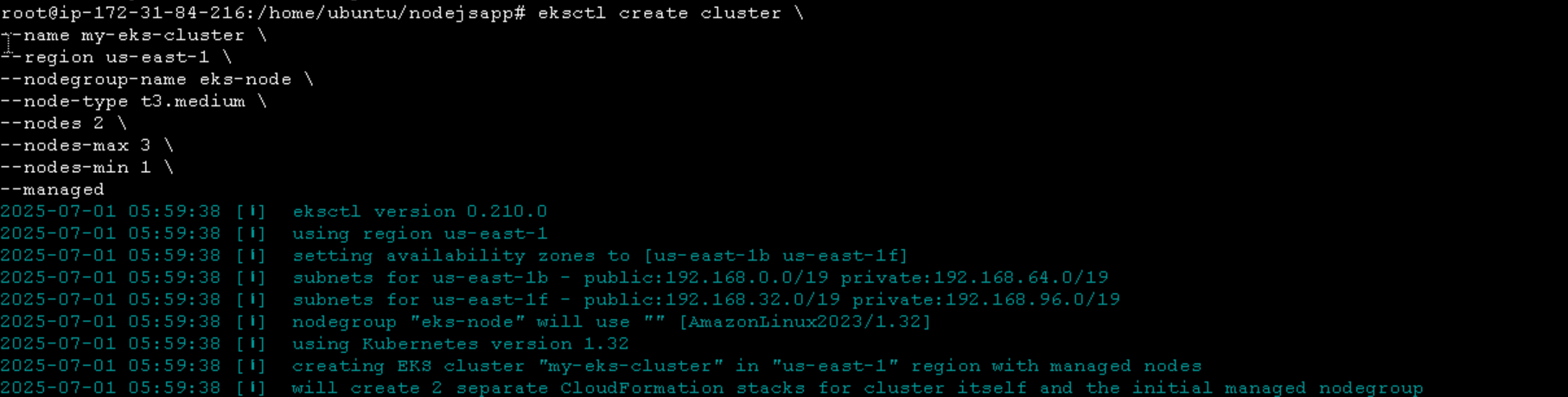
**AWS EXAM1**

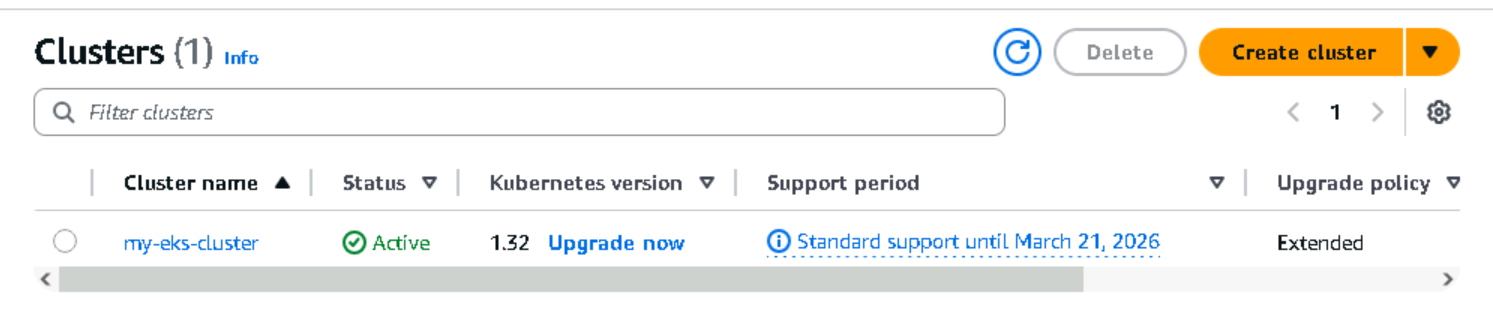
**TASK1 : PROVISIONING AWS RESOURCES**

1. **CREATE ECR REPOSITORY**

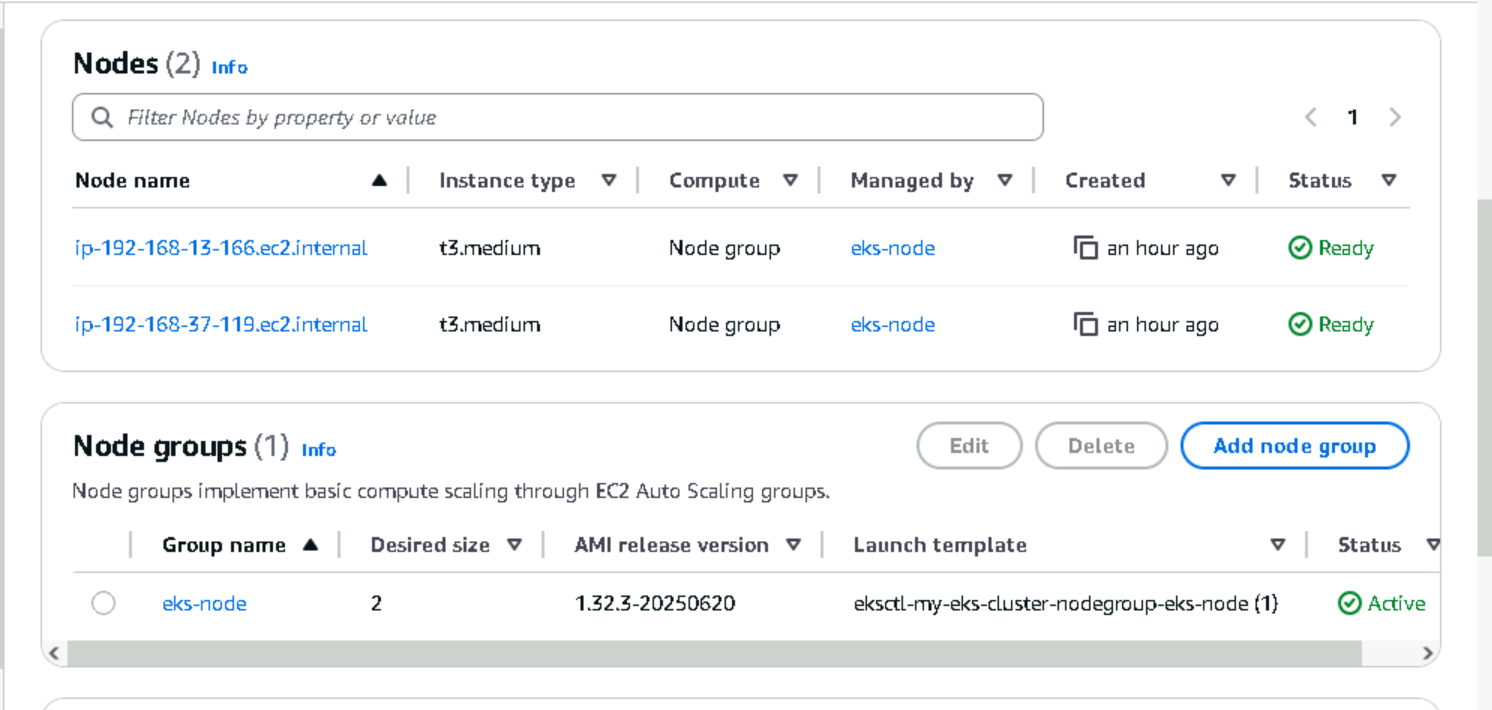


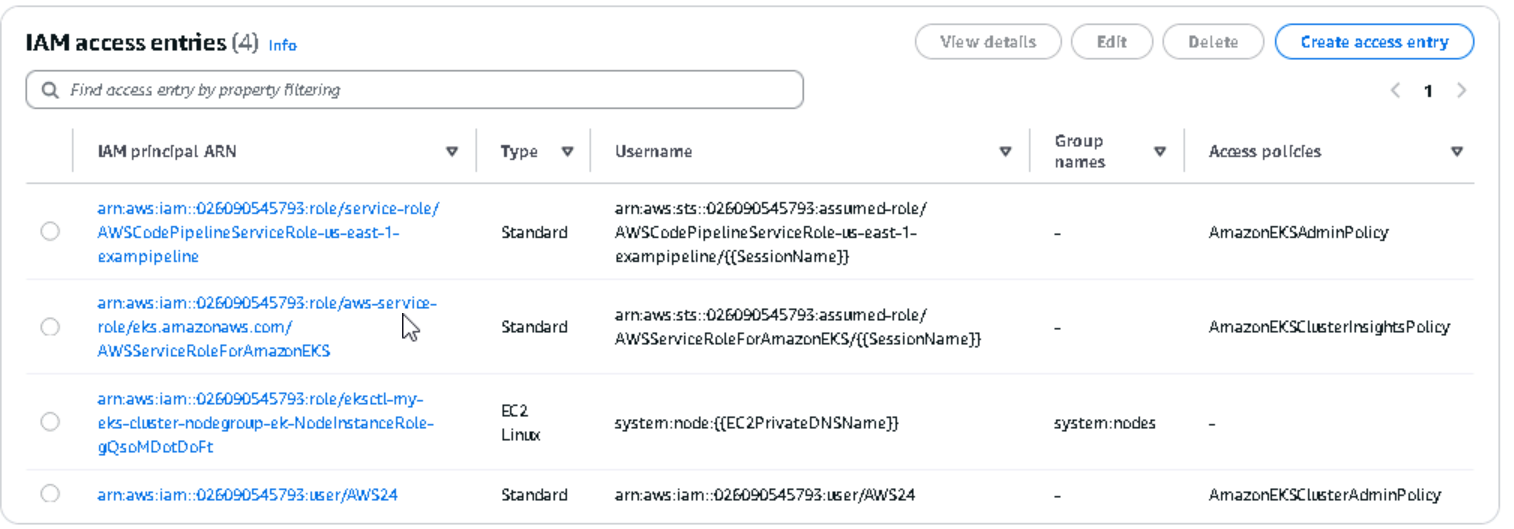
1. **CREATE EKS CLUSTER**





1. **EKS CLUSTER INTEGRAGION with all the nodes, node groups, role etc**





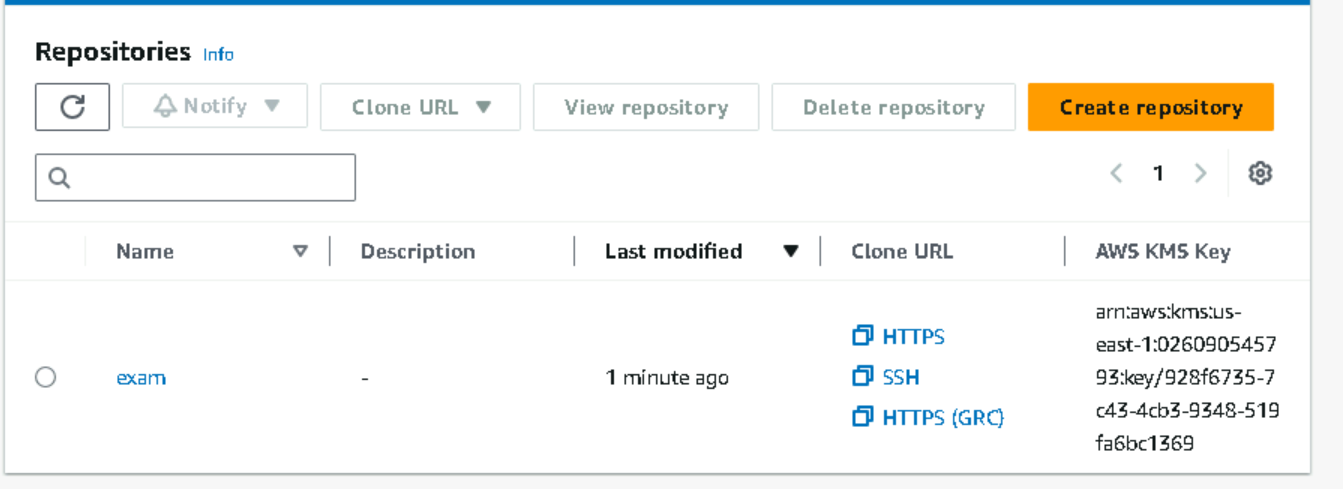
**TASK2: AWS CICD ENVIRONMENT**

**Steps:**

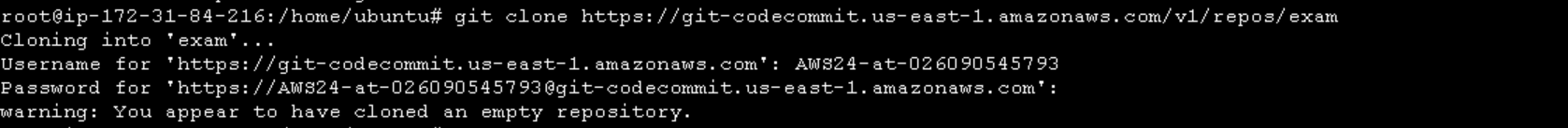
1. CLONE THE REPO GIVEN

git clone <https://github.com/devopsexamus/nodejs>

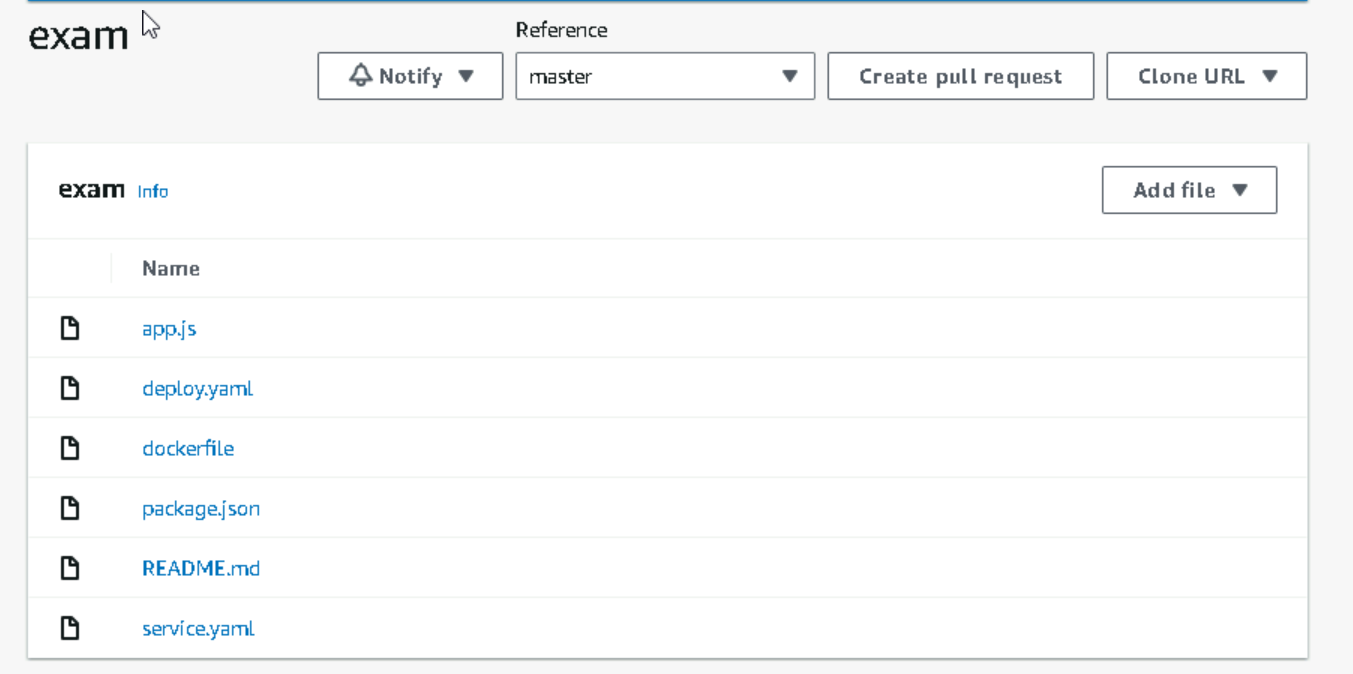
1. Create a code commit repository and copy all files from git repo to code commit repo.
2. **Since don’t have github access for the exam, I used code commit for source stage**
3. **Create a code commit repository**



1. **Cloning code commit repo**



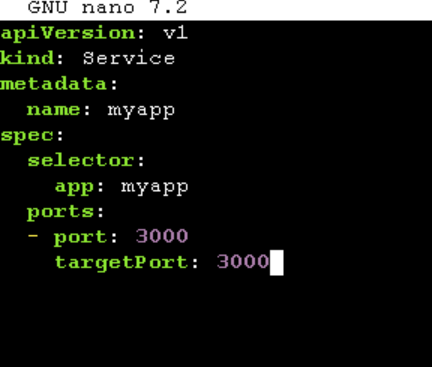
1. **Push all the file to code commit repo**



**Deployment yaml file for deploy**



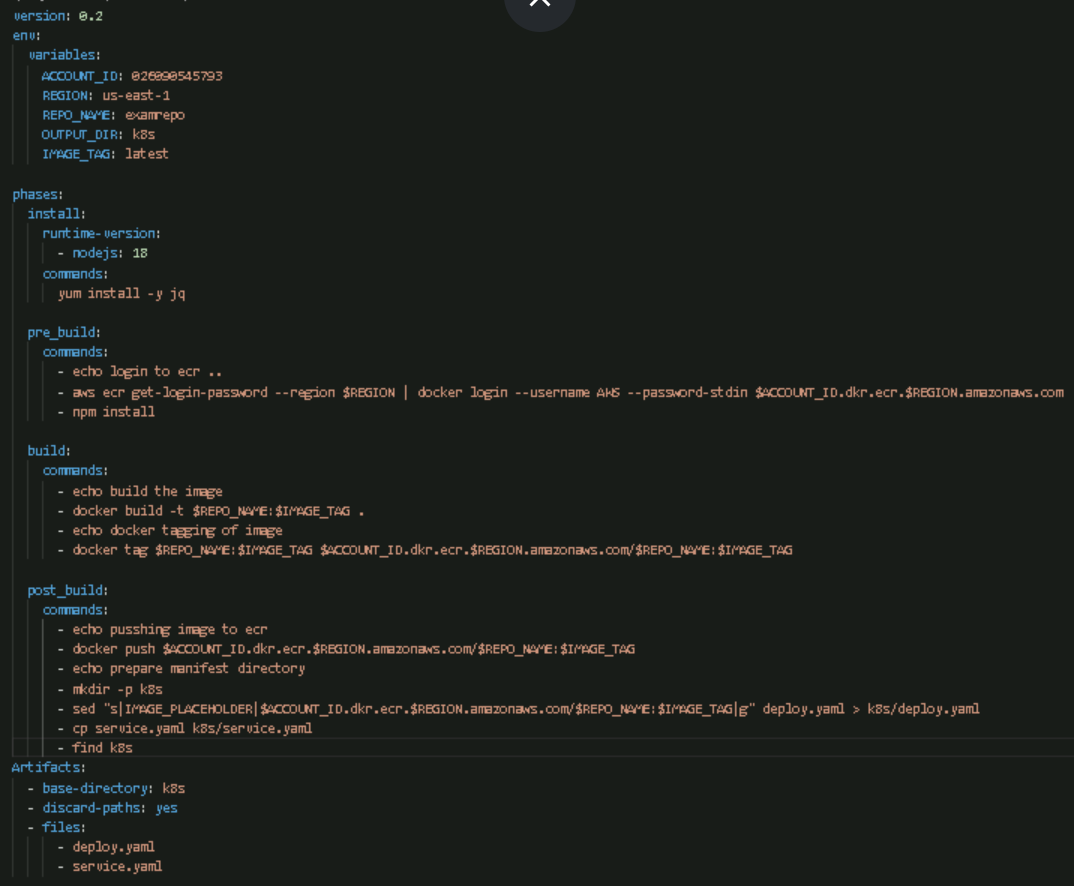
**yaml file for service**



**buildspec.yaml for building and deploying**

**This includes**

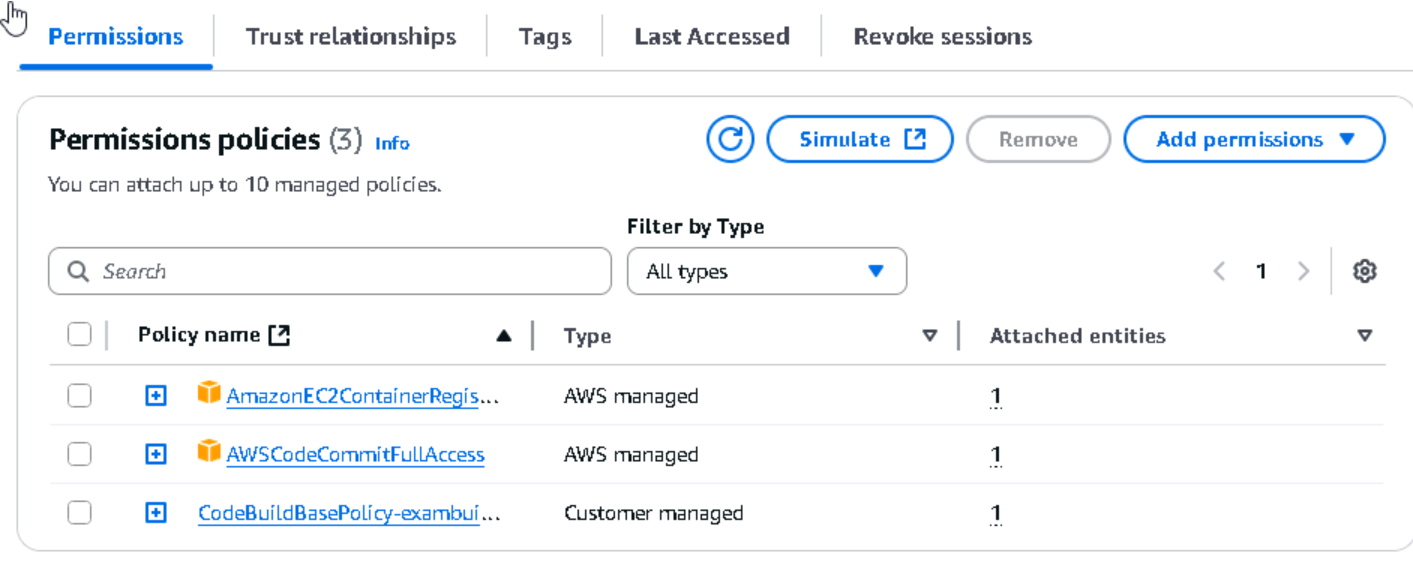
* **Required Installations for the deployment**
* **ECR repository login**
* **Docker file build for docker image**
* **Docker tagging the image to latest**
* **Push the image to ecr repository**
* **Create a build artifact manifest directory for yaml file to store (k8s)**
* **Copy all file to manifest directory**
* **Apply all files for deploy to eks cluster**



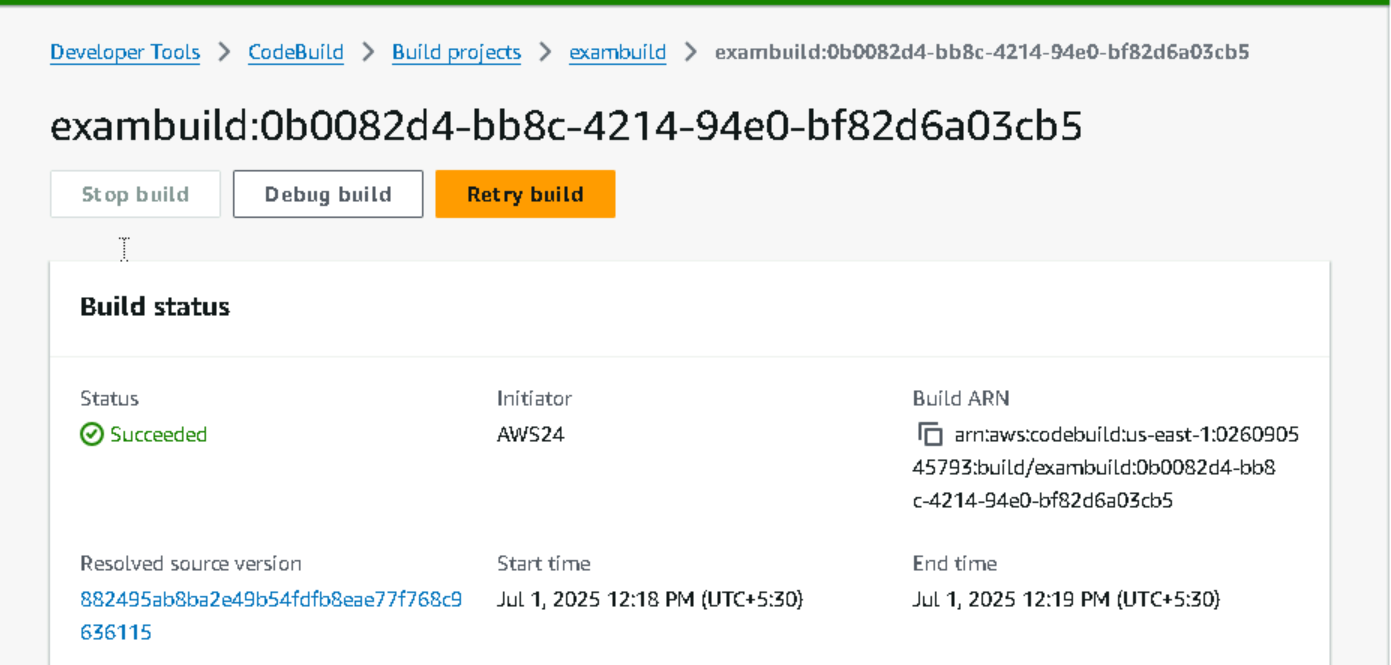
**TASK3: CONFIGURE CICD USING AWS CODE PIPELINE**

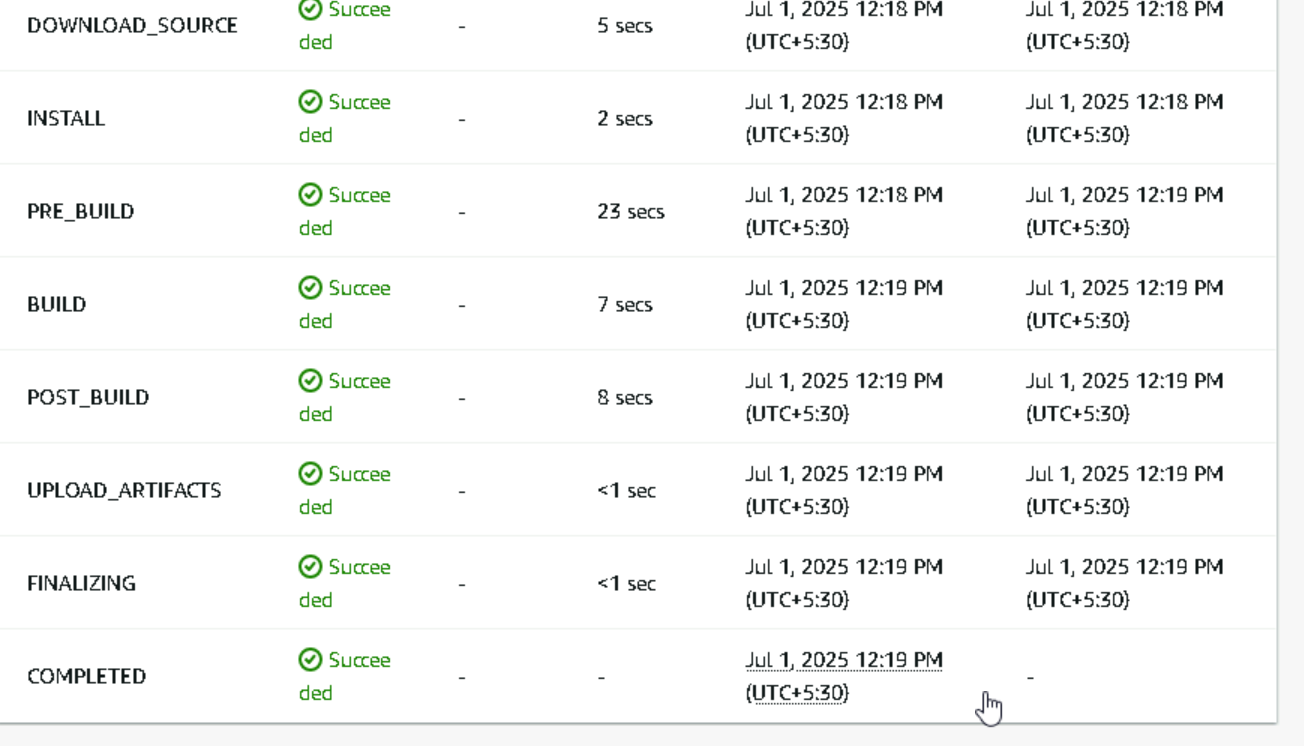
1. **CREATE A CODE BUILD**

GIVE PERMISSIONS FOR CODE BUILD FOR ACCESSING CODE COMMIT AND ECR REGISTRY



CODE BUILD SUCCESS



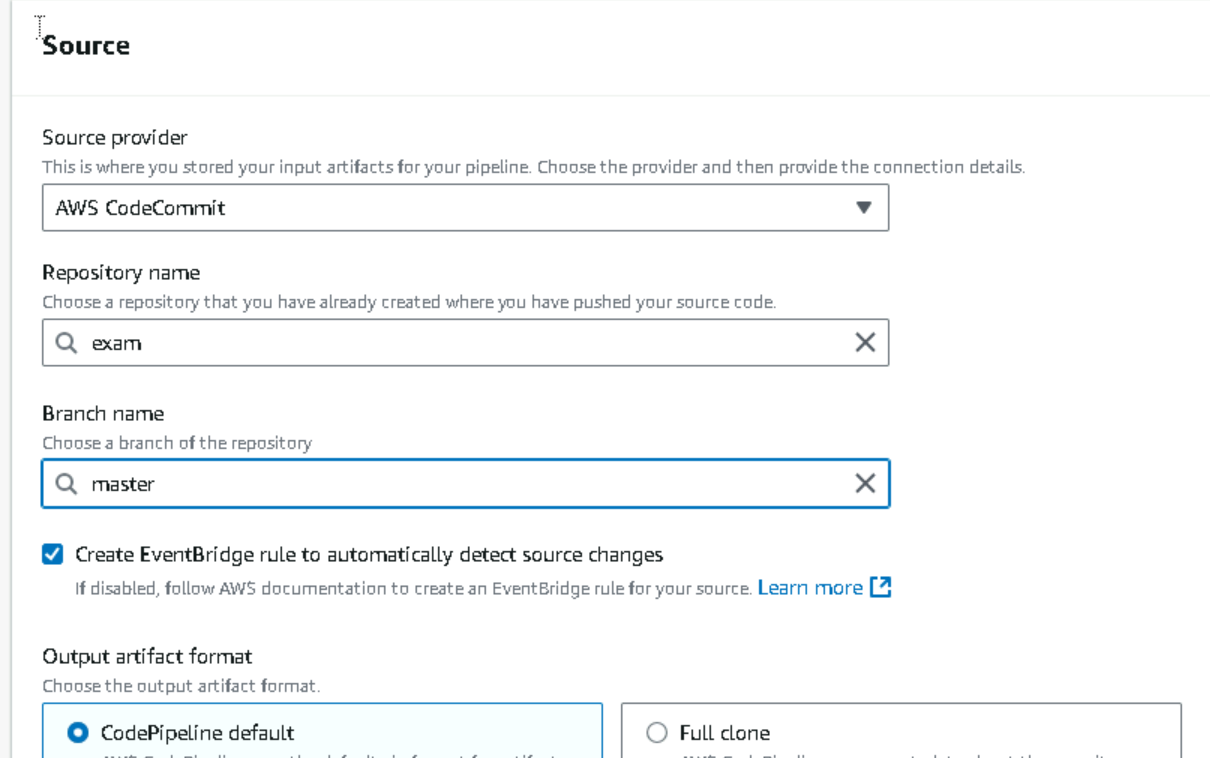


1. **CREATE CODE PIPELINE FOR DEPLOYMENT**

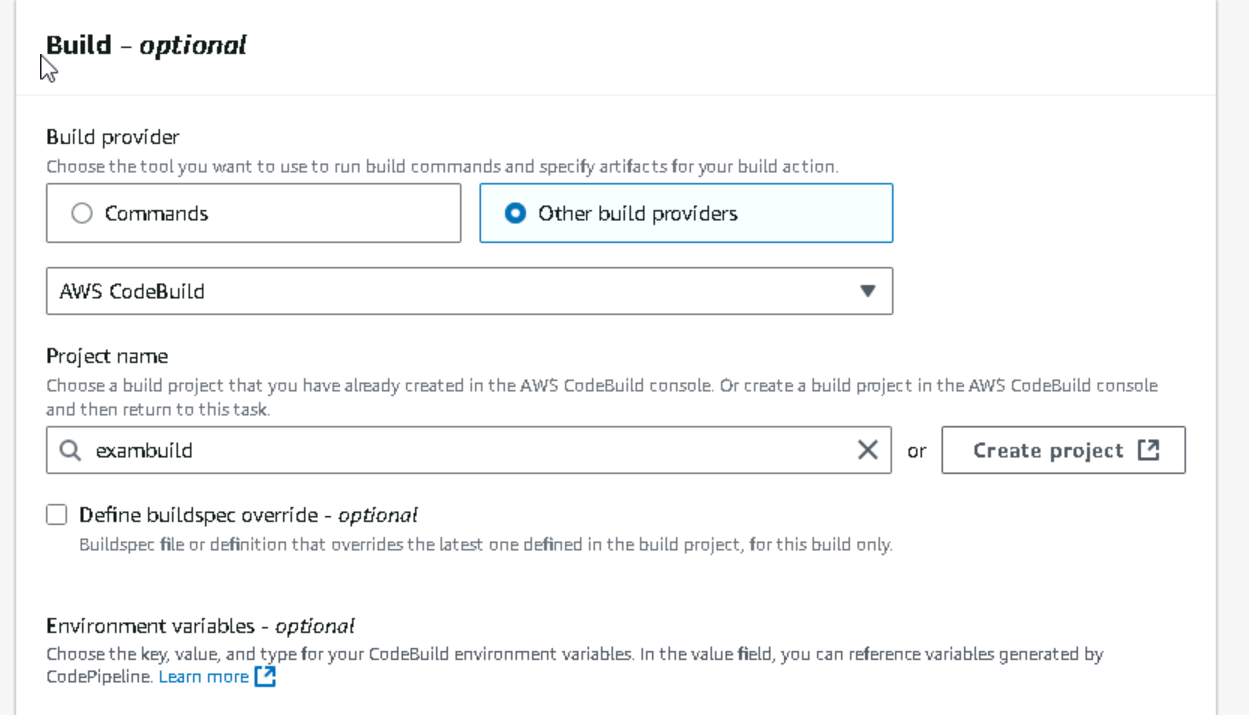
**STEPS:**

* **SOURCE PROVIDER : CODE COMMIT**
* **BUILD PROVIDER : CODE BUILD**
* **DEPLOY PROVIDER: EKS CLUSTER**

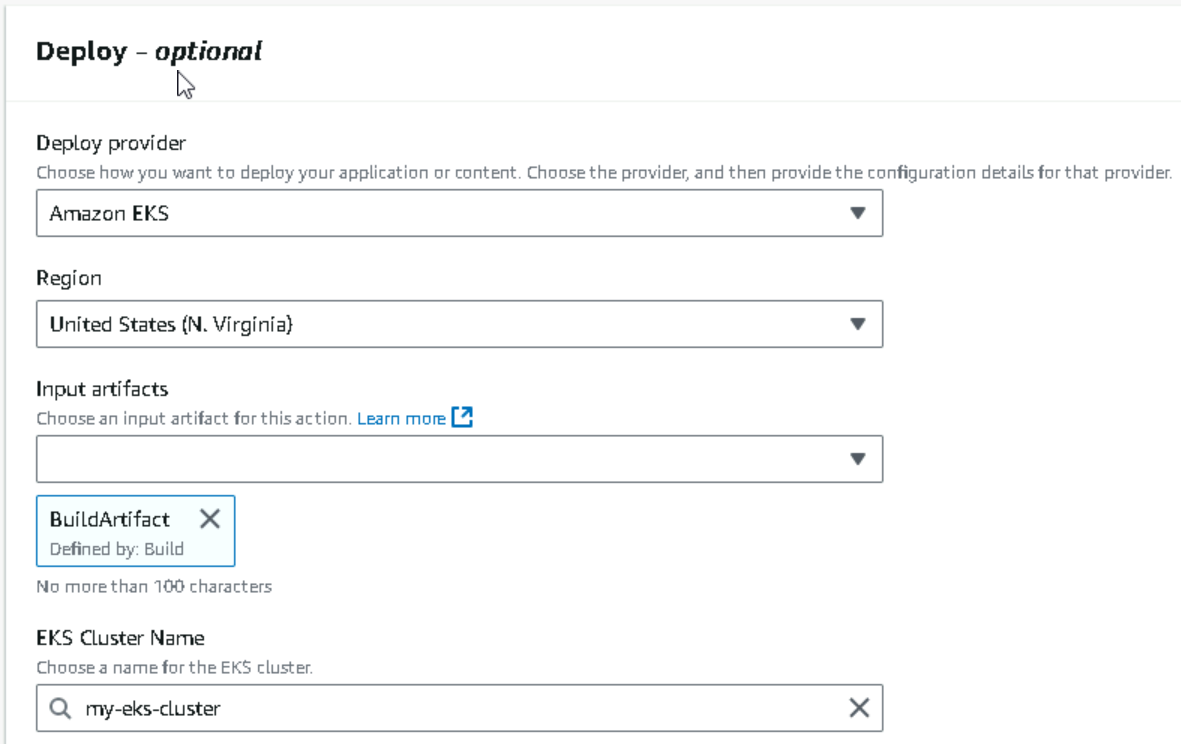
**SOURCE – CODE COMMIT**



**BUILD – CODE BUILD AND SELECT YOUR BUILD PROJECT CREATED**

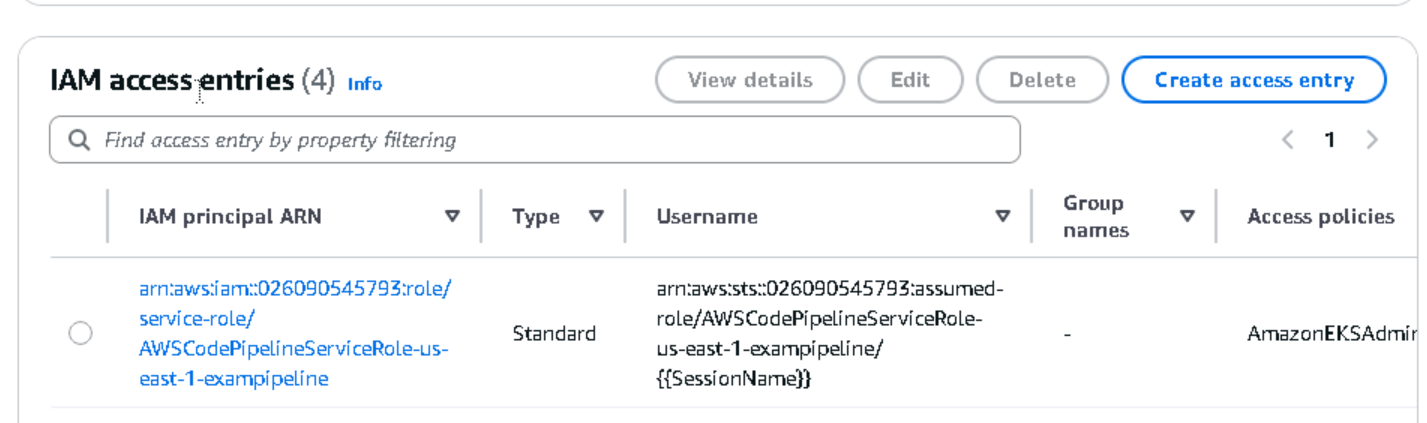


**DEPLOY – EKS CLUSTER AND SELECT YOUR EKS CLUSTER CREATED**



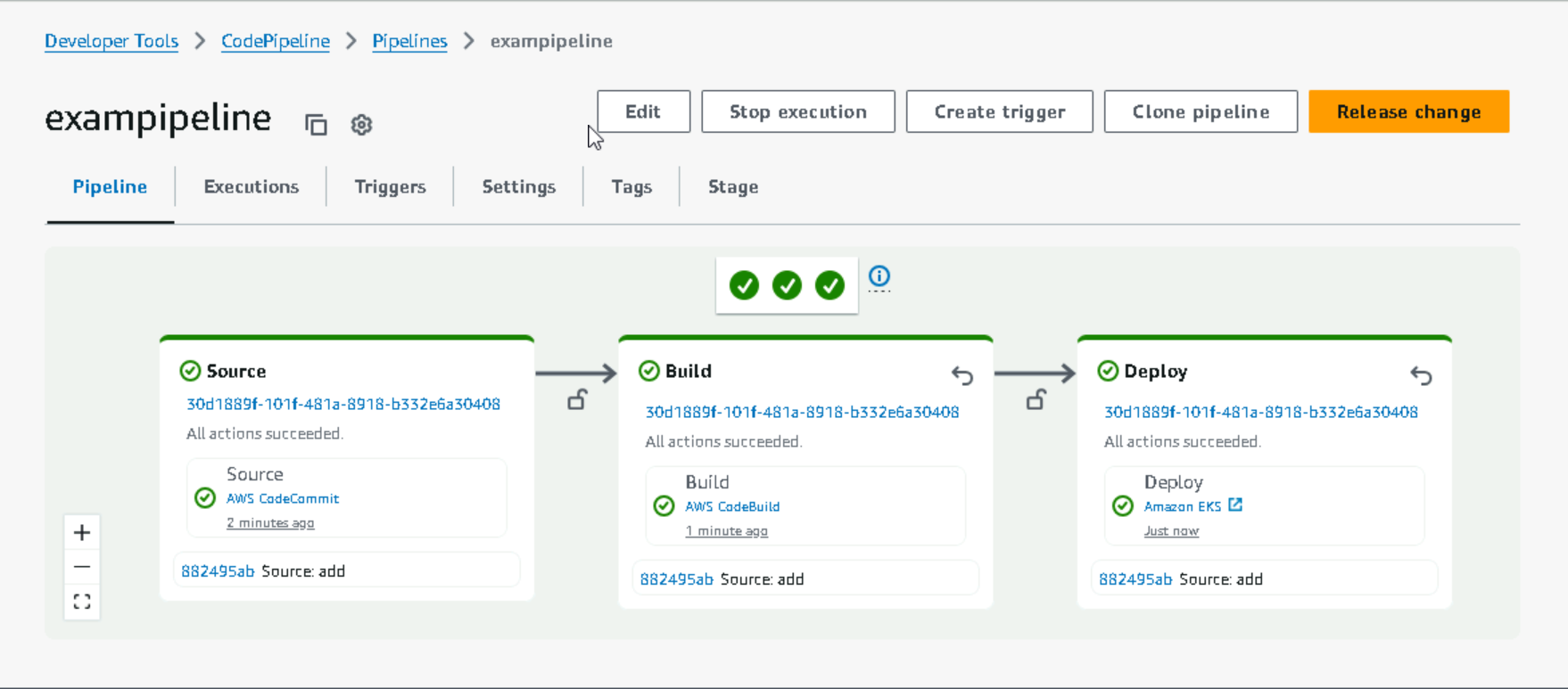
**PERMISSION FOR CODE PIPELINE TO ACCESS EKS CLUSTER**

* Give an access entry in eks cluster with eks admin policy permission

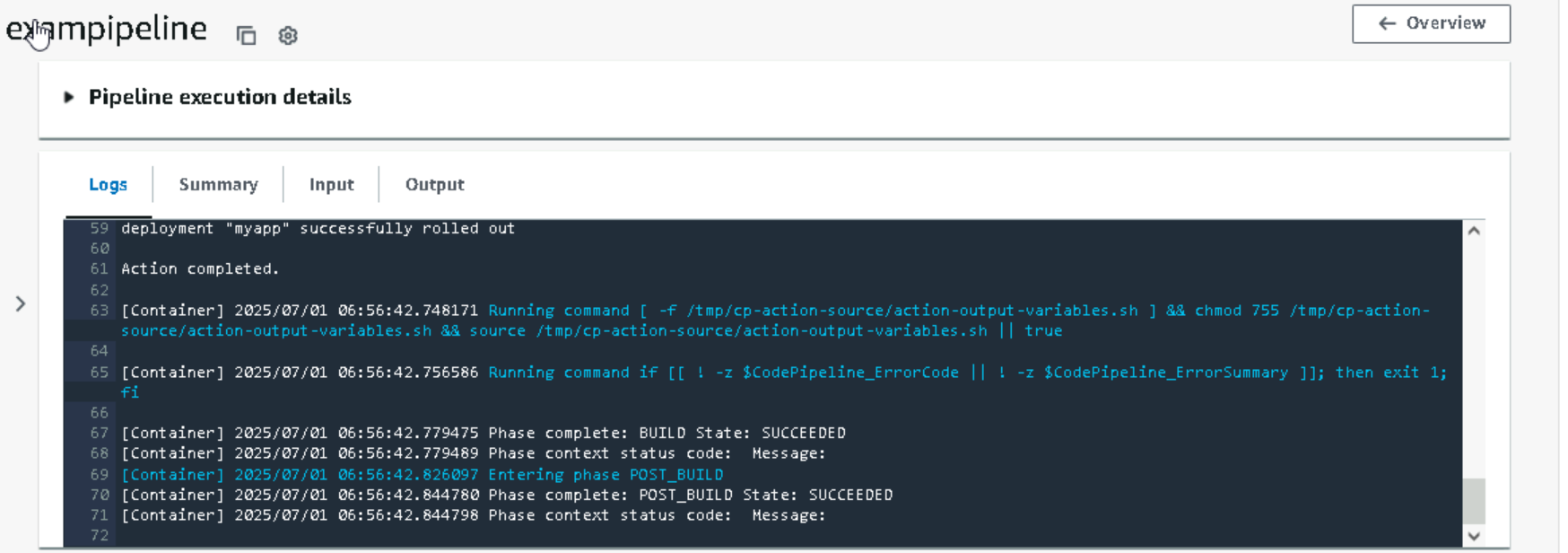


**TASK4: OUTPUT**

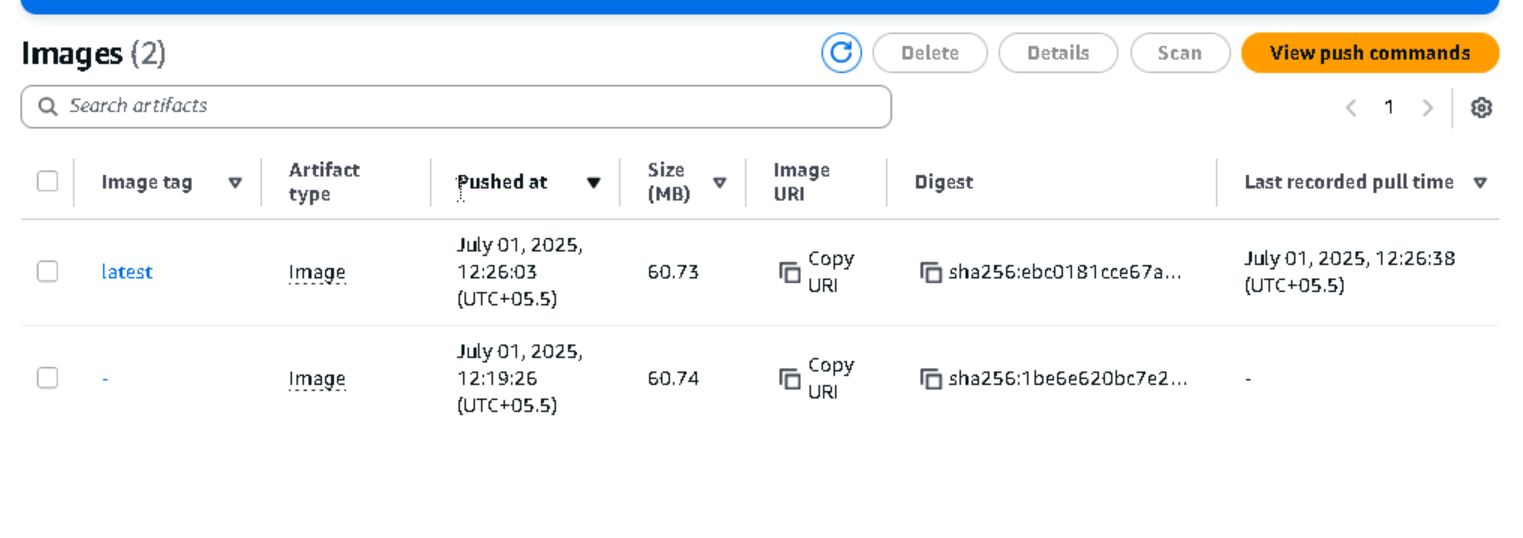
**CODE PIPELINE SUCCESS**



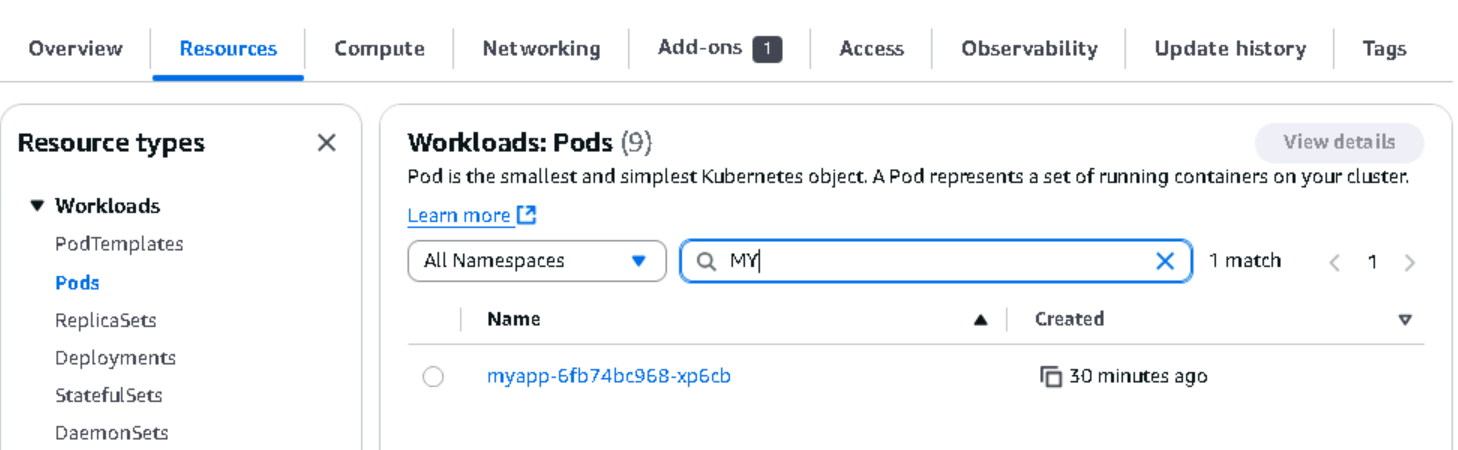
**LOGS OF SUCCESS PIPLEINE**



**IN ECR REPO SEE THE IMAGE PUSHED THROUGH CODE PIPELINE**

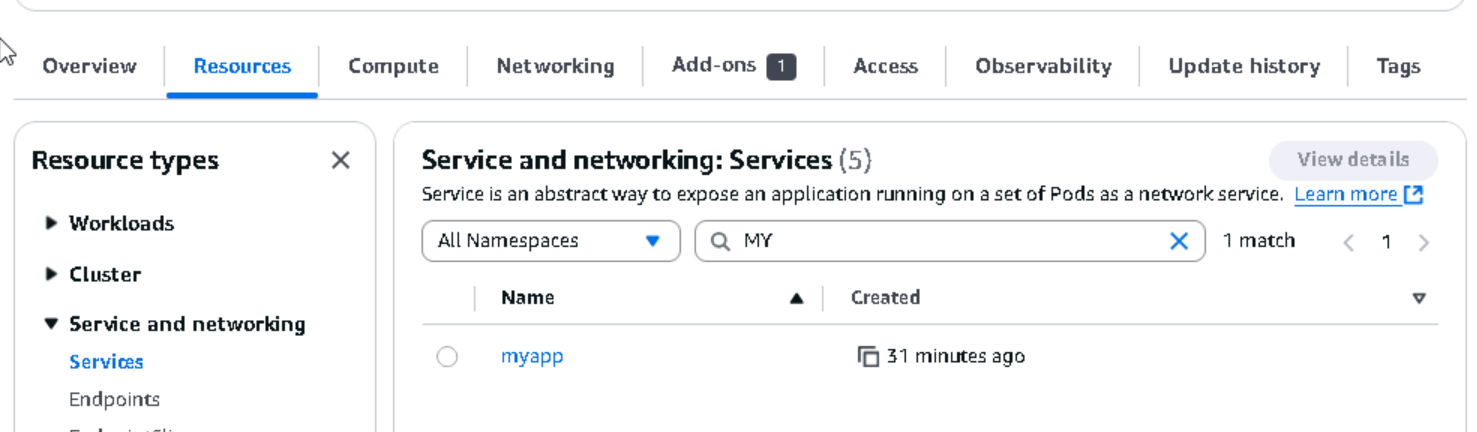


**IN EKS CLUSTER WE CAN SEE THE POD CREATED THROUGH PIPELINE**

****



**EKS CLUSTER WE CAN SEE THE SERVICE CREATED ALSO (myapp)**



**NOTE**

Steps:

* Create ecr repository
* Create eks cluster
* Give all permissions need for eks cluster
* Clone the git repo
* Create a code commit repository
* Push all the codes to code commit repository
* Created files need for build and deployment : deploy.yaml, service.yaml, buildspec.yaml (dockerfile is already given in repo)
* Create a code build for build stage
* Create a code pipeline for cicd
* Source : code commit
* Build: code build
* Deploy: eks cluster
* Verify all the output (pods , service created and image pushed)

In source I used code commit as source, instead of that I can use github also.i have no access to github so I used code commit. If I am using github, I need to connect the aws to my github through connector option seen in code pipeline first stage.