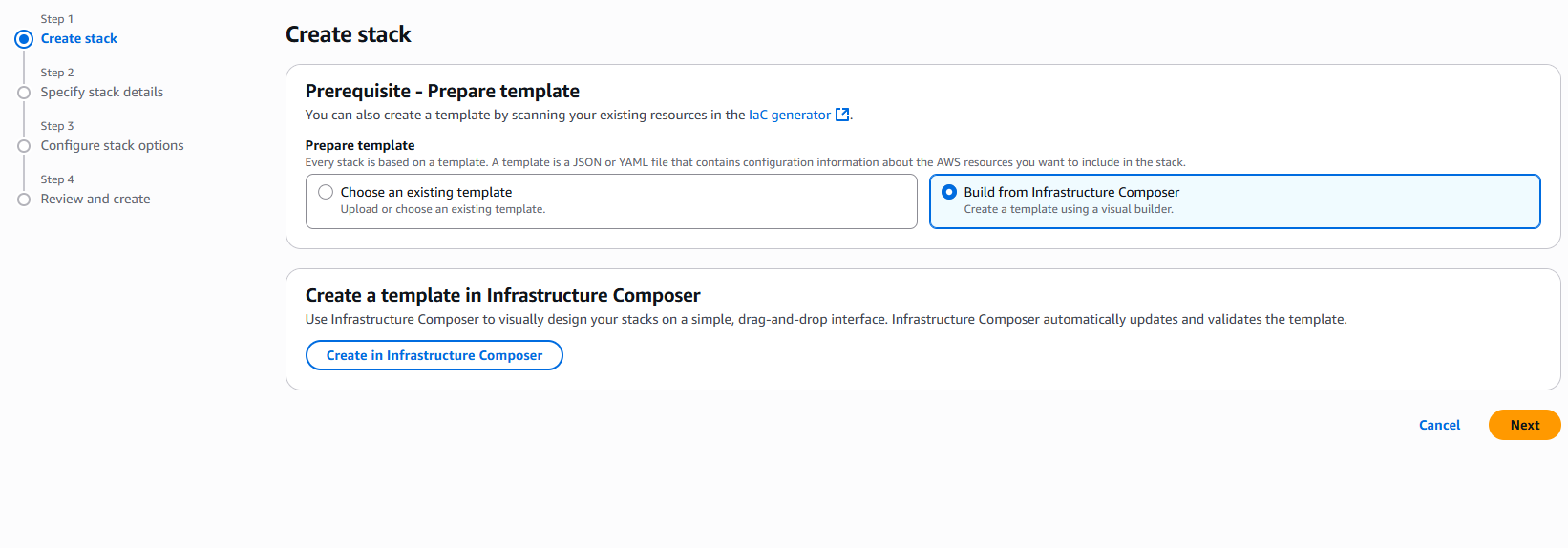
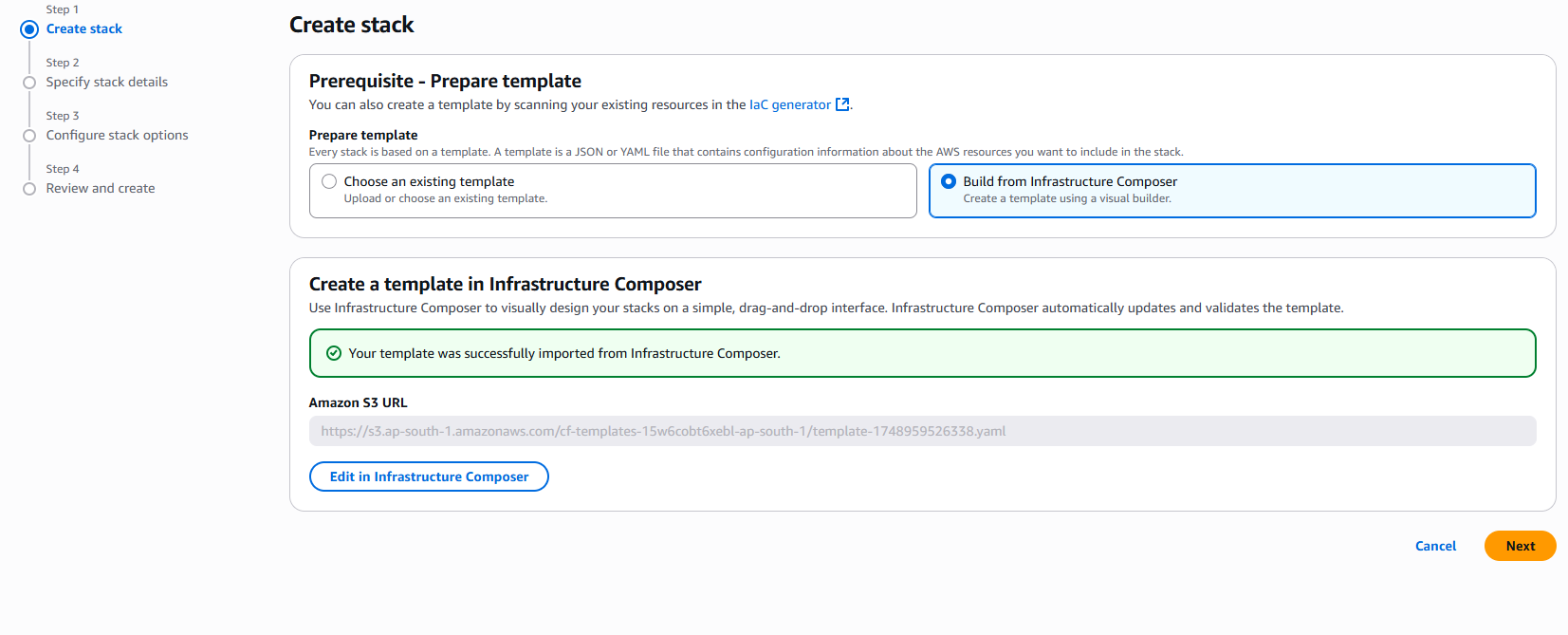
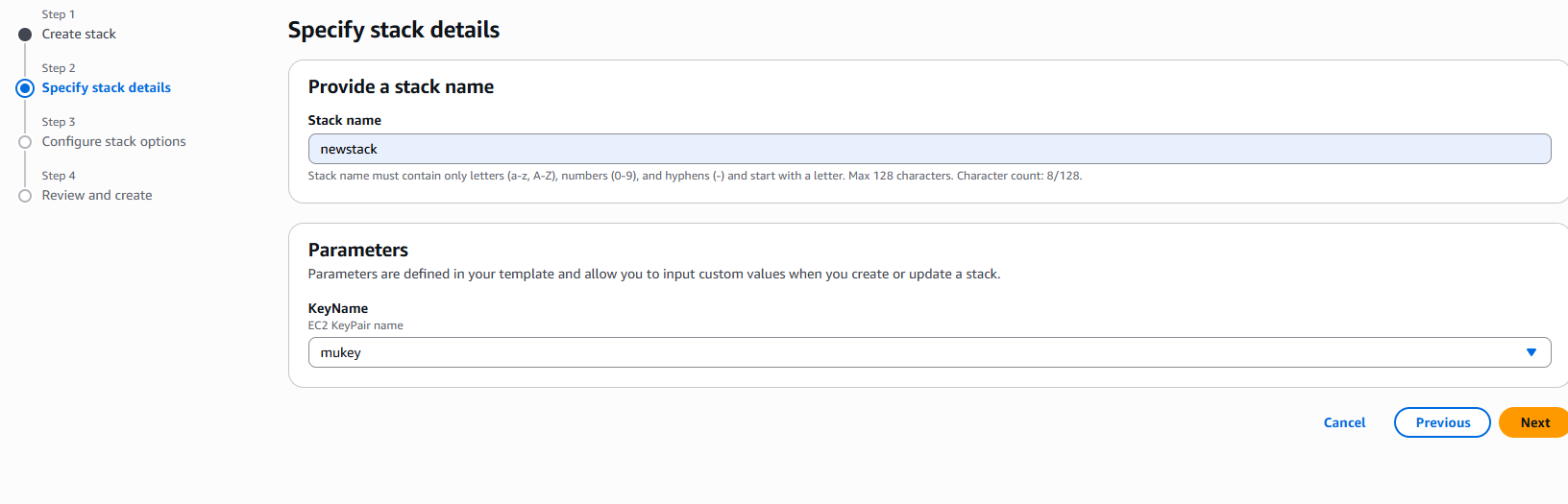
Internship Task

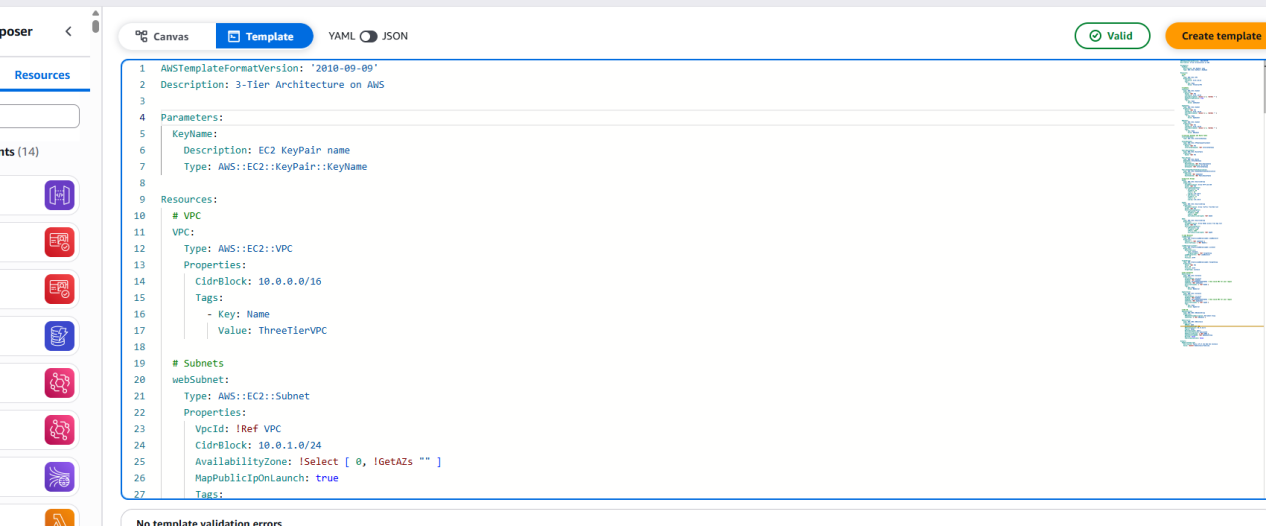
Que: 1. Deploy 3-tier architecture using cloud formation.

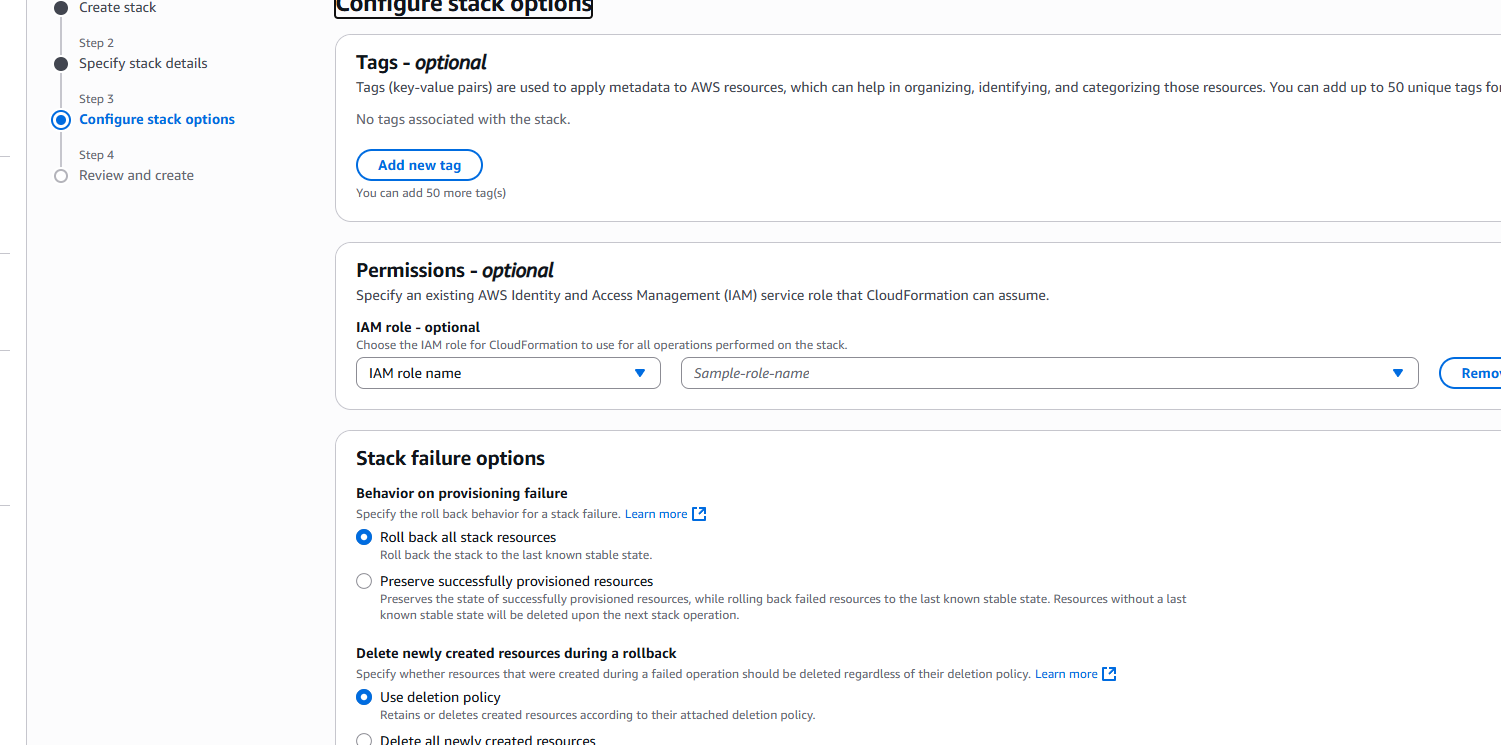
ANS:

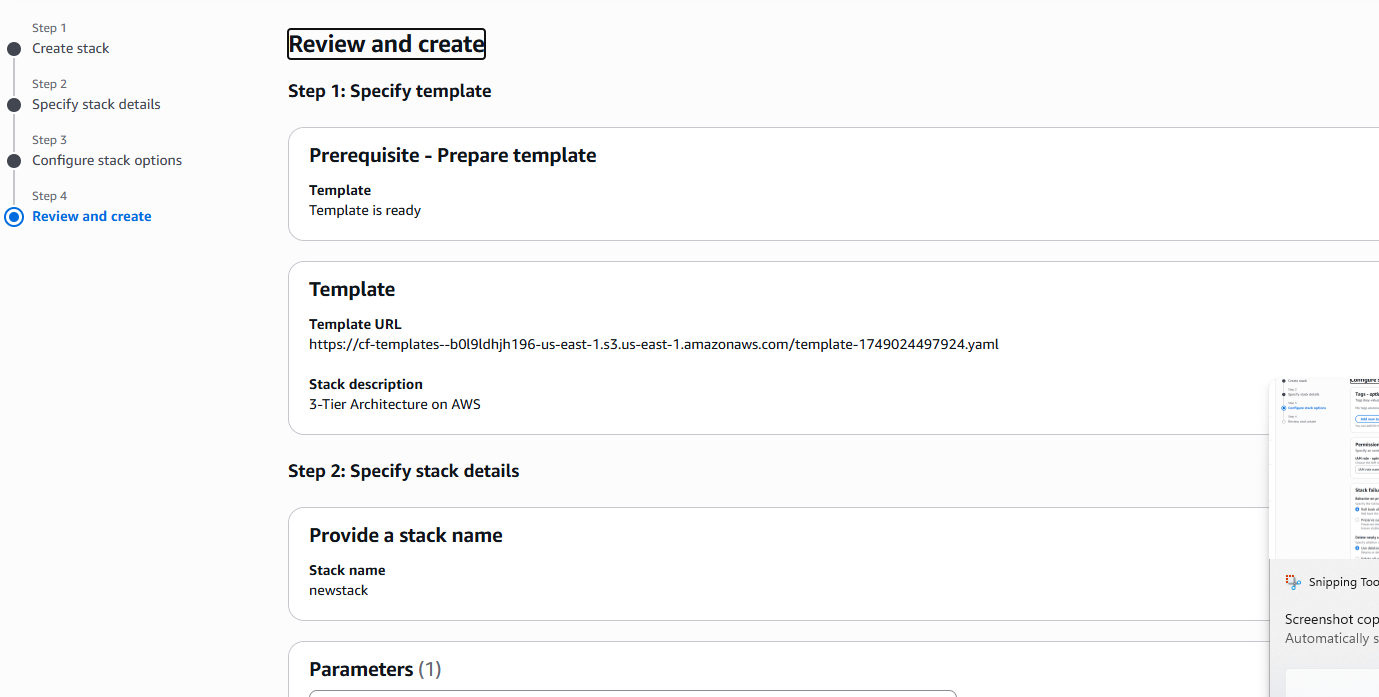
Create stack on aws cloud formation

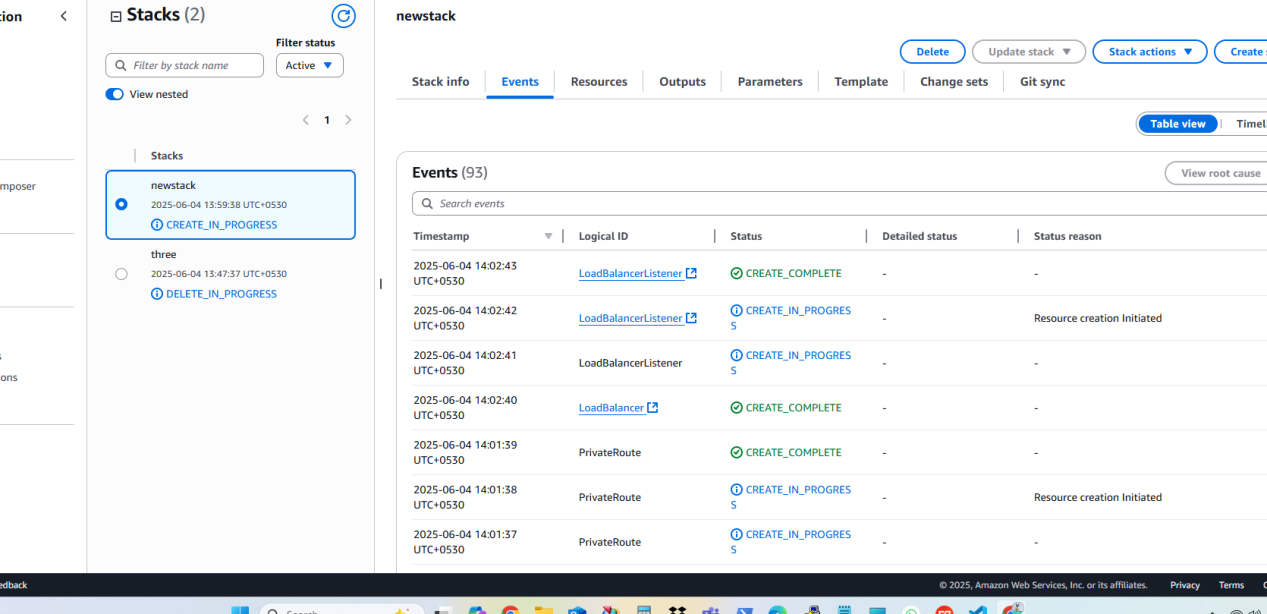


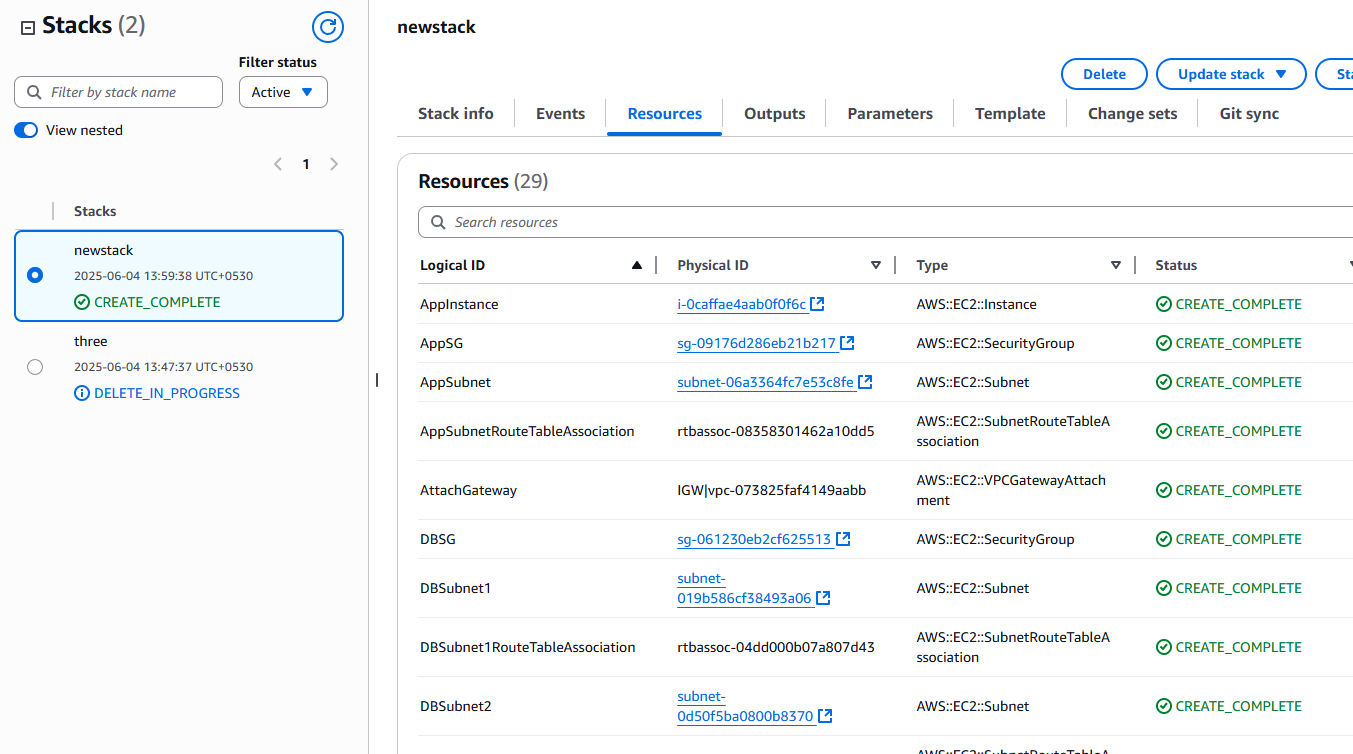


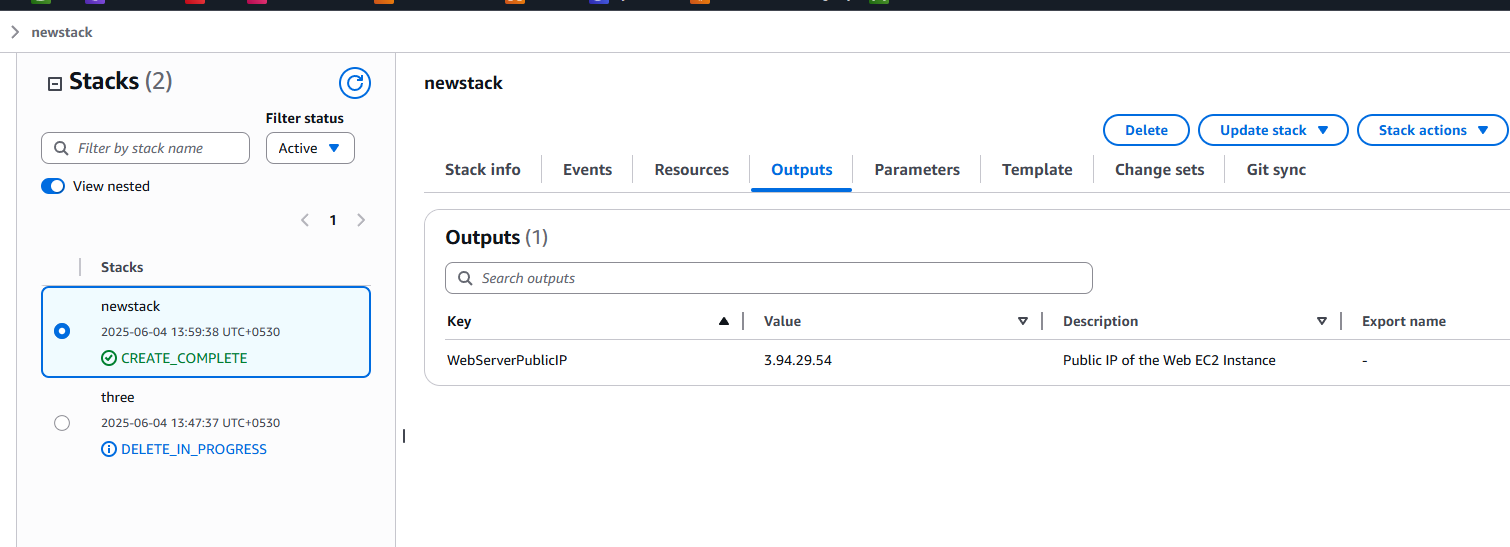


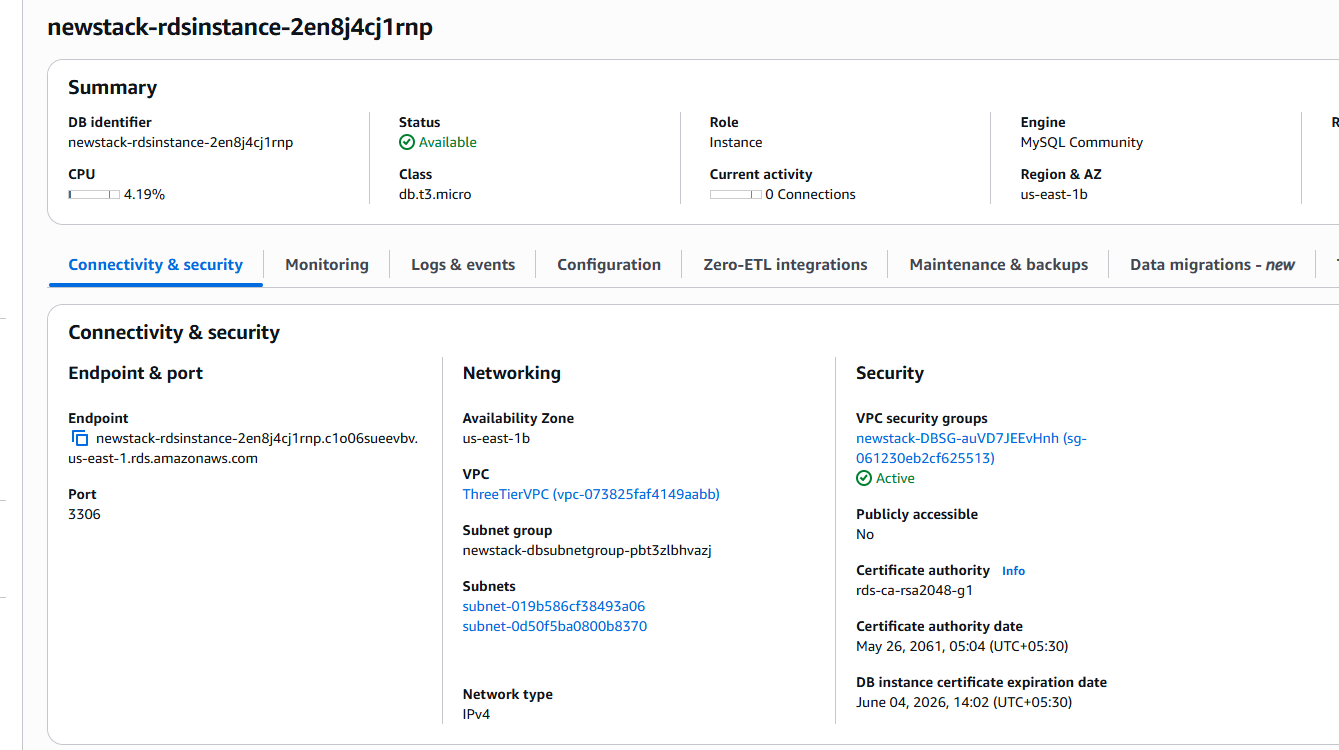


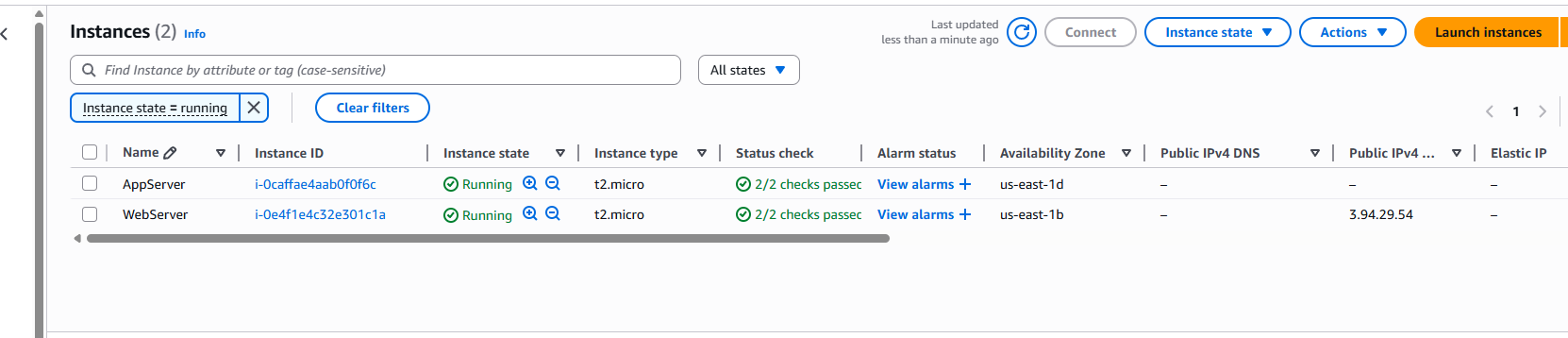








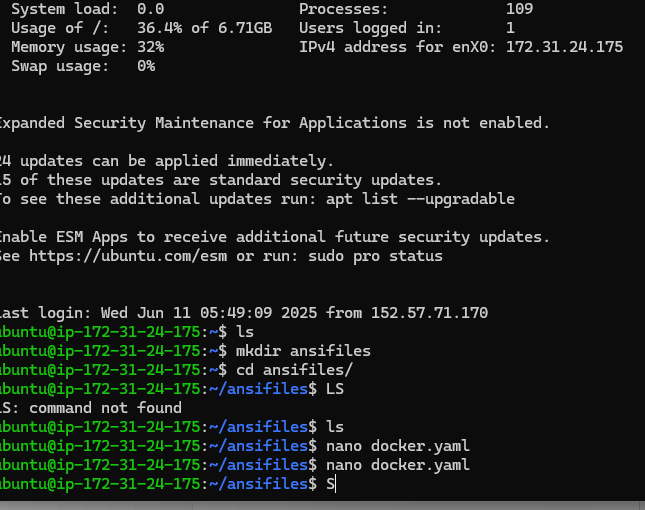


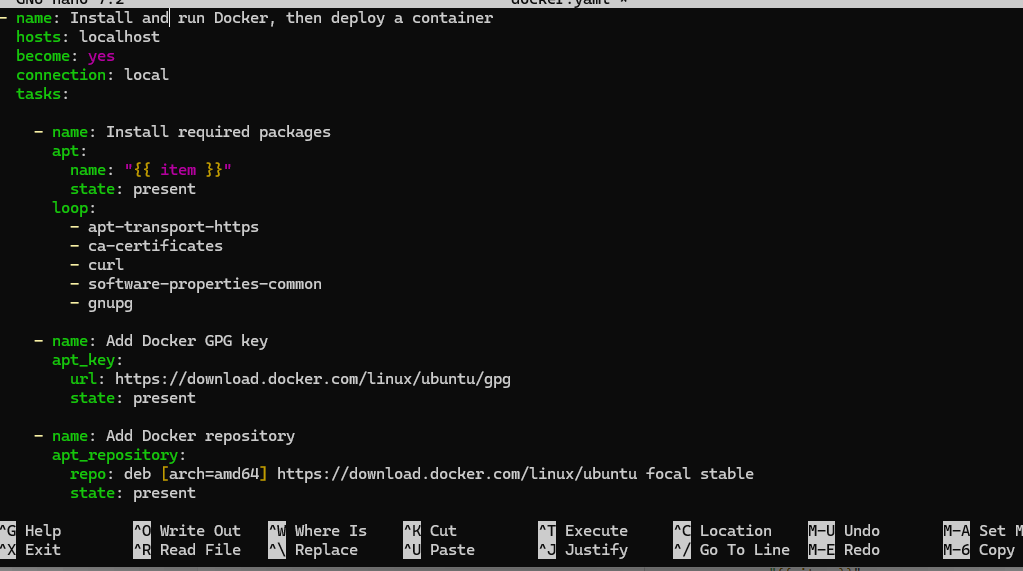


Q.2 automating Docker setup with ansible playbook

Ans: 1. launch Ubuntu instance

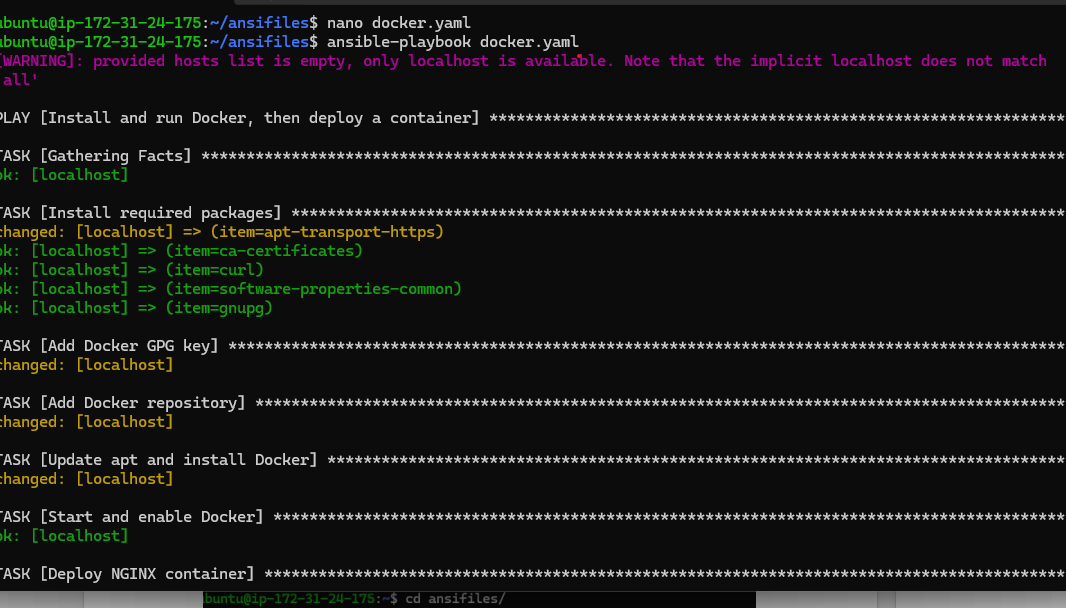
1. install Ansible on it and create directory
2. Create playbook file to install docker and create container

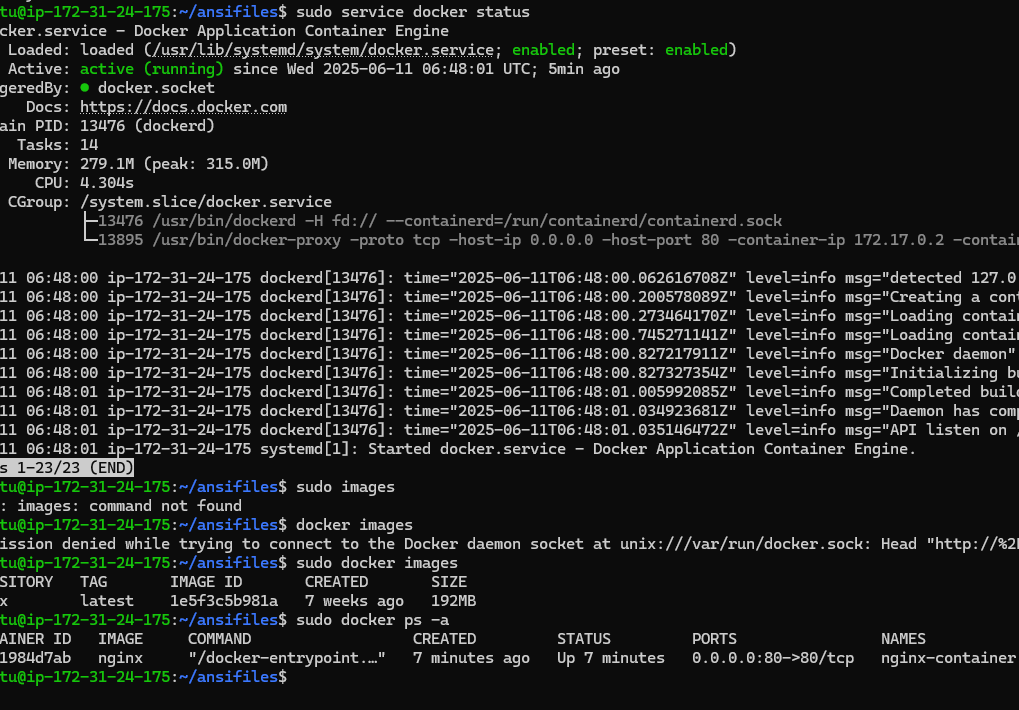




Run --- ansifile-playbook docker.yaml

{I install docker on local machine if we used remote server than we have to create inventory file for that in that add username, private I/p and key-pair of remote-server}





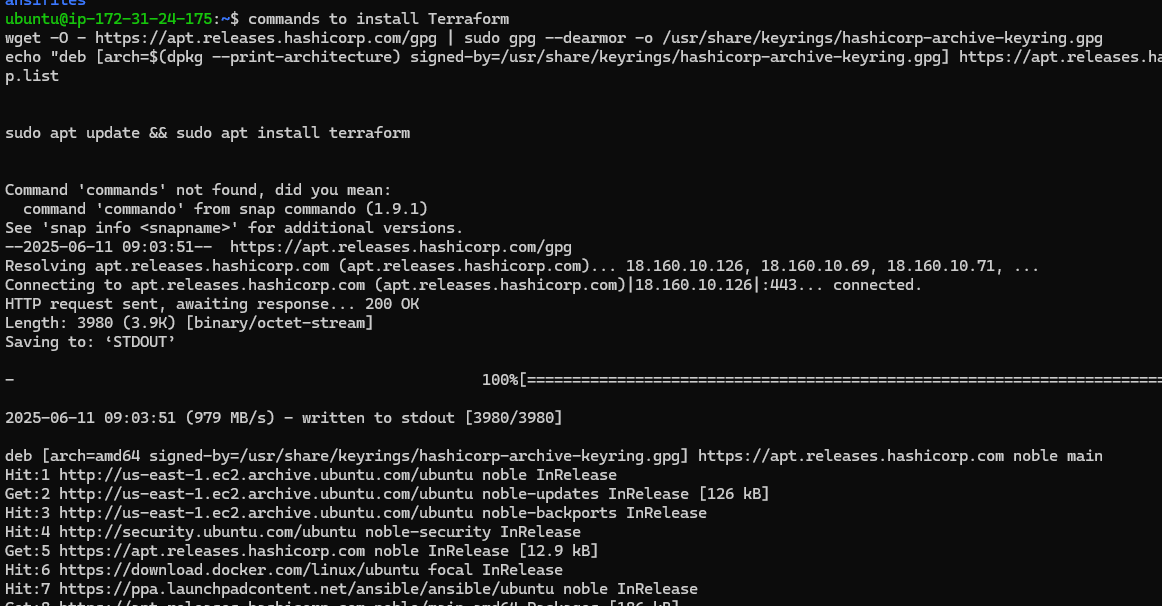
Q.3 setup the worker node and master node using Terraform and also cloud formation.

Ans.

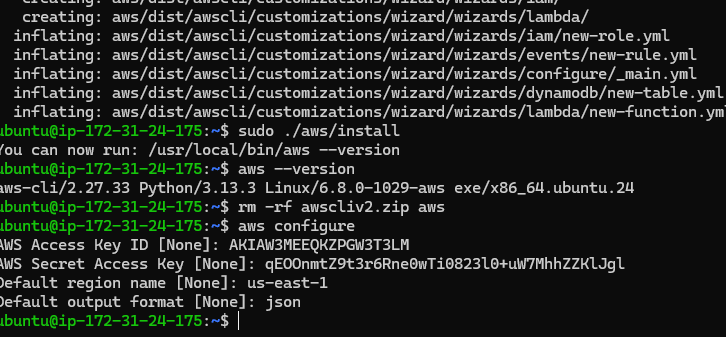
Setup worker node and master node using terraform

1. First we need to install terraform on local machine through that we can create kubernetes cluster (worker node and master node)

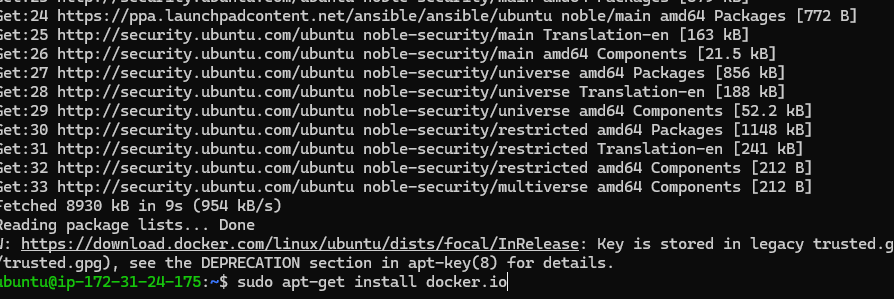
Terraform installed

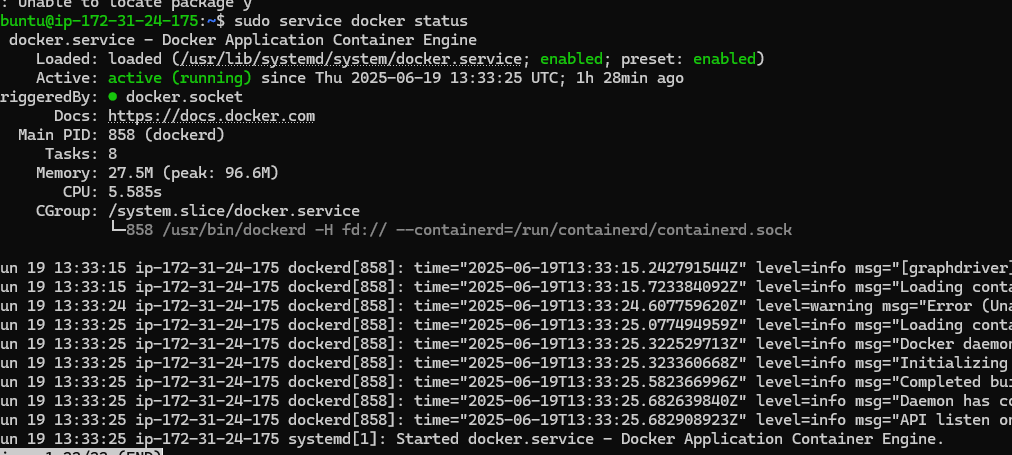


Install AWS-cli to configure AWS provider



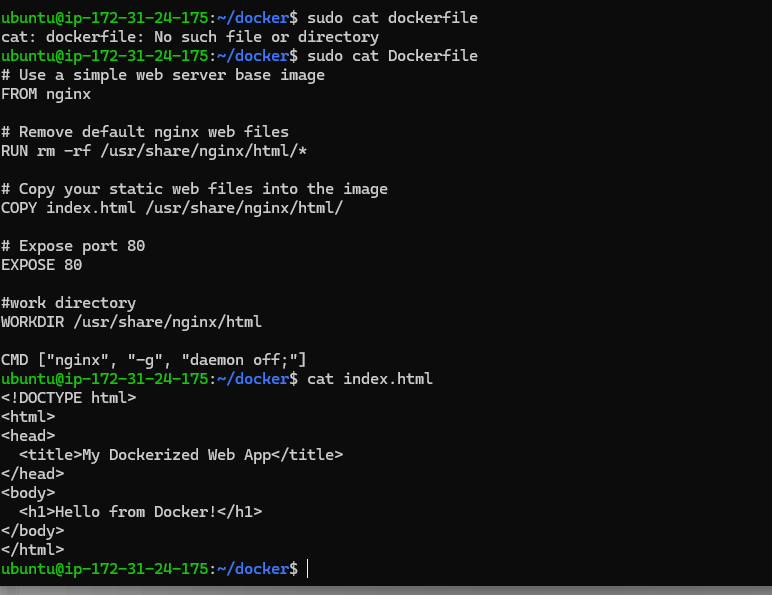
1. Q.4 Docker Installation and Setup
   * Install Docker on your local machine or a designated server.
   * Verify that Docker is correctly installed and running.

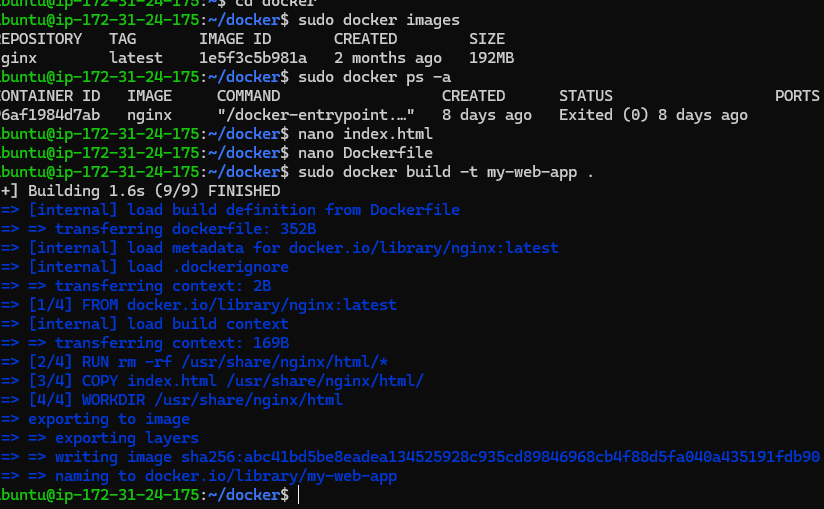


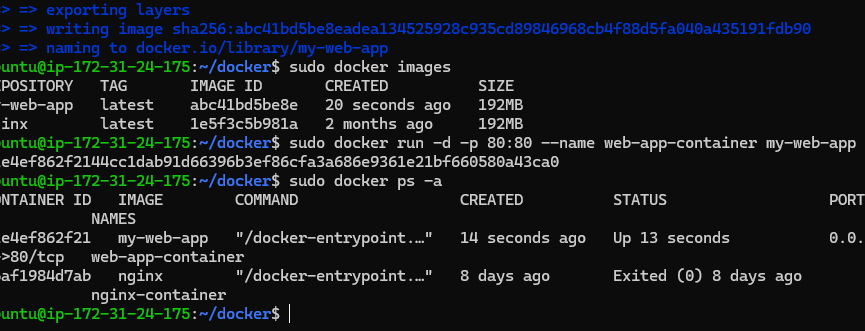


Containerizing the Web Application

* + Choose a simple web application (e.g., a basic HTML/JavaScript web page) or use a sample application.
  + Create a Dockerfile for the web application. Ensure that it specifies the base image, sets up the necessary environment, copies the application files, and defines the startup command.Build a Docker image from the Dockerfile.



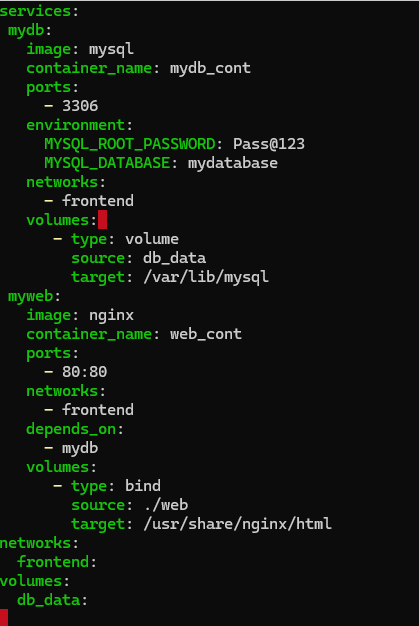


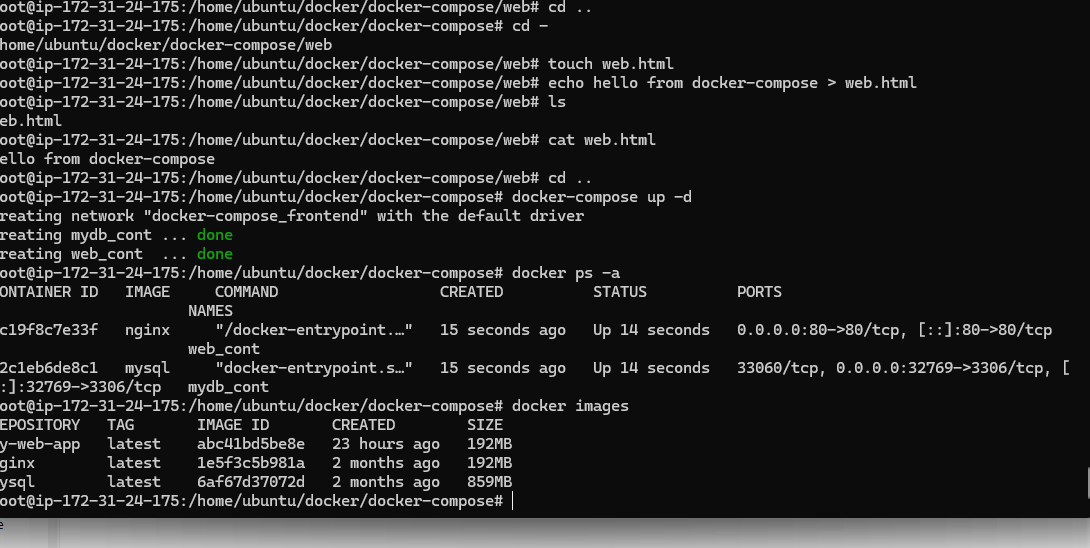


* Docker Compose
  + Create a docker-compose.yml file to define a multi-container application stack.
  + Include the container for your web application and a database container (e.g., MySQL or PostgreSQL) in the Docker Compose file.
  + Set environment variables, network configurations, and dependencies between the containers.
  + Ensure that the database container is configured to persist data outside of the container.

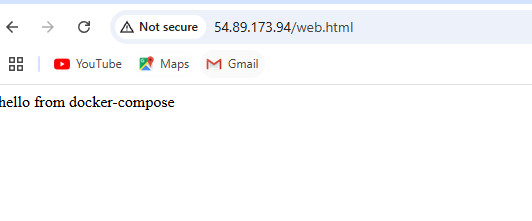
Mkdir docker-compose

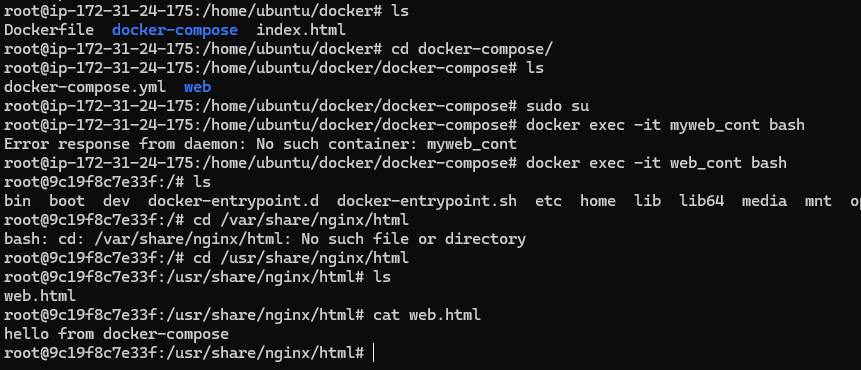
Nano docker-compose.yml



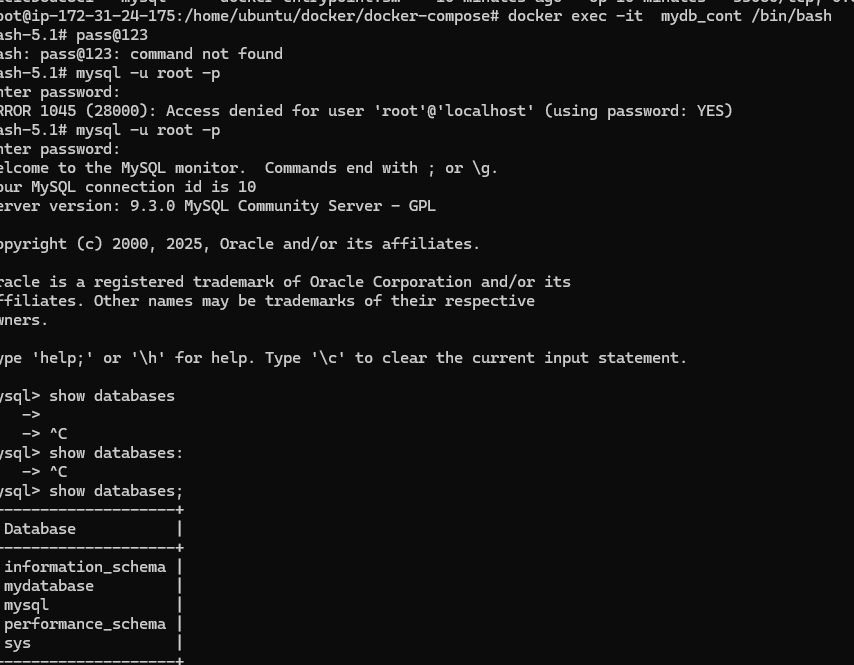


Add 3306 and 80 in your instance sg

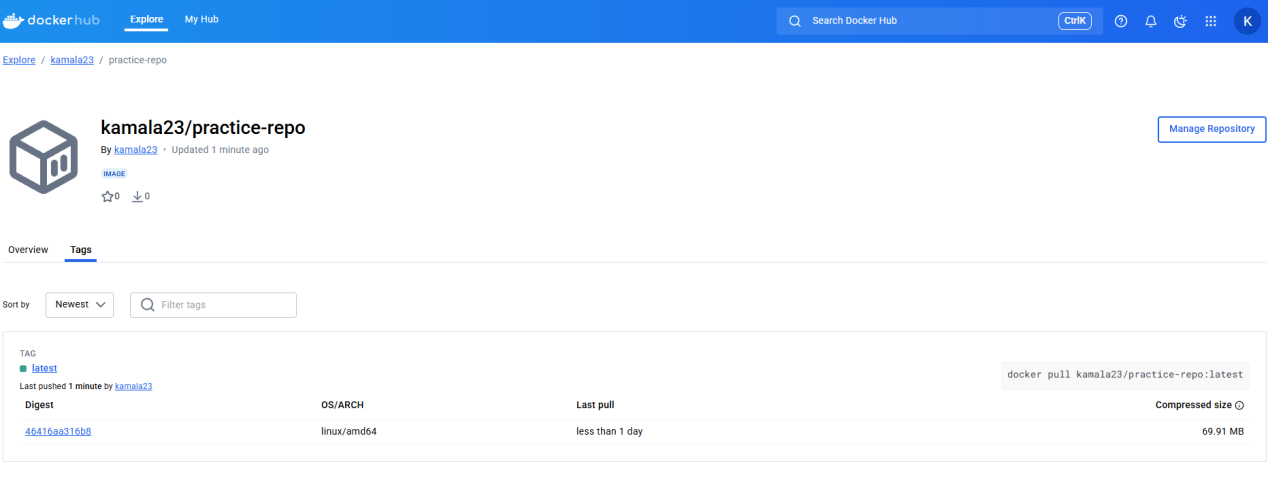
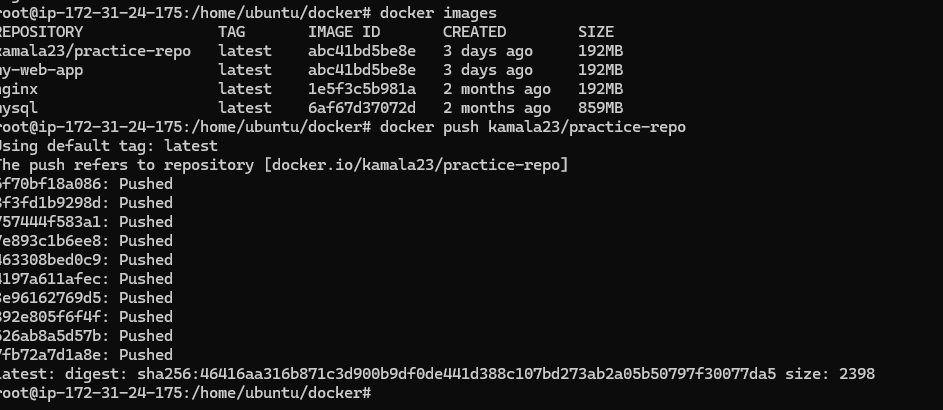
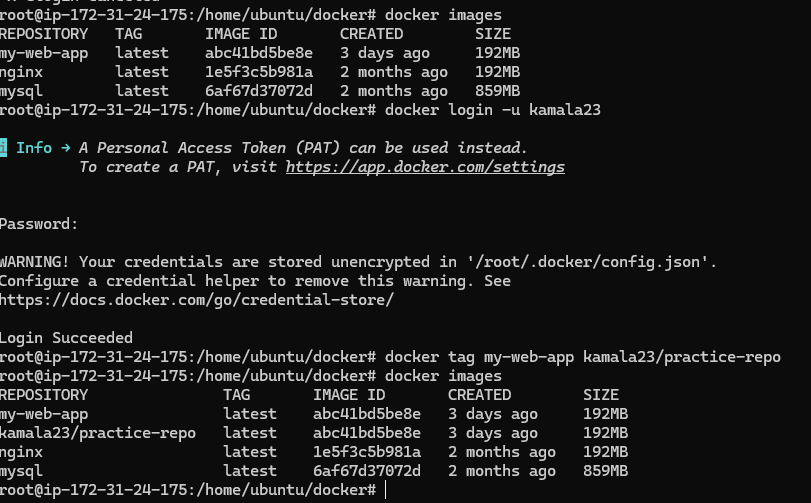




* Local Development
  + Use Docker Compose to launch the multi-container application stack on your local machine.
  + Verify that the web application is accessible in your web browser.
  + Test the functionality and data persistence of the application.



* Docker Hub or Container Registry
  + Push your web application Docker image to Docker Hub or a container registry of your choice.
  + Provide a link to the image repository in your documentation.



My dockerhub repo link is ----

<https://hub.docker.com/repository/docker/kamala23/practice-repo/>

Q.5 Configure VPC flow log and log should be saved in S3, Use IAM role for the same.

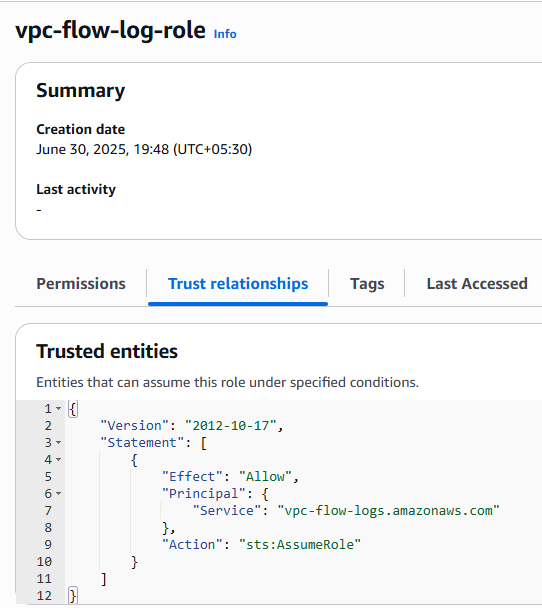
ANS. Create s3 bucakte for vpc flow logs

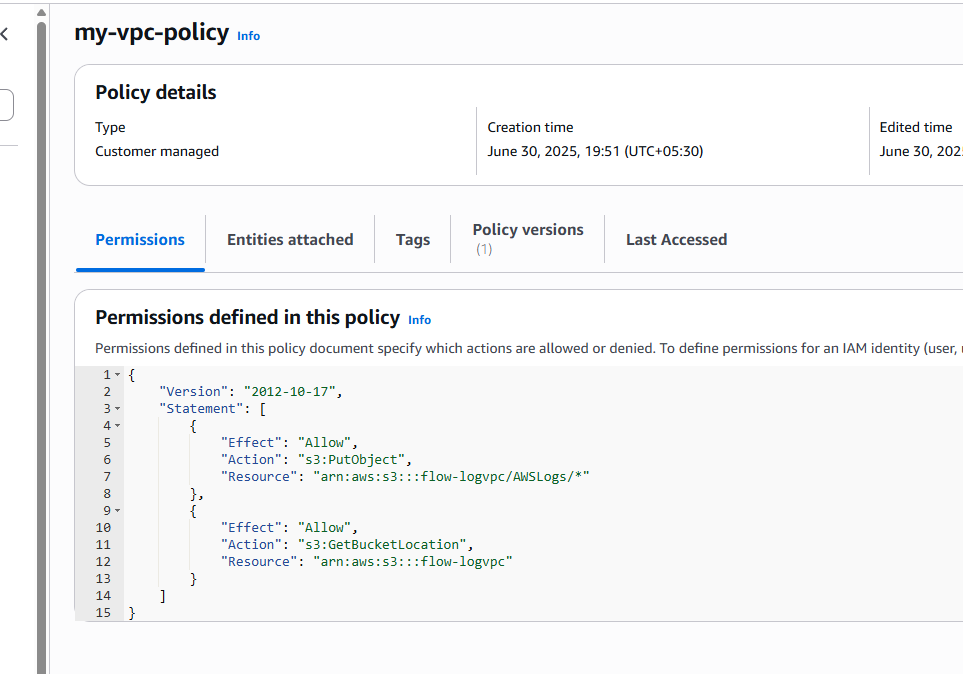
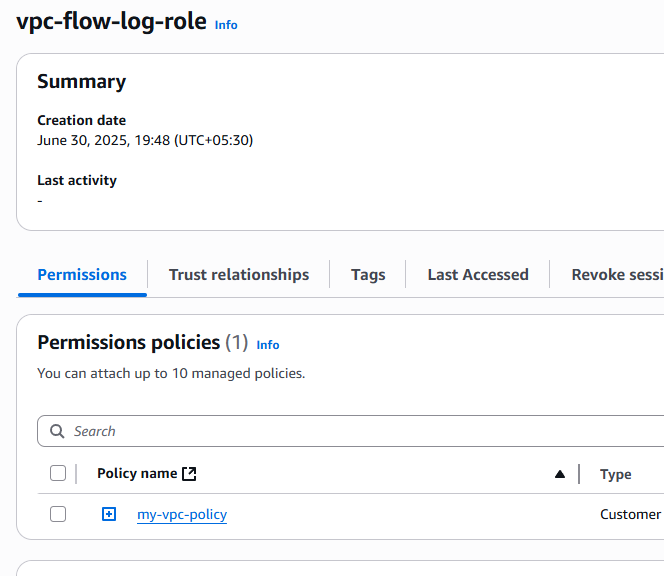
· Go to S3 → Create bucket

· Name it: flow-logvpc

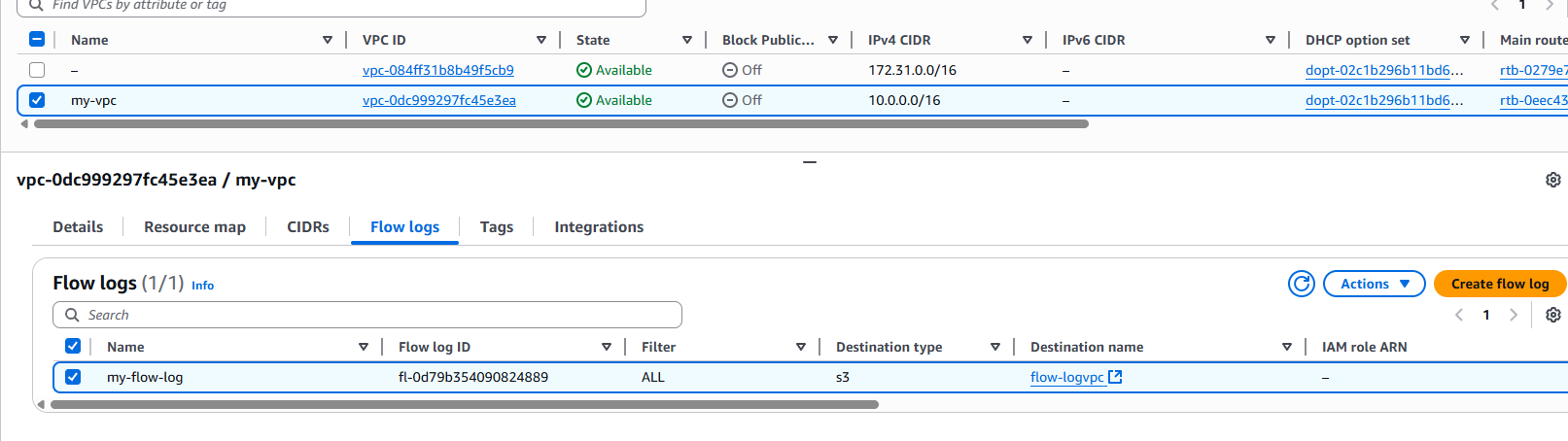
· Enable "Block all public access" and default settings.

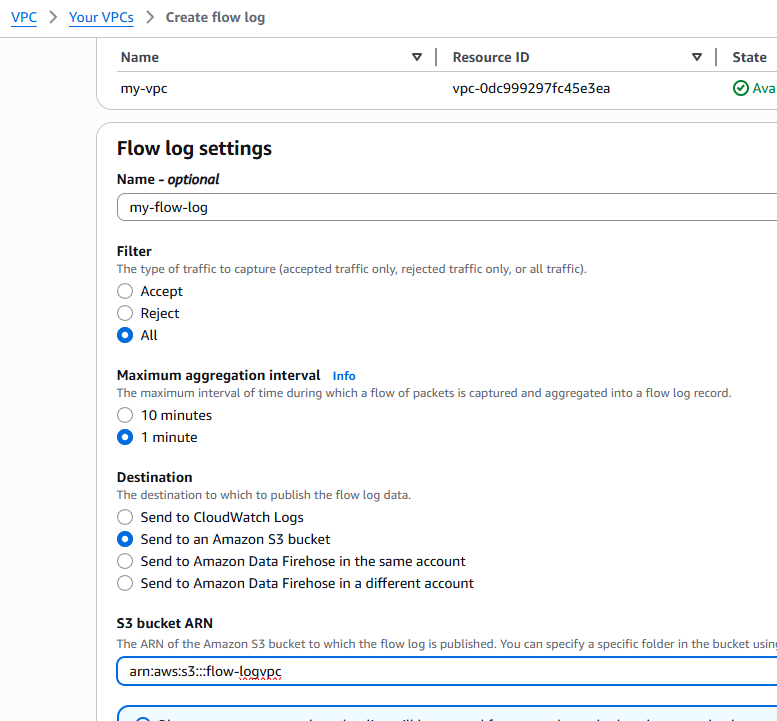
Create IAM role with related permission and policies

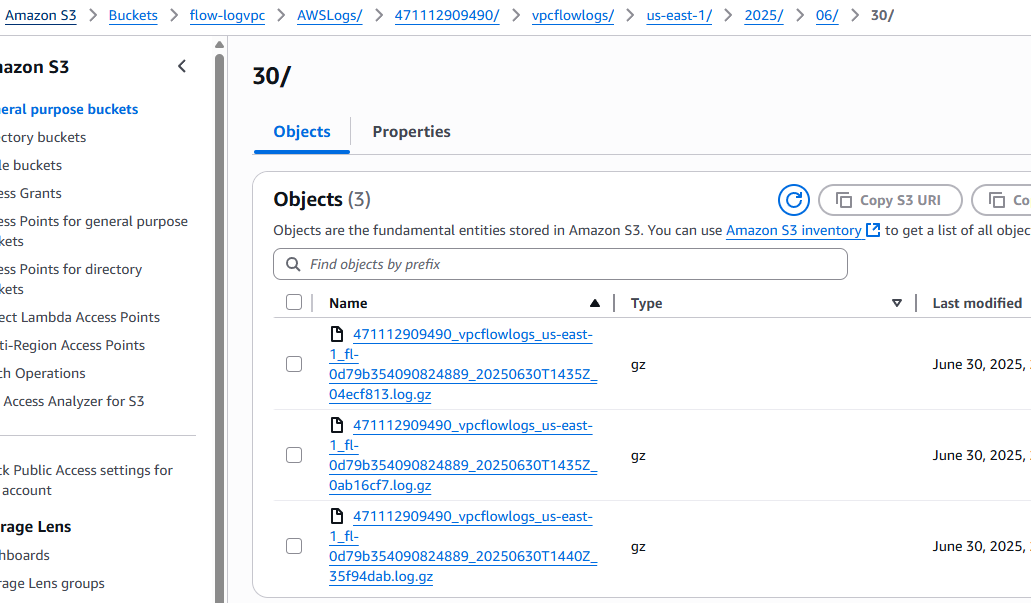




Create vpc-flow log and attached to s3 buckate







Now vpc flow log is successfully configured. To see the log files you can download them using download option available there.

Q.6 Prepare cloud watch dashboard for detailed monitoring for below point

- Billing and cost dashboard

- Logs dashboard

- Network performance dashboard

- Security and compliance dashboard

ANS.6