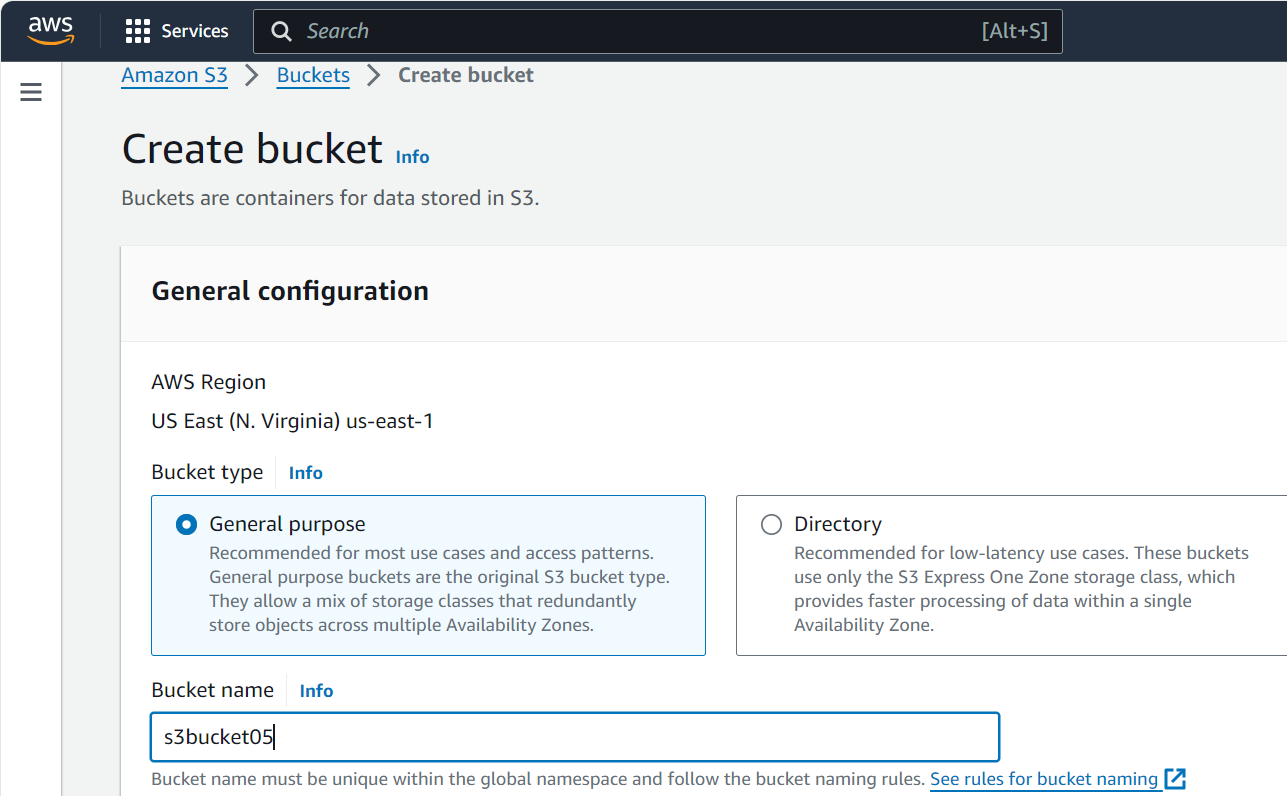
**Easy**

**Deploy a static website on AWS S3 bucket using GitHub action. when developer is making some changes in website code. New changes should be reflected on my website**

**Step 1: Create a bucket**

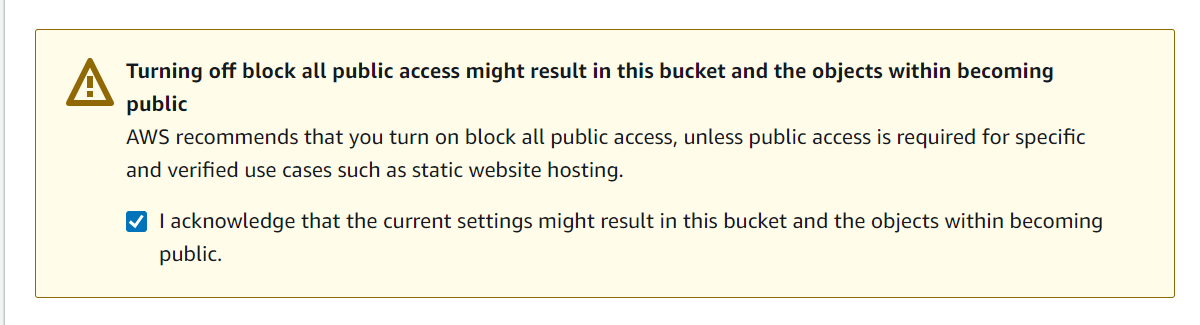
****

**Uncheck the check box for block all public access**

**A screenshot of a computer

Description automatically generated**

**Click on the acknowledge checkbox**

****

**Step 2: Enable static website hosting**

1. **In the Buckets list, choose the name of the bucket that you want to enable static website hosting for.**
2. Choose **Properties**.
3. **Under Static website hosting, choose Edit.**

**A screenshot of a computer

Description automatically generated**

1. **Under Buckets, choose the name of your bucket.**
2. **Choose Permissions.**
3. **Under Bucket Policy, choose Edit.**

**A screenshot of a computer

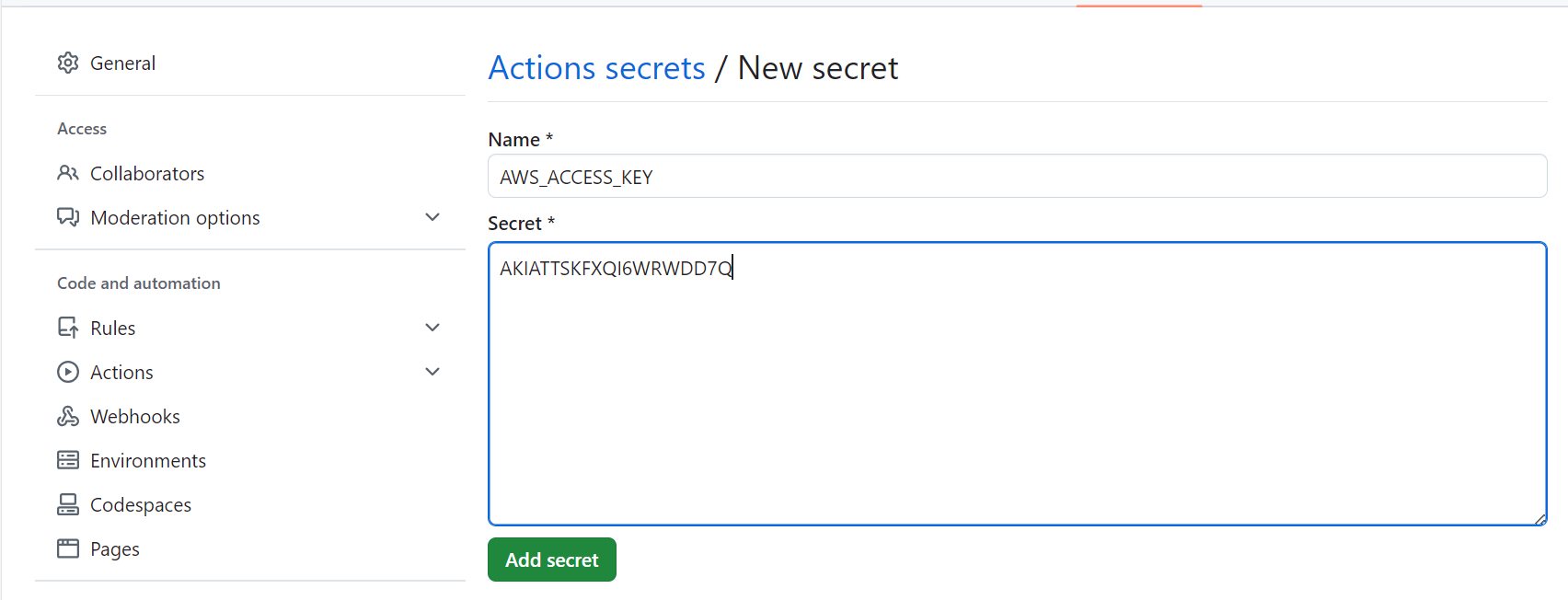
Description automatically generated**

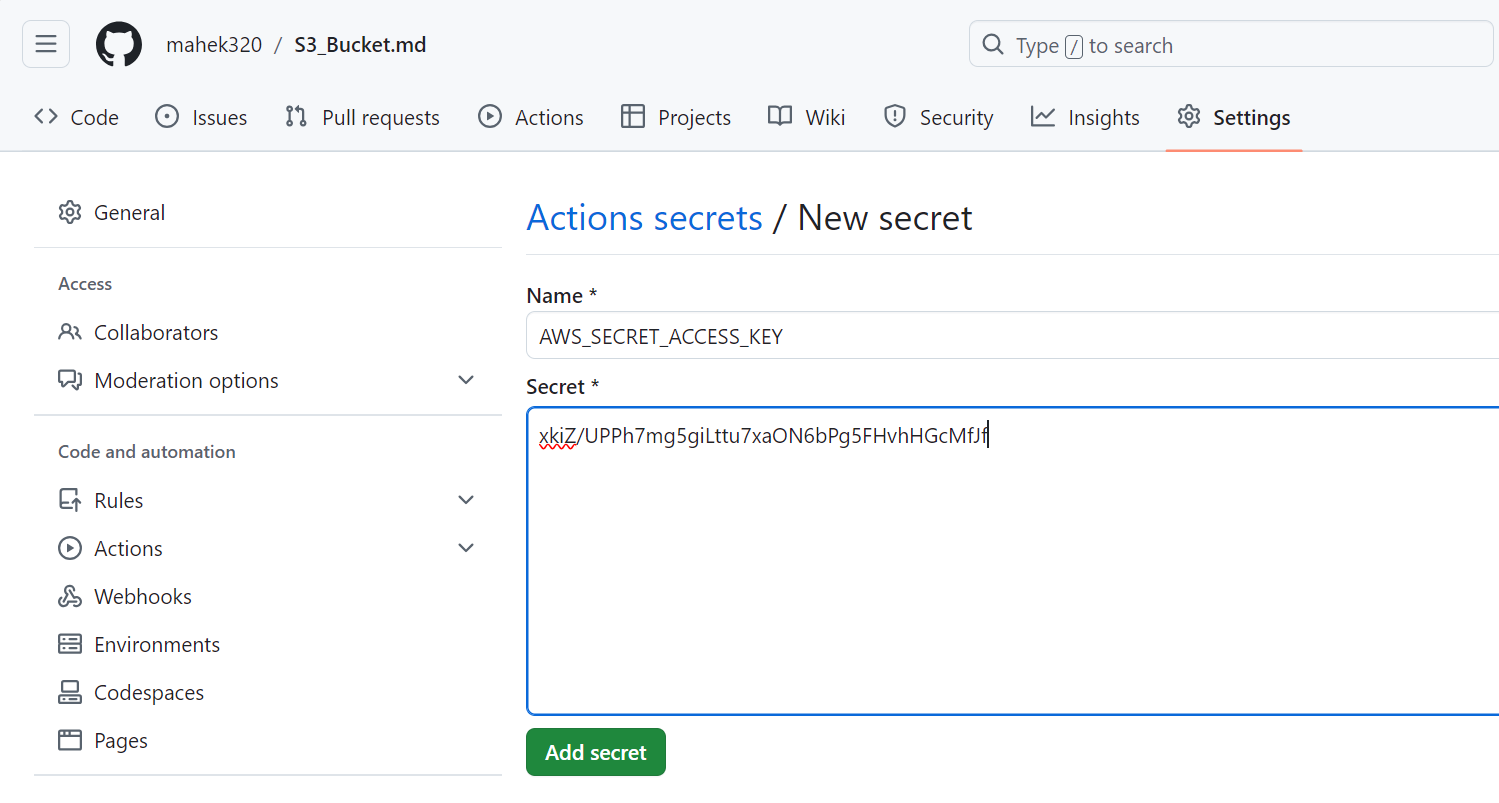
**STEP 3: Create a new repository in your GitHub account**

**A screenshot of a computer

Description automatically generated**

**Go to settings >> secrets and variables >> actions >> add new repository secret  
  
Here add access key and secret key**

****

****

**A screenshot of a computer

Description automatically generated**

**Change the bucket name and region and remove public**

**A computer screen shot of a computer code

Description automatically generated**

**Create an index file in the same repository**

**A screenshot of a computer

Description automatically generated**

**Click on actions in the git repository**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Under Static website hosting, click on the Endpoint URL.**

**Copy paste the endpoint url**

**A screenshot of a computer

Description automatically generated**

Easy 1

Pull the Ubuntu image from Docker hub and launch a web application in the container on port no. 8080 and this application should be reachable globally.

Launch an EC2 instance for docker and connect it on the terminal

A screenshot of a computer

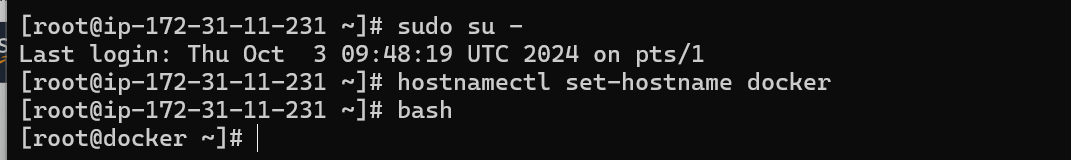
Description automatically generated

Add port 8080 in the security group of the instance

A screenshot of a computer

Description automatically generated

Set the hostname



Run the docker commands on the terminal to install docker



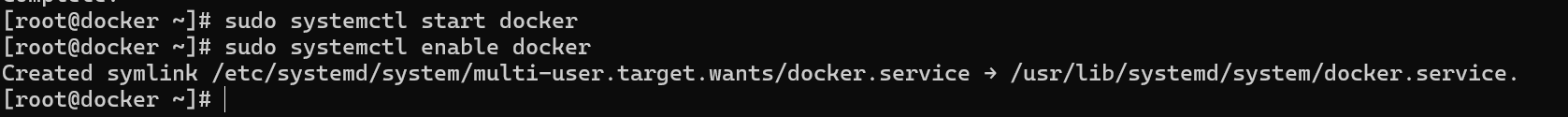
A screenshot of a computer

Description automatically generated

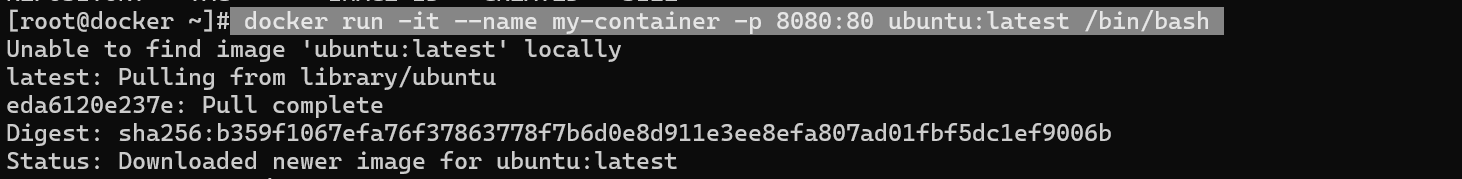
A black screen with white text

Description automatically generated

We will now use the start docker command to start docker



We will pull the ubuntu image



We will install all the packages and update them

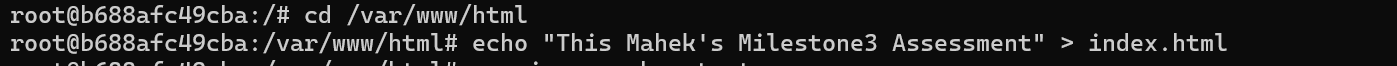
A screen shot of a computer

Description automatically generated

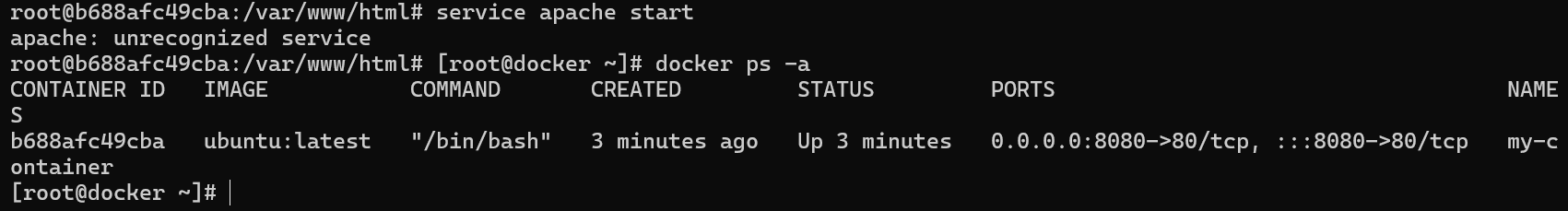
A computer screen with white text

Description automatically generated

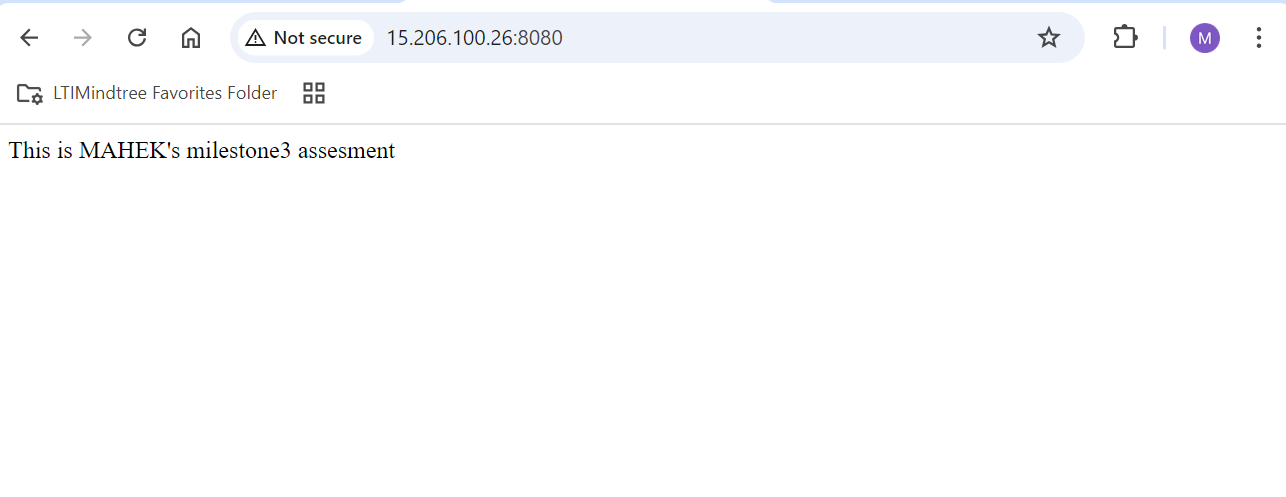
We will create a html file



We will start the start the apache



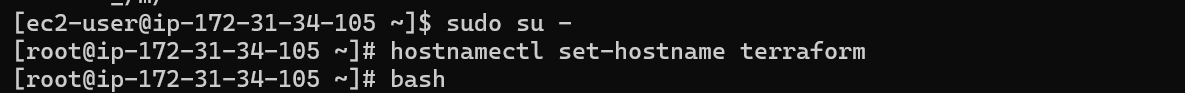
The image has been successfully hosted on 8080



Medium

Create a custom VPC and create two subnets like public and private in that subnet you need to deploy one web server and another is database server using IAC tool terraform

Created an ec2 instance and connected to terminal



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

We have configured aws and install aws cli

A screenshot of a computer program

Description automatically generated

We created a vim vpc.tf file

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

We have then validated the terraform

A black background with white text

Description automatically generated

We have now used the terraform plan command

A screenshot of a computer

Description automatically generated

We have applied all the changes

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

As we can see that all the changes have been successfully implemented

A screenshot of a computer program

Description automatically generated

Medium 1

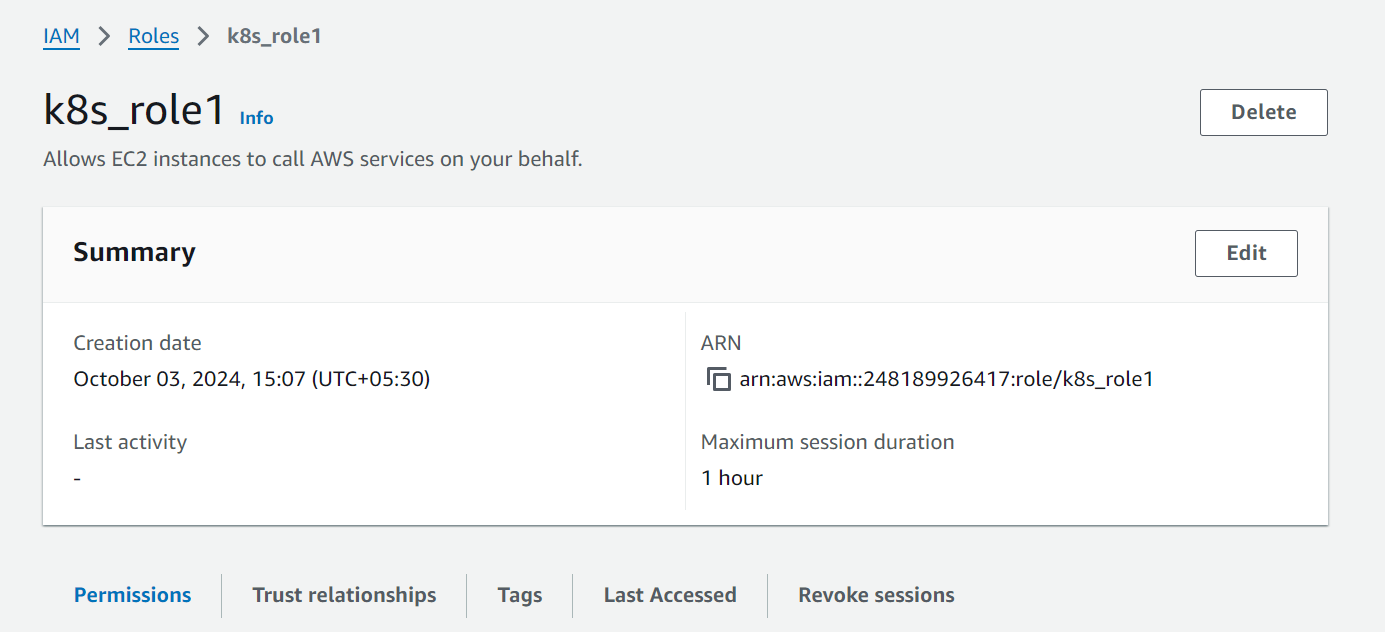
Deploy a web application in the Kubernetes pod. And create a replica set. In any case load is going to increase on your replica set. increase the number of replicas of the pods.

Create an instance

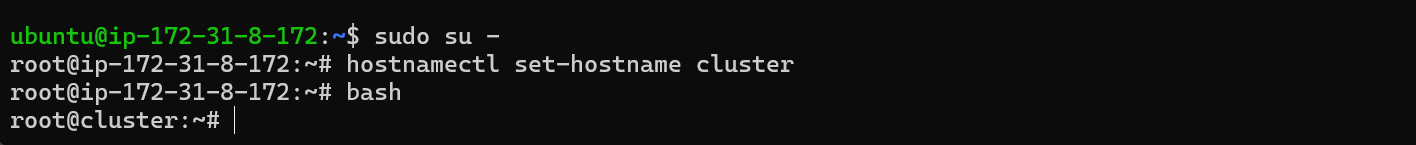
A screenshot of a computer

Description automatically generated

Create a role



Connect to the terminal

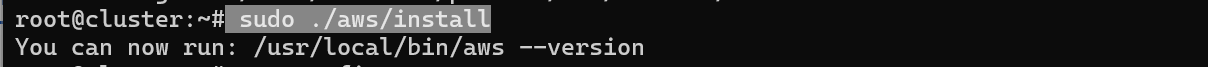


Unzip the necessary files for installation

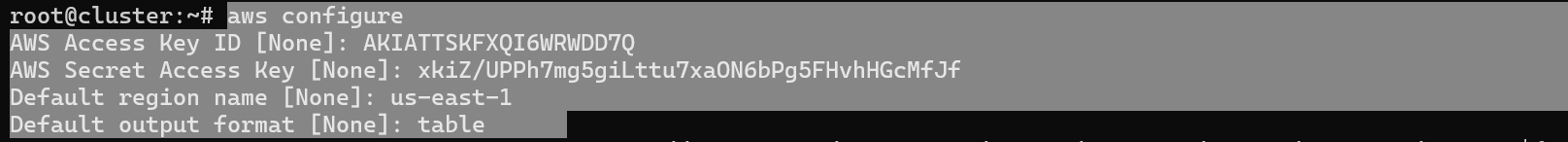
A black and white screen

Description automatically generated

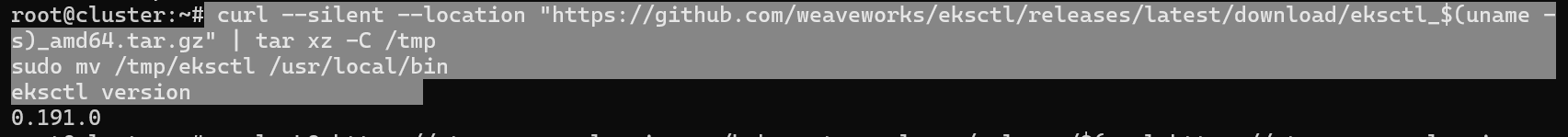
Install aws



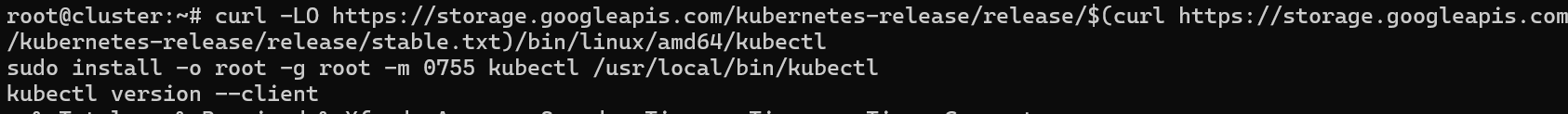
Configure aws



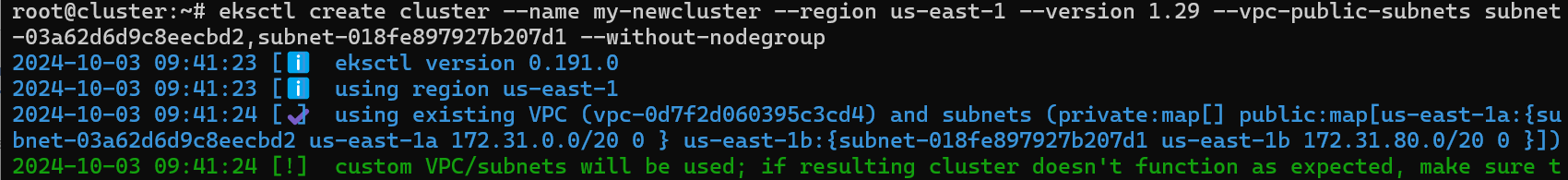
Install EKS tool



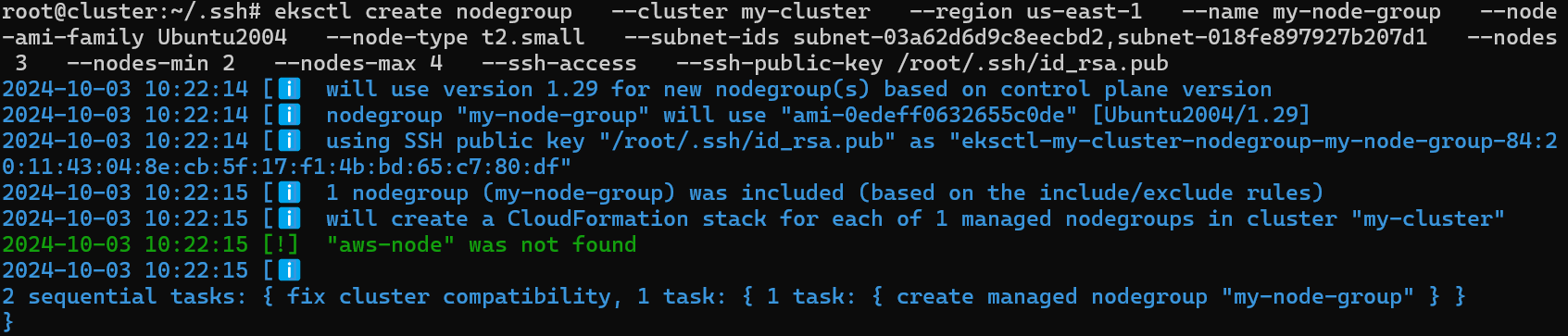
Install Kubectl



Create a cluster



Creating node group

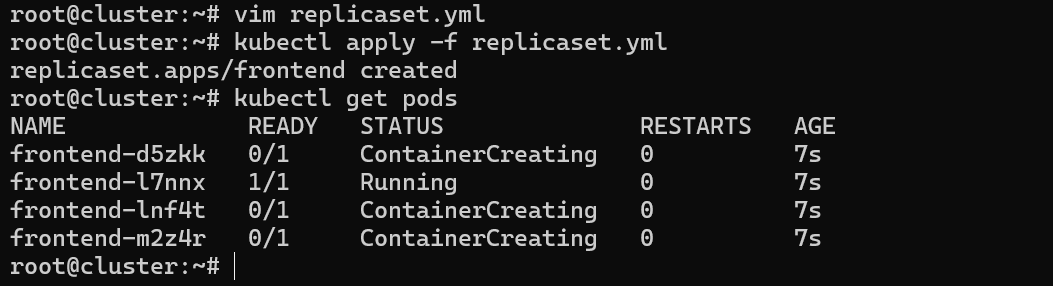


Create a vim replicaset.yml

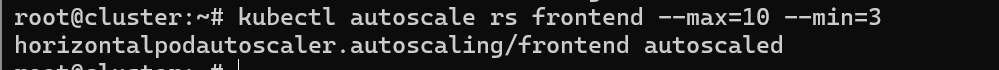
A screenshot of a computer

Description automatically generated

Here we see that the pods are created



We can increase or decrease the number of replicaset by the below command automatically



Inorder to increase it manually

A black screen with white text

Description automatically generated