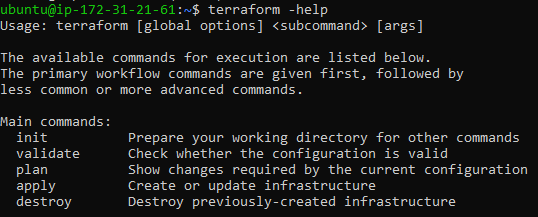
**DEVOPS COURSE CAPSTONE PROJECT**

**TASK 0: INSTALLING REQUIRED SOFTWARES**

1. AWS CLI:

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"  
 unzip awscliv2.zip  
 sudo ./aws/install  
 aws --version  
https://lh6.googleusercontent.com/uPOGV033J9rQI4iZ3zKsTRk7vqfTFiPAtIM8d1qJ_trpHEkfhRMYnveIas4c_W7EocedJNzkthwhvl8Wc607X_t0GNsDULFKzWRwPEol3hnkQLsr_xoGraPz9urW3GN5YXb2SANW

1. TERRAFORM:

sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl  
 curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -  
 sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb\_release -cs) main"  
 sudo apt-get update && sudo apt-get install terraform -y  
 terraform -help  


1. KUBECTL:

curl -o kubectl <https://amazon-eks.s3.us-west-2.amazonaws.com/1.21.2/2021-07-05/bin/linux/amd64/kubectl>curl -o kubectl.sha256 <https://amazon-eks.s3.us-west-2.amazonaws.com/1.21.2/2021-07-05/bin/linux/amd64/kubectl.sha256>openssl sha1 -sha256 kubectl  
 chmod +x ./kubectl  
 mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin  
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc  
kubectl version --short --client

https://lh3.googleusercontent.com/Vx2qXOZ09Q75q3J62UkI-lXd2YYZyBY8_aN4Ju-gLqILV8D3IH7liXK7uBy0hbiAFlmtzUppH7gFasEMoW9RfzEHIe101yPG0xA7NZOIpefk4VXFlEWILhWS8gMbch8Sd0Sbn99H

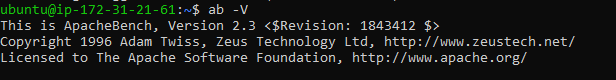
1. EKSTCL:

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp  
 sudo mv /tmp/eksctl /usr/local/bin  
 eksctl version  
https://lh4.googleusercontent.com/Q6dJPDqYAXcTCiwlbDnB_KHzuxx13XZWlcdj9O2BGdjIZ_2ErAuXWWuSewoUvbO8RgFKfEMIAgAVBaovfkAgYeyyfHA8AgjPnNPuDFxnvi4vekrmAaP8tegHqr1a8H-JnvdiANn0

1. HELM:

curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -  
 sudo apt-get install apt-transport-https --yes  
 echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list  
 sudo apt-get update  
 sudo apt-get install helm -y  
https://lh3.googleusercontent.com/vHYYZpNERq6t-4wfmu3XgzTGUXaQTw5b4WHingQCtFPnVGiN6ZocqoDOlgcCTbmYTYJRanM334IdBdex4SSpyzmmNbI7nEBDlktVagt93dHhplUV8rSibSTW0fZtgsi-d57fvxLm

1. AB:  
   apt-get update  
   apt-get install apache2-utils -y  
   ab -V



**TASK 1: EKS CLUSTER SETUP**

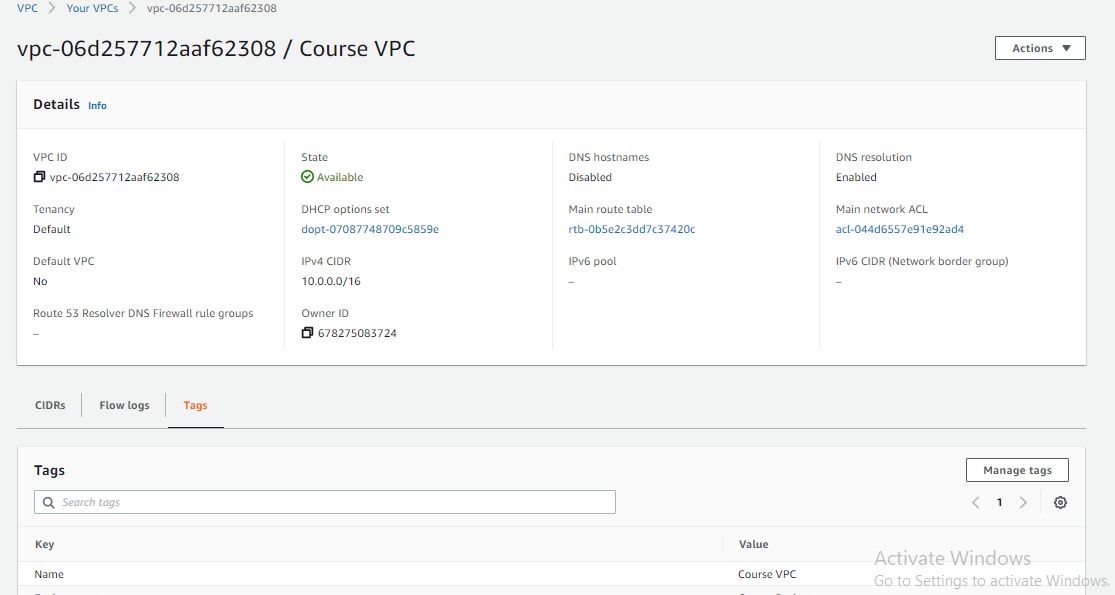
set up aws cli using “aws configure” command

Then perform three commands i.e:

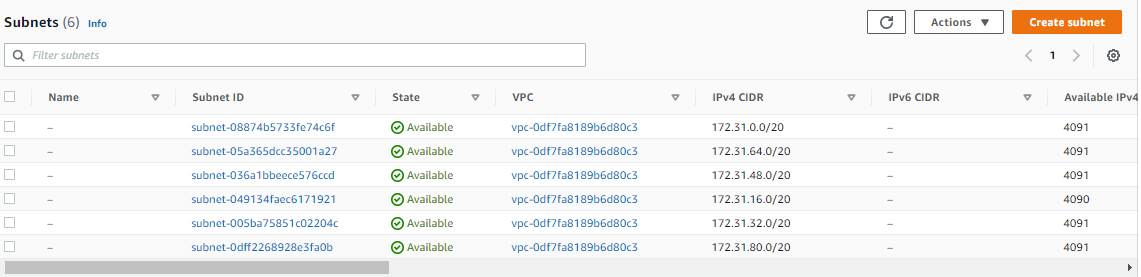
Terraform init  
Terraform plan

Terraform apply.

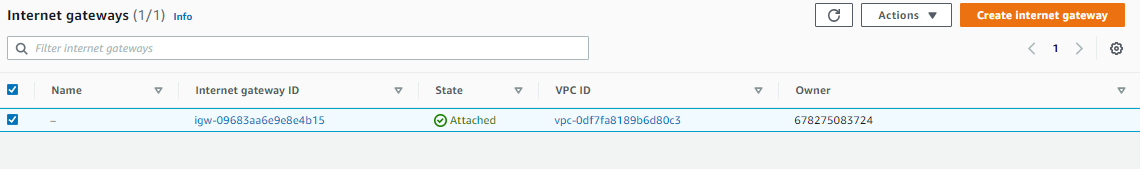
VPC

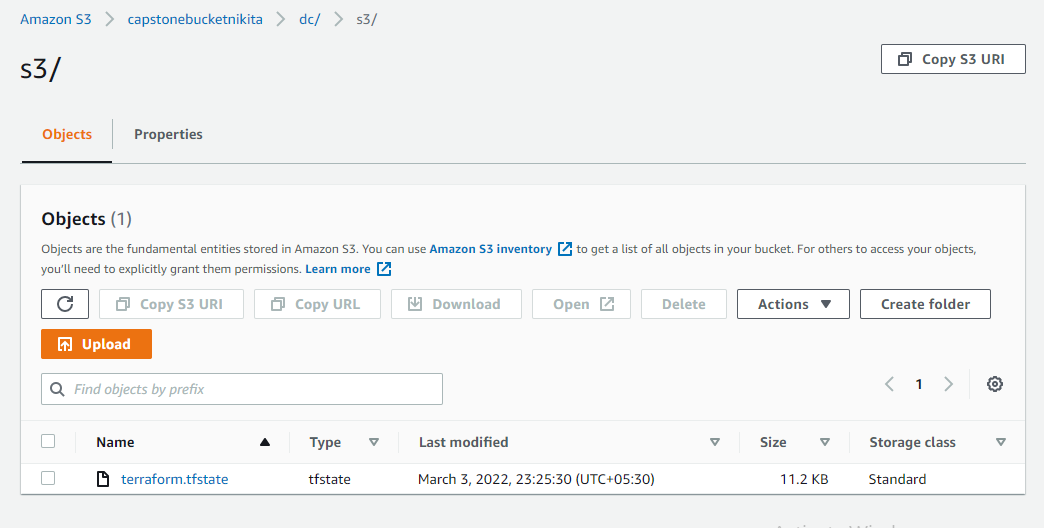


SUBNETS

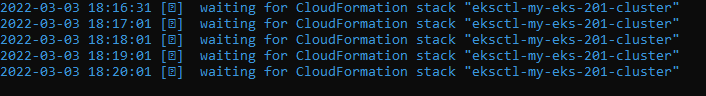


INTERNET GATEWAYS



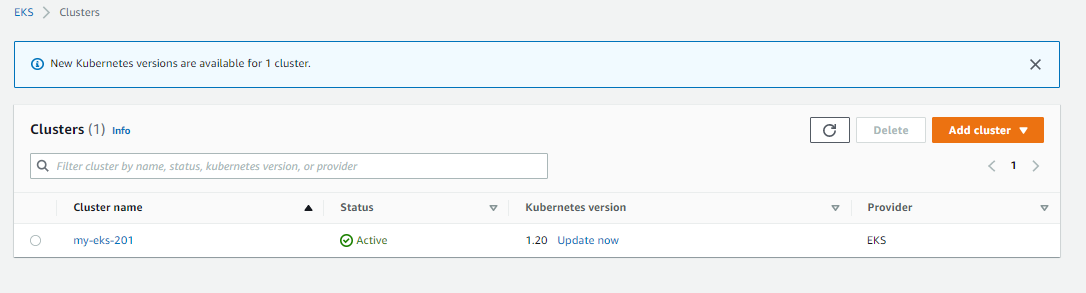
TF STATE BACKUP IN S3  


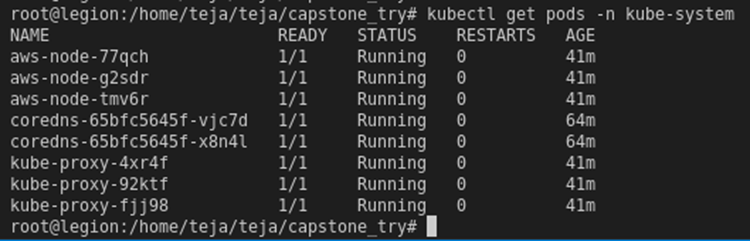
FOR CREATING EKS CLUSTER  
  
Ekstcl create cluster -f my-eks-conf-yaml



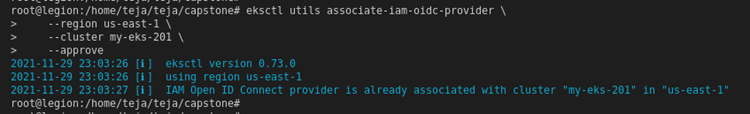
Cluster creation in progress

After Creation it can be checked in EKS in AWS



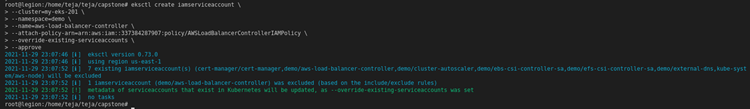
Available Pods after kubectl get pods –n Kube-System:  


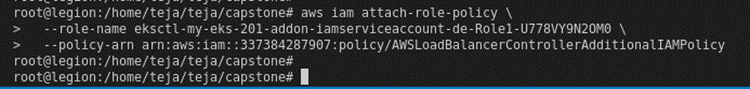
OIDC Provider



ALB Policy

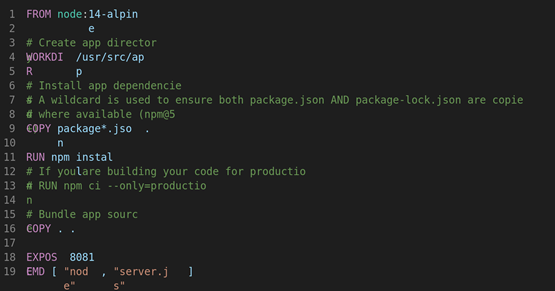


ALB Service Account  


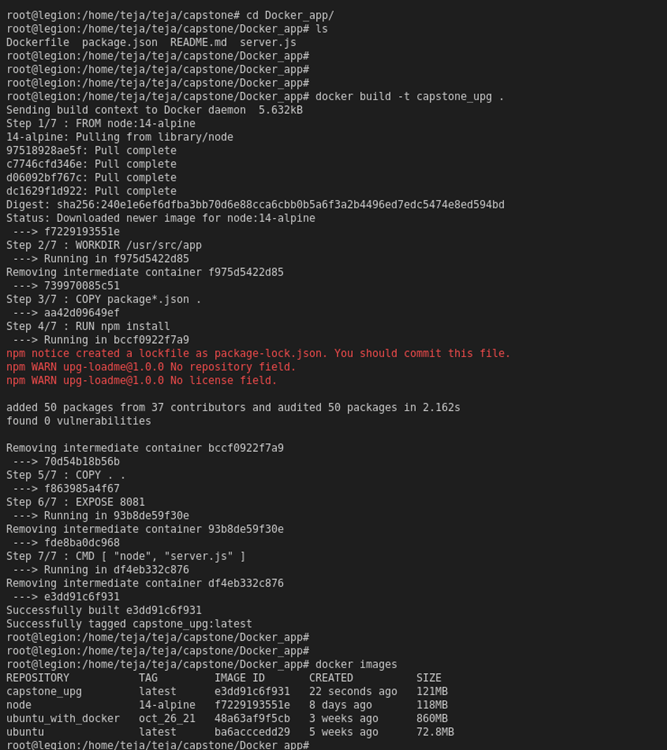
Amazon EKS Load Balancer Controller Role  


**TASK 2:  SAMPLE APPLICATION DEPLOYMENT**

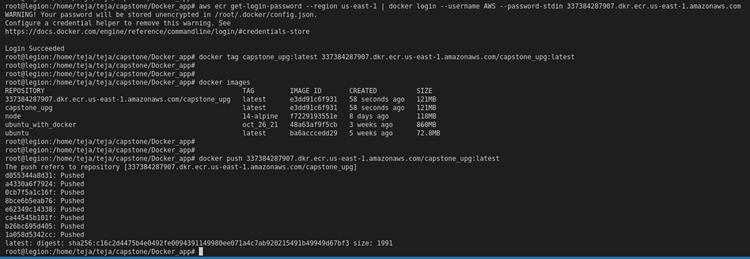
First create ECR repo in aws and then the dockerfile



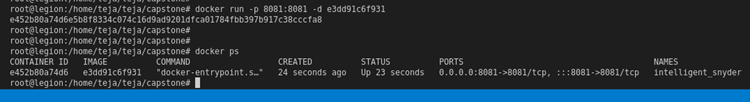
After creation build the file

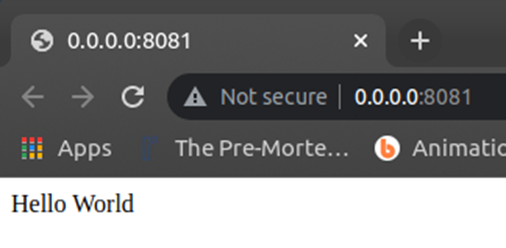


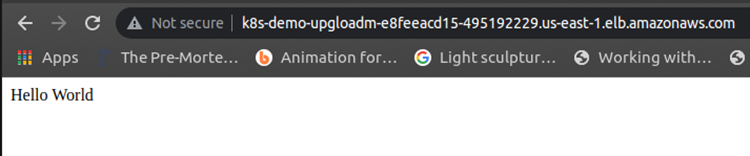
Authentication of ecr repor using aws cli or docker push

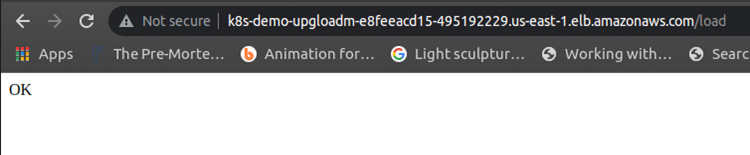


Docker image to check application



Application run   






**TASK 3: DEPLOYING REDIS ON KUBERNATES**

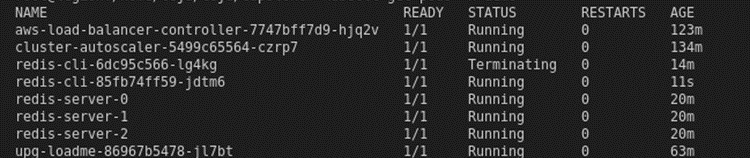
Create EBS volume with desired memory and then create and deploy PersistentVolume, PersistentVolumeClaim, config-map for redis, stateful redis-set redis server



Listing all pods and getting into redis-cli



Restarting the deployment

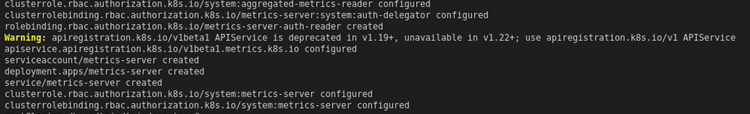


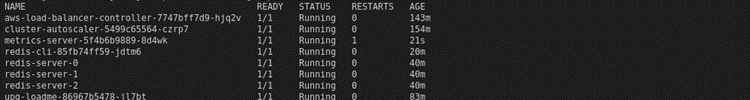
Check for new redis cli pod and then login to check the previously set foo key value is persisted

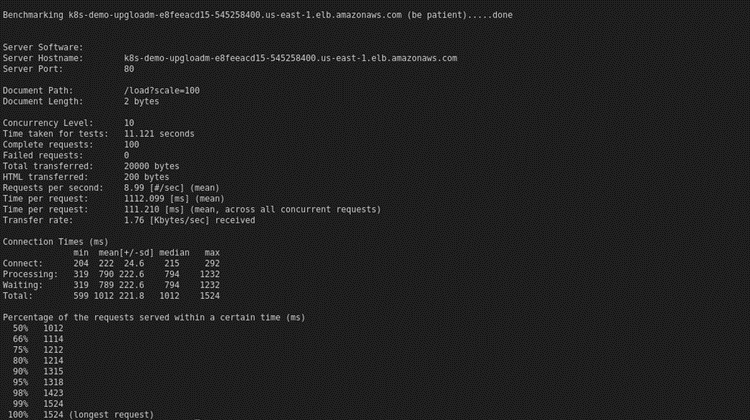


**TASK 4: TESTING AUTOSCALING OF APPLICATION**

Deploying metric server system







OUTPUT OF get hpa command  
https://lh6.googleusercontent.com/6YlguglhVfRle9F4fL2jJkFsUiuXC16-Fk0XGmnhyZ8f-pJZ6xrIKl22HogSjLX6MI-43Nzra9KPlLf4EHPTgzPAf2M9Id7nkAlHy5qdbdU-uL5-l4_48TYkFkCCuTdN2yOMy3eo

Application Load

