Deploying your application

Phase 5 Practice Project Set 4

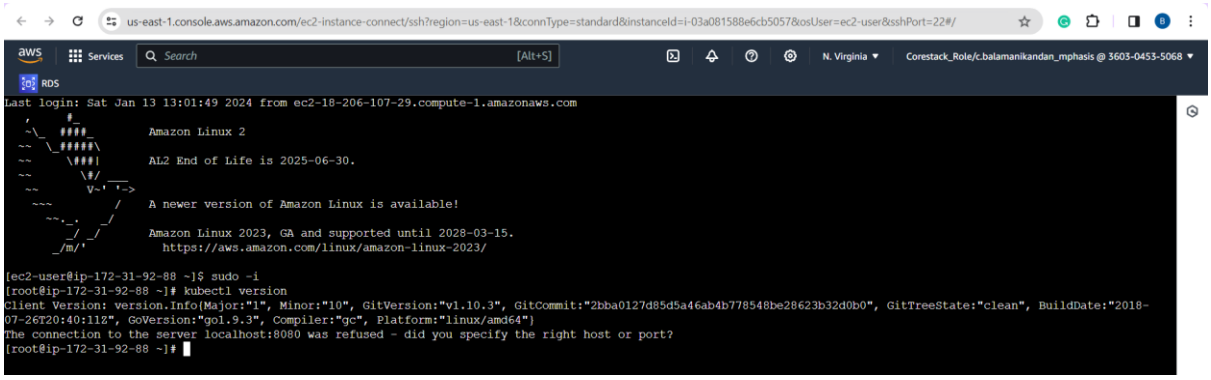
Name: C Balamanikandan

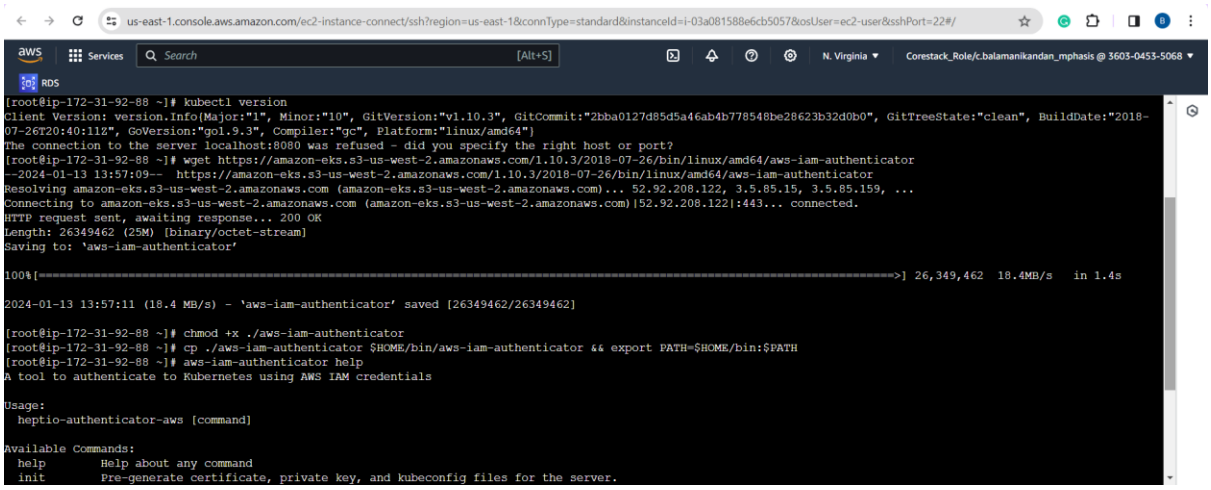
- Follow the steps shown below in the screenshot.

```
wget https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/kubectl

chmod +x kubectl

./kubectl
```

- Configure **kubectl** in PATH variable to call **kubectl** command globally. Follow the set of commands given below to configure PATH variable:

```
mkdir bin

cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH

kubectl version

kubectl version --short --client
```





- Install **EKS CTL command line** to create an EKS cluster.

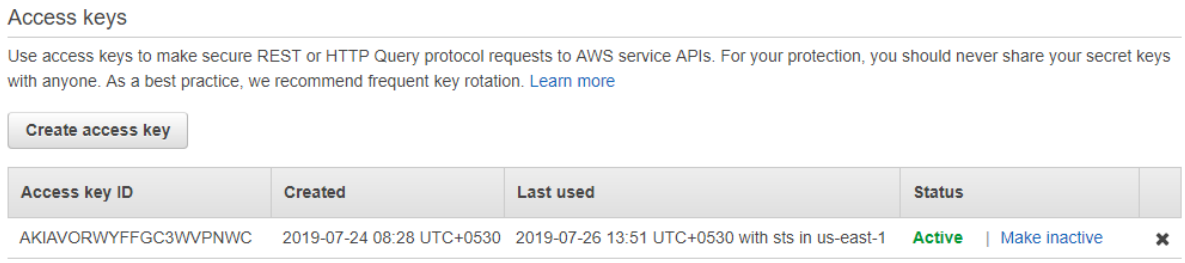
```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/download/latest_release/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

mv /tmp/eksctl /usr/local/bin

eksctl version
```

```
root@ip-172-31-86-69:~# curl --silent --location "https://github.com/weaveworks/eksctl/r
p
root@ip-172-31-86-69:~#  mv /tmp/eksctl /usr/local/bin
root@ip-172-31-86-69:~# eksctl version
[â
  ¹]  version.Info{BuiltAt:"", GitCommit:"", GitTag:"0.2.1"}
root@ip-172-31-86-69:~#
```

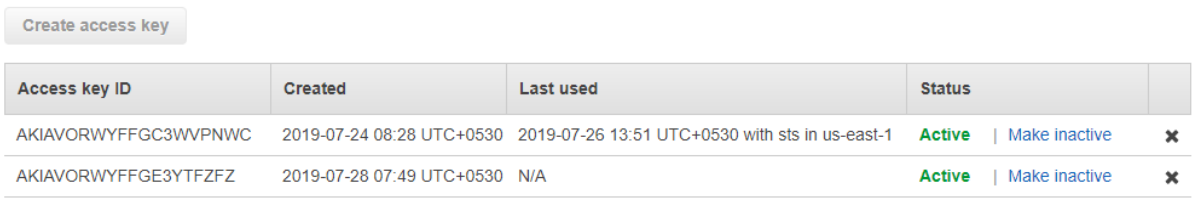
- Install AWS CLI using the sequence of commands given below.

```
apt install python-pip

pip install awscli

aws --version
```

- Configure AWS CLI. We need to create **Access Keys** in AWS IAM Console.



- Click on **Create Access key** and keep the keys safe with you.



- Configure AWS CLI and provide **Access Keys and Secret Access Keys** while configuring AWS CLI.

```
root@ip-172-31-17-73:~# aws configure
AWS Access Key ID [None]: AKIAVORWYFFGE3YTFZFZ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
root@ip-172-31-17-73:~#
```

Step 5.4.2: Creating an EKS cluster using eksctl command line

- Create an EKS Cluster using the command below:

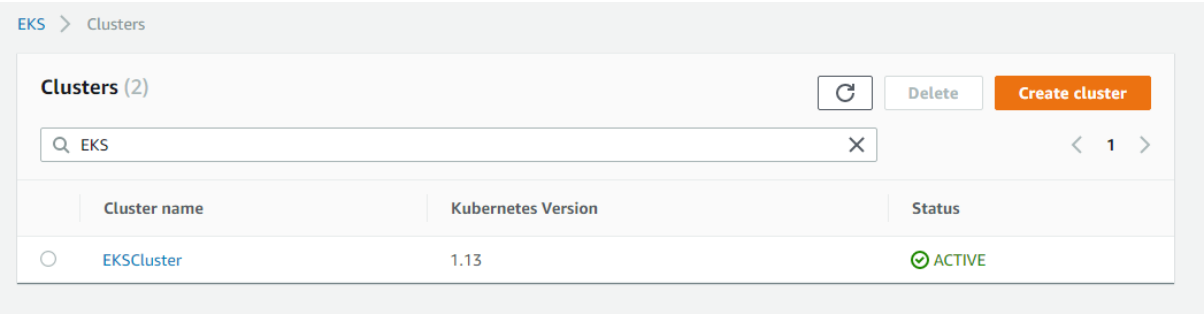
eksctl create cluster --name=EKSCluster --nodes=2 --region=us-west-2

```
root@ip-172-31-86-69:~# eksctl create cluster --name=EKSCluster --nodes=2 --region=us-west-2
[ℹ
  ¹]  using region us-west-2
[ℹ
  ¹]  setting availability zones to [us-west-2c us-west-2d us-west-2b]
[ℹ
  ¹]  subnets for us-west-2c - public:192.168.0.0/19 private:192.168.96.0/19
[ℹ
  ¹]  subnets for us-west-2d - public:192.168.32.0/19 private:192.168.128.0/19
[ℹ
  ¹]  subnets for us-west-2b - public:192.168.64.0/19 private:192.168.160.0/19
[ℹ
  ¹]  nodegroup "ng-c8e07a6f" will use "ami-03a55127c613349a7" [AmazonLinux2/1.13]
[ℹ
  ¹]  using Kubernetes version 1.13
[ℹ
  ¹]  creating EKS cluster "EKSCluster" in "us-west-2" region
[ℹ
  ¹]  will create 2 separate CloudFormation stacks for cluster itself and the initial nodegroup
[ℹ
  ¹]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-west-2 --name=EKSCluster'
[ℹ
  ¹]  2 sequential tasks: { create cluster control plane "EKSCluster", create nodegroup "ng-c8e07a6f" }
[ℹ
  ¹]  building cluster stack "eksctl-EKSCluster-cluster"
[ℹ
  ¹]  deploying stack "eksctl-EKSCluster-cluster"
```

```
[ℹ]  all EKS cluster resource for "EKSCluster" had been created
[ℹ]  saved kubeconfig as "/root/.kube/config"
[ℹ
  ¹]  adding role "arn:aws:iam::130374862735:role/eksctl-EKSCluster-nodegroup-ng-c8-NodeInstanceRole-1FKZC9GNJUUMU" to auth ConfigMap
[ℹ
  ¹]  nodegroup "ng-c8e07a6f" has 0 node(s)
[ℹ
  ¹]  waiting for at least 2 node(s) to become ready in "ng-c8e07a6f"
[ℹ
  ¹]  nodegroup "ng-c8e07a6f" has 2 node(s)
[ℹ
  ¹]  node "ip-192-168-28-149.us-west-2.compute.internal" is ready
[ℹ
  ¹]  node "ip-192-168-76-186.us-west-2.compute.internal" is ready
[ℹ
  ¹]  kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
[ℹ]  EKS cluster "EKSCluster" in "us-west-2" region is ready
```

- Validate the cluster using **kubectl get node** command through AWS Console.

```
root@ip-172-31-86-69:~# kubectl get node
NAME                                           STATUS    ROLES     AGE       VERSION
ip-192-168-28-149.us-west-2.compute.internal   Ready     <none>    5m        v1.13.7-eks-c57ff8
ip-192-168-76-186.us-west-2.compute.internal   Ready     <none>    5m        v1.13.7-eks-c57ff8
root@ip-172-31-86-69:~#
```



Step 5.4.3: Deploying an application to AWS EKS cluster

- Create Kubernetes deployment and service using the set of commands mentioned below:

kubectl run kubernetes-bootcamp --image=docker.io/jocatalin/kubernetes-bootcamp:v1 --port=8080

kubectl expose deployment/kubernetes-bootcamp --port=8080 --target-port=8080 --type=NodePort