

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
```

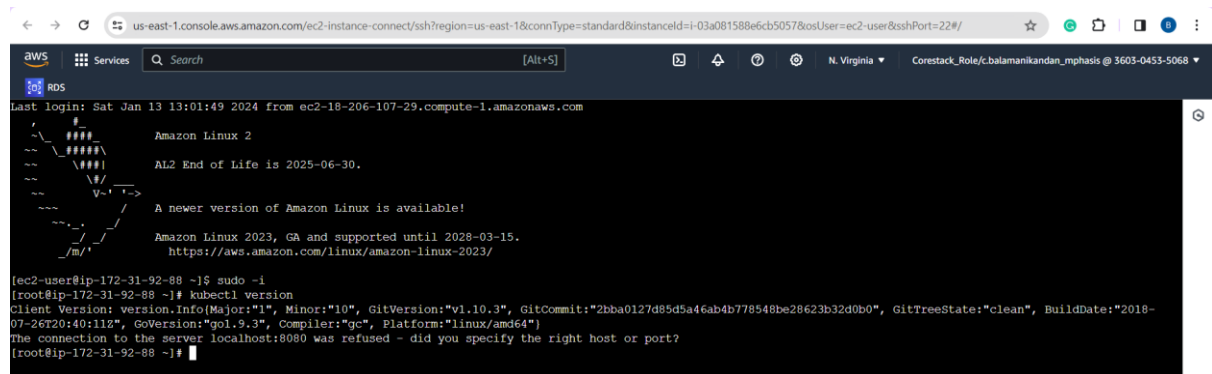
- Configure **kubectl** in PATH variable to call **kubectl** command globally. Follow the set of commands given below to configure PATH variable:

```
mkdir bin
```

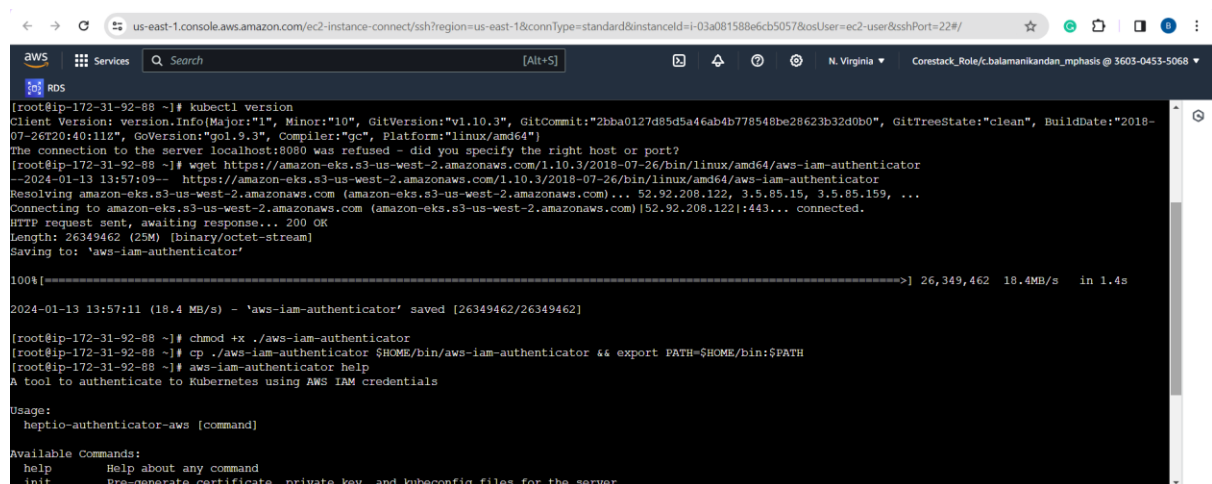
```
cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

```
kubectl version
```

```
kubectl version --short --client
```



```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-03a081588e6cb5057&osUser=ec2-user&sshPort=22#/  
[Alt+S]  
N. Virginia  
Corestack_Role/c.balamananandan_mphasis @ 3603-0453-5068  
last login: Sat Jan 13 13:01:49 2024 from ec2-18-206-107-29.compute-1.amazonaws.com  
Amazon Linux 2  
AL2 End of Life is 2025-06-30.  
A newer version of Amazon Linux is available!  
Amazon Linux 2023, GA and supported until 2028-03-15.  
https://aws.amazon.com/linux/amazon-linux-2023/  
[ec2-user@ip-172-31-92-88 ~]$ sudo -i  
[root@ip-172-31-92-88 ~]# kubectl version  
client Version: version.Info{Major:"1", Minor:"10", GitVersion:"v1.10.3", GitCommit:"2bba0127d85d5a46ab4b778548be28623b32d0b0", GitTreeState:"clean", BuildDate:"2018-07-26T20:40:11Z", GoVersion:"go1.9.3", Compiler:"gc", Platform:"linux/amd64"}  
The connection to the server localhost:8080 was refused - did you specify the right host or port?  
[root@ip-172-31-92-88 ~]#
```



```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-03a081588e6cb5057&osUser=ec2-user&sshPort=22#/  
[Alt+S]  
N. Virginia  
Corestack_Role/c.balamananandan_mphasis @ 3603-0453-5068  
[root@ip-172-31-92-88 ~]# kubectl version  
client Version: version.Info{Major:"1", Minor:"10", GitVersion:"v1.10.3", GitCommit:"2bba0127d85d5a46ab4b778548be28623b32d0b0", GitTreeState:"clean", BuildDate:"2018-07-26T20:40:11Z", GoVersion:"go1.9.3", Compiler:"gc", Platform:"linux/amd64"}  
The connection to the server localhost:8080 was refused - did you specify the right host or port?  
[root@ip-172-31-92-88 ~]# wget https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator  
--2024-01-13 13:57:09-- https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator  
Resolving amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)... 52.92.208.122, 3.5.85.159, 3.5.85.159, ...  
Connecting to amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)|52.92.208.122|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 26349462 (25M) [binary/octet-stream]  
Saving to: 'aws-iam-authenticator'  
100%[=====] 26,349,462 18.4MB/s in 1.4s  
2024-01-13 13:57:11 (18.4 MB/s) - 'aws-iam-authenticator' saved [26349462/26349462]  
[root@ip-172-31-92-88 ~]# chmod +x ./aws-iam-authenticator  
[root@ip-172-31-92-88 ~]# cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH  
[root@ip-172-31-92-88 ~]# aws-iam-authenticator help  
A tool to authenticate to Kubernetes using AWS IAM credentials  
Usage:  
aws-iam-authenticator-aws [command]  
Available Commands:  
help      Help about any command  
init      Pre-generate certificate, private key, and kubeconfig files for the server.
```

- 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
  ^] 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.

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

Create access key

Access key ID	Created	Last used	Status	
AKIAVORWYFFGC3WVPNWC	2019-07-24 08:28 UTC+0530	2019-07-26 13:51 UTC+0530 with sts in us-east-1	Active	Make inactive ✕

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

Create access key

Access key ID	Created	Last used	Status	
AKIAVORWYFFGC3WVPNWC	2019-07-24 08:28 UTC+0530	2019-07-26 13:51 UTC+0530 with sts in us-east-1	Active	Make inactive ✕
AKIAVORWYFFGE3YTFZFZ	2019-07-28 07:49 UTC+0530	N/A	Active	Make inactive ✕

- 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]: AKIAVORWYFFGE3YTF2FZ
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
[â] using region us-west-2
[â] setting availability zones to [us-west-2c us-west-2d us-west-2b]
[â] subnets for us-west-2c - public:192.168.0.0/19 private:192.168.96.0/19
[â] subnets for us-west-2d - public:192.168.32.0/19 private:192.168.128.0/19
[â] subnets for us-west-2b - public:192.168.64.0/19 private:192.168.160.0/19
[â] nodegroup "ng-c8e07a6f" will use "ami-03a55127c613349a7" [AmazonLinux2/1.13]
[â] using Kubernetes version 1.13
[â] creating EKS cluster "EKScluster" in "us-west-2" region
[â] will create 2 separate CloudFormation stacks for cluster itself and the initial nodegroup
[â] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-west-2 --name=EKScluster'
[â] 2 sequential tasks: { create cluster control plane "EKScluster", create nodegroup "ng-c8e07a6f" }
[â] building cluster stack "eksctl-EKScluster-cluster"
[â] deploying stack "eksctl-EKScluster-cluster"

```

```

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

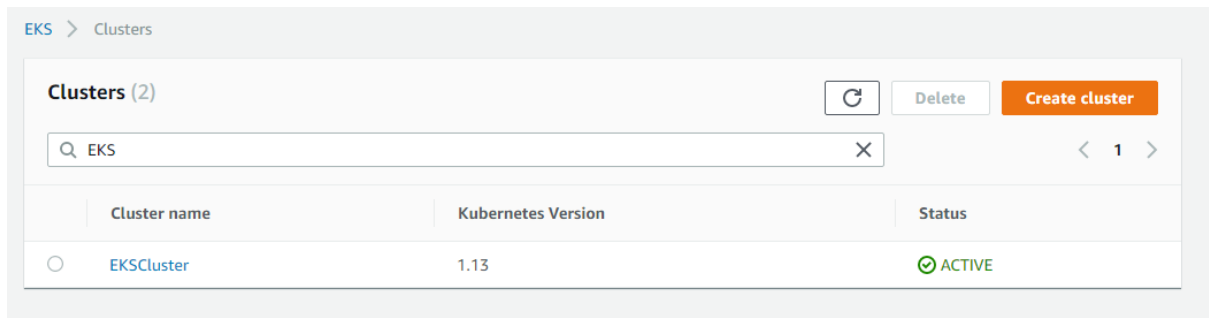
```

- 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
```