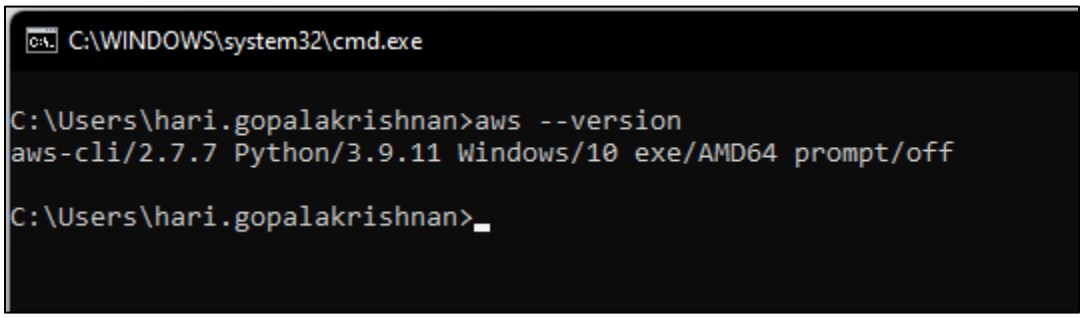
Assignment 8 :: Deploy EKS cluster using GUI and CLI on AWS. Deploy deployment using httpd image on EKS.

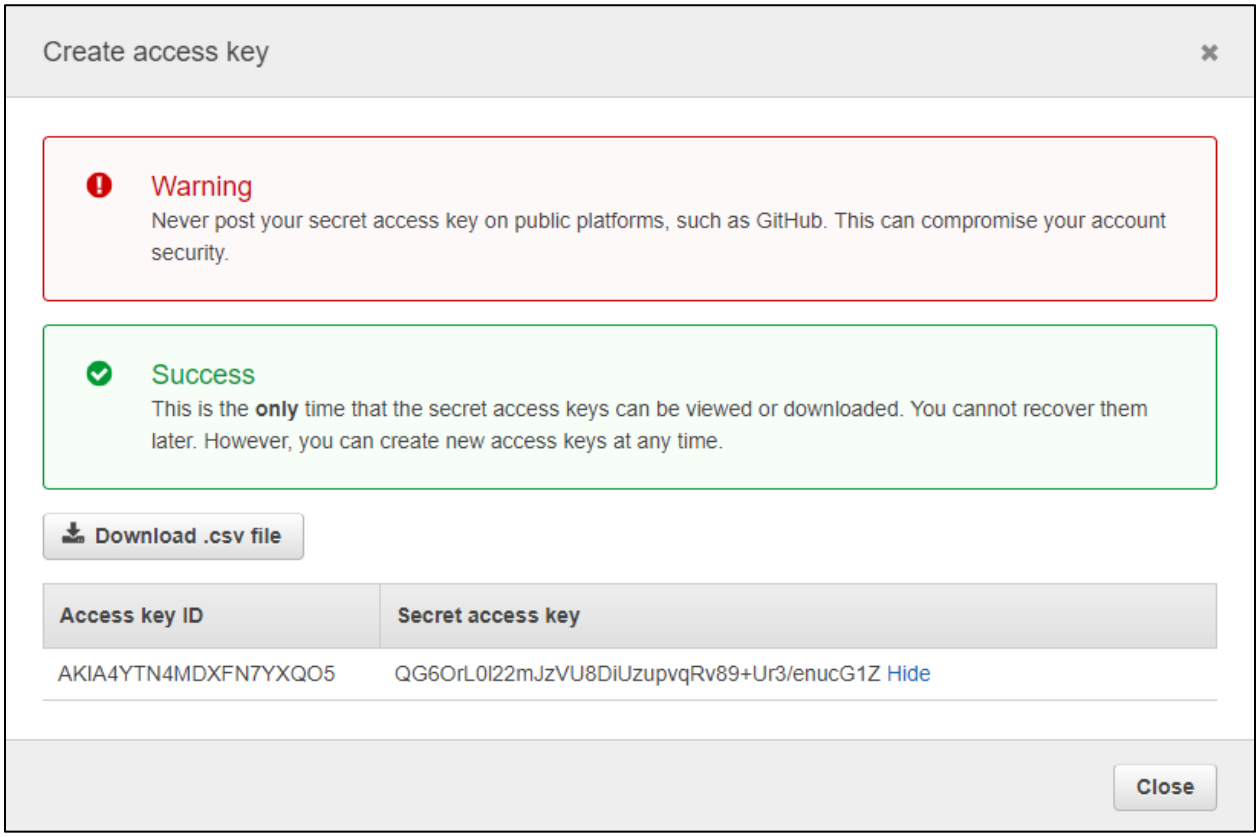
Step 1 :: Install AWS CLI

Step 1.1 :: Check for aws version



Step 1.2 :: Configure AWS Command Line using Security Credentials

- Go to AWS Management Console --> Services --> IAM
- Select the IAM User: hari.gopalakrishnan@globallogic.com
- Click on Security credentials tab
- Click on Create access key
- Copy Access ID and Secret access key



- Run “**aws configure**”

```
C:\Users\hari.gopalakrishnan>aws configure
AWS Access Key ID [****************4GJA]: AKIA4YTN4MDXFN7YXQO5
AWS Secret Access Key [****************VTLH]:
Default region name [us-east-1]:
Default output format [None]:

C:\Users\hari.gopalakrishnan>
```

- Test if aws cli is working fine

Step 2 :: Install kubectl CLI

Install and check the version

- **mkdir kubectlbinary**

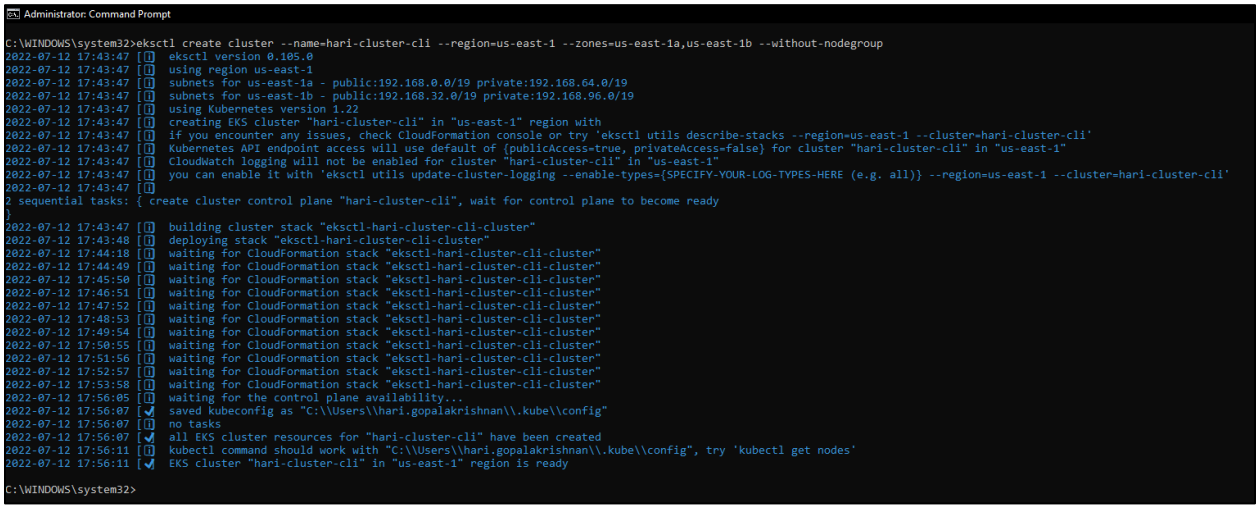
- **cd kubectlbinary**
- **curl -o kubectl.exe https://amazon-eks.s3.us-west-2.amazonaws.com/1.16.8/2020-04-16/bin/windows/amd64/kubectl.exe**
- Update the system Path environment variable
    - C:\Users\hari.gopalakrishnan\kubectlbinary
- Verify the kubectl client version
    - **kubectl version --short –client**
    - **kubectl version --client**

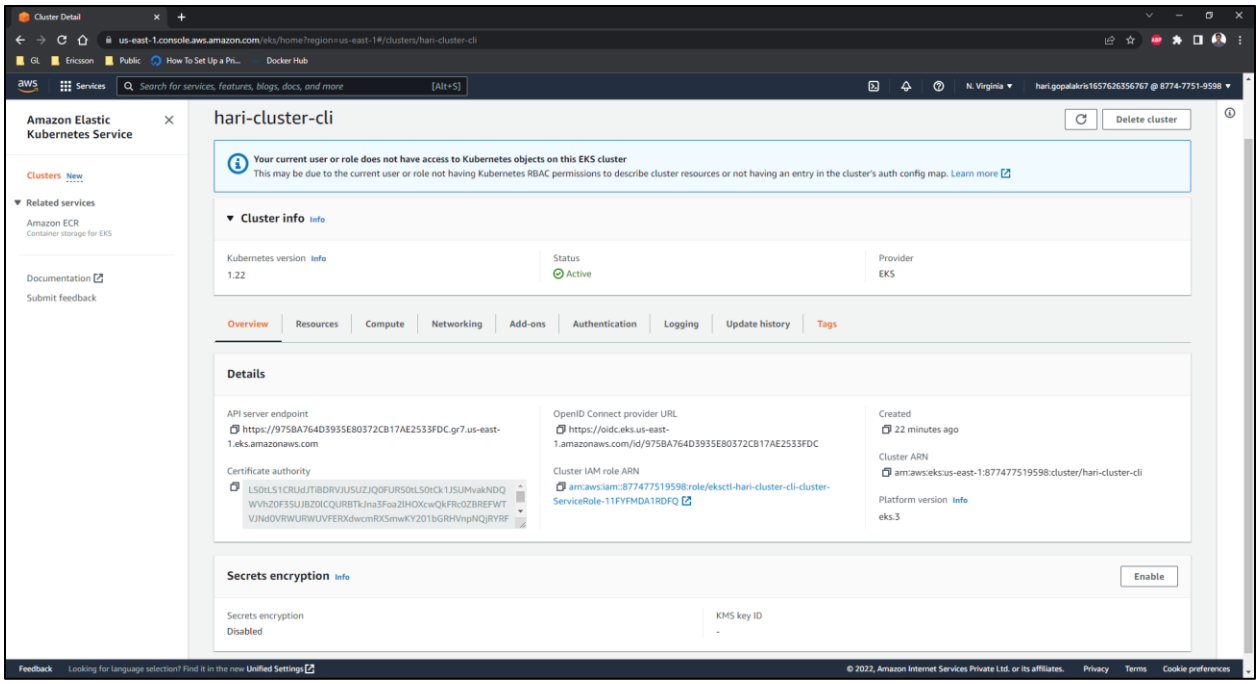
Step 3 :: Install eksctl

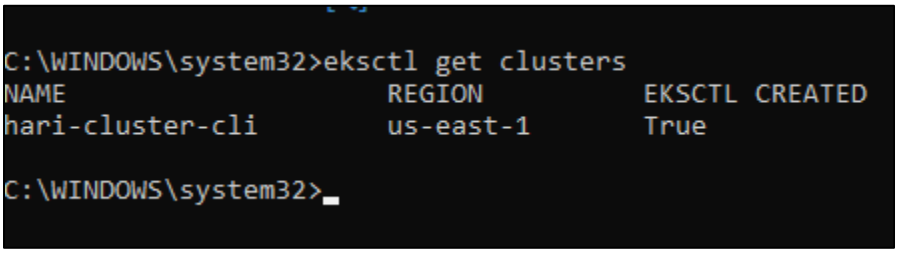
- **@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"**
- **choco install -y eksctl**
- **eksctl version**

Step 4 : : Creating a cluster

eksctl create cluster --name=hari-cluster-cli --region=us-east-1 --zones=us-east-1a,us-east-1b --without-nodegroup





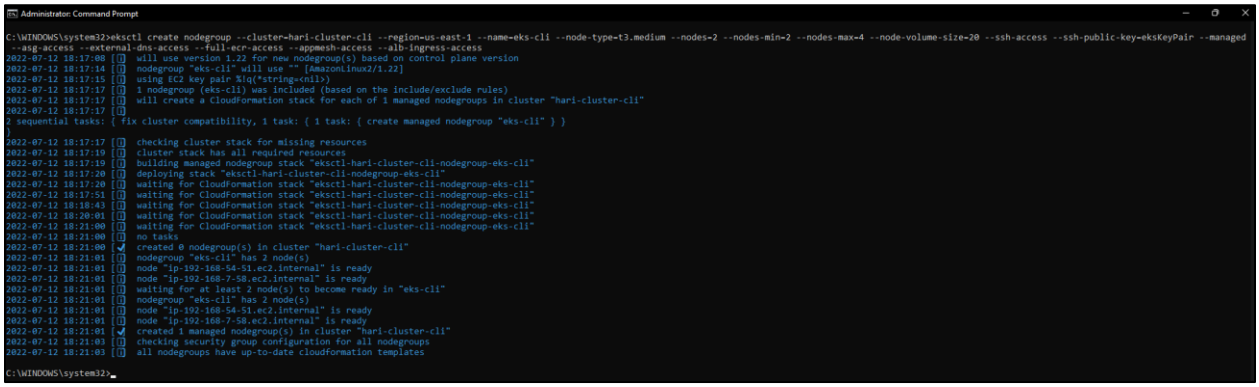


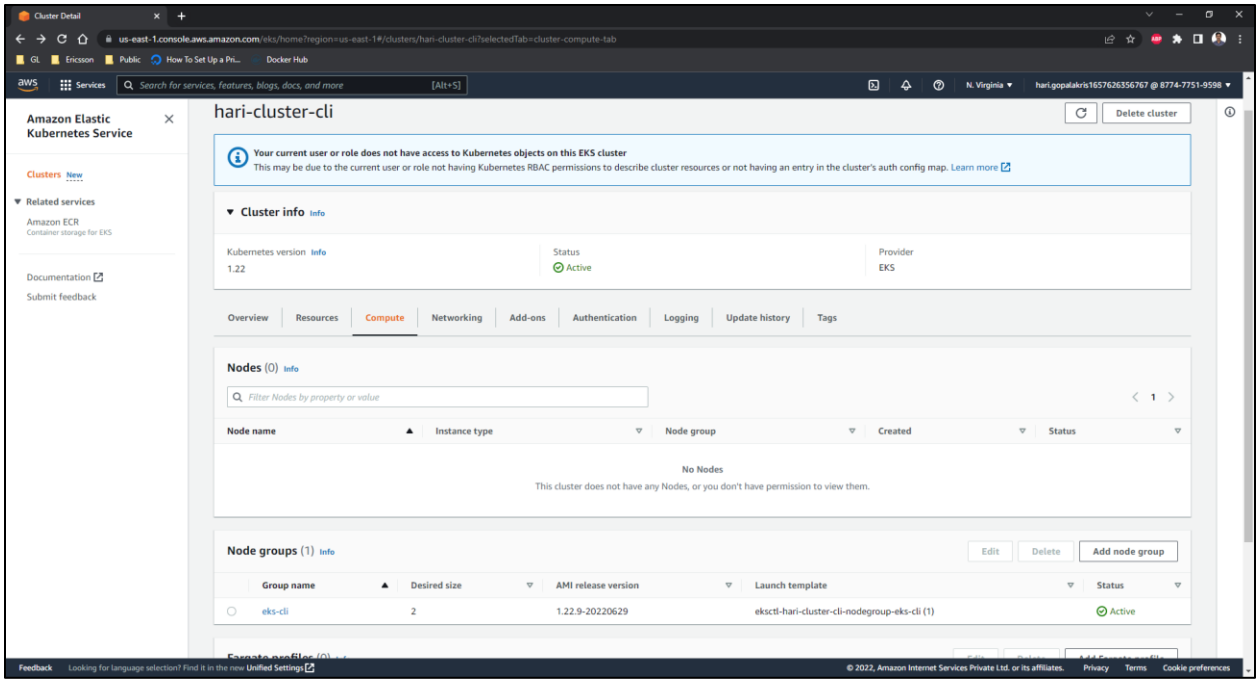
Step 5 :: Create nodes to the eks cluster

Step 5.1 :: Create a ssh key

Step 5.2 :: Create nodes

eksctl create nodegroup --cluster=hari-cluster-cli --region=us-east-1 --name=eks-cli --node-type=t3.medium --nodes=2 --nodes-min=2 --nodes-max=4 --node-volume-size=20 --ssh-access --ssh-public-key=eksKeyPair --managed --asg-access --external-dns-access --full-ecr-access --appmesh-access --alb-ingress-access






```
Administrator: Command Prompt

C:\WINDOWS\system32>kubectl get nodes -o wide
NAME                          STATUS   ROLES    AGE     VERSION              INTERNAL-IP     EXTERNAL-IP     OS-IMAGE         KERNEL-VERSION                CONTAINER-RUNTIME
ip-192-168-54-51.ec2.internal   Ready    <none>   4m9s    v1.22.9-eks-810597c   192.168.54.51   3.221.170.129   Amazon Linux 2   5.4.196-108.356.amzn2.x86_64   docker://20.10.13
ip-192-168-7-58.ec2.internal    Ready    <none>   4m15s   v1.22.9-eks-810597c   192.168.7.58    18.209.172.61   Amazon Linux 2   5.4.196-108.356.amzn2.x86_64   docker://20.10.13

C:\WINDOWS\system32>
```

Step 5.3 :: Create a httpd pod and check its status

kubectl run httpd --image=httpd
kubectl get pods -o wide

```
Administrator: Command Prompt

C:\WINDOWS\system32>kubectl get pods -o wide
NAME                    READY   STATUS    RESTARTS   AGE     IP              NODE                             NOMINATED NODE   READINESS GATES
httpd-86f8b7f7cd-fwbrf   1/1     Running   0          3m56s   192.168.39.187   ip-192-168-54-51.ec2.internal   <none>           <none>

C:\WINDOWS\system32>
```