

## Kubernetes Task-2

### Installing AWS CLI:

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
sureshk@SSMS:~$ aws --version
aws-cli/2.25.6 Python/3.12.9 Linux/5.15.167.4-microsoft-standard-WSL2 exe/x86_64.ubuntu.24
sureshk@SSMS:~$
```

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
sureshk@SSMS:~$ aws --version
aws-cli/2.25.6 Python/3.12.9 Linux/5.15.167.4-microsoft-standard-WSL2 exe/x86_64.ubuntu.24
sureshk@SSMS:~$ aws configure
AWS Access Key ID [None]: AKIA2RP6IRNVJAXYRAE2
AWS Secret Access Key [None]: none!tcU6JLGQD717tDc0FqVxlbuloSWUDRGwxge
Default region name [None]: Mumbai
Default output format [None]:
sureshk@SSMS:~$
```

### Installing Eksctl:

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ curl -sL "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin/
sureshk@SSMS:~$ eksctl version
0.206.0
sureshk@SSMS:~$
```

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ curl -sL "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin/
sureshk@SSMS:~$ eksctl version
0.206.0
sureshk@SSMS:~$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
sureshk@SSMS:~$
```

## Creating an EKS Cluster:

```
Windows PowerShell x sureshk@SSMS: ~ x + v
2025-03-30 18:21:57 [i] creating addon: metrics-server
2025-03-30 18:21:57 [i] successfully created addon: metrics-server
2025-03-30 18:21:59 [i] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure
the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon cre
ation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2025-03-30 18:21:59 [i] creating addon: vpc-cni
2025-03-30 18:21:59 [i] successfully created addon: vpc-cni
2025-03-30 18:22:02 [i] creating addon: kube-proxy
2025-03-30 18:22:03 [i] successfully created addon: kube-proxy
2025-03-30 18:22:04 [i] creating addon: coredns
2025-03-30 18:22:04 [i] successfully created addon: coredns
2025-03-30 18:24:23 [i] building managed nodegroup stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-03-30 18:24:25 [i] deploying stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-03-30 18:24:25 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-03-30 18:25:01 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-03-30 18:25:35 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-03-30 18:27:30 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-03-30 18:27:30 [i] waiting for the control plane to become ready
2025-03-30 18:27:31 [i] saved kubeconfig as "/home/sureshk/.kube/config"
2025-03-30 18:27:31 [i] no tasks
2025-03-30 18:27:31 [i] all EKS cluster resources for "my-cluster" have been created
2025-03-30 18:27:32 [i] nodegroup "my-nodes" has 2 node(s)
2025-03-30 18:27:32 [i] node "ip-192-168-18-213.ec2.internal" is ready
2025-03-30 18:27:32 [i] node "ip-192-168-59-128.ec2.internal" is ready
2025-03-30 18:27:32 [i] waiting for at least 2 node(s) to become ready in "my-nodes"
2025-03-30 18:27:33 [i] nodegroup "my-nodes" has 2 node(s)
2025-03-30 18:27:33 [i] node "ip-192-168-18-213.ec2.internal" is ready
2025-03-30 18:27:33 [i] node "ip-192-168-59-128.ec2.internal" is ready
2025-03-30 18:27:33 [i] created 1 managed nodegroup(s) in cluster "my-cluster"
2025-03-30 18:27:34 [i] kubectl command should work with "/home/sureshk/.kube/config", try 'kubectl get nodes'
2025-03-30 18:27:34 [i] EKS cluster "my-cluster" in "us-east-1" region is ready
sureshk@SSMS:~$
sureshk@SSMS:~$
sureshk@SSMS:~$
sureshk@SSMS:~$
sureshk@SSMS:~$
sureshk@SSMS:~$
```

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-18-213.ec2.internal     Ready    <none>   2m33s v1.32.1-eks-5d632ec
ip-192-168-59-128.ec2.internal     Ready    <none>   2m39s v1.32.1-eks-5d632ec
sureshk@SSMS:~$
```

## Deploying Nginx on EKS:

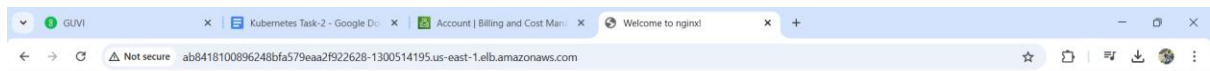
```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ vi nginx-deployment.yaml
sureshk@SSMS:~$ cat nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80

sureshk@SSMS:~$
```

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
sureshk@SSMS:~$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    2/2     2            2           23s
sureshk@SSMS:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-96b9d695-dv2ts     1/1     Running   0           34s
nginx-deployment-96b9d695-qdhpq     1/1     Running   0           34s
sureshk@SSMS:~$
```

```
Windows PowerShell x sureshk@SSMS: ~ x + v
sureshk@SSMS:~$ vi nginx-service.yaml
sureshk@SSMS:~$ cat nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  type: LoadBalancer
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80

sureshk@SSMS:~$ kubectl apply -f nginx-service.yaml
service/nginx-service created
sureshk@SSMS:~$ kubectl get svc nginx-service
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
nginx-service  LoadBalancer  10.100.96.6      ab8418100896248bfa579eaa2f922628-1300514195.us-east-1.elb.amazonaws.com  80:32492/TCP  20s
sureshk@SSMS:~$
```



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

---