

# Kubernetes Task - 2

Eksctl Install

```
ec2-user@ip-172-31-90-145:~ X + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hraj0\Desktop> cd ..
PS C:\Users\hraj0> cd downloads
PS C:\Users\hraj0\downloads> ssh -i hp.pem ec2-user@54.175.57.24

#_
~\_ #####_ Amazon Linux 2023
~~ \#####\
~~ \###|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '->
~~~
~~.-.-
~~/_/_/_
_/_/_'

Last login: Fri Apr 4 15:26:43 2025 from 223.185.61.4
[ec2-user@ip-172-31-90-145 ~]$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin
[ec2-user@ip-172-31-90-145 ~]$ eksctl version
0.207.0
[ec2-user@ip-172-31-90-145 ~]$ |
```

# Kubectl Install

```
[ec2-user@ip-172-31-90-145 ~]$ eksctl version
0.207.0
[ec2-user@ip-172-31-90-145 ~]$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
chmod +x kubectl
sudo mv kubectl /usr/local/bin
```

% Total		% Received		% Xferd		Average Speed		Time	Time	Time	Current
						Dload	Upload	Total	Spent	Left	Speed
100	138	100	138	0	0	3268	0	--:--:--	--:--:--	--:--:--	3285
100	54.6M	100	54.6M	0	0	70.4M	0	--:--:--	--:--:--	--:--:--	70.4M

```
[ec2-user@ip-172-31-90-145 ~]$ |
```

# AWS EKS Cluster Create

```
ec2-user@ip-172-31-90-145:~$ aws configure
[ec2-user@ip-172-31-90-145 ~]$ aws configure
AWS Access Key ID [*****]: AKIAU6PKIZLAHJOLXWDL
AWS Secret Access Key [*****]: 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
Default region name [us-east-1]: us-east-1
Default output format [json]: json
[ec2-user@ip-172-31-90-145 ~]$ eksctl create cluster --name my-cluster --region us-east-1 --nodegroup-name my-nodes --node-type t3.medium --nodes 2
2025-04-04 18:37:47 [I] eksctl version 0.207.0
2025-04-04 18:37:47 [I] using region us-east-1
2025-04-04 18:37:47 [I] setting availability zones to [us-east-1b us-east-1c]
2025-04-04 18:37:47 [I] subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
2025-04-04 18:37:47 [I] subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
2025-04-04 18:37:47 [I] nodegroup "my-nodes" will use "" [AmazonLinux2/1.32]
2025-04-04 18:37:47 [I] using Kubernetes version 1.32
2025-04-04 18:37:47 [I] creating EKS cluster "my-cluster" in "us-east-1" region with managed nodes
2025-04-04 18:37:47 [I] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-04-04 18:37:47 [I] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=my-cluster'
2025-04-04 18:37:47 [I] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "my-cluster" in "us-east-1"
2025-04-04 18:37:47 [I] CloudWatch logging will not be enabled for cluster "my-cluster" in "us-east-1"
2025-04-04 18:37:47 [I] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=my-cluster'
2025-04-04 18:37:47 [I] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-04-04 18:37:47 [I]
2 sequential tasks: { create cluster control plane "my-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "my-nodes",
  },
}
2025-04-04 18:37:47 [I] building cluster stack "eksctl-my-cluster-cluster"
2025-04-04 18:37:47 [I] deploying stack "eksctl-my-cluster-cluster"
2025-04-04 18:38:17 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:38:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:39:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:40:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:41:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:42:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:43:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:44:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:45:48 [I] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:45:49 [!] 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 creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2025-04-04 18:45:49 [I] creating addon: vpc-cni
2025-04-04 18:45:50 [I] successfully created addon: vpc-cni
2025-04-04 18:45:50 [I] creating addon: kube-proxy
2025-04-04 18:45:50 [I] successfully created addon: kube-proxy
2025-04-04 18:45:51 [I] creating addon: coredns
2025-04-04 18:45:51 [I] successfully created addon: coredns
2025-04-04 18:45:51 [I] creating addon: metrics-server
2025-04-04 18:45:51 [I] successfully created addon: metrics-server
```

```
ec2-user@ip-172-31-90-145:~$ cat eksctl.yml
1 task: { create addons },
  wait for control plane to become ready,
},
create managed nodegroup "my-nodes",
}
}

2025-04-04 18:37:47 [i] building cluster stack "eksctl-my-cluster-cluster"
2025-04-04 18:37:47 [i] deploying stack "eksctl-my-cluster-cluster"
2025-04-04 18:38:17 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:38:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:39:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:40:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:41:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:42:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:43:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:44:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:45:48 [i] waiting for CloudFormation stack "eksctl-my-cluster-cluster"
2025-04-04 18:45:49 [!] 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 creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2025-04-04 18:45:49 [i] creating addon: vpc-cni
2025-04-04 18:45:50 [i] successfully created addon: vpc-cni
2025-04-04 18:45:50 [i] creating addon: kube-proxy
2025-04-04 18:45:50 [i] successfully created addon: kube-proxy
2025-04-04 18:45:51 [i] creating addon: coredns
2025-04-04 18:45:51 [i] successfully created addon: coredns
2025-04-04 18:45:51 [i] creating addon: metrics-server
2025-04-04 18:45:51 [i] successfully created addon: metrics-server
2025-04-04 18:47:52 [i] building managed nodegroup stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:47:52 [i] deploying stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:47:52 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:48:22 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:49:04 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:49:47 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:51:34 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:51:34 [i] waiting for the control plane to become ready
2025-04-04 18:51:34 [i] saved kubeconfig as "/home/ec2-user/.kube/config"
2025-04-04 18:51:34 [i] no tasks
2025-04-04 18:51:34 [i] all EKS cluster resources for "my-cluster" have been created
2025-04-04 18:51:34 [i] nodegroup "my-nodes" has 2 node(s)
2025-04-04 18:51:34 [i] node "ip-192-168-30-181.ec2.internal" is ready
2025-04-04 18:51:34 [i] node "ip-192-168-38-32.ec2.internal" is ready
2025-04-04 18:51:34 [i] waiting for at least 2 node(s) to become ready in "my-nodes"
2025-04-04 18:51:34 [i] nodegroup "my-nodes" has 2 node(s)
2025-04-04 18:51:34 [i] node "ip-192-168-30-181.ec2.internal" is ready
2025-04-04 18:51:34 [i] node "ip-192-168-38-32.ec2.internal" is ready
2025-04-04 18:51:34 [i] created 1 managed nodegroup(s) in cluster "my-cluster"
2025-04-04 18:51:35 [i] kubectl command should work with "/home/ec2-user/.kube/config", try 'kubectl get nodes'
2025-04-04 18:51:35 [i] EKS cluster "my-cluster" in "us-east-1" region is ready
[ec2-user@ip-172-31-90-145 ~]$
```

# nginx-deployment.yaml

```
ec2-user@ip-172-31-90-145:~  
GNU nano 8.3 nginx-deployment.yaml Modified  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: nginx-deployment  
  labels:  
    app: nginx  
spec:  
  replicas: 2  
  selector:  
    matchLabels:  
      app: nginx  
  template:  
    metadata:  
      labels:  
        app: nginx  
    spec:  
      containers:  
      - name: nginx  
        image: nginx:latest  
        ports:  
        - containerPort: 80  
---  
apiVersion: v1  
kind: Service  
metadata:  
  name: nginx-service  
spec:  
  selector:  
    app: nginx  
  type: LoadBalancer # Ye LoadBalancer type ka hoga taaki bahar access ho sake  
  ports:  
  - protocol: TCP  
    port: 80  
    targetPort: 80  
|  
  
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line ^E Redo
```



# Check deployment and service

```
ec2-user@ip-172-31-90-145:~$ kubectl apply -f eksctl-my-cluster-nodegroup-my-nodes.yaml
2025-04-04 18:47:52 [i] deploying stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:47:52 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:48:22 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:49:04 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:49:47 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:51:34 [i] waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-my-nodes"
2025-04-04 18:51:34 [i] waiting for the control plane to become ready
2025-04-04 18:51:34 [✓] saved kubeconfig as "/home/ec2-user/.kube/config"
2025-04-04 18:51:34 [i] no tasks
2025-04-04 18:51:34 [✓] all EKS cluster resources for "my-cluster" have been created
2025-04-04 18:51:34 [i] nodegroup "my-nodes" has 2 node(s)
2025-04-04 18:51:34 [i] node "ip-192-168-30-181.ec2.internal" is ready
2025-04-04 18:51:34 [i] node "ip-192-168-38-32.ec2.internal" is ready
2025-04-04 18:51:34 [i] waiting for at least 2 node(s) to become ready in "my-nodes"
2025-04-04 18:51:34 [i] nodegroup "my-nodes" has 2 node(s)
2025-04-04 18:51:34 [i] node "ip-192-168-30-181.ec2.internal" is ready
2025-04-04 18:51:34 [i] node "ip-192-168-38-32.ec2.internal" is ready
2025-04-04 18:51:34 [✓] created 1 managed nodegroup(s) in cluster "my-cluster"
2025-04-04 18:51:35 [i] kubectl command should work with "/home/ec2-user/.kube/config", try 'kubectl
get nodes'
2025-04-04 18:51:35 [✓] EKS cluster "my-cluster" in "us-east-1" region is ready
[ec2-user@ip-172-31-90-145 ~]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE      VERSION
ip-192-168-30-181.ec2.internal      Ready    <none>    2m35s    v1.32.1-eks-5d632ec
ip-192-168-38-32.ec2.internal      Ready    <none>    2m35s    v1.32.1-eks-5d632ec
[ec2-user@ip-172-31-90-145 ~]$ kubectl cluster-info
Kubernetes control plane is running at https://6C693CF7492EB3E332706D812582E17A.gr7.us-east-1.eks.amaz
onaws.com
CoreDNS is running at https://6C693CF7492EB3E332706D812582E17A.gr7.us-east-1.eks.amazonaws.com/api/v1/
namespaces/kube-system/services/kube-dns/dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[ec2-user@ip-172-31-90-145 ~]$ nano nginx-deployment.yaml
[ec2-user@ip-172-31-90-145 ~]$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
service/nginx-service created
[ec2-user@ip-172-31-90-145 ~]$ kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
nginx-deployment-96b9d695-2qbwj     1/1      Running    0            9s
nginx-deployment-96b9d695-pqv7w     1/1      Running    0            9s
[ec2-user@ip-172-31-90-145 ~]$ kubectl get svc
NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP
kubernetes                          ClusterIP   10.100.0.1     <none>
nginx-service                       LoadBalancer 10.100.94.143 ae81c31dff91a49dea2ffa176ddc8b08-2135406246.us-east-1.e
lb.amazonaws.com 80:32557/TCP 9s
[ec2-user@ip-172-31-90-145 ~]$ kubectl get svc nginx-service
NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP
nginx-service                       LoadBalancer 10.100.94.143 ae81c31dff91a49dea2ffa176ddc8b08-2135406246.us-east-1.e
lb.amazonaws.com 80:32557/TCP 21s
[ec2-user@ip-172-31-90-145 ~]$
```

deployment successful

