

Deployment:

```
ubuntu@ip-172-31-33-4: ~/ek X Settings X + v
ubuntu@ip-172-31-33-4:~/eks-setup$ cat eks.yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: shaurya-cluster
  region: ap-south-1

nodeGroups:
  - name: ng-1
    instanceType: t2.micro
    desiredCapacity: 3
ubuntu@ip-172-31-33-4:~/eks-setup$ |
```

```
ubuntu@ip-172-31-33-4: ~/ek X Settings X + v
ubuntu@ip-172-31-33-4:~/eks-setup$ cat eks.yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: shaurya-cluster
  region: ap-south-1

nodeGroups:
  - name: ng-1
    instanceType: t2.micro
    desiredCapacity: 3
ubuntu@ip-172-31-33-4:~/eks-setup$ eksctl create cluster -f eks.yaml
2025-02-28 09:37:59 [I] eksctl version 0.204.0
2025-02-28 09:37:59 [I] using region ap-south-1
2025-02-28 09:37:59 [I] skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.micro
2025-02-28 09:37:59 [I] setting availability zones to [ap-south-1a ap-south-1b]
2025-02-28 09:37:59 [I] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2025-02-28 09:37:59 [I] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2025-02-28 09:37:59 [I] nodegroup "ng-1" will use "ami-0c5c573feb08cfa4d" [AmazonLinux2/1.30]
2025-02-28 09:37:59 [I] using Kubernetes version 1.30
2025-02-28 09:37:59 [I] creating EKS cluster "shaurya-cluster" in "ap-south-1" region with un-managed nodes
2025-02-28 09:37:59 [I] 1 nodegroup (ng-1) was included (based on the include/exclude rules)
2025-02-28 09:37:59 [I] will create a CloudFormation stack for cluster itself and 1 nodegroup stack(s)
2025-02-28 09:37:59 [I] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=shaurya-cluster'
2025-02-28 09:37:59 [I] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "shaurya-cluster" in "ap-south-1"
2025-02-28 09:37:59 [I] CloudWatch logging will not be enabled for cluster "shaurya-cluster" in "ap-south-1"
2025-02-28 09:37:59 [I] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=shaurya-cluster'
2025-02-28 09:37:59 [I] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-02-28 09:37:59 [I]
2 sequential tasks: { create cluster control plane "shaurya-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create nodegroup "ng-1",
  },
}
2025-02-28 09:37:59 [I] building cluster stack "eksctl-shaurya-cluster-cluster"
2025-02-28 09:37:59 [I] deploying stack "eksctl-shaurya-cluster-cluster"
2025-02-28 09:38:29 [I] waiting for CloudFormation stack "eksctl-shaurya-cluster-cluster"
2025-02-28 09:38:59 [I] waiting for CloudFormation stack "eksctl-shaurya-cluster-cluster"
```

Created The EKS Cluster:

```
ubuntu@ip-172-31-33-4: ~/ek X Settings X + v
ubuntu@ip-172-31-33-4:~/eks-setup$ eksctl get cluster --region ap-south-1
NAME          REGION  EKSCtl CREATED
shaurya-cluster ap-south-1 True
ubuntu@ip-172-31-33-4:~/eks-setup$ eksctl get nodegroup --cluster shaurya-cluster --region ap-south-1
CLUSTER      NODEGROUP  STATUS  CREATED          MIN SIZE  MAX SIZE  DESIRED CAPACITY  INSTANCE TYPE  IMAGE ID  ASG NAME                                     TYPE
shaurya-cluster ng-1      CREATE_COMPLETE  2025-02-28T09:48:02Z  3         3         3                  t2.micro       ami-0c5c573feb08cfa4d  eksctl-shaurya-cluster-nodegroup-ng-1-NodeGroup-1x1x7XQ51w33  unmanaged
ubuntu@ip-172-31-33-4:~/eks-setup$ |
```

The YAML file to be deployed:

```
ubuntu@ip-172-31-33-4: ~/ek X Settings X + v
ubuntu@ip-172-31-33-4:~/eks-setup$ cat shaurya-nginx.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  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
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
ubuntu@ip-172-31-33-4:~/eks-setup$
```

Deployed the pods on the Cluster:

```
ubuntu@ip-172-31-33-4: ~/ek X Settings X + v
ubuntu@ip-172-31-33-4:~/eks-setup$ kubectl apply -f shaurya-nginx.yaml
deployment.apps/nginx-deployment created
service/nginx-service created
ubuntu@ip-172-31-33-4:~/eks-setup$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-576c6b7b6-b9zds 1/1     Running   0           5s
pod/nginx-deployment-576c6b7b6-k4j1v 0/1     ContainerCreating   0           5s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kubernetes                  ClusterIP     10.100.0.1   <none>         443/TCP          22m
service/nginx-service               LoadBalancer 10.100.74.27 ab0c017a5615849da8668d59a85fe863-189230015.ap-south-1.elb.amazonaws.com 80:31550/TCP 5s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment    1/2     2             1           6s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-576c6b7b6 2         2         1       6s
ubuntu@ip-172-31-33-4:~/eks-setup$
```