Tasks To Be Performed:

1. Deploy a Kubernetes cluster for 3 nodes
2. Create a NGINX deployment of 3 replicas

1. Deploy a Kubernetes cluster for 3 nodes

```
EKS $eksctl create cluster \
  --name my-eks-cluster \
  --region ap-south-1 \
  --nodegroup-name worker-nodes \
  --node-type t3.medium \
  --nodes 3 \
  --nodes-min 3 \
  --nodes-max 3 \
  --managed
2025-10-13 08:01:32 [ℹ]  eksctl version 0.215.0
2025-10-13 08:01:32 [ℹ]  using region ap-south-1
2025-10-13 08:01:32 [ℹ]  setting availability zones to [ap-south-1b ap-south-1c ap-south-1a]
2025-10-13 08:01:32 [ℹ]  subnets for ap-south-1b - public:192.168.0.0/19 private:192.168.96.0/19
2025-10-13 08:01:32 [ℹ]  subnets for ap-south-1c - public:192.168.32.0/19 private:192.168.128.0/19
2025-10-13 08:01:32 [ℹ]  subnets for ap-south-1a - public:192.168.64.0/19 private:192.168.160.0/19
2025-10-13 08:01:32 [ℹ]  nodegroup "worker-nodes" will use "" [AmazonLinux2023/1.32]
2025-10-13 08:01:32 [!]  Auto Mode will be enabled by default in an upcoming release of eksctl. This means managed node groups and
longer be created by default. To maintain current behavior, explicitly set 'autoModeConfig.enabled: false' in your cluster configu
o/usage/auto-mode/
2025-10-13 08:01:32 [ℹ]  using Kubernetes version 1.32
2025-10-13 08:01:32 [ℹ]  creating EKS cluster "my-eks-cluster" in "ap-south-1" region with managed nodes
2025-10-13 08:01:32 [ℹ]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-10-13 08:01:32 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap
2025-10-13 08:01:32 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "
2025-10-13 08:01:32 [ℹ]  CloudWatch logging will not be enabled for cluster "my-eks-cluster" in "ap-south-1"
2025-10-13 08:01:32 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e
ster=my-eks-cluster'
2025-10-13 08:01:32 [ℹ]  default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS a
2025-10-13 08:01:32 [ℹ]
2 sequential tasks: { create cluster control plane "my-eks-cluster",
    2 sequential sub-tasks: {
        2 sequential sub-tasks: {
            1 task: { create addons },
            wait for control plane to become ready,
        }
```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or

```
2025-10-13 08:15:01 [ℹ]  node "ip-192-168-61-122.ap-south-1.compute.internal" is ready
2025-10-13 08:15:01 [ℹ]  node "ip-192-168-69-62.ap-south-1.compute.internal" is ready
2025-10-13 08:15:01 [✔]  created 1 managed nodegroup(s) in cluster "my-eks-cluster"
2025-10-13 08:15:01 [ℹ]  kubectl command should work with "/home/ec2-user/.kube/config", try 'kubectl get nodes'
2025-10-13 08:15:01 [✔]  EKS cluster "my-eks-cluster" in "ap-south-1" region is ready
EKS $kubectl get nodes
NAME                                           STATUS   ROLES    AGE     VERSION
ip-192-168-16-102.ap-south-1.compute.internal   Ready    <none>   2m48s   v1.32.9-eks-113cf36
ip-192-168-61-122.ap-south-1.compute.internal   Ready    <none>   2m50s   v1.32.9-eks-113cf36
ip-192-168-69-62.ap-south-1.compute.internal    Ready    <none>   2m51s   v1.32.9-eks-113cf36
EKS $
```

2. Create a NGINX deployment of 3 replicas

```
EKS $kubectl create deployment nginx --image=nginx --replicas=3
deployment.apps/nginx created
EKS $kubectl expose deployment nginx --type=LoadBalancer --port=80
service/nginx exposed
EKS $kubectl get deployment nginx
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   3/3     3            3           82s
EKS $
```