Task 4: Build a Production-Ready EKS Cluster With Private Nodes + Public ALB. With diagram.

Key Activities

· Create EKS cluster with private subnets only.

· Create public subnets only for ALB.

· Enable VPC CNI custom networking.

· Deploy:

  o ALB Ingress Controller

  o NGINX App using Ingress

· Ensure worker nodes have no public IP.

· Test access only through ALB.

➔

subnet
➔ public Subnet A/B ➔ ALB only
Private Subnet A/B ➔ EKS worker nodes

Create ->

1. **EKS control plane**: Private endpoint only
2.**Worker nodes**:
    1.In **private subnets**
    2.No public IP
3.**Public subnets**:
- Used **only by ALB**

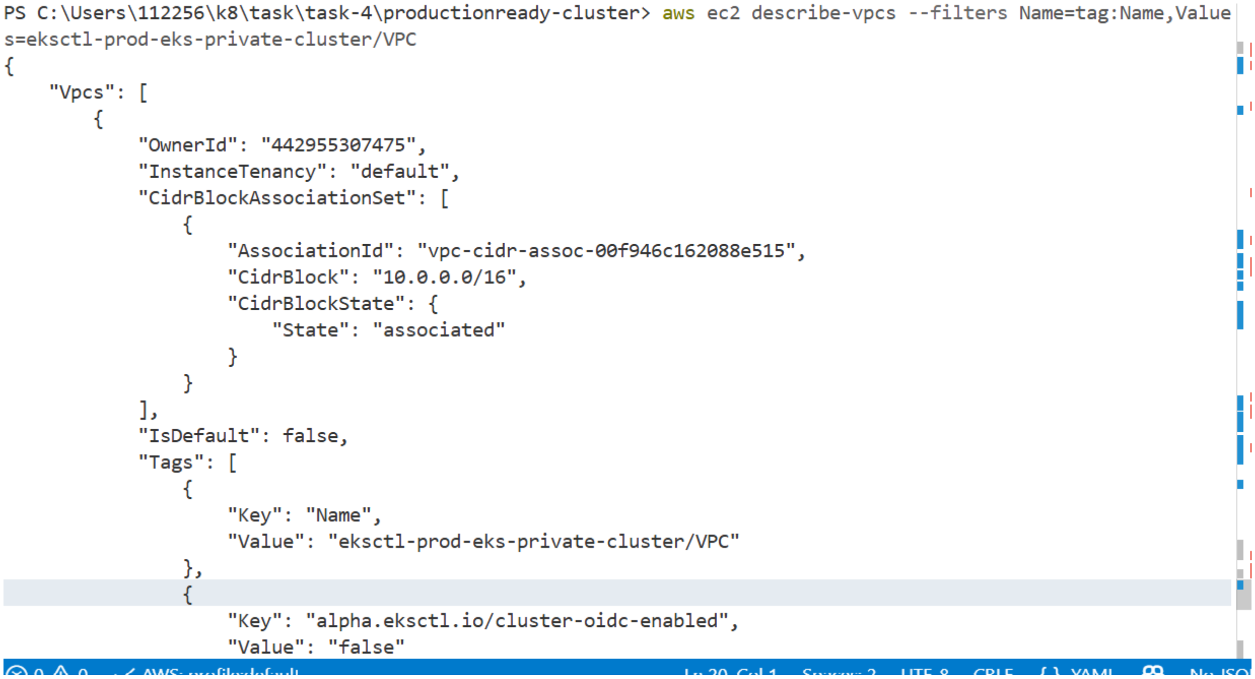
4.**Ingress**: AWS ALB Ingress Controller
5.**Application**: NGINX exposed **only via ALB**
6. **Networking**: VPC CNI Custom Networking enabled
7.**Access**: App reachable **only through ALB DNS**

Internet
↓
Public ALB (Public Subnets)
↓
Worker nodes (Private Subnets)
↓
Vpc

01-vpc.yaml ->

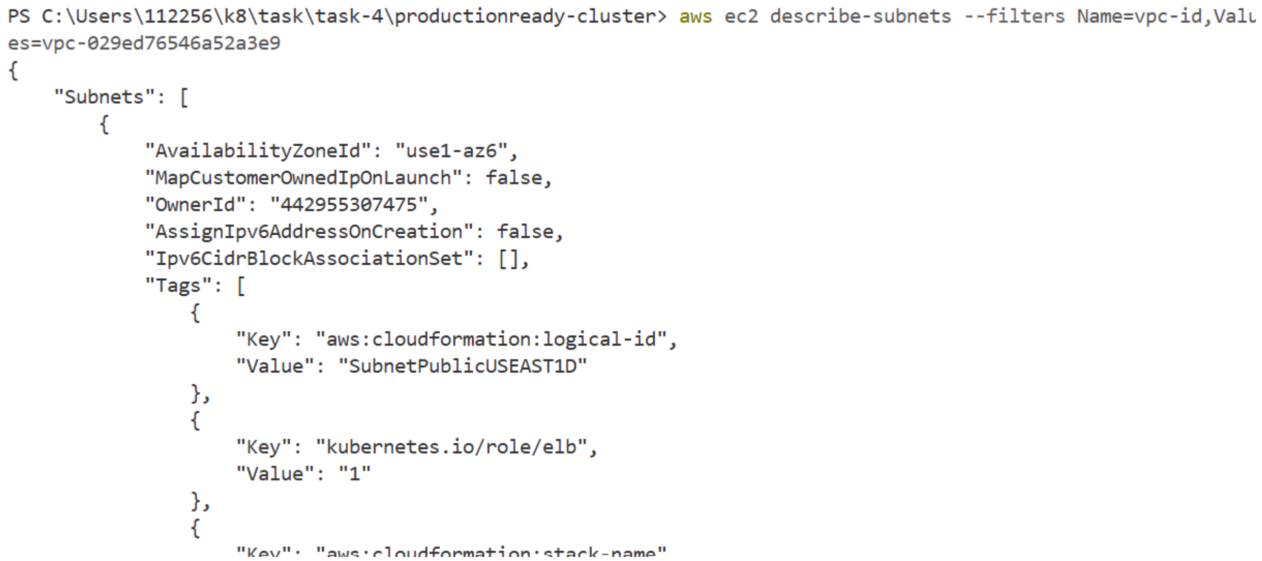
□ 1 VPC
□ 2 Public subnets (ALB)
□ 2 Private subnets (nodes)
□ 1 NAT Gateway

verify
aws ec2 describe-vpcs --filters Name=tag:Name,Values=eksctl-prod-eks-private-cluster/VPC



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> aws ec2 describe-vpcs --filters Name=tag:Name,Value
s=eksctl-prod-eks-private-cluster/VPC
{
    "Vpcs": [
        {
            "OwnerId": "442955307475",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-00f946c162088e515",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "eksctl-prod-eks-private-cluster/VPC"
                },
                {
                    "Key": "alpha.eksctl.io/cluster-oidc-enabled",
                    "Value": "false"
```

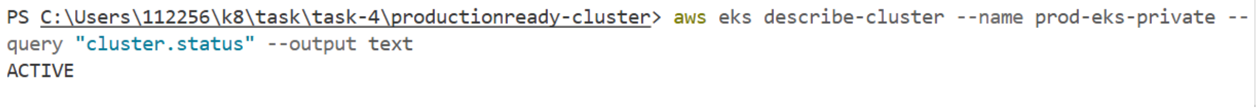
aws ec2 describe-subnets \
--filters Name=vpc-id,Values=<VPC-ID>



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> aws ec2 describe-subnets --filters Name=vpc-id,Valu
es=vpc-029ed76546a52a3e9
{
    "Subnets": [
        {
            "AvailabilityZoneId": "use1-az6",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "442955307475",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "Tags": [
                {
                    "Key": "aws:cloudformation:logical-id",
                    "Value": "SubnetPublicUSEAST1D"
                },
                {
                    "Key": "kubernetes.io/role/elb",
                    "Value": "1"
                },
                {
                    "Key": "aws:cloudformation:stack-name"
```

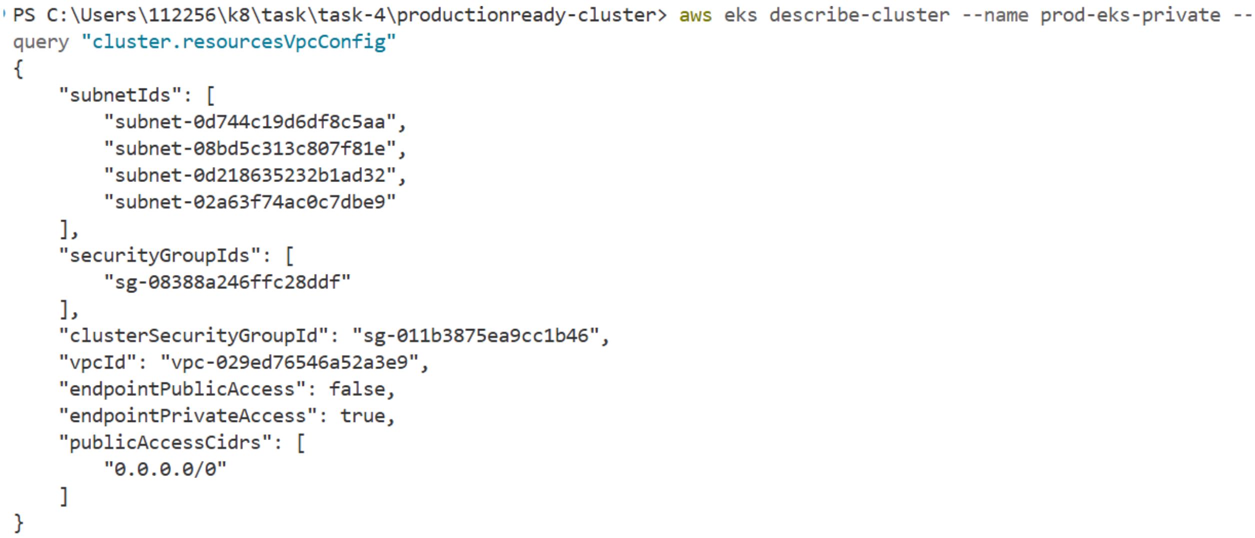
02-eks-cluster.yaml

aws eks describe-cluster --name prod-eks-private --query "cluster.status" --output text



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> aws eks describe-cluster --name prod-eks-private --
query "cluster.status" --output text
ACTIVE
```

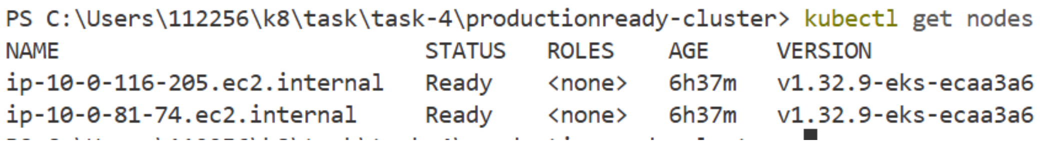
aws eks describe-cluster --name prod-eks-private --query "cluster.resourcesVpcConfig"



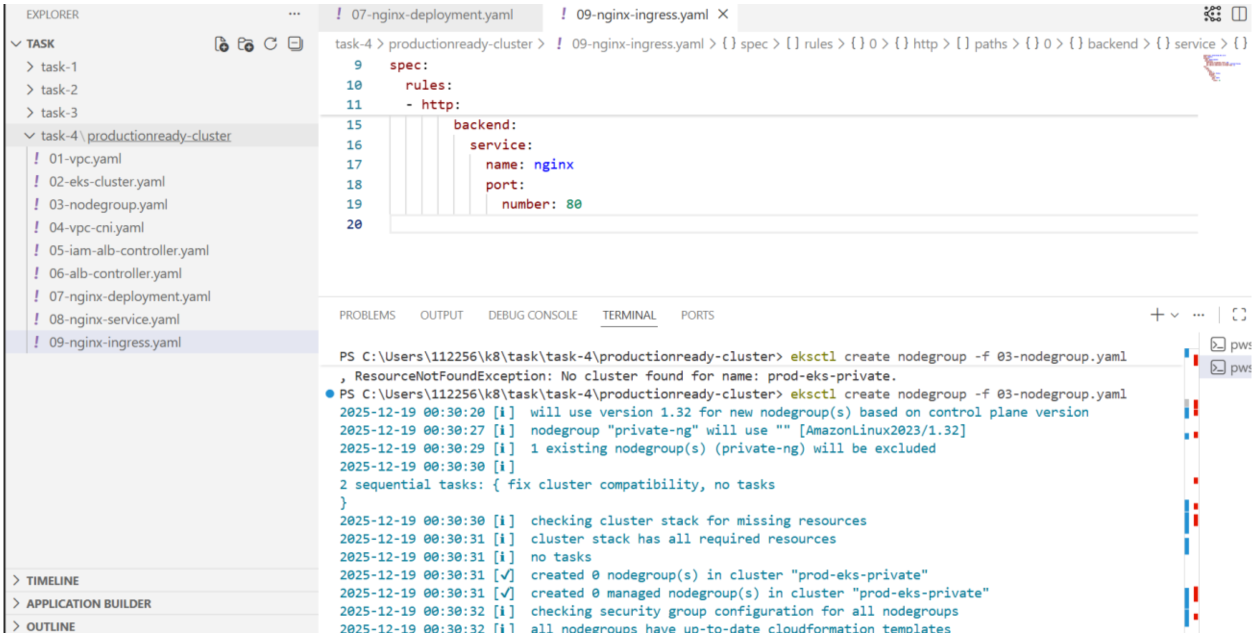
```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> aws eks describe-cluster --name prod-eks-private --
query "cluster.resourcesVpcConfig"
{
    "subnetIds": [
        "subnet-0d744c19d6df8c5aa",
        "subnet-08bd5c313c807f81e",
        "subnet-0d218635232b1ad32",
        "subnet-02a63f74ac0c7dbe9"
    ],
    "securityGroupIds": [
        "sg-08388a246ffc28ddf"
    ],
    "clusterSecurityGroupId": "sg-011b3875ea9cc1b46",
    "vpcId": "vpc-029ed76546a52a3e9",
    "endpointPublicAccess": false,
    "endpointPrivateAccess": true,
    "publicAccessCidrs": [
        "0.0.0.0/0"
    ]
}
```

03-nodegroup.yaml

kubectl get nodes



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get nodes
NAME                            STATUS   ROLES    AGE     VERSION
ip-10-0-116-205.ec2.internal    Ready    <none>   6h37m   v1.32.9-eks-ecaa3a6
ip-10-0-81-74.ec2.internal      Ready    <none>   6h37m   v1.32.9-eks-ecaa3a6
```

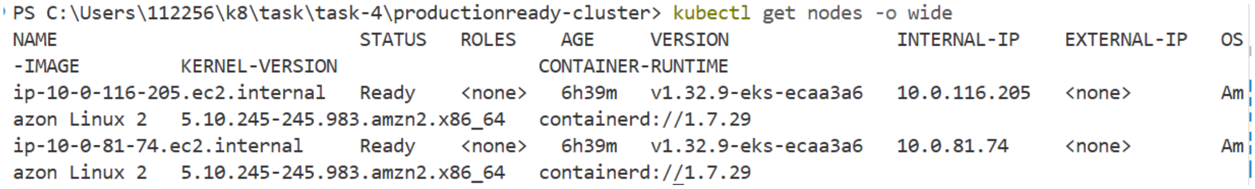


```
spec:
  rules:
  - http:
          backend:
            service:
              name: nginx
              port:
                number: 80

```

```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> eksctl create nodegroup -f 03-nodegroup.yaml
, ResourceNotFoundException: No cluster found for name: prod-eks-private.
PS C:\Users\112256\k8\task\task-4\productionready-cluster> eksctl create nodegroup -f 03-nodegroup.yaml
2025-12-19 00:30:20 [ℹ]  will use version 1.32 for new nodegroup(s) based on control plane version
2025-12-19 00:30:27 [ℹ]  nodegroup "private-ng" will use "" [AmazonLinux2023/1.32]
2025-12-19 00:30:29 [ℹ]  1 existing nodegroup(s) (private-ng) will be excluded
2025-12-19 00:30:30 [ℹ]
2 sequential tasks: { fix cluster compatibility, no tasks
}
2025-12-19 00:30:30 [ℹ]  checking cluster stack for missing resources
2025-12-19 00:30:31 [ℹ]  cluster stack has all required resources
2025-12-19 00:30:31 [ℹ]  no tasks
2025-12-19 00:30:31 [✓]  created 0 nodegroup(s) in cluster "prod-eks-private"
2025-12-19 00:30:31 [✓]  created 0 managed nodegroup(s) in cluster "prod-eks-private"
2025-12-19 00:30:32 [ℹ]  checking security group configuration for all nodegroups
2025-12-19 00:30:32 [ℹ]  all nodegroups have up-to-date cloudformation templates
```

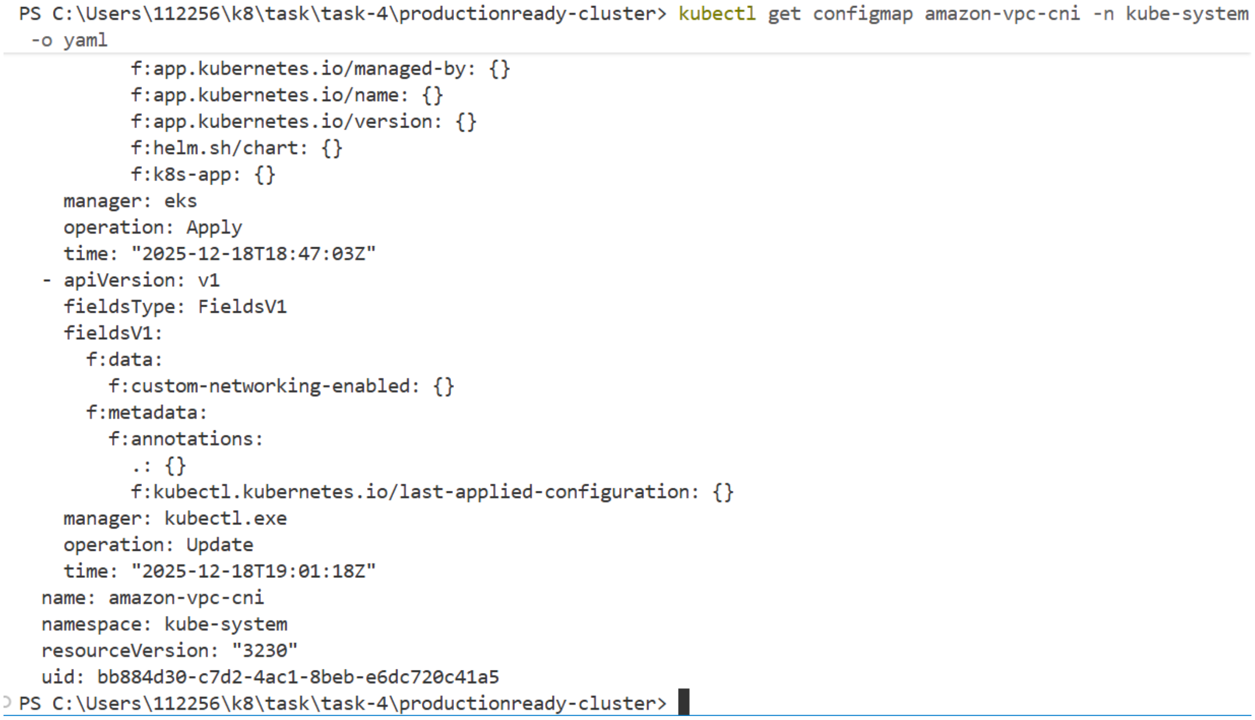
kubectl get nodes -o wide



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get nodes -o wide
NAME                          STATUS   ROLES    AGE     VERSION               INTERNAL-IP    EXTERNAL-IP   OS
-IMAGE         KERNEL-VERSION                   CONTAINER-RUNTIME
ip-10-0-116-205.ec2.internal   Ready    <none>   6h39m   v1.32.9-eks-ecaa3a6   10.0.116.205   <none>        Am
azon Linux 2   5.10.245-245.983.amzn2.x86_64   containerd://1.7.29
ip-10-0-81-74.ec2.internal     Ready    <none>   6h39m   v1.32.9-eks-ecaa3a6   10.0.81.74     <none>        Am
azon Linux 2   5.10.245-245.983.amzn2.x86_64   containerd://1.7.29
```

04-vpc-cni.yaml — Verify Custom Networking

kubectl get configmap amazon-vpc-cni -n kube-system -o yaml

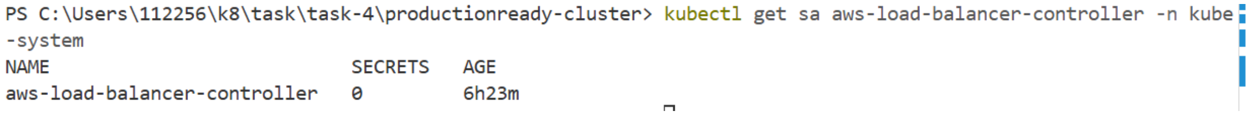


```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get configmap amazon-vpc-cni -n kube-system
 -o yaml
          f:app.kubernetes.io/managed-by: {}
          f:app.kubernetes.io/name: {}
          f:app.kubernetes.io/version: {}
          f:helm.sh/chart: {}
          f:k8s-app: {}
    manager: eks
    operation: Apply
    time: "2025-12-18T18:47:03Z"
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:data:
        f:custom-networking-enabled: {}
      f:metadata:
        f:annotations:
          .: {}
          f:kubectl.kubernetes.io/last-applied-configuration: {}
    manager: kubectl.exe
    operation: Update
    time: "2025-12-18T19:01:18Z"
  name: amazon-vpc-cni
  namespace: kube-system
  resourceVersion: "3230"
  uid: bb884d30-c7d2-4ac1-8beb-e6dc720c41a5
PS C:\Users\112256\k8\task\task-4\productionready-cluster>
```

05-iam-alb-controller.yaml — Verify IAM Service Account

Verify ServiceAccount

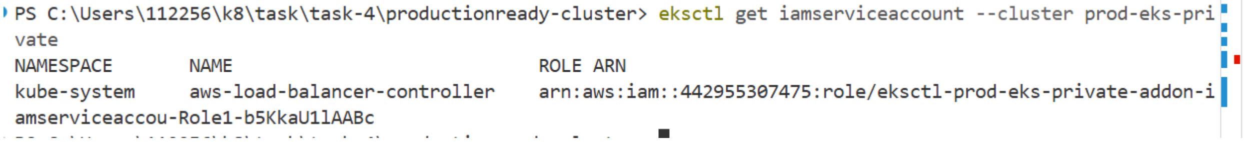
kubectl get sa aws-load-balancer-controller -n kube-system



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get sa aws-load-balancer-controller -n kube
-system
NAME                           SECRETS   AGE
aws-load-balancer-controller   0         6h23m
```

Verify IAM role attached

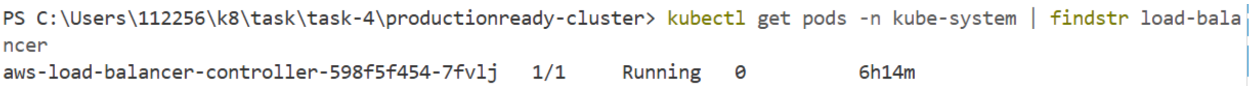
eksctl get iamserviceaccount --cluster prod-eks-private



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> eksctl get iamserviceaccount --cluster prod-eks-pri
vate
NAMESPACE       NAME                            ROLE ARN
kube-system     aws-load-balancer-controller    arn:aws:iam::442955307475:role/eksctl-prod-eks-private-addon-i
amserviceaccou-Role1-b5KkaU1lAABc
```

06-alb-controller.yaml — Verify ALB Controller

kubectl get pods -n kube-system | findstr load-balancer

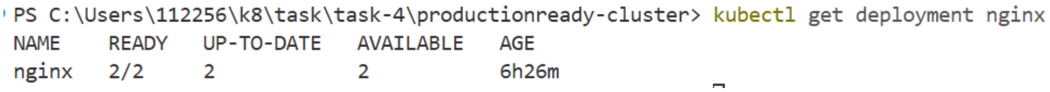


```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get pods -n kube-system | findstr load-bala
ncer
aws-load-balancer-controller-598f5f454-7fvlj   1/1     Running   0          6h14m
```

07-nginx-deployment.yaml — Verify App Pods

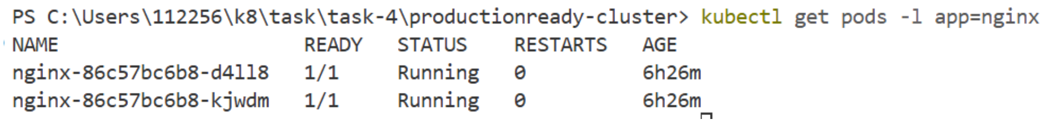
Verify deployment

kubectl get deployment nginx



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get deployment nginx
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   2/2     2            2           6h26m
```

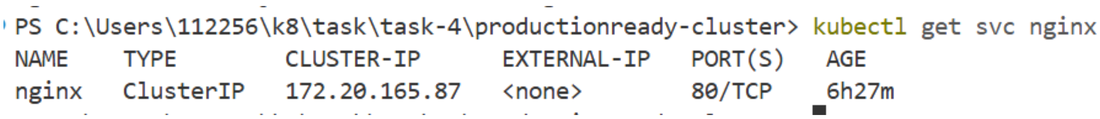
Verify pods -
kubectl get pods -l app=nginx



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get pods -l app=nginx
NAME                     READY   STATUS    RESTARTS   AGE
nginx-86c57bc6b8-d4ll8   1/1     Running   0          6h26m
nginx-86c57bc6b8-kjwdm   1/1     Running   0          6h26m
```

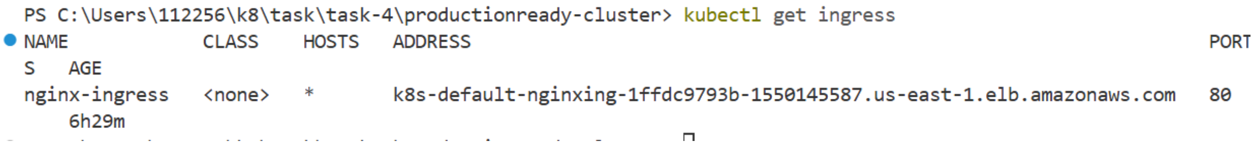
8-nginx-service.yaml
Verify Service

kubectl get svc nginx



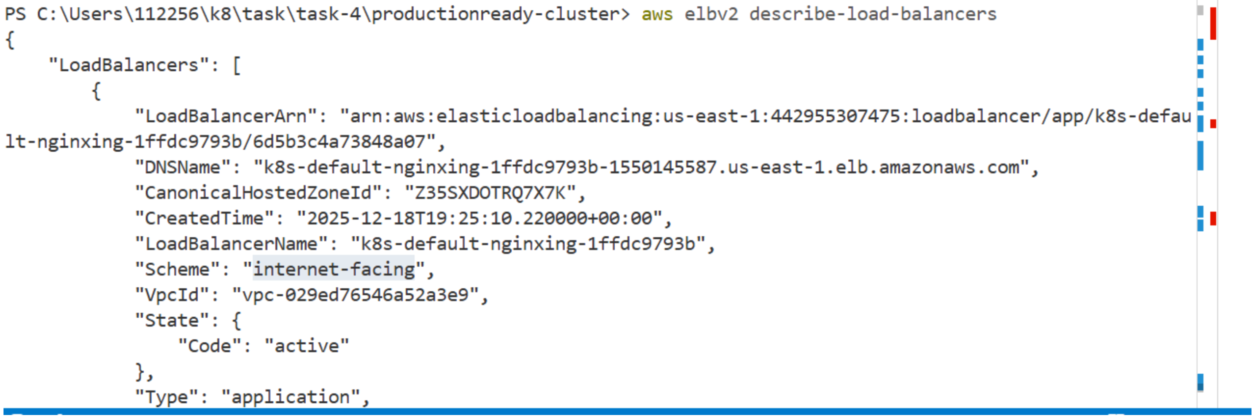
```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get svc nginx
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx   ClusterIP   172.20.165.87   <none>        80/TCP    6h27m
```

09-nginx-ingress.yaml — Verify ALB Creation
kubectl get ingress



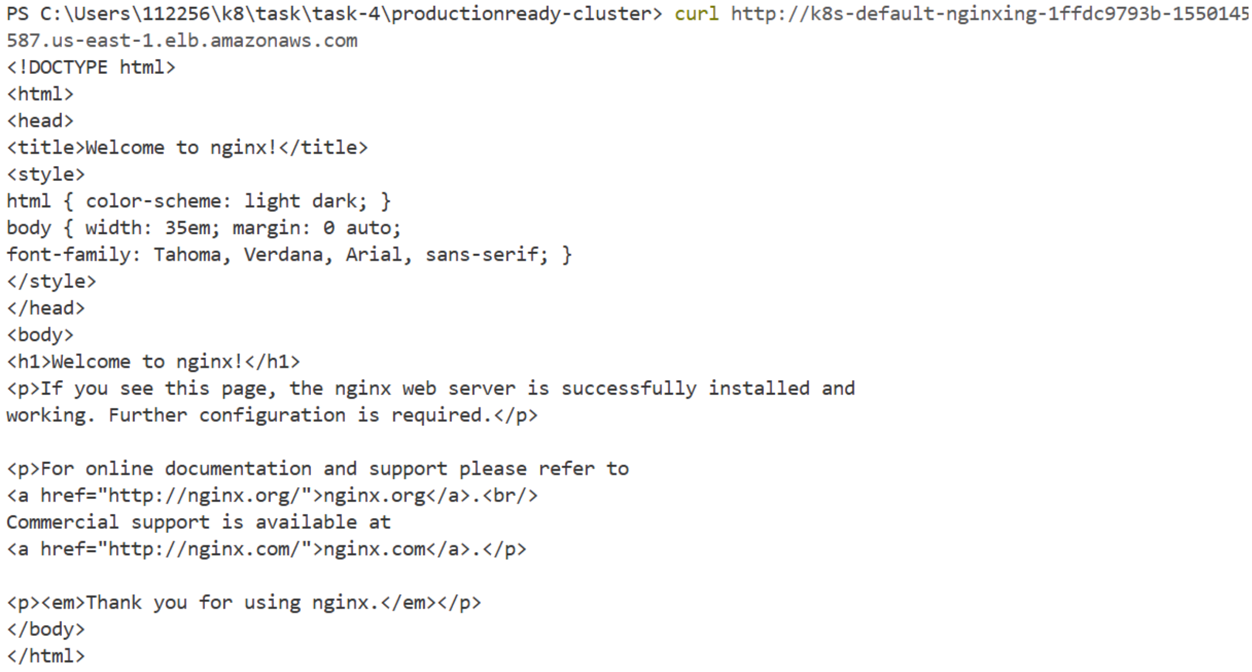
```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> kubectl get ingress
NAME            CLASS    HOSTS   ADDRESS                                                                  PORT
S   AGE
nginx-ingress   <none>   *       k8s-default-nginxing-1ffdc9793b-1550145587.us-east-1.elb.amazonaws.com   80
    6h29m
```

Verify ALB in AWS



```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> aws elbv2 describe-load-balancers
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:442955307475:loadbalancer/app/k8s-defau
lt-nginxing-1ffdc9793b/6d5b3c4a73848a07",
            "DNSName": "k8s-default-nginxing-1ffdc9793b-1550145587.us-east-1.elb.amazonaws.com",
            "CanonicalHostedZoneId": "Z35SXDOTRQ7X7K",
            "CreatedTime": "2025-12-18T19:25:10.220000+00:00",
            "LoadBalancerName": "k8s-default-nginxing-1ffdc9793b",
            "Scheme": "internet-facing",
            "VpcId": "vpc-029ed76546a52a3e9",
            "State": {
                "Code": "active"
            },
            "Type": "application",
```

ALB access

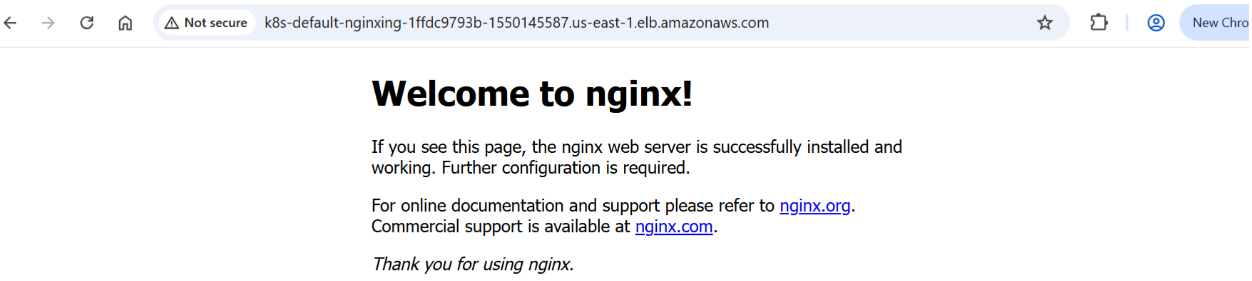


```
PS C:\Users\112256\k8\task\task-4\productionready-cluster> curl http://k8s-default-nginxing-1ffdc9793b-1550145
587.us-east-1.elb.amazonaws.com
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

http://k8s-default-nginxing-1ffdc9793b-1550145587.us-east-1.elb.amazonaws.com/


Not secure
k8s-default-nginxing-1ffdc9793b-1550145587.us-east-1.elb.amazonaws.com
New Chro

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.
Commercial support is available at nginx.com.

Thank you for using nginx.