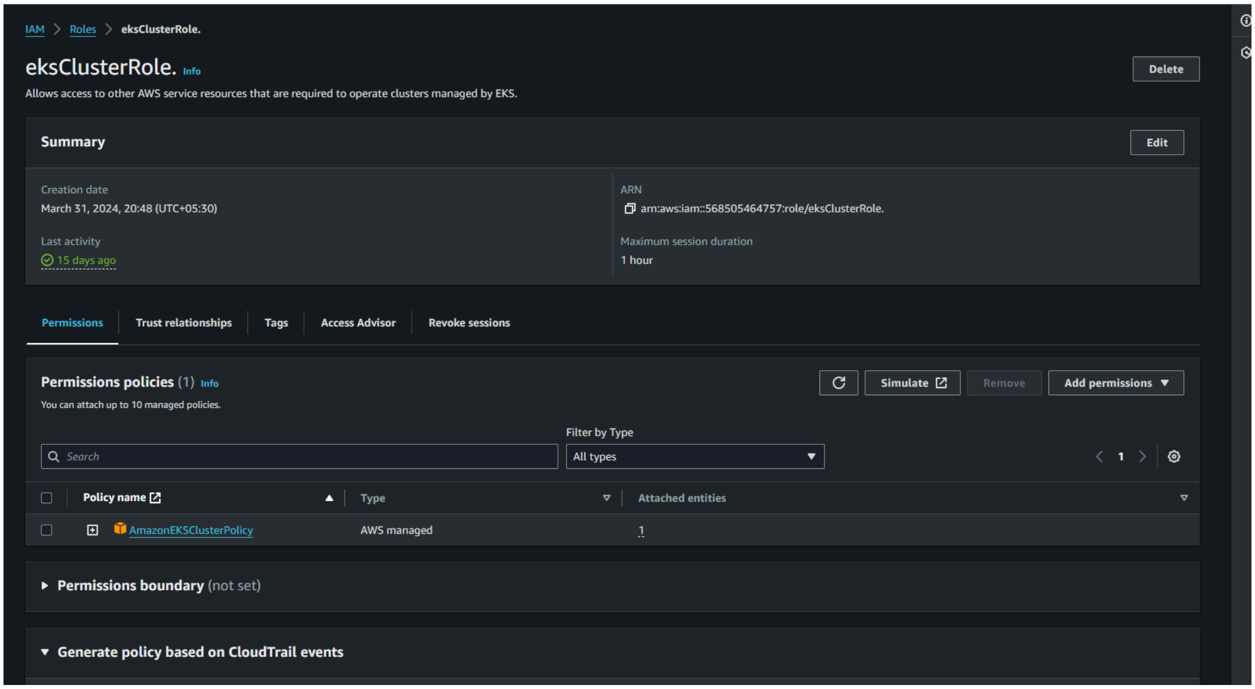
Creating Cluster role



IAM > Roles > eksClusterRole.

eksClusterRole. Info

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

Summary

Creation date
March 31, 2024, 20:48 (UTC+05:30)

ARN
arn:aws:iam::568505464757:role/eksClusterRole.

Last activity
15 days ago

Maximum session duration
1 hour

Permissions policies (1) Info

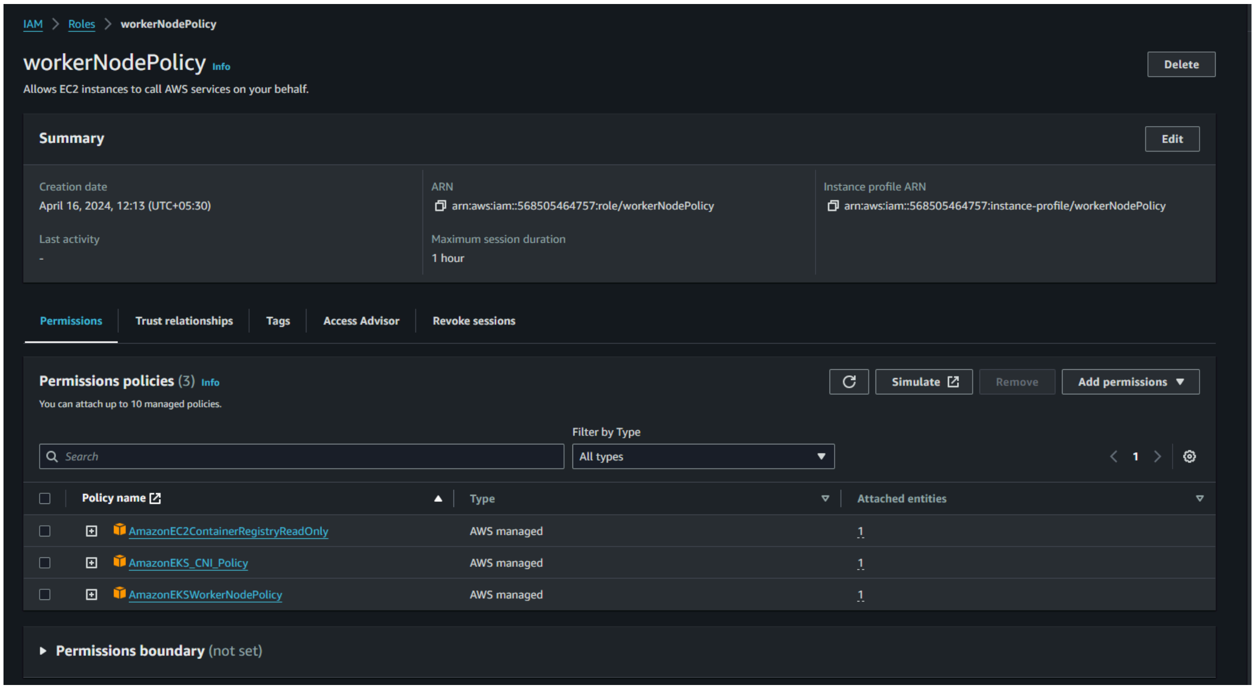
You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEKSClusterPolicy	AWS managed	1

▸ **Permissions boundary** (not set)

▾ **Generate policy based on CloudTrail events**

Creating worker node policy



IAM > Roles > workerNodePolicy

workerNodePolicy Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
April 16, 2024, 12:13 (UTC+05:30)

ARN
arn:aws:iam::568505464757:role/workerNodePolicy

Instance profile ARN
arn:aws:iam::568505464757:instance-profile/workerNodePolicy

Last activity
-

Maximum session duration
1 hour

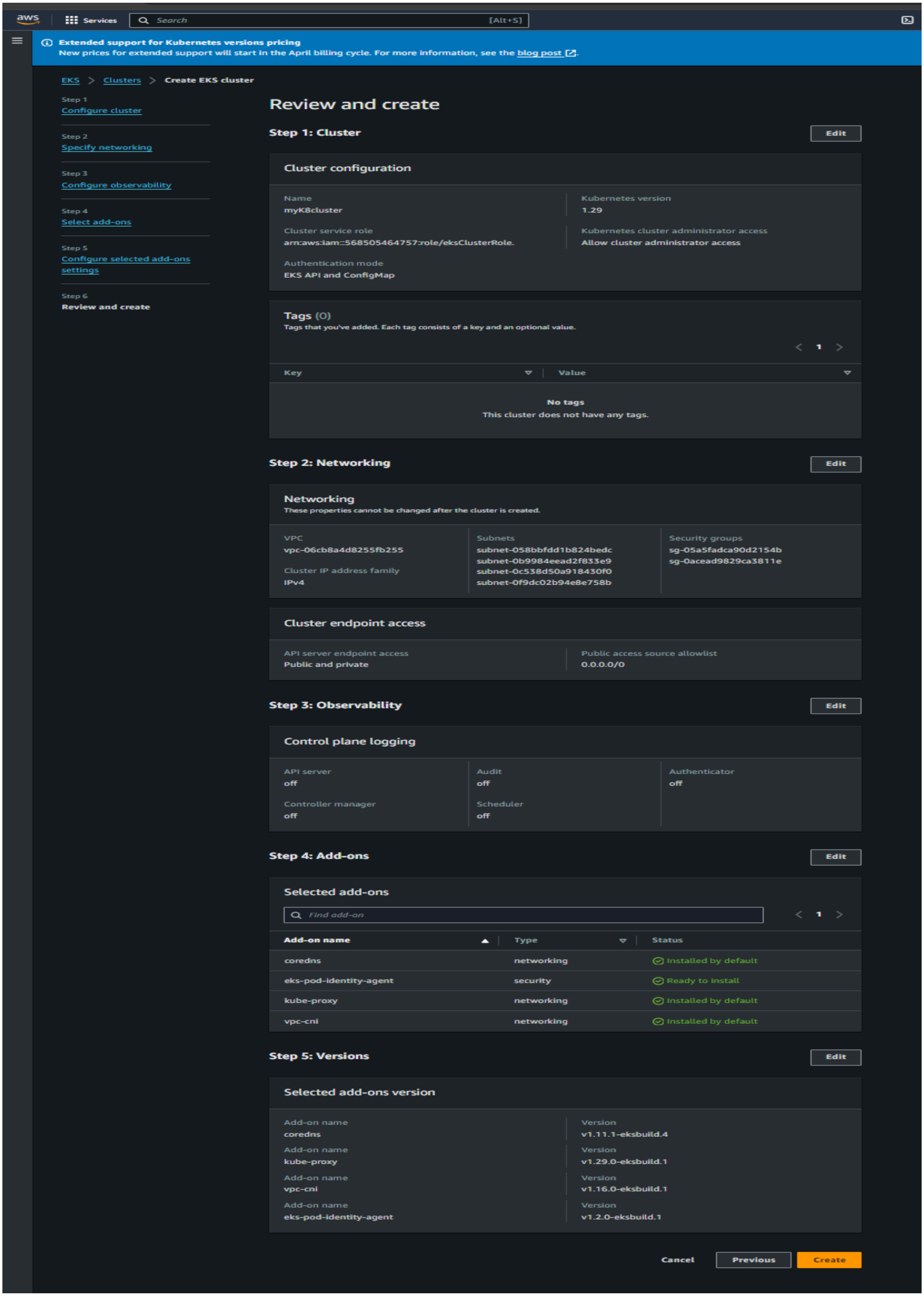
Permissions policies (3) Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryReadOnly	AWS managed	1
AmazonEKS_CNI_Policy	AWS managed	1
AmazonEKSWorkerNodePolicy	AWS managed	1

▸ **Permissions boundary** (not set)

Creating AWS EKS cluster



Extended support for Kubernetes versions pricing
New prices for extended support will start in the April billing cycle. For more information, see the blog post.

EKS > Clusters > Create EKS cluster

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure observability

Step 4
Select add-ons

Step 5
Configure selected add-ons settings

Step 6
Review and create

Review and create

Step 1: Cluster

Cluster configuration

Name
myK8cluster

Kubernetes version
1.29

Cluster service role
arn:aws:iam::568505464757:role/eksClusterRole.

Kubernetes cluster administrator access
Allow cluster administrator access

Authentication mode
EKS API and ConfigMap

Tags (0)
Tags that you've added. Each tag consists of a key and an optional value.

Key	Value

No tags
This cluster does not have any tags.

Step 2: Networking

Networking
These properties cannot be changed after the cluster is created.

VPC
vpc-06cb8a4d8255fb255

Cluster IP address family
IPv4

Subnets
subnet-058bbfdd1b824bedc
subnet-0b9984eead2f833e9
subnet-0c538d50a918430f0
subnet-0f9dc02b94e8e758b

Security groups
sg-05a5fadca90d2154b
sg-0acead9829ca3811e

Cluster endpoint access

API server endpoint access
Public and private

Public access source allowlist
0.0.0.0/0

Step 3: Observability

Control plane logging

API server
off

Controller manager
off

Audit
off

Scheduler
off

Authenticator
off

Step 4: Add-ons

Selected add-ons

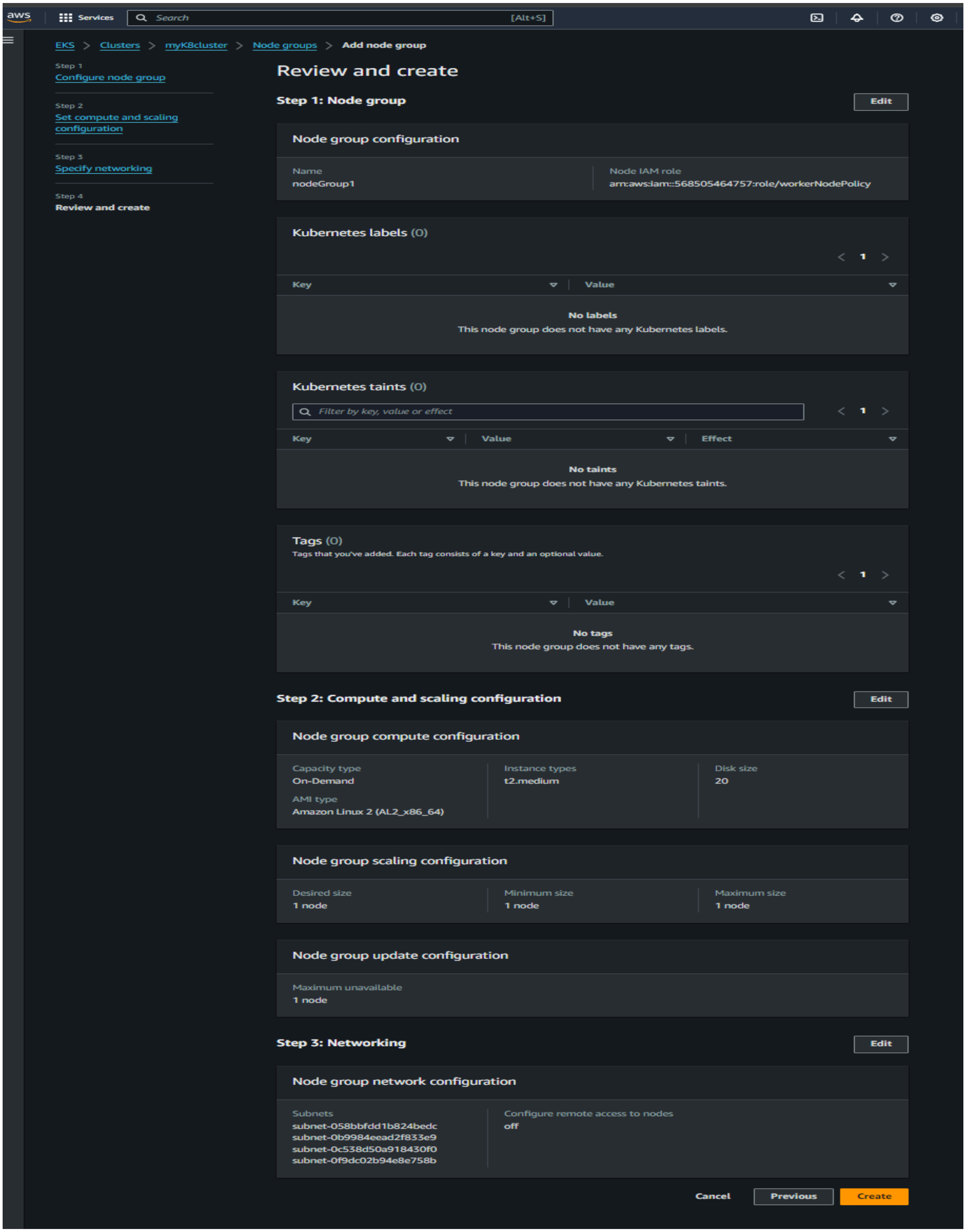
Add-on name	Type	Status
coredns	networking	✓ Installed by default
eks-pod-identity-agent	security	✓ Ready to install
kube-proxy	networking	✓ Installed by default
vpc-cni	networking	✓ Installed by default

Step 5: Versions

Selected add-ons version

Add-on name	Version
coredns	v1.11.1-eksbuild.4
kube-proxy	v1.29.0-eksbuild.1
vpc-cni	v1.16.0-eksbuild.1
eks-pod-identity-agent	v1.2.0-eksbuild.1

Create worker node group



Review and create

Step 1: Node group

Node group configuration

Name
nodeGroup1

Node IAM role
arn:aws:iam::568505464757:role/workerNodePolicy

Kubernetes labels (0)

Key	Value

No labels
This node group does not have any Kubernetes labels.

Kubernetes taints (0)

Filter by key, value or effect

Key	Value	Effect

No taints
This node group does not have any Kubernetes taints.

Tags (0)
Tags that you've added. Each tag consists of a key and an optional value.

Key	Value

No tags
This node group does not have any tags.

Step 2: Compute and scaling configuration

Node group compute configuration

Capacity type
On-Demand

AMI type
Amazon Linux 2 (AL2_x86_64)

Instance types
t2.medium

Disk size
20

Node group scaling configuration

Desired size
1 node

Minimum size
1 node

Maximum size
1 node

Node group update configuration

Maximum unavailable
1 node

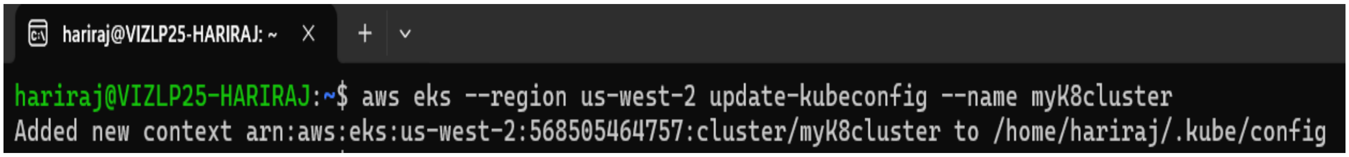
Step 3: Networking

Node group network configuration

Subnets
subnet-058bbfdd1b824bedc
subnet-0b9984eead2f833e9
subnet-0c538d50a918430f0
subnet-0f9dc02b94e8e758b

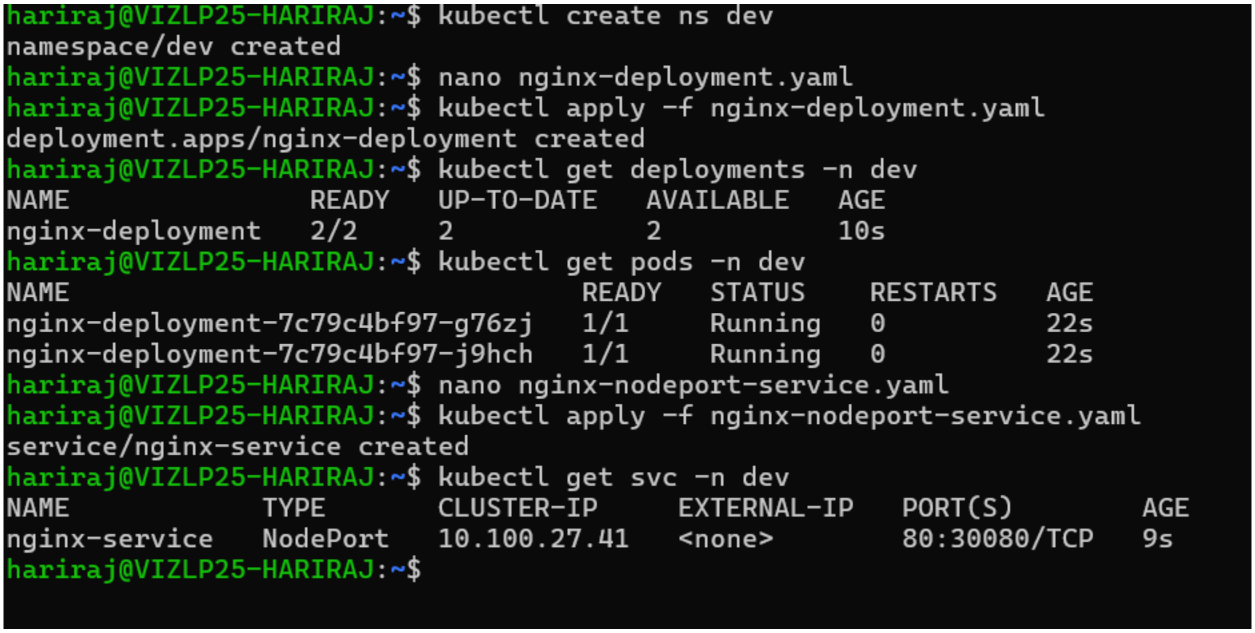
Configure remote access to nodes
off

Connecting cluster through AWS CLI



```
hariraj@VIZLP25-HARIRAJ:~$ aws eks --region us-west-2 update-kubeconfig --name myK8cluster
Added new context arn:aws:eks:us-west-2:568505464757:cluster/myK8cluster to /home/hariraj/.kube/config
```

Nginx deployment



```
hariraj@VIZLP25-HARIRAJ:~$ kubectl create ns dev
namespace/dev created
hariraj@VIZLP25-HARIRAJ:~$ nano nginx-deployment.yaml
hariraj@VIZLP25-HARIRAJ:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
hariraj@VIZLP25-HARIRAJ:~$ kubectl get deployments -n dev
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           10s
hariraj@VIZLP25-HARIRAJ:~$ kubectl get pods -n dev
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-7c79c4bf97-g76zj   1/1     Running   0          22s
nginx-deployment-7c79c4bf97-j9hch   1/1     Running   0          22s
hariraj@VIZLP25-HARIRAJ:~$ nano nginx-nodeport-service.yaml
hariraj@VIZLP25-HARIRAJ:~$ kubectl apply -f nginx-nodeport-service.yaml
service/nginx-service created
hariraj@VIZLP25-HARIRAJ:~$ kubectl get svc -n dev
NAME            TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
nginx-service   NodePort   10.100.27.41   <none>        80:30080/TCP   9s
hariraj@VIZLP25-HARIRAJ:~$
```

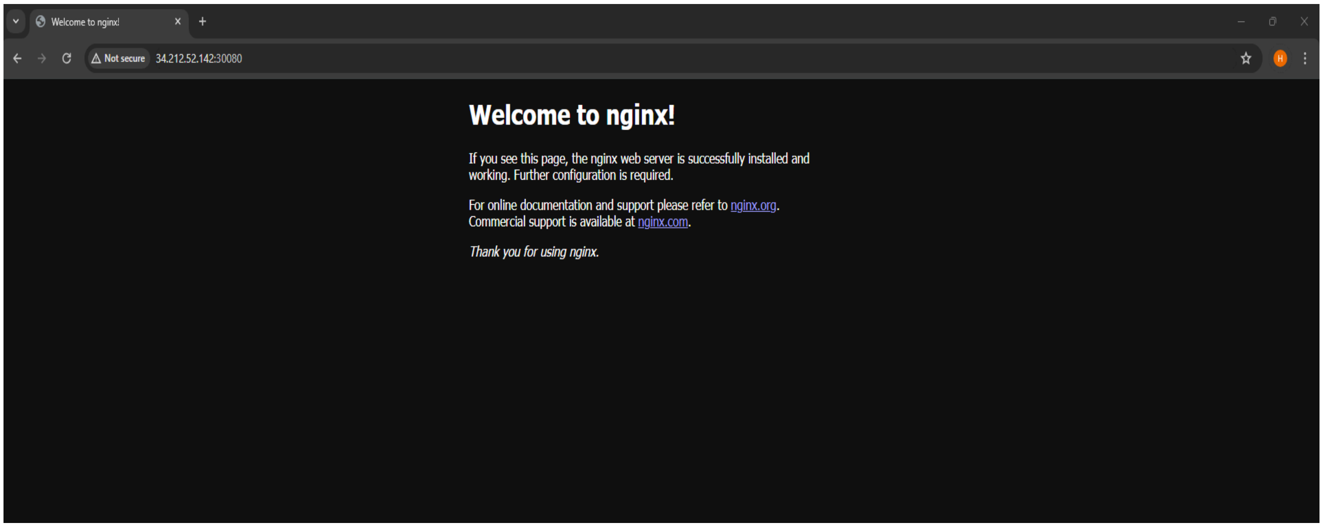
nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: dev
spec:
  replicas: 2  # Number of replicas
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

nginx-nodeport-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: dev
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080  # Optional: specify a nodePort or let Kubernetes assign one
```

Output



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.