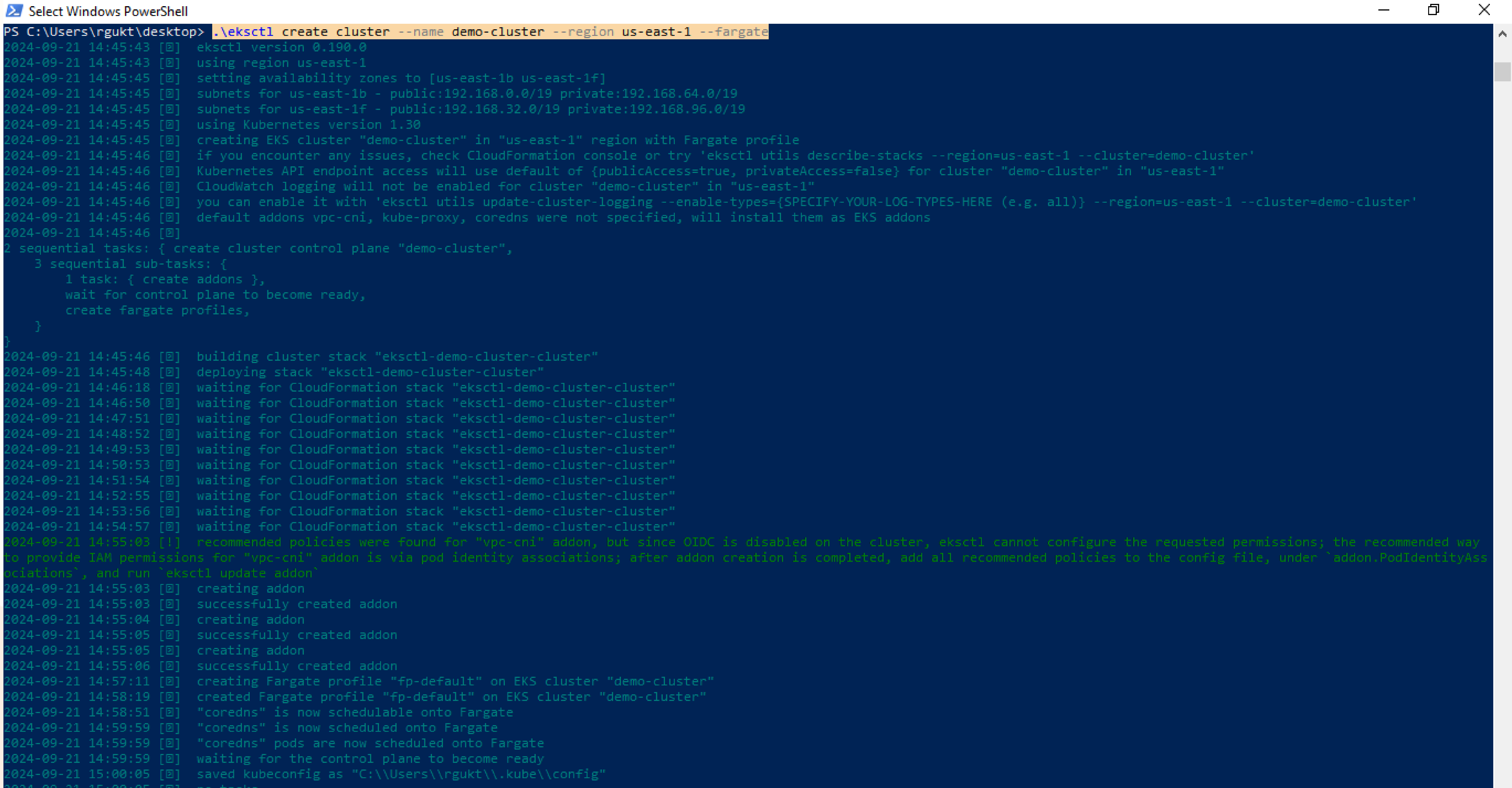
Title : Creating an EKS cluster and deploying an application using Terraform

step-1

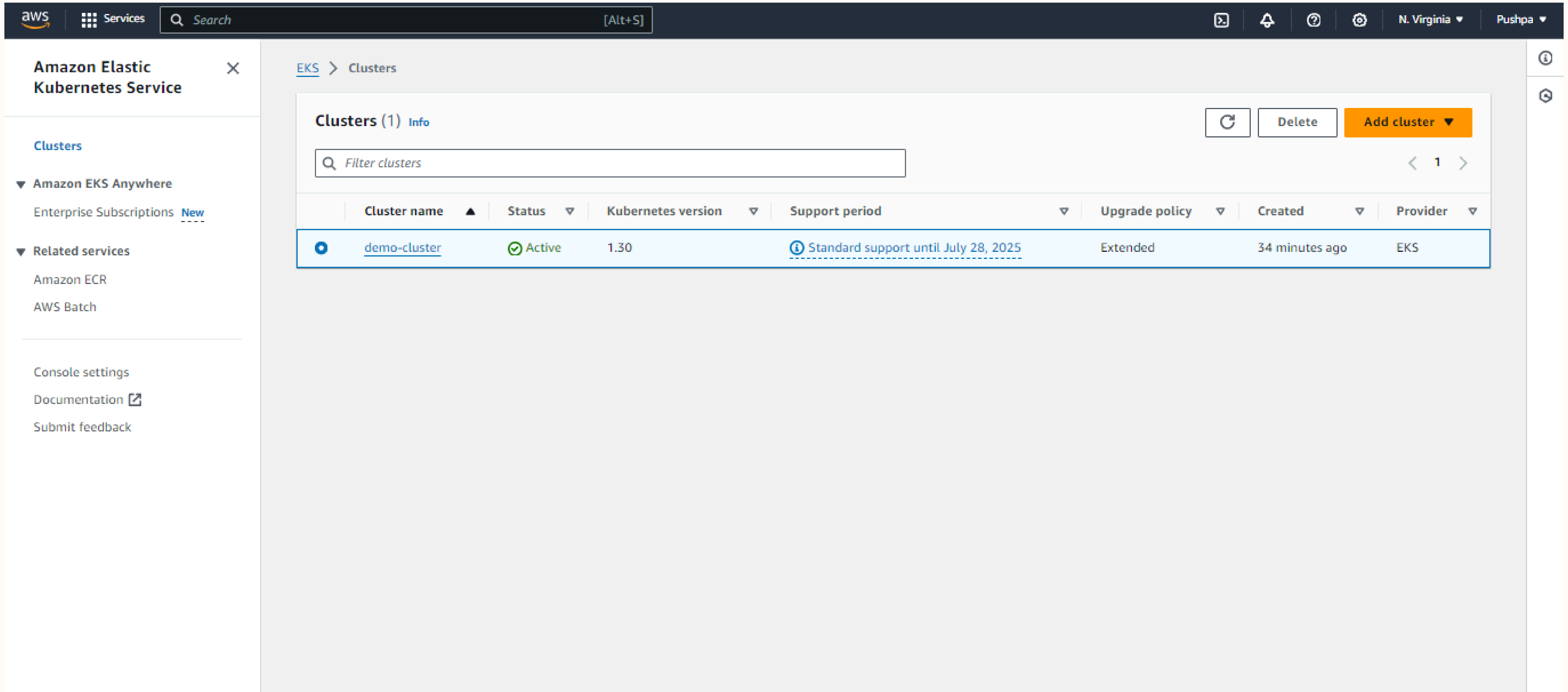
we have to configure to the aws console by using access key and secret access key

step-2 creating EKS cluster



```
PS C:\Users\rgukt\desktop> .\eksctl create cluster --name demo-cluster --region us-east-1 --fargate
2024-09-21 14:45:43 [⍰]  eksctl version 0.190.0
2024-09-21 14:45:43 [⍰]  using region us-east-1
2024-09-21 14:45:45 [⍰]  setting availability zones to [us-east-1b us-east-1f]
2024-09-21 14:45:45 [⍰]  subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
2024-09-21 14:45:45 [⍰]  subnets for us-east-1f - public:192.168.32.0/19 private:192.168.96.0/19
2024-09-21 14:45:45 [⍰]  using Kubernetes version 1.30
2024-09-21 14:45:45 [⍰]  creating EKS cluster "demo-cluster" in "us-east-1" region with Fargate profile
2024-09-21 14:45:46 [⍰]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=demo-cluster'
2024-09-21 14:45:46 [⍰]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "demo-cluster" in "us-east-1"
2024-09-21 14:45:46 [⍰]  CloudWatch logging will not be enabled for cluster "demo-cluster" in "us-east-1"
2024-09-21 14:45:46 [⍰]  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=demo-cluster'
2024-09-21 14:45:46 [⍰]  default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2024-09-21 14:45:46 [⍰]
2 sequential tasks: { create cluster control plane "demo-cluster",
    3 sequential sub-tasks: {
        1 task: { create addons },
        wait for control plane to become ready,
        create fargate profiles,
    }
}
2024-09-21 14:45:46 [⍰]  building cluster stack "eksctl-demo-cluster-cluster"
2024-09-21 14:45:48 [⍰]  deploying stack "eksctl-demo-cluster-cluster"
2024-09-21 14:46:18 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:46:50 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:47:51 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:48:52 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:49:53 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:50:53 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:51:54 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:52:55 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:53:56 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:54:57 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-09-21 14:55:03 [!]  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.PodIdentityAss
ociations`, and run `eksctl update addon`
2024-09-21 14:55:03 [⍰]  creating addon
2024-09-21 14:55:03 [⍰]  successfully created addon
2024-09-21 14:55:04 [⍰]  creating addon
2024-09-21 14:55:05 [⍰]  successfully created addon
2024-09-21 14:55:05 [⍰]  creating addon
2024-09-21 14:55:06 [⍰]  successfully created addon
2024-09-21 14:57:11 [⍰]  creating Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-09-21 14:58:19 [⍰]  created Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-09-21 14:58:51 [⍰]  "coredns" is now schedulable onto Fargate
2024-09-21 14:59:59 [⍰]  "coredns" is now scheduled onto Fargate
2024-09-21 14:59:59 [⍰]  "coredns" pods are now scheduled onto Fargate
2024-09-21 14:59:59 [⍰]  waiting for the control plane to become ready
2024-09-21 15:00:05 [⍰]  saved kubeconfig as "C:\\Users\\rgukt\\.kube\\config"
```

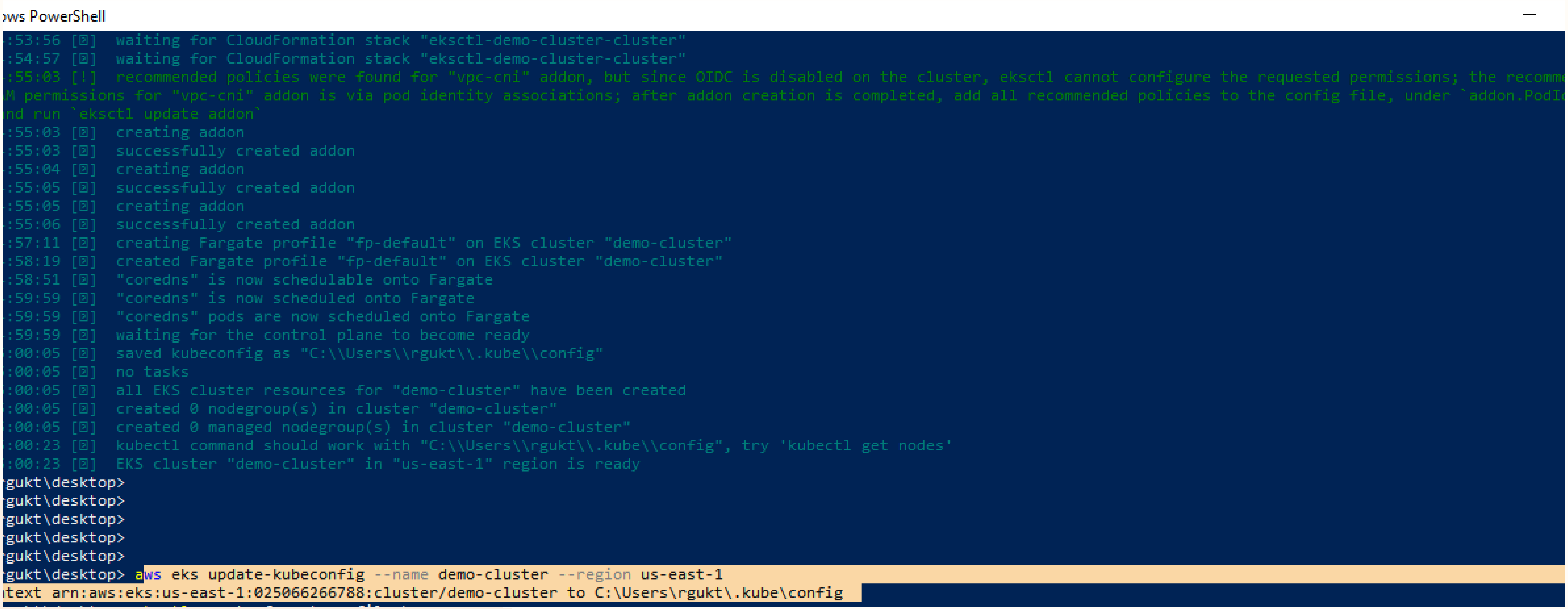
This is the EKS cluster we created



Clusters (1)

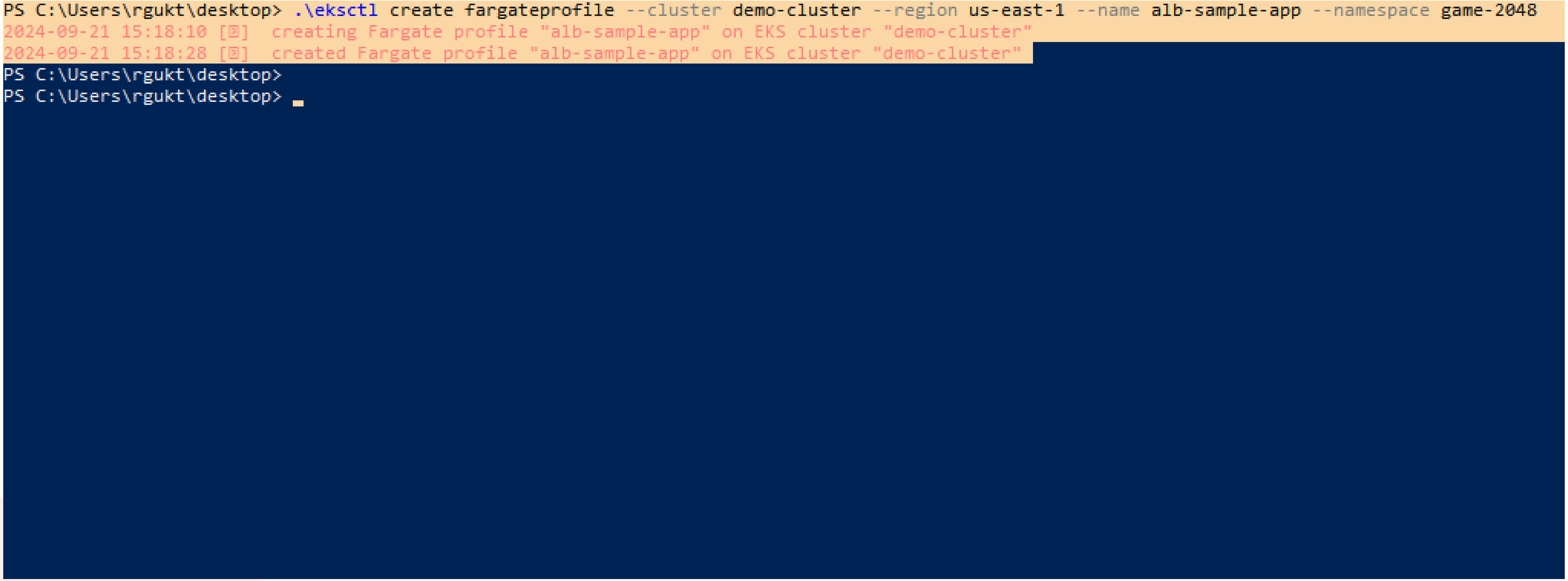
	Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
●	demo-cluster	Active	1.30	Standard support until July 28, 2025	Extended	34 minutes ago	EKS

step-3 updating the kube configuration of cluster



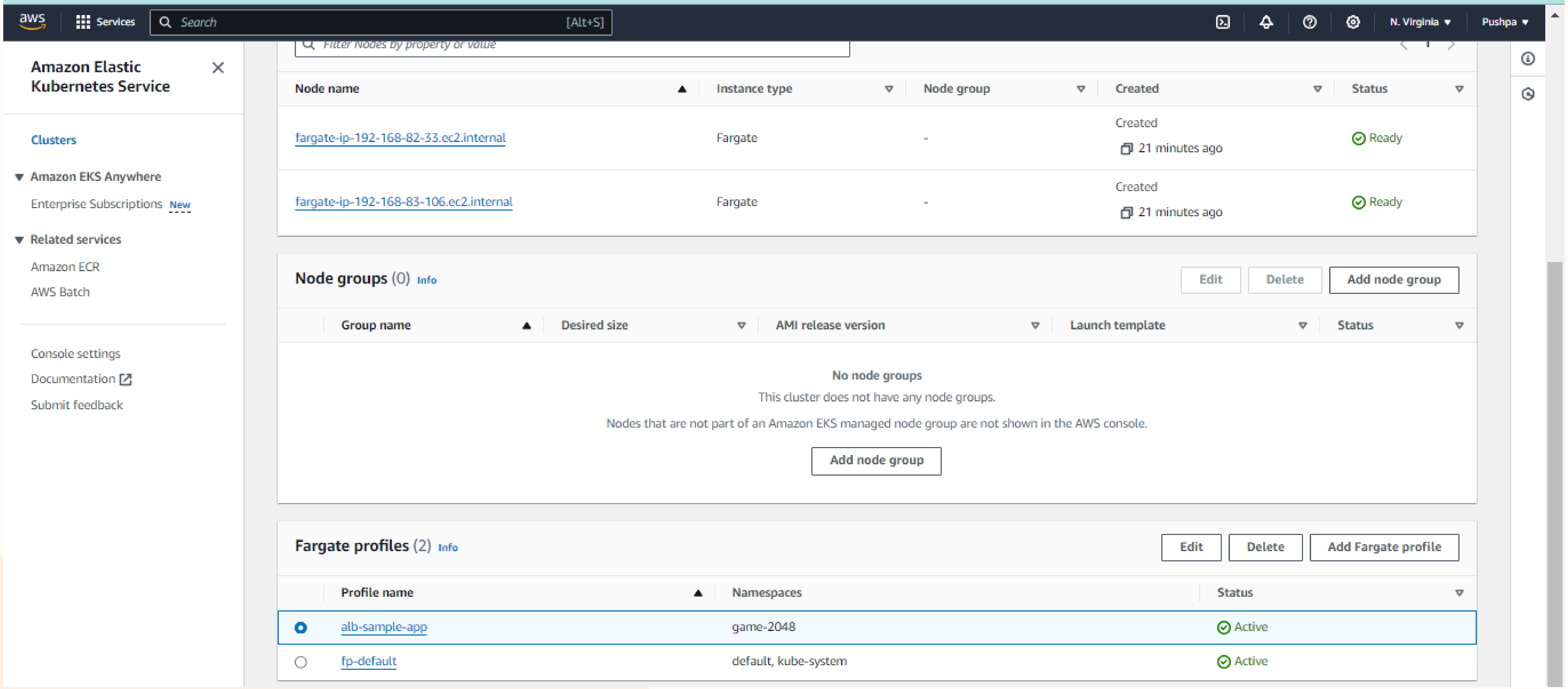
```
:53:56 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
:54:57 [⍰]  waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
:55:03 [!]  recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recomme
M 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.PodIc
nd run `eksctl update addon`
:55:03 [⍰]  creating addon
:55:03 [⍰]  successfully created addon
:55:04 [⍰]  creating addon
:55:05 [⍰]  successfully created addon
:55:05 [⍰]  creating addon
:55:06 [⍰]  successfully created addon
:57:11 [⍰]  creating Fargate profile "fp-default" on EKS cluster "demo-cluster"
:58:19 [⍰]  created Fargate profile "fp-default" on EKS cluster "demo-cluster"
:58:51 [⍰]  "coredns" is now schedulable onto Fargate
:59:59 [⍰]  "coredns" is now scheduled onto Fargate
:59:59 [⍰]  "coredns" pods are now scheduled onto Fargate
:59:59 [⍰]  waiting for the control plane to become ready
:00:05 [⍰]  saved kubeconfig as "C:\\Users\\rgukt\\.kube\\config"
:00:05 [⍰]  no tasks
:00:05 [⍰]  all EKS cluster resources for "demo-cluster" have been created
:00:05 [⍰]  created 0 nodegroup(s) in cluster "demo-cluster"
:00:05 [⍰]  created 0 managed nodegroup(s) in cluster "demo-cluster"
:00:23 [⍰]  kubectl command should work with "C:\\Users\\rgukt\\.kube\\config", try 'kubectl get nodes'
:00:23 [⍰]  EKS cluster "demo-cluster" in "us-east-1" region is ready
gukt\desktop>
gukt\desktop>
gukt\desktop>
gukt\desktop>
gukt\desktop>
gukt\desktop> aws eks update-kubeconfig --name demo-cluster --region us-east-1
text arn:aws:eks:us-east-1:025066266788:cluster/demo-cluster to C:\Users\rgukt\.kube\config
```

step-4 creating fargate profile



```
PS C:\Users\rgukt\desktop> .\eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-09-21 15:18:10 [ℹ]  creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-09-21 15:18:28 [ℹ]  created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
PS C:\Users\rgukt\desktop>
PS C:\Users\rgukt\desktop> _
```

This is the fargate profile we created



Node name	Instance type	Node group	Created	Status
fargate-ip-192-168-82-33.ec2.internal	Fargate	-	Created 21 minutes ago	Ready
fargate-ip-192-168-83-106.ec2.internal	Fargate	-	Created 21 minutes ago	Ready

Node groups (0) Info

Edit Delete Add node group

Group name	Desired size	AMI release version	Launch template	Status

No node groups

This cluster does not have any node groups.

Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.

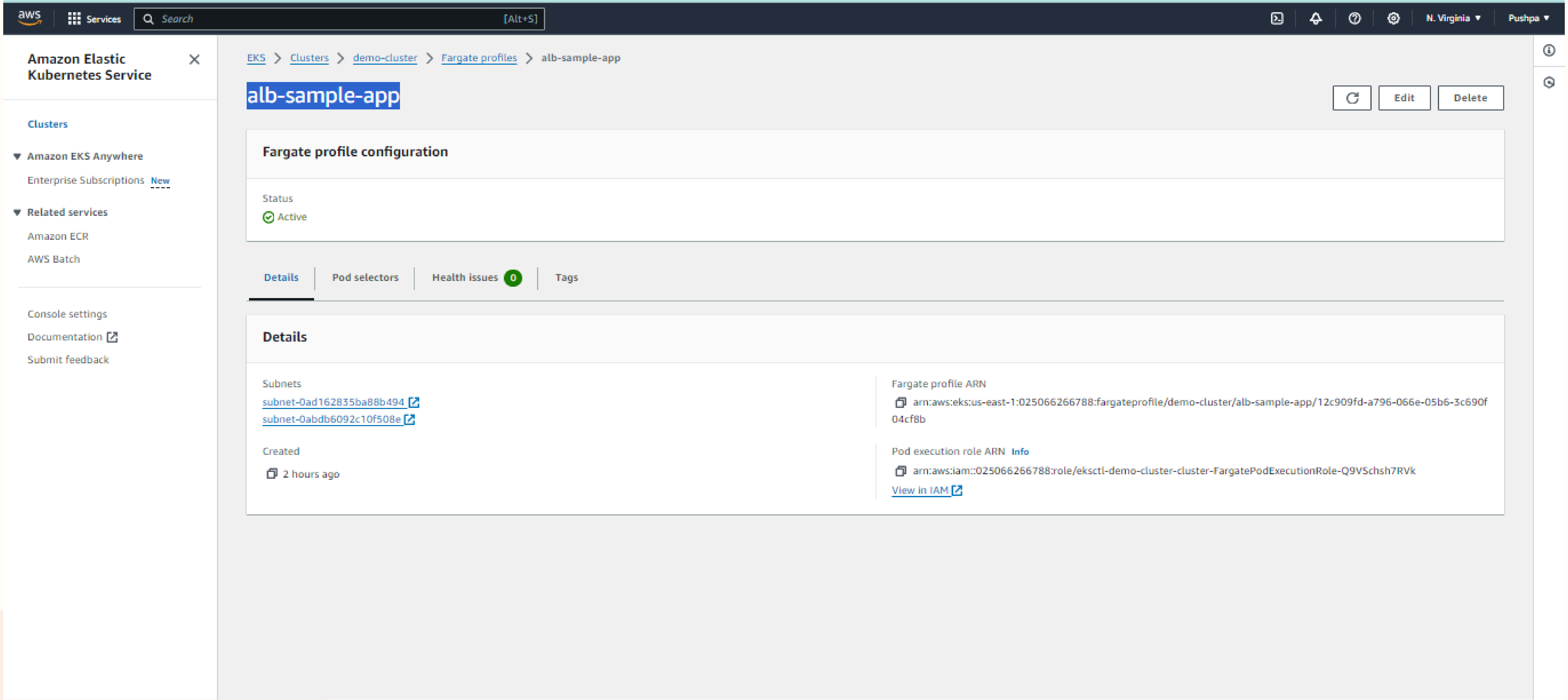
Add node group

Fargate profiles (2) Info

Edit Delete Add Fargate profile

	Profile name	Namespaces	Status
●	alb-sample-app	game-2048	Active
○	fp-default	default, kube-system	Active

The fargate details



alb-sample-app

Fargate profile configuration

Status
Active

Details

Subnets
subnet-0ad162835ba88b494
subnet-0abdb6092c10f508e

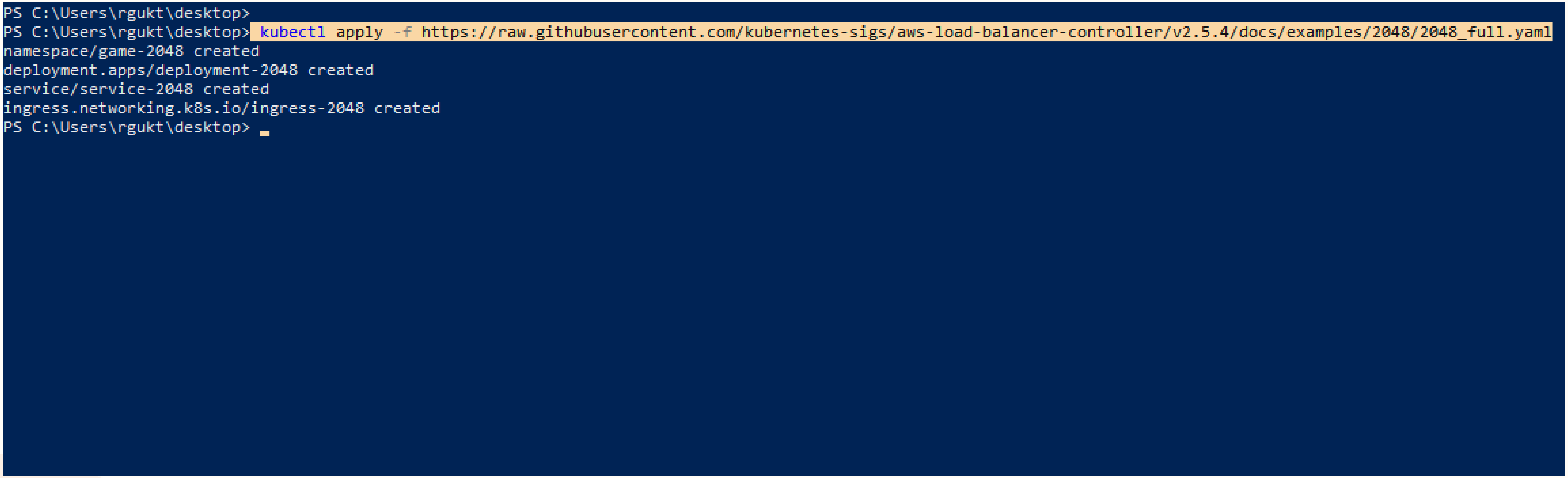
Created
2 hours ago

Fargate profile ARN
arn:aws:eks:us-east-1:025066266788:fargateprofile/demo-cluster/alb-sample-app/12c909fd-a796-066e-05b6-3c690f04cf8b

Pod execution role ARN Info
arn:aws:iam::025066266788:role/eksctl-demo-cluster-cluster-FargatePodExecutionRole-Q9VSchsh7RVk
View in IAM

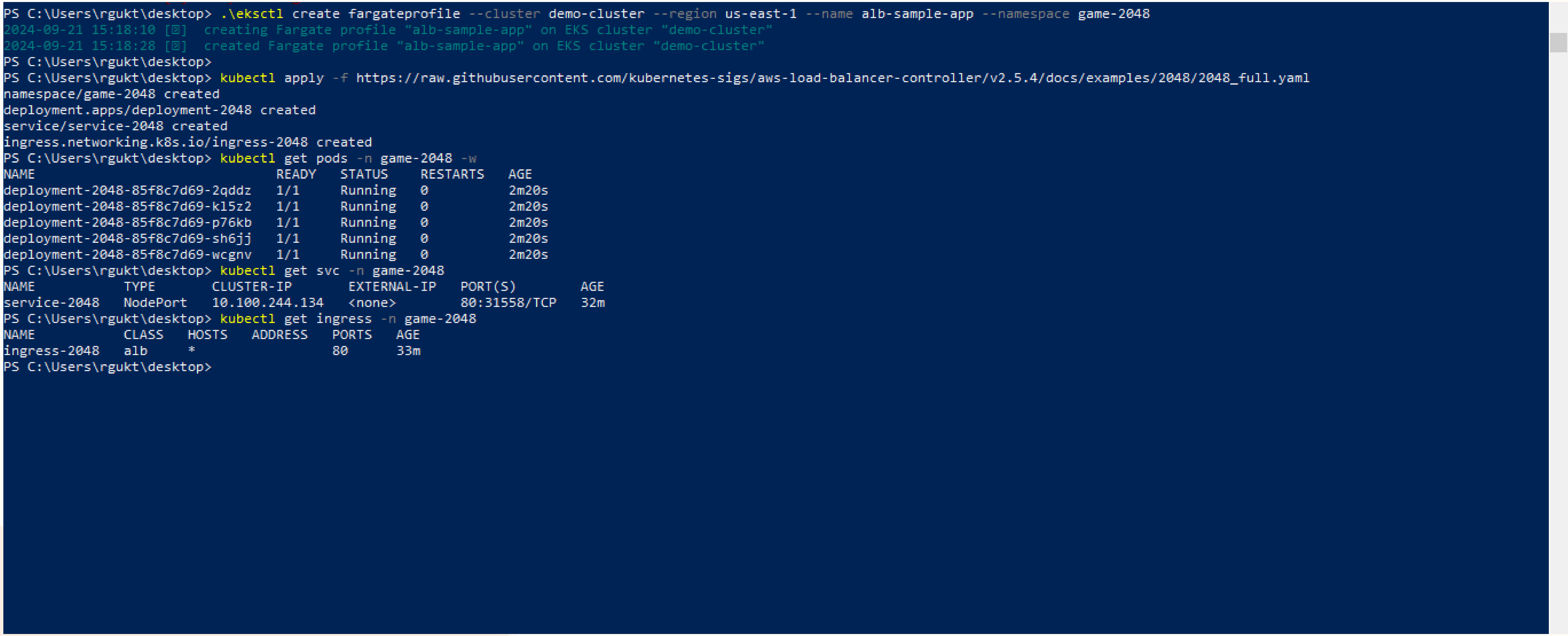
step-5

creating a namespace , deployment , service , ingress



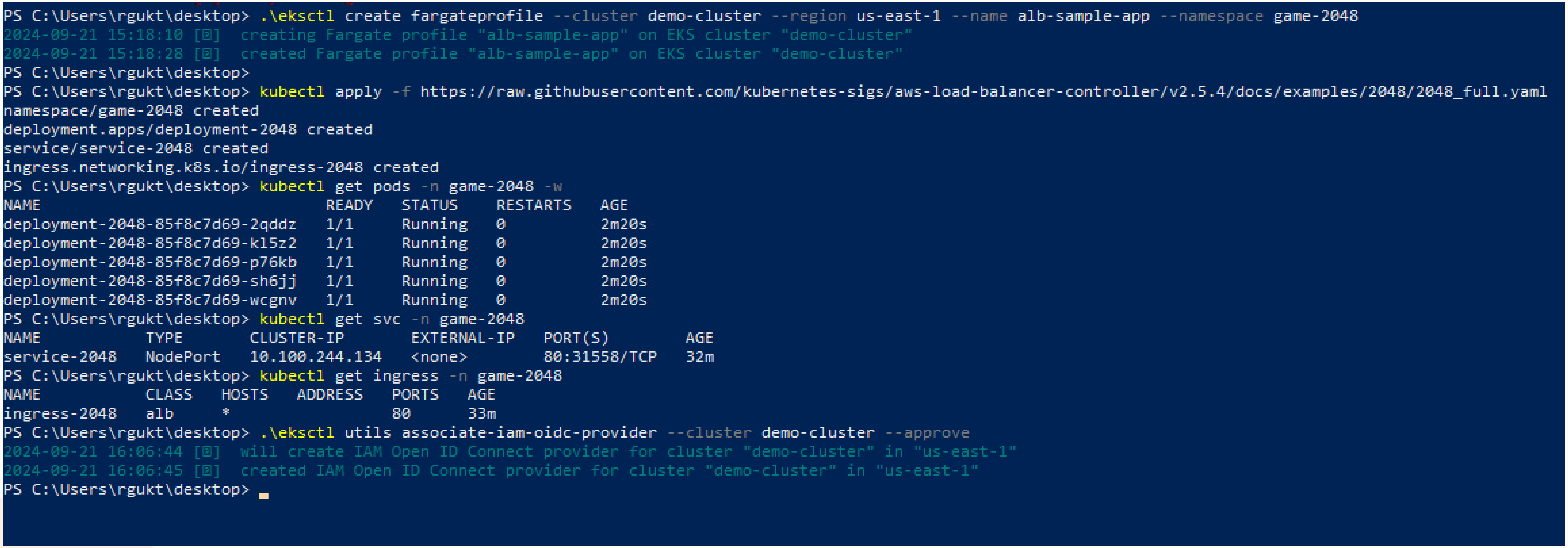
```
PS C:\Users\rgukt\desktop>
PS C:\Users\rgukt\desktop> kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created
PS C:\Users\rgukt\desktop> _
```

We can observe the pods , Services and ingress rule running inside the namespace i.e “game-2048” we can observe that there is no address in the ingress



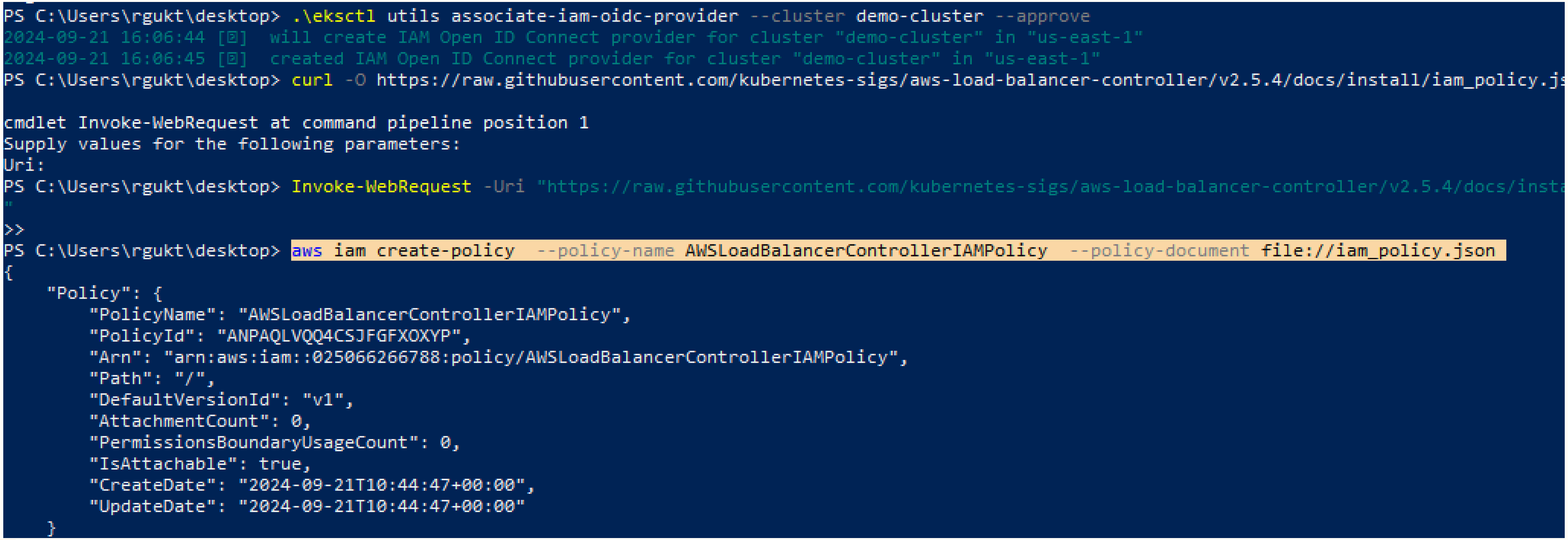
```
PS C:\Users\rgukt\desktop> .\eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-09-21 15:18:10 [⍰]  creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-09-21 15:18:28 [⍰]  created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
PS C:\Users\rgukt\desktop>
PS C:\Users\rgukt\desktop> kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created
PS C:\Users\rgukt\desktop> kubectl get pods -n game-2048 -w
NAME                              READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-2qddz   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-kl5z2   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-p76kb   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-sh6jj   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-wcgnv   1/1     Running   0          2m20s
PS C:\Users\rgukt\desktop> kubectl get svc -n game-2048
NAME           TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service-2048   NodePort   10.100.244.134   <none>        80:31558/TCP   32m
PS C:\Users\rgukt\desktop> kubectl get ingress -n game-2048
NAME           CLASS   HOSTS   ADDRESS   PORTS   AGE
ingress-2048   alb     *                 80      33m
PS C:\Users\rgukt\desktop>
```

step-6 Creating IAM open ID connect provider for the cluster



```
PS C:\Users\rgukt\desktop> .\eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-09-21 15:18:10 [⍰]  creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-09-21 15:18:28 [⍰]  created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
PS C:\Users\rgukt\desktop>
PS C:\Users\rgukt\desktop> kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created
PS C:\Users\rgukt\desktop> kubectl get pods -n game-2048 -w
NAME                              READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-2qddz   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-kl5z2   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-p76kb   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-sh6jj   1/1     Running   0          2m20s
deployment-2048-85f8c7d69-wcgnv   1/1     Running   0          2m20s
PS C:\Users\rgukt\desktop> kubectl get svc -n game-2048
NAME           TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service-2048   NodePort   10.100.244.134   <none>        80:31558/TCP   32m
PS C:\Users\rgukt\desktop> kubectl get ingress -n game-2048
NAME           CLASS   HOSTS   ADDRESS   PORTS   AGE
ingress-2048   alb     *                 80      33m
PS C:\Users\rgukt\desktop> .\eksctl utils associate-iam-oidc-provider --cluster demo-cluster --approve
2024-09-21 16:06:44 [⍰]  will create IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"
2024-09-21 16:06:45 [⍰]  created IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"
PS C:\Users\rgukt\desktop> _
```

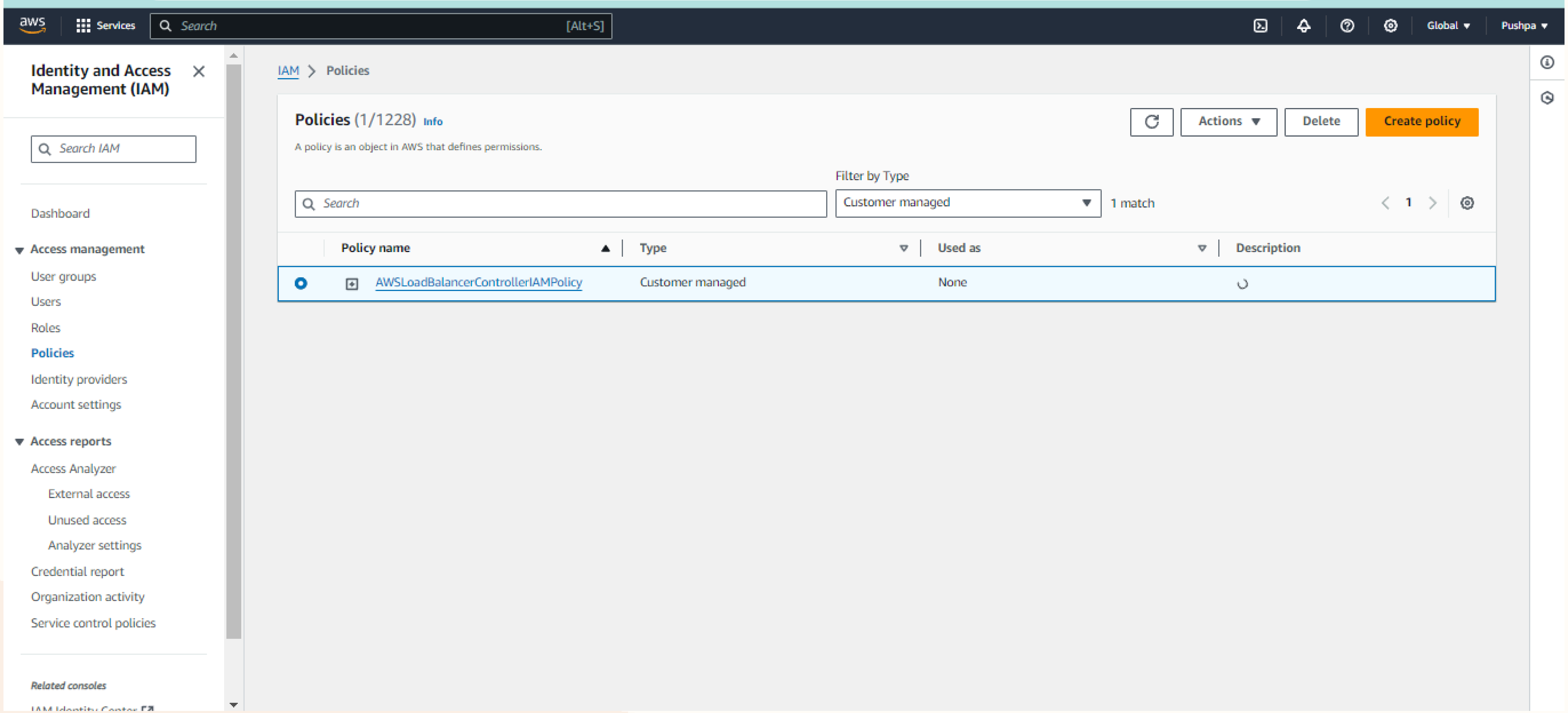
step-7 creating an IAM policy



```
PS C:\Users\rgukt\desktop> .\eksctl utils associate-iam-oidc-provider --cluster demo-cluster --approve
2024-09-21 16:06:44 [ℹ]  will create IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"
2024-09-21 16:06:45 [ℹ]  created IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"
PS C:\Users\rgukt\desktop> curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/install/iam_policy.js

cmdlet Invoke-WebRequest at command pipeline position 1
Supply values for the following parameters:
Uri:
PS C:\Users\rgukt\desktop> Invoke-WebRequest -Uri "https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/insta
"
>>
PS C:\Users\rgukt\desktop> aws iam create-policy  --policy-name AWSLoadBalancerControllerIAMPolicy  --policy-document file://iam_policy.json
{
    "Policy": {
        "PolicyName": "AWSLoadBalancerControllerIAMPolicy",
        "PolicyId": "ANPAQLVQQ4CSJFGFXOXYP",
        "Arn": "arn:aws:iam::025066266788:policy/AWSLoadBalancerControllerIAMPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2024-09-21T10:44:47+00:00",
        "UpdateDate": "2024-09-21T10:44:47+00:00"
    }
```

This is the policy we have created



Policies (1/1228) Info

A policy is an object in AWS that defines permissions.

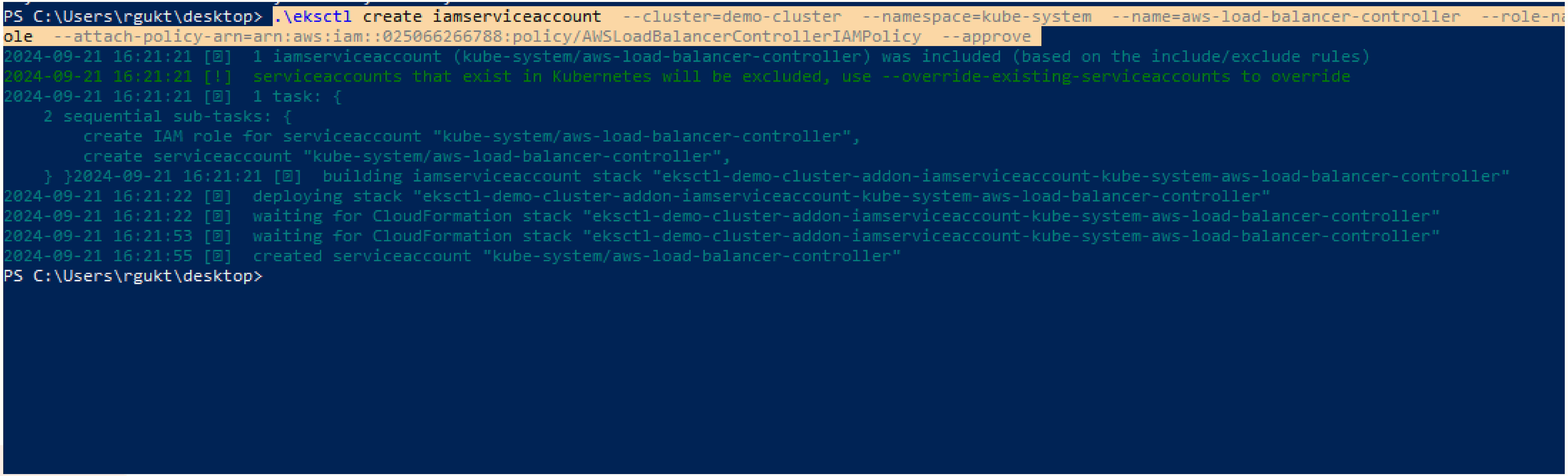
Filter by Type: Customer managed

1 match

	Policy name	Type	Used as	Description
●	AWSLoadBalancerControllerIAMPolicy	Customer managed	None	

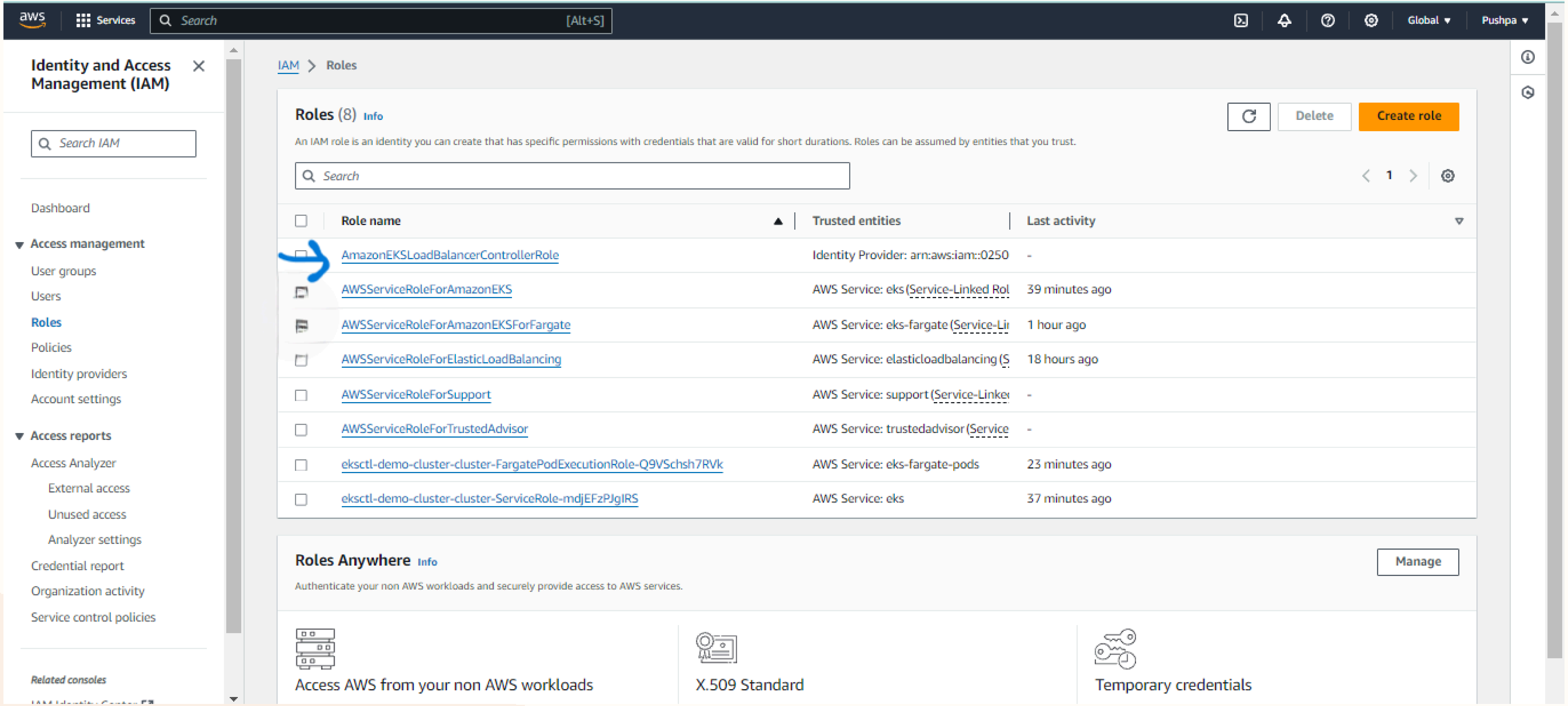
step-8

creating an IAM role



```
PS C:\Users\rgukt\desktop> .\eksctl create iamserviceaccount  --cluster=demo-cluster  --namespace=kube-system  --name=aws-load-balancer-controller  --role-na
ole  --attach-policy-arn=arn:aws:iam::025066266788:policy/AWSLoadBalancerControllerIAMPolicy  --approve
2024-09-21 16:21:21 [ℹ]  1 iamserviceaccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2024-09-21 16:21:21 [!]  serviceaccounts that exist in Kubernetes will be excluded, use --override-existing-serviceaccounts to override
2024-09-21 16:21:21 [ℹ]  1 task: {
    2 sequential sub-tasks: {
        create IAM role for serviceaccount "kube-system/aws-load-balancer-controller",
        create serviceaccount "kube-system/aws-load-balancer-controller",
    } }2024-09-21 16:21:21 [ℹ]  building iamserviceaccount stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-09-21 16:21:22 [ℹ]  deploying stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-09-21 16:21:22 [ℹ]  waiting for CloudFormation stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-09-21 16:21:53 [ℹ]  waiting for CloudFormation stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-09-21 16:21:55 [ℹ]  created serviceaccount "kube-system/aws-load-balancer-controller"
PS C:\Users\rgukt\desktop>
```

This is the role we have created



Roles (8) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities	Last activity
□	AmazonEKSLoadBalancerControllerRole	Identity Provider: arn:aws:iam::0250	-
□	AWSServiceRoleForAmazonEKS	AWS Service: eks (Service-Linked Rol	39 minutes ago
□	AWSServiceRoleForAmazonEKSForFargate	AWS Service: eks-fargate (Service-Li	1 hour ago
□	AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S	18 hours ago
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	eksctl-demo-cluster-cluster-FargatePodExecutionRole-Q9VSchsh7RVk	AWS Service: eks-fargate-pods	23 minutes ago
□	eksctl-demo-cluster-cluster-ServiceRole-mdjEFzPJgIRS	AWS Service: eks	37 minutes ago

Roles Anywhere Info

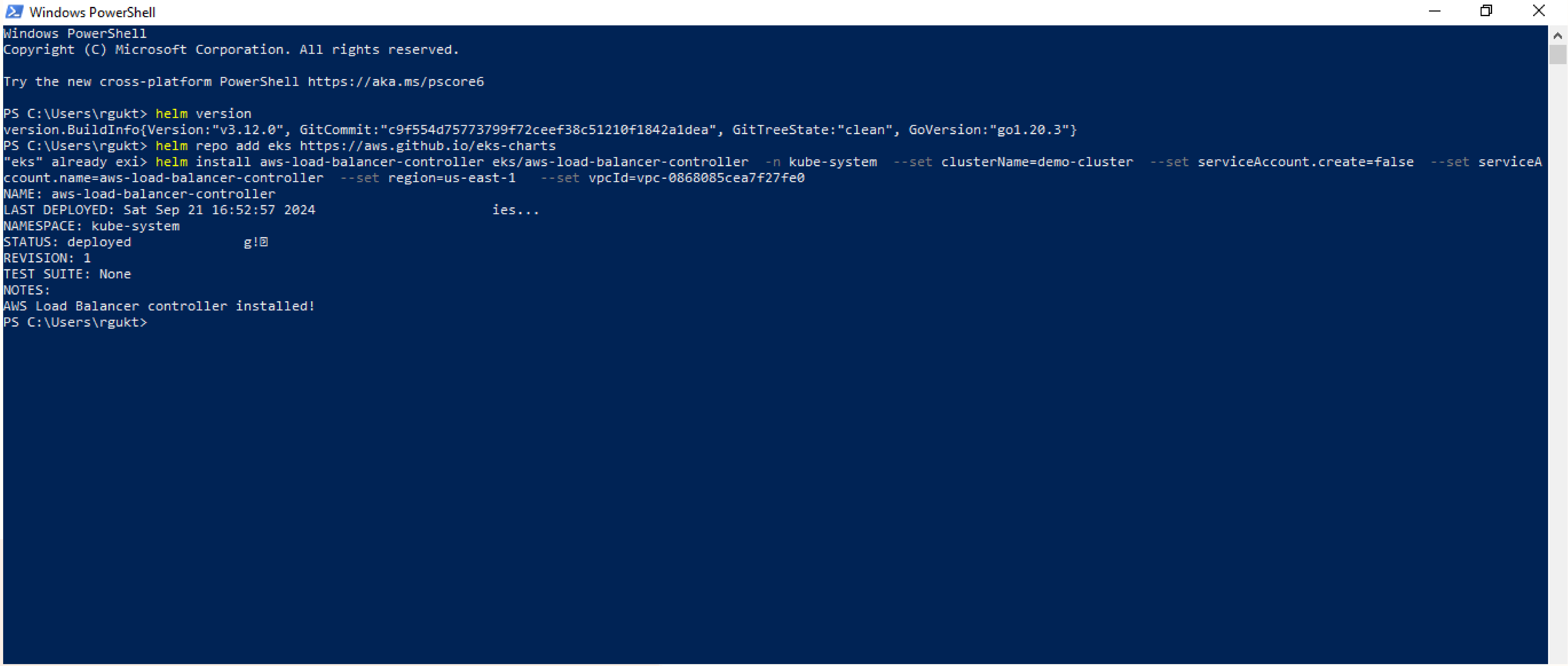
Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

X.509 Standard

Temporary credentials

step-9 creating load balancer controller using helm

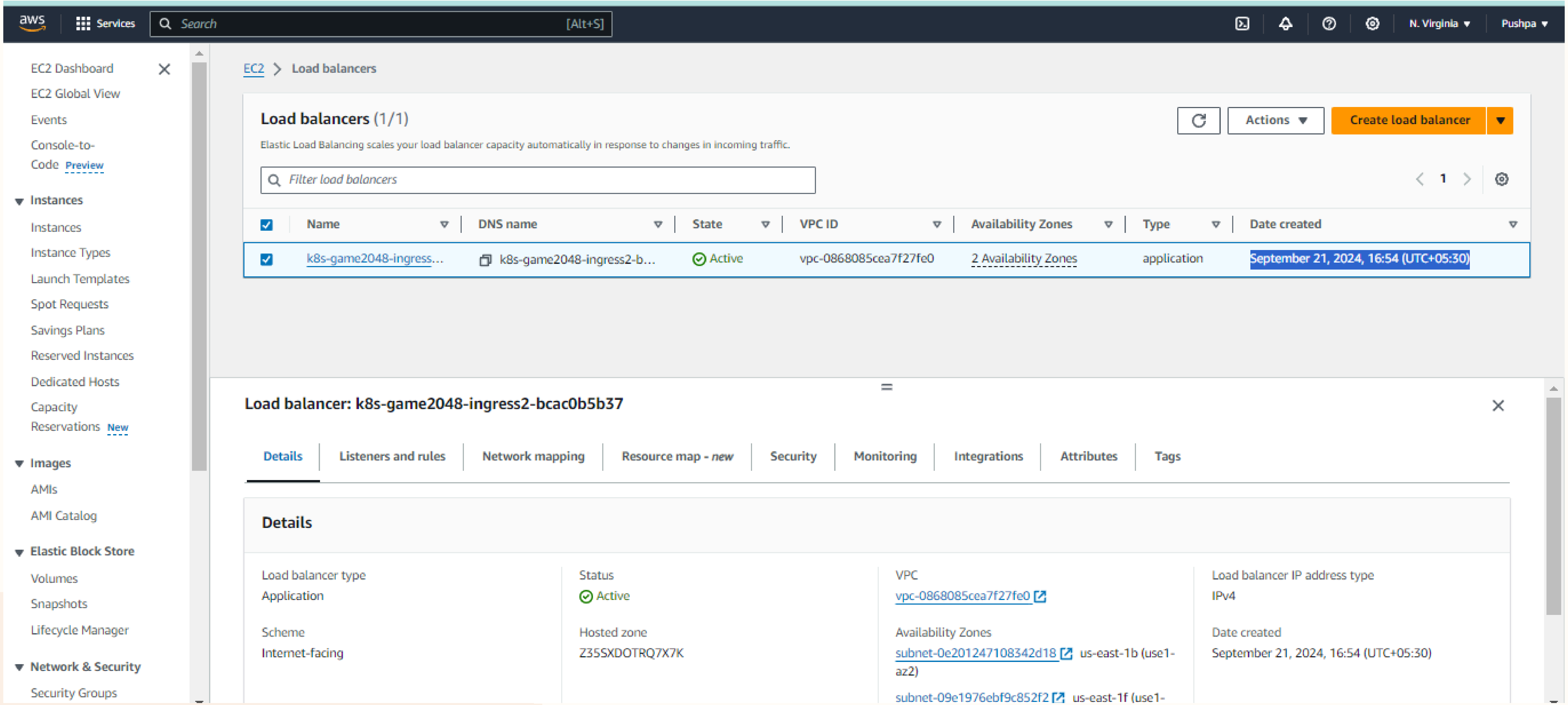


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\rgukt> helm version
version.BuildInfo{Version:"v3.12.0", GitCommit:"c9f554d75773799f72ceef38c51210f1842a1dea", GitTreeState:"clean", GoVersion:"go1.20.3"}
PS C:\Users\rgukt> helm repo add eks https://aws.github.io/eks-charts
"eks" already exi> helm install aws-load-balancer-controller eks/aws-load-balancer-controller  -n kube-system  --set clusterName=demo-cluster  --set serviceAccount.create=false  --set serviceA
ccount.name=aws-load-balancer-controller  --set region=us-east-1   --set vpcId=vpc-0868085cea7f27fe0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Sep 21 16:52:57 2024                      ies...
NAMESPACE: kube-system
STATUS: deployed              g!⍰
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
PS C:\Users\rgukt>
```

This is the load balancer we created



Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☑	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
☑	k8s-game2048-ingress...	k8s-game2048-ingress2-b...	✓ Active	vpc-0868085cea7f27fe0	2 Availability Zones	application	September 21, 2024, 16:54 (UTC+05:30)

Load balancer: k8s-game2048-ingress2-bcac0b5b37

Details

Load balancer type
Application

Scheme
Internet-facing

Status
✓ Active

Hosted zone
Z35SXDOTRQ7X7K

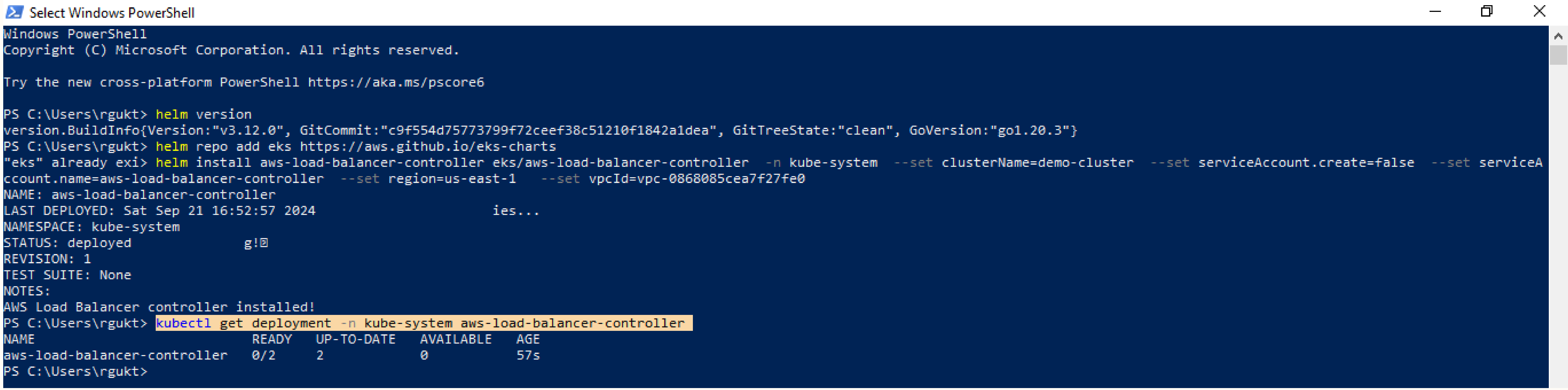
VPC
vpc-0868085cea7f27fe0

Availability Zones
subnet-0e201247108342d18 us-east-1b (use1-az2)
subnet-09e1976ebf9c852f2 us-east-1f (use1-

Load balancer IP address type
IPv4

Date created
September 21, 2024, 16:54 (UTC+05:30)

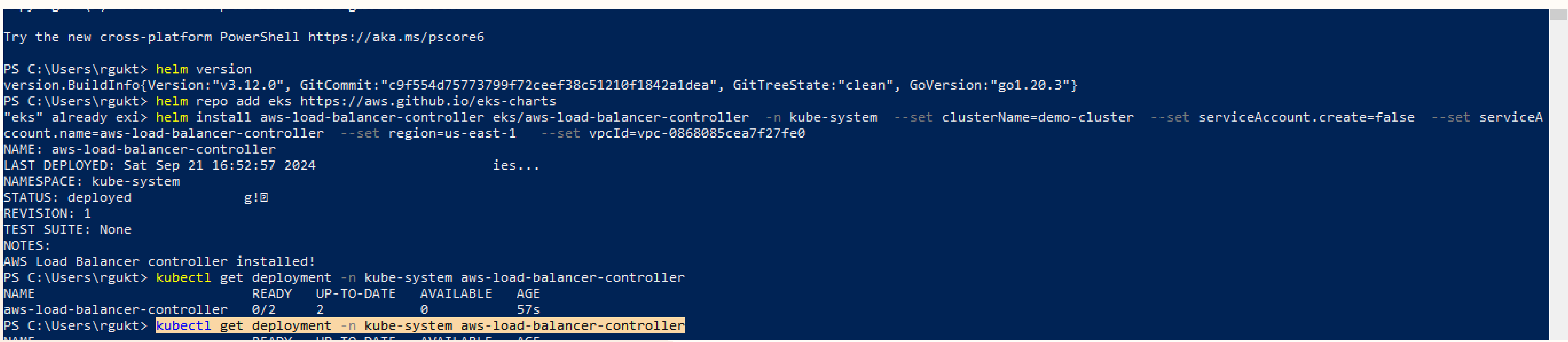
step-10 we get the deployment through load balancer ,once it is running



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\rgukt> helm version
version.BuildInfo{Version:"v3.12.0", GitCommit:"c9f554d75773799f72ceef38c51210f1842a1dea", GitTreeState:"clean", GoVersion:"go1.20.3"}
PS C:\Users\rgukt> helm repo add eks https://aws.github.io/eks-charts
"eks" already exi> helm install aws-load-balancer-controller eks/aws-load-balancer-controller  -n kube-system  --set clusterName=demo-cluster  --set serviceAccount.create=false  --set serviceA
ccount.name=aws-load-balancer-controller  --set region=us-east-1   --set vpcId=vpc-0868085cea7f27fe0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Sep 21 16:52:57 2024                      ies...
NAMESPACE: kube-system
STATUS: deployed              g!⊡
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   0/2     2            0           57s
PS C:\Users\rgukt>
```

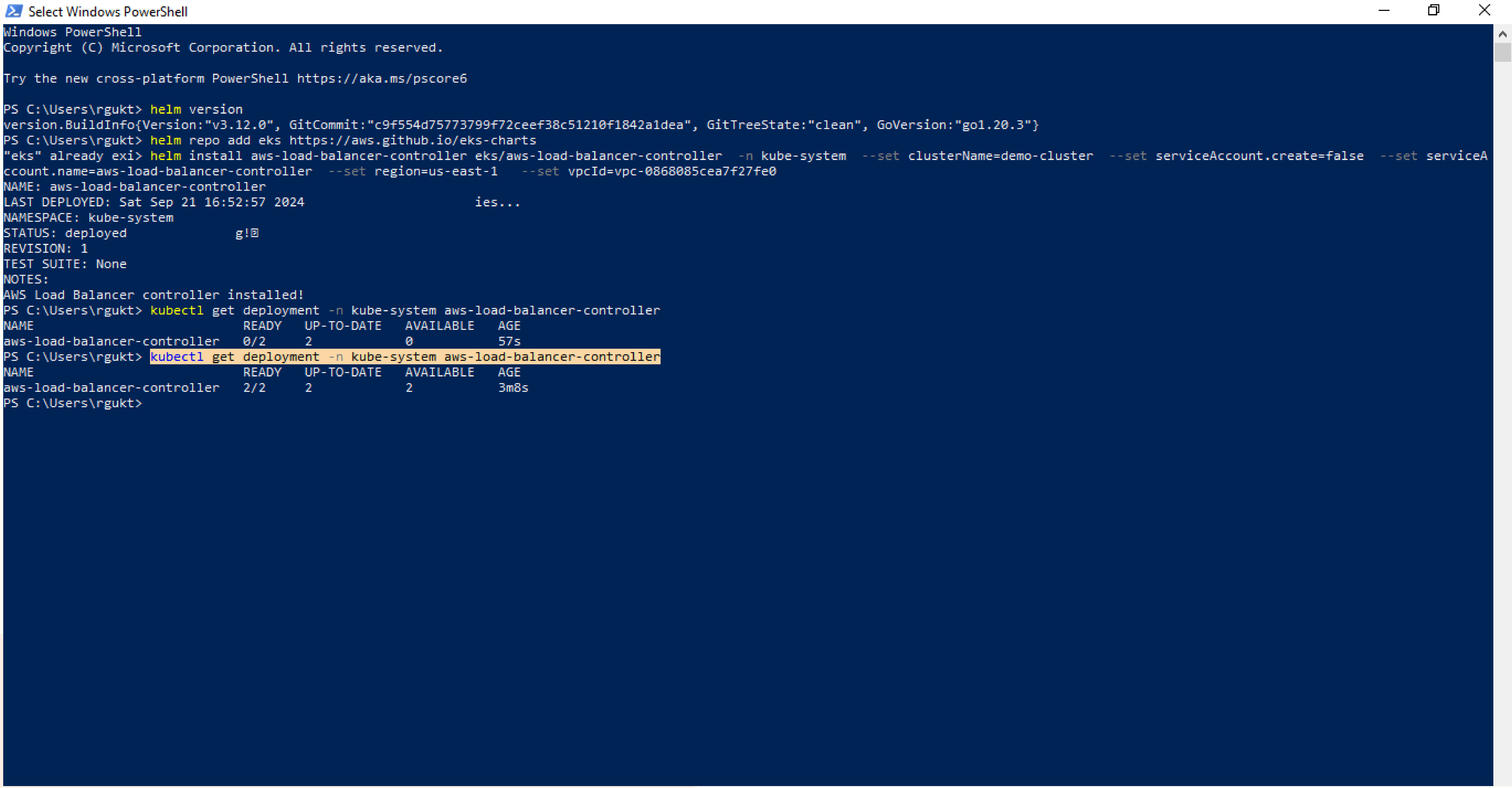


```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\rgukt> helm version
version.BuildInfo{Version:"v3.12.0", GitCommit:"c9f554d75773799f72ceef38c51210f1842a1dea", GitTreeState:"clean", GoVersion:"go1.20.3"}
PS C:\Users\rgukt> helm repo add eks https://aws.github.io/eks-charts
"eks" already exi> helm install aws-load-balancer-controller eks/aws-load-balancer-controller  -n kube-system  --set clusterName=demo-cluster  --set serviceAccount.create=false  --set serviceA
ccount.name=aws-load-balancer-controller  --set region=us-east-1   --set vpcId=vpc-0868085cea7f27fe0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Sep 21 16:52:57 2024                      ies...
NAMESPACE: kube-system
STATUS: deployed              g!⊡
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   0/2     2            0           57s
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
```

step-11 Now if we run ingress ,we will get the address

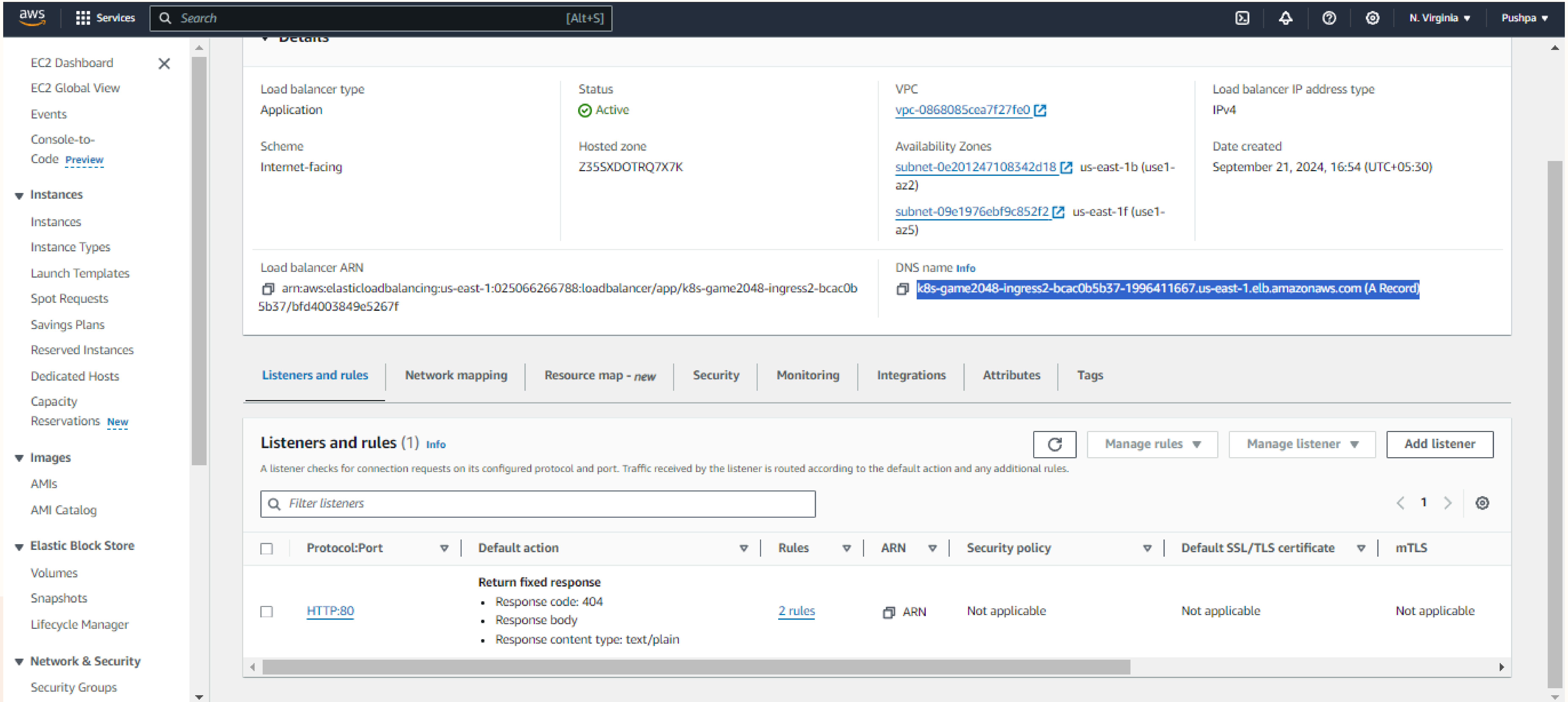


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\rgukt> helm version
version.BuildInfo{Version:"v3.12.0", GitCommit:"c9f554d75773799f72ceef38c51210f1842a1dea", GitTreeState:"clean", GoVersion:"go1.20.3"}
PS C:\Users\rgukt> helm repo add eks https://aws.github.io/eks-charts
"eks" already exi> helm install aws-load-balancer-controller eks/aws-load-balancer-controller  -n kube-system  --set clusterName=demo-cluster  --set serviceAccount.create=false  --set serviceA
ccount.name=aws-load-balancer-controller  --set region=us-east-1   --set vpcId=vpc-0868085cea7f27fe0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Sep 21 16:52:57 2024                      ies...
NAMESPACE: kube-system
STATUS: deployed              g!⍰
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   0/2     2            0           57s
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2            2           3m8s
PS C:\Users\rgukt>
```

we can observe that same address is in the dns name of the load balancer



Load balancer type
Application

Status
Active

VPC
vpc-0868085cea7f27fe0

Load balancer IP address type
IPv4

Scheme
Internet-facing

Hosted zone
Z35SXDOTRQ7X7K

Availability Zones
subnet-0e201247108342d18 us-east-1b (use1-az2)
subnet-09e1976ebf9c852f2 us-east-1f (use1-az5)

Date created
September 21, 2024, 16:54 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:025066266788:loadbalancer/app/k8s-game2048-ingress2-bcac0b5b37/bfd4003849e5267f

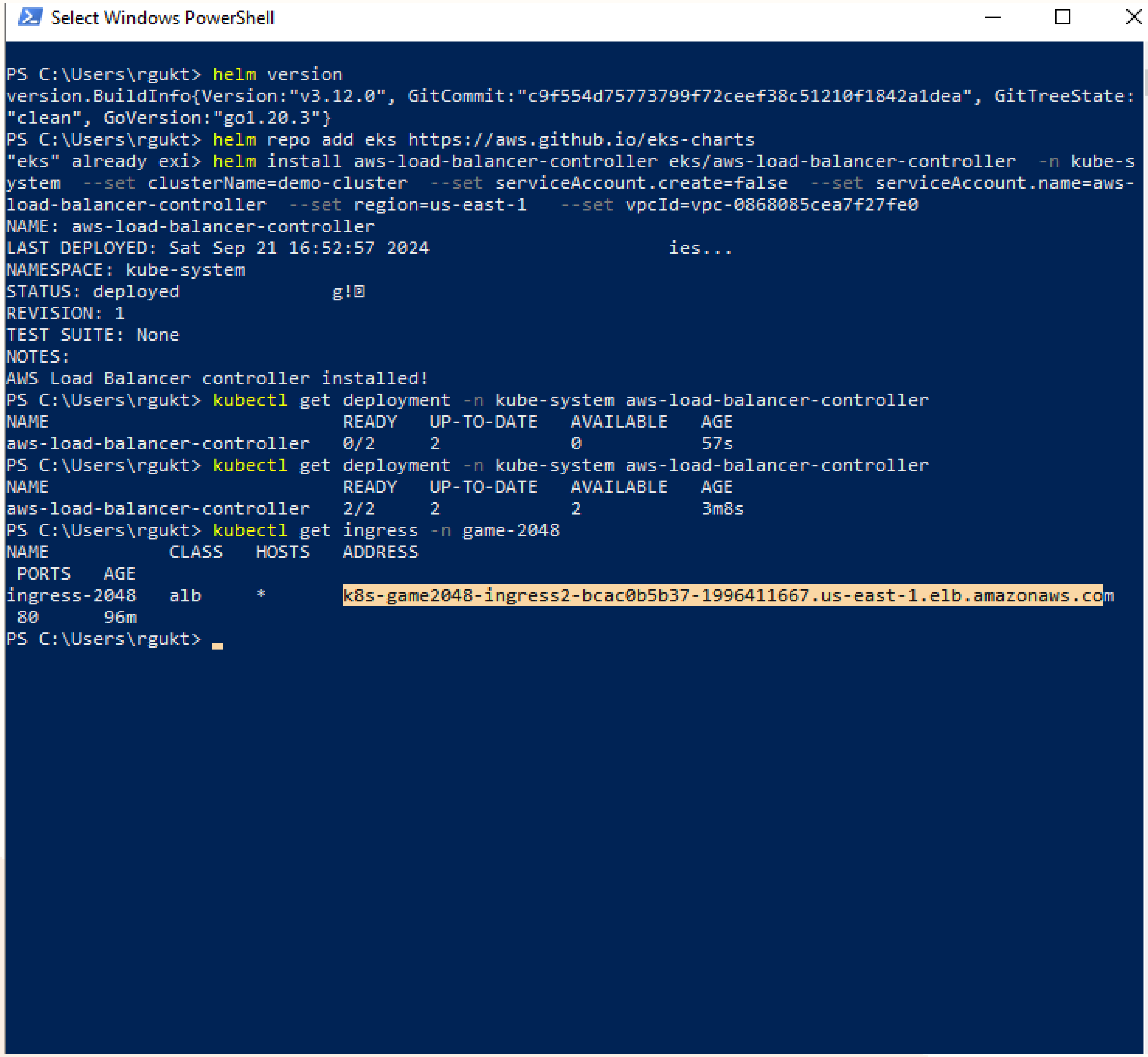
DNS name Info
k8s-game2048-ingress2-bcac0b5b37-1996411667.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

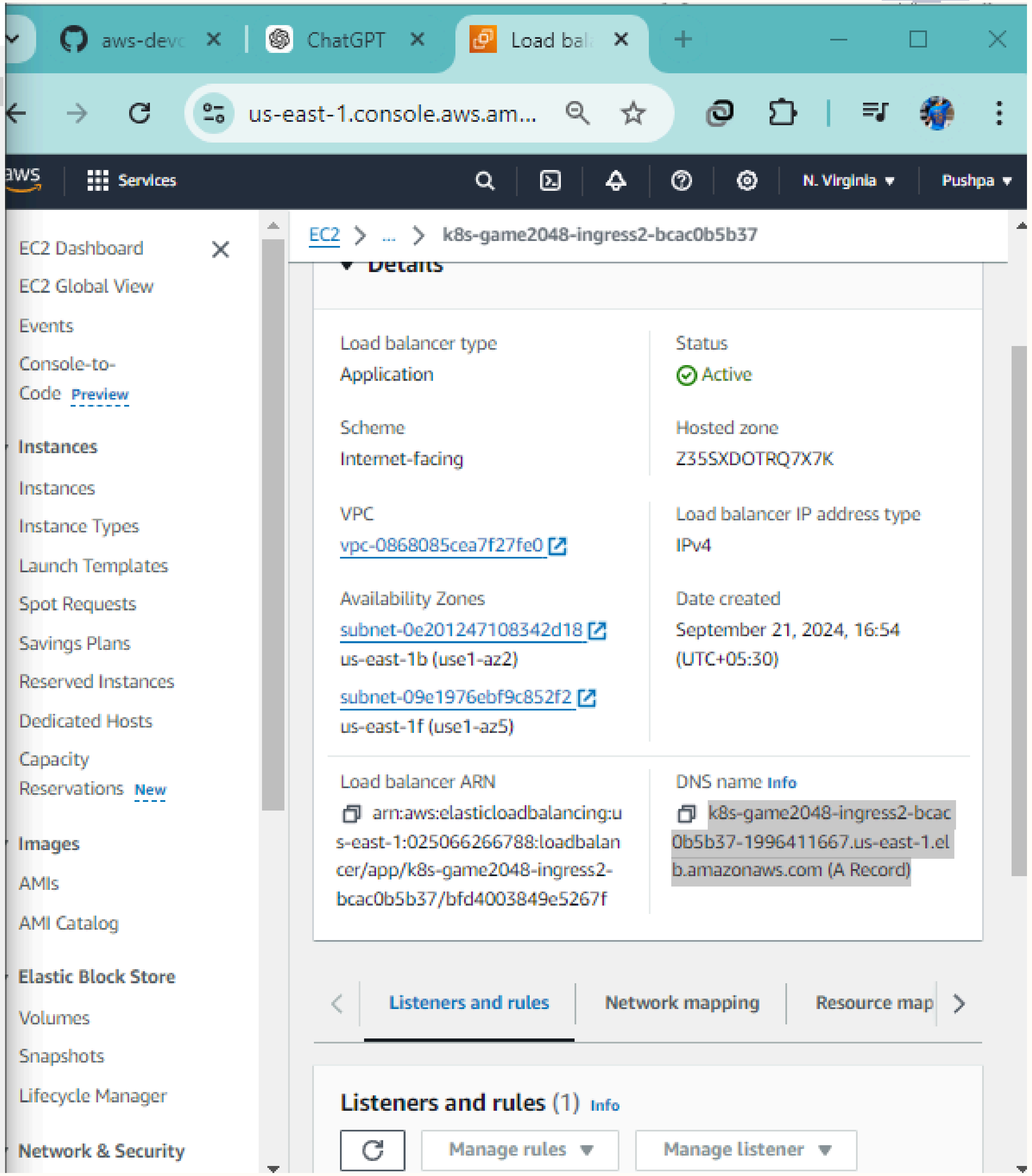
□	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS
□	HTTP:80	**Return fixed response** • Response code: 404 • Response body • Response content type: text/plain	2 rules	ARN	Not applicable	Not applicable	Not applicable

The comparison



Select Windows PowerShell

```
PS C:\Users\rgukt> helm version
version.BuildInfo{Version:"v3.12.0", GitCommit:"c9f554d75773799f72ceef38c51210f1842a1dea", GitTreeState:
"clean", GoVersion:"go1.20.3"}
PS C:\Users\rgukt> helm repo add eks https://aws.github.io/eks-charts
"eks" already exi> helm install aws-load-balancer-controller eks/aws-load-balancer-controller  -n kube-s
ystem  --set clusterName=demo-cluster  --set serviceAccount.create=false  --set serviceAccount.name=aws-
load-balancer-controller  --set region=us-east-1   --set vpcId=vpc-0868085cea7f27fe0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Sep 21 16:52:57 2024                      ies...
NAMESPACE: kube-system
STATUS: deployed             g!⍰
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   0/2     2            0           57s
PS C:\Users\rgukt> kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2            2           3m8s
PS C:\Users\rgukt> kubectl get ingress -n game-2048
NAME           CLASS   HOSTS   ADDRESS
 PORTS   AGE
ingress-2048   alb     *       k8s-game2048-ingress2-bcac0b5b37-1996411667.us-east-1.elb.amazonaws.com
 80      96m
PS C:\Users\rgukt> _
```



▾ Details

Load balancer type
Application

Status
Active

Scheme
Internet-facing

Hosted zone
Z35SXDOTRQ7X7K

VPC
vpc-0868085cea7f27fe0

Load balancer IP address type
IPv4

Availability Zones
subnet-0e201247108342d18
us-east-1b (use1-az2)
subnet-09e1976ebf9c852f2
us-east-1f (use1-az5)

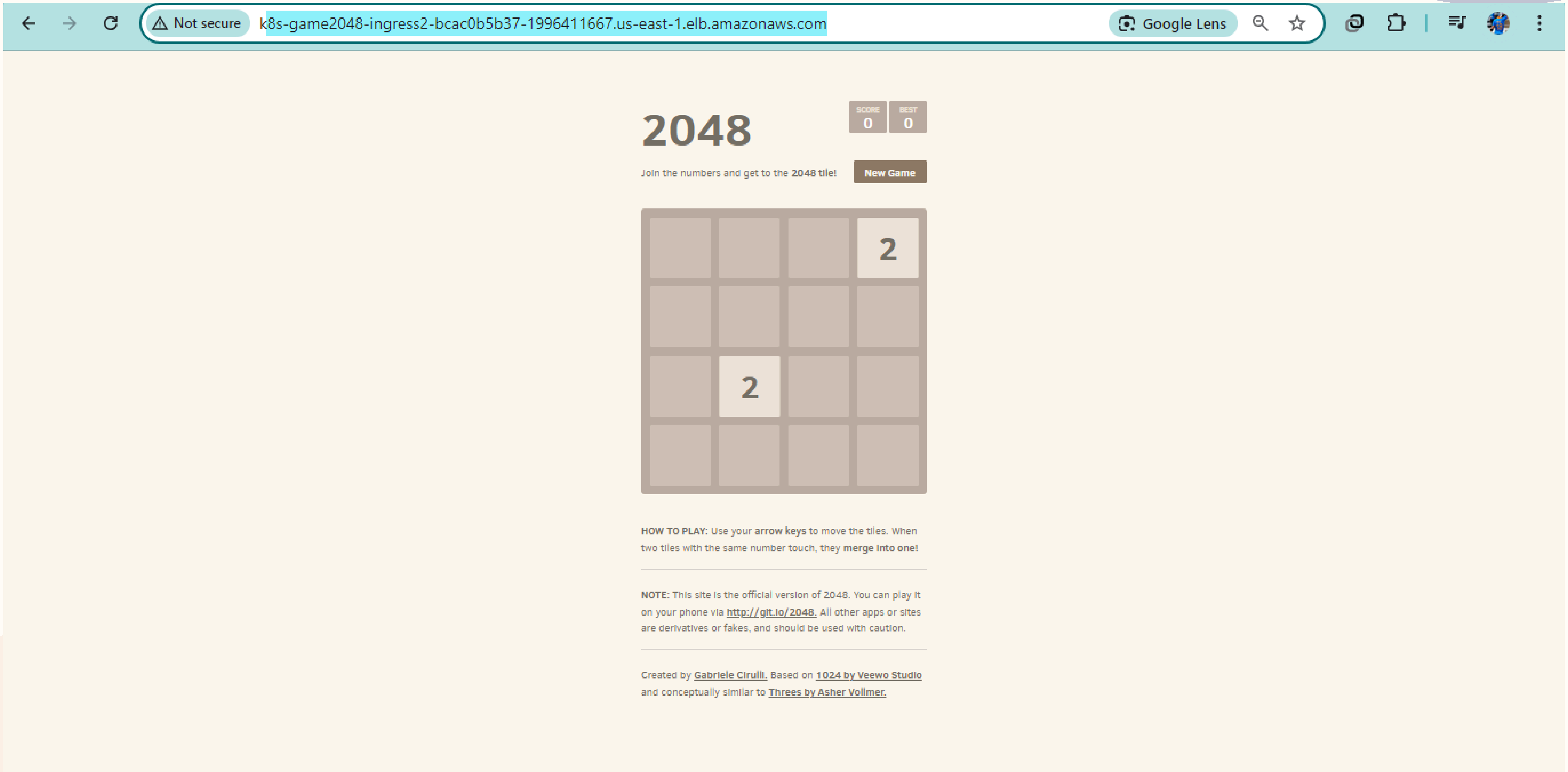
Date created
September 21, 2024, 16:54 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:025066266788:loadbalancer/app/k8s-game2048-ingress2-bcac0b5b37/bfd4003849e5267f

DNS name Info
k8s-game2048-ingress2-bcac0b5b37-1996411667.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

If we browse the dns name , we will get our output i.e the “game-2048”



Not secure
k8s-game2048-ingress2-bcac0b5b37-1996411667.us-east-1.elb.amazonaws.com
Google Lens
2048
SCORE
0
BEST
0
New Game
2
2

Join the numbers and get to the **2048 tile!**

HOW TO PLAY: Use your **arrow keys** to move the tiles. When two tiles with the same number touch, they **merge into one!**

NOTE: This site is the official version of 2048. You can play it on your phone via http://git.io/2048. All other apps or sites are derivatives or fakes, and should be used with caution.

Created by Gabriele Cirulli. Based on 1024 by Veewo Studio and conceptually similar to Threes by Asher Vollmer.

THANK YOU