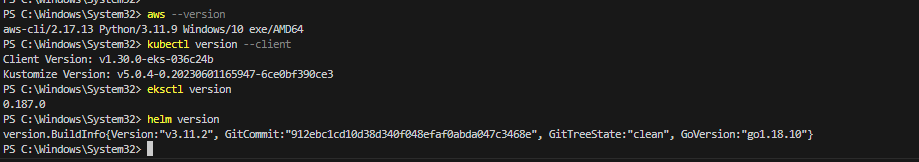
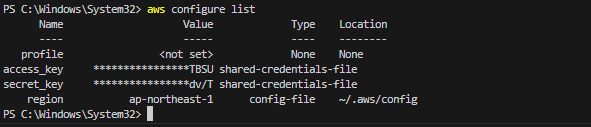
AWS-EKS Sample application deployment

1. Install AWS cli, kubectl, eksctl, helm in the local system.



1. Configure aws user Access and secret key

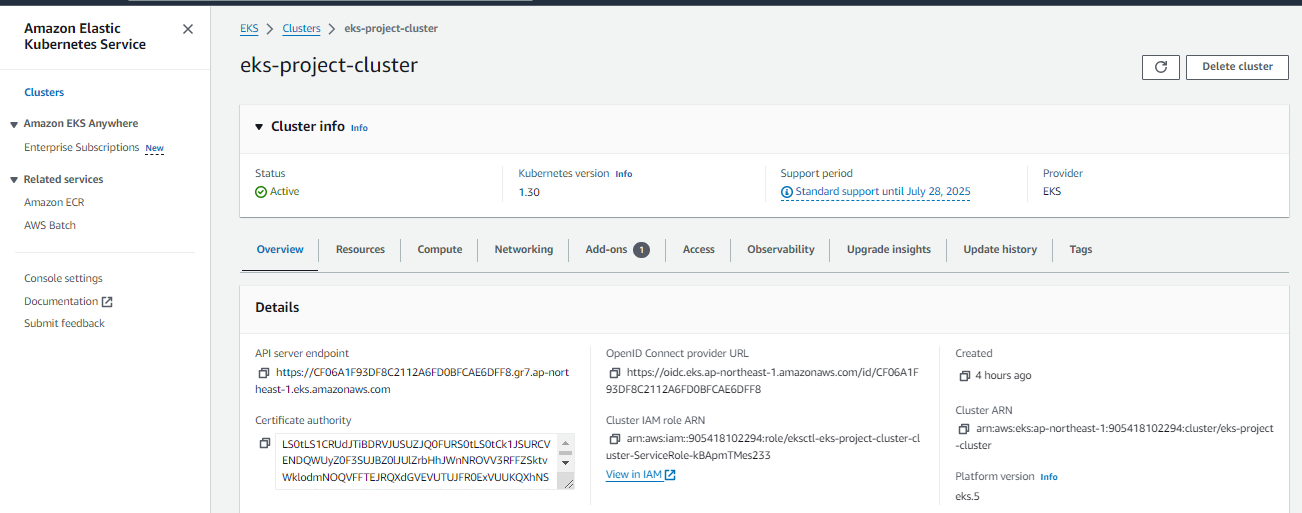


1. Instead of creating cluster manually from console, I will use eksctl and fargate to skip all the manual configurations.(but takes 20-30 min)

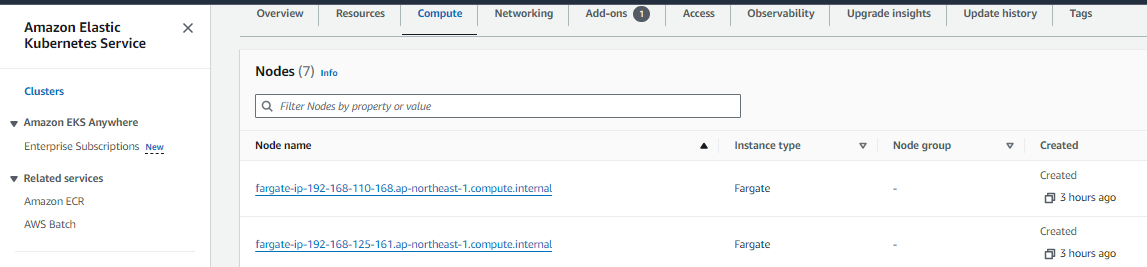
**CMD**: eksctl create cluster --name eks-project-cluster --region ap-northeast-1 –fargate

AWS Fargate offers a serverless approach to running containers, abstracting away the management of EC2 instances. It simplifies deployment by allowing you to focus solely on defining container resources without worrying about infrastructure scaling or maintenance. In contrast, manually setting up node groups and instances in Amazon EKS provides greater control and customization over instance types, networking, and software configurations, making it suitable for workloads with specific requirements or where direct infrastructure management is preferred. Each option balances simplicity and control depending on your application's needs and operational preferences.

1. We can get to see the cluster from the console after above cmd.



1. We can get to see the virtual machines of farget(not ec2 machine)



1. To connect & use EKS from the remote with the local machine

**CMD: aws eks update-kubeconfig --name eks-project-cluster --region ap-northeast-1**

1. Need to create a foregate profile to deploy the application in a new namespace.

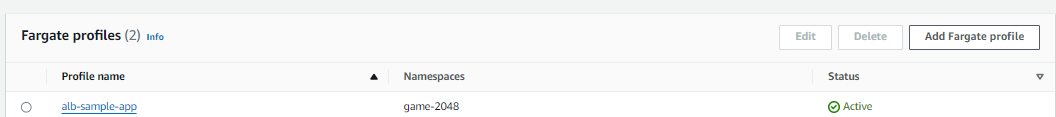
**CMD: eksctl create fargateprofile `**

**--cluster eks-project-cluster `**

**--region ap-northeast-1 `**

**--name alb-sample-app `**

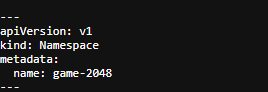
**--namespace game-2048**

****

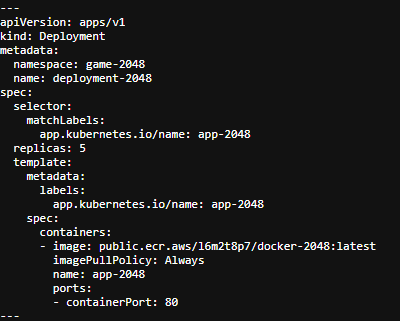
1. **Lets create namespace & deploy the application(pod), Service(Nodeport), ingress(Route traffic inside the cluster and has configuration of ALB).**

**CMD: kubectl apply -f** [**https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048\_full.yaml**](https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml)

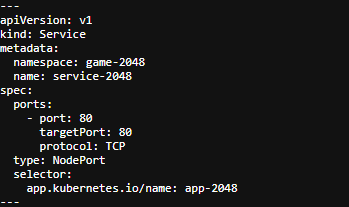
**Namespace:**

****

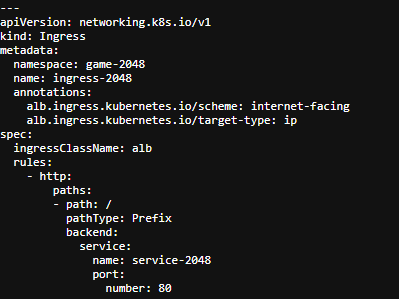
**Pod :**

****

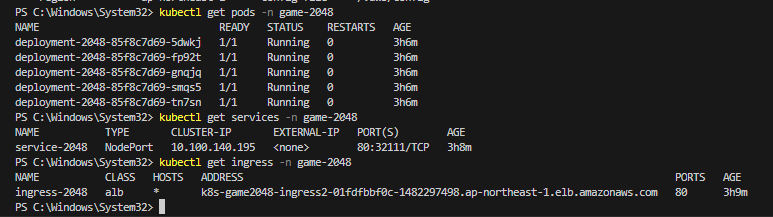
**Service:**

****

**Ingress(internal networking & config of alb resource ):**

****

1. Check if all of them got created

****

1. Lets create a ingress controller, which reads ingress resource and creates load balancer,, target groups, port, etc,.

But before that, for ingress controller to create aws resources, it need to have an IAM role. So we will set up the IAM OIDC provider.

**CMD: eksctl utils associate-iam-oidc-provider --cluster eks-project-cluster—approve**

And now the controller: **CMD:**

* **Invoke-WebRequest -Uri "https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/install/iam\_policy.json" -OutFile "iam\_policy.json"**
* **aws iam create-policy `**

**--policy-name AWSLoadBalancerControllerIAMPolicy `**

**--policy-document** [**file://iam\_policy.json**](file://iam_policy.json)

* **eksctl create iamserviceaccount `**

**--cluster=elk-project-cluster `**

**--namespace=kube-system `**

**--name=aws-load-balancer-controller `**

**--role-name AmazonEKSLoadBalancerControllerRole `**

**--attach-policy-arn=arn:aws:iam905418102294:policy/AWSLoadBalancerControllerIAMPolicy `**

**--approve**

* **helm repo add eks** [**https://aws.github.io/eks-charts**](https://aws.github.io/eks-charts)
* **helm install aws-load-balancer-controller eks/aws-load-balancer-controller `**

**-n kube-system `**

**--set clusterName=eks-project-cluster `**

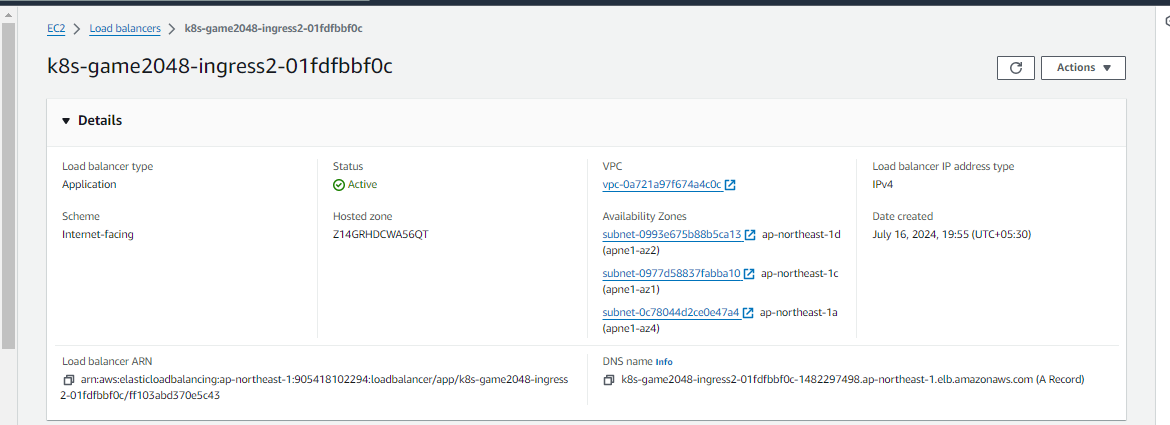
**--set serviceAccount.create=false `**

**--set serviceAccount.name=aws-load-balancer-controller `**

**--set region=ap-northeast-1 `**

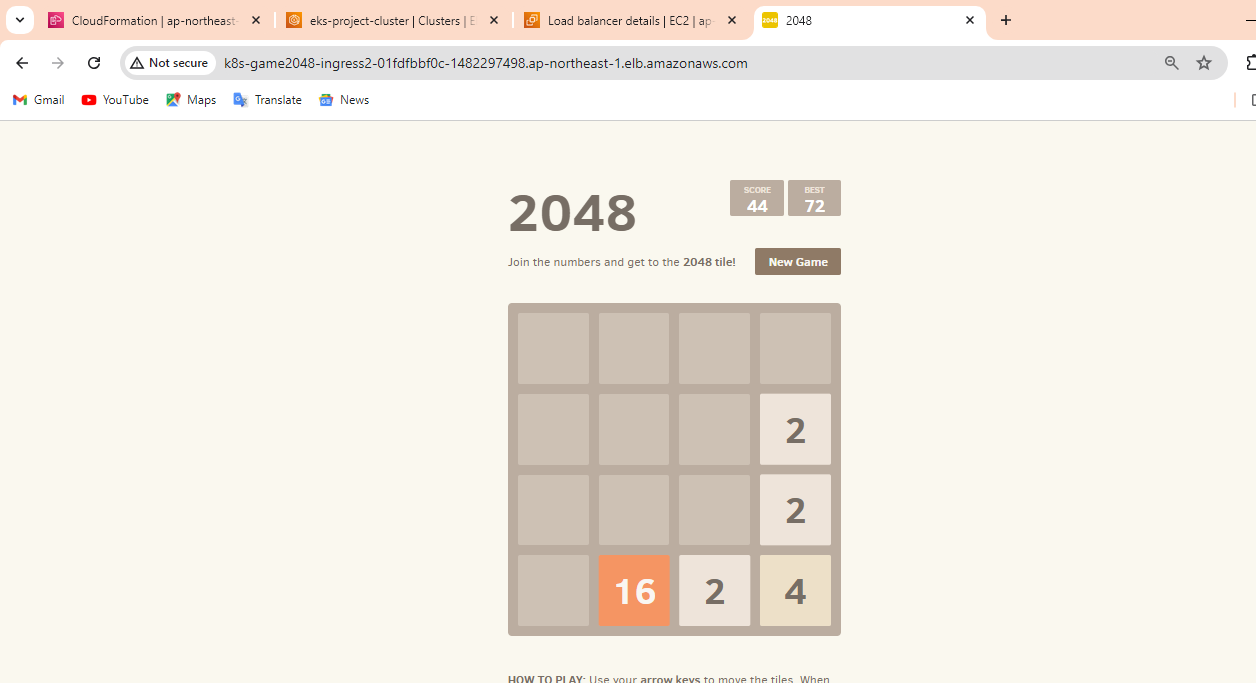
**--set vpcId=** [**vpc-0a721a97f674a4c0c**](https://ap-northeast-1.console.aws.amazon.com/vpc/home?#VpcDetails:VpcId=vpc-0a721a97f674a4c0c)**0000**

**11) Check if ALB has been created from console**

****

****

**Check the al burl in browser:**

****

Hence, successfully deployed a application in EKS cluster with ingress.

To delete the cluster created:

**CMD:** eksctl delete cluster --name eks-project-cluster --region ap-northeast-1

