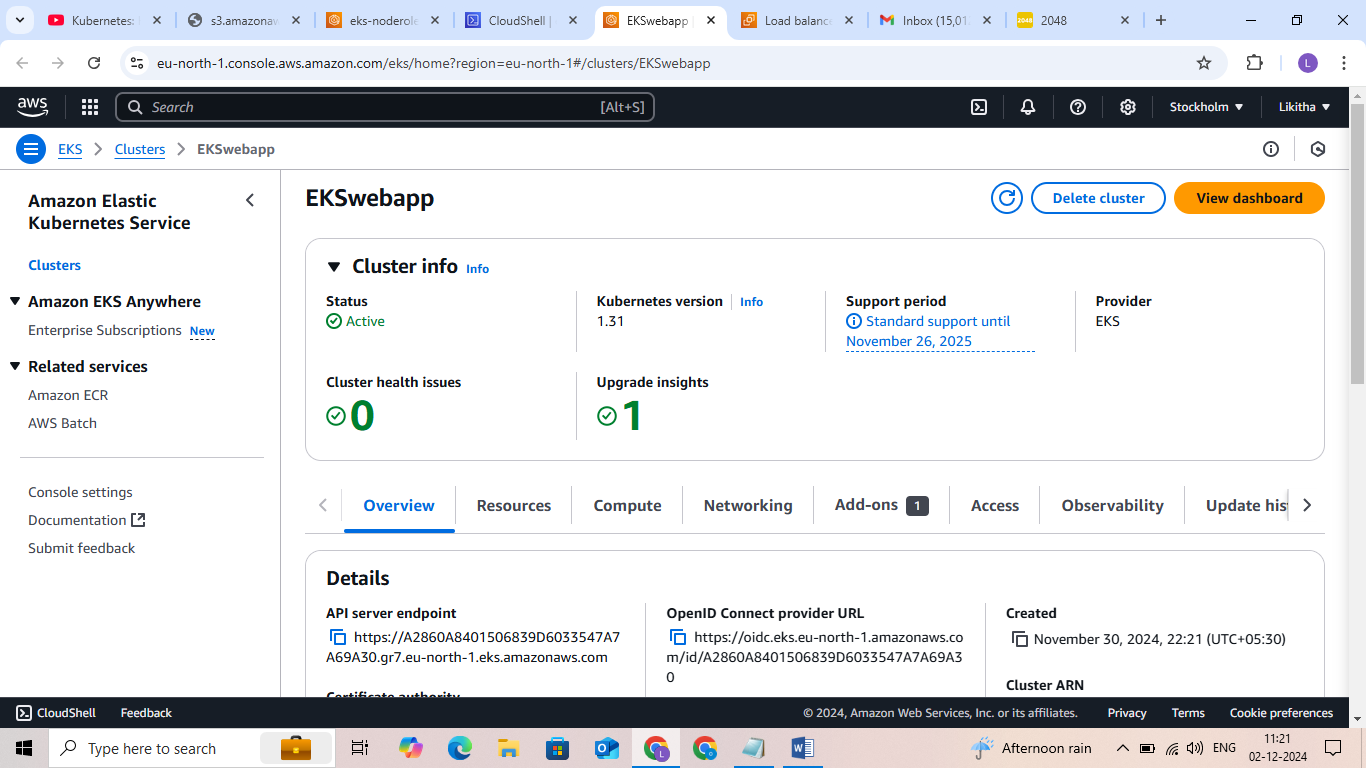
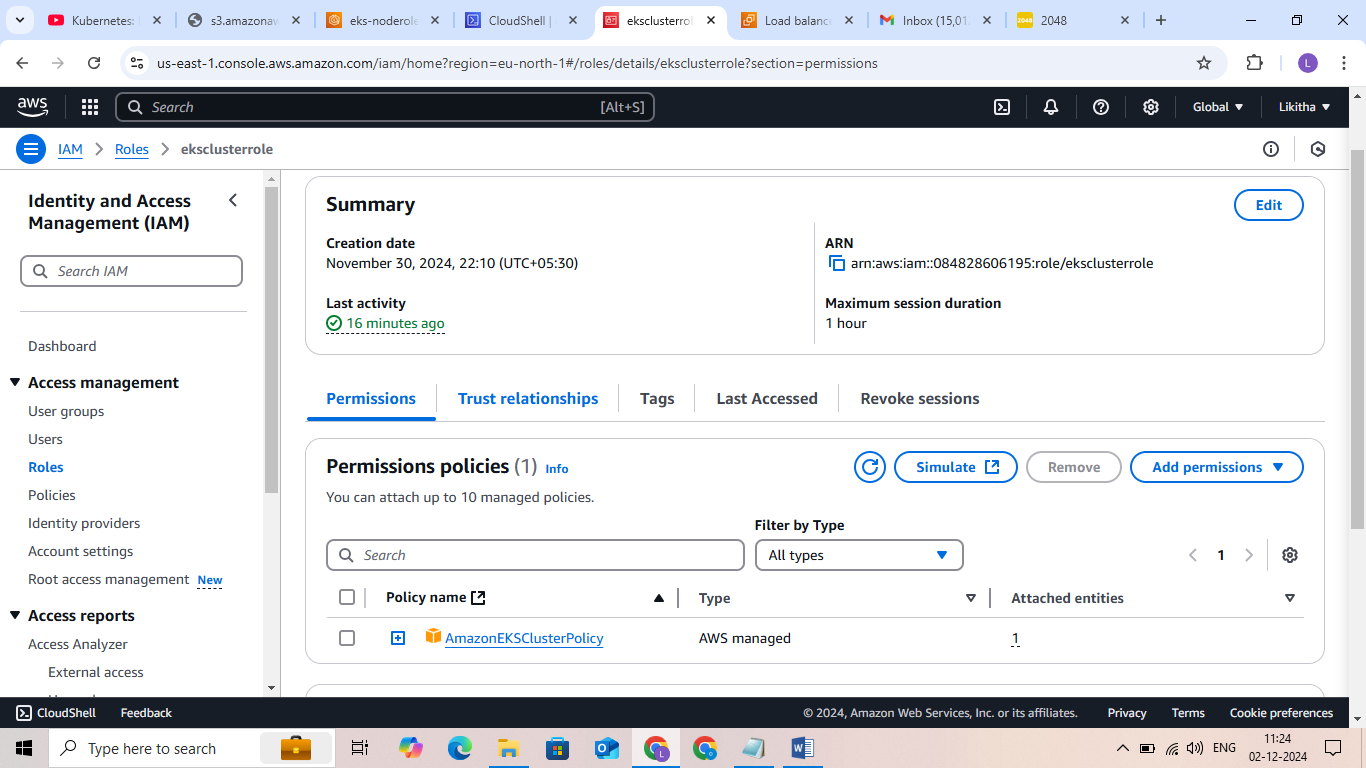
**2048 GAME**

**Create an EKS cluster and deploy 2048 game into that cluster**

**Task 1: Create an EKS cluster**

****

Created an IAM role 'eksclusterrole' with 1 policy attached: AmazonEKSClusterPolicy



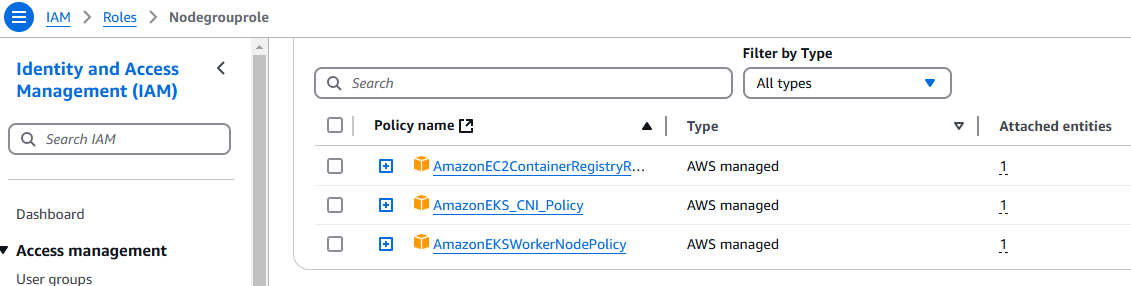
Create another IAM role Nodegrouprole' with 3 policies attached:

(Allows EC2 instances to call AWS services on your behalf.)

- AmazonEKSWorkerNodePolicy

- AmazonEC2ContainerRegistryReadOnly

- AmazonEKS\_CNI\_Policy



Choosen default VPC, with 2 or 3 subnets

Created a security group which open the ports 22, 80, 8080

**Task 2: Add Node Groups to our cluster**

Now, added the worker nodes where the pods can run

Open the cluster > Compute > Add NodeGrp

Created a Node group

**Task 3: Authenticate to this cluster**

Used Cloud shell in AWS

Created a kubeconfig file where it stores the credentials for EKS:

* aws eks update-kubeconfig --region region-code --name my-cluster

aws eks update-kubeconfig --region eu-north-1 --name EKSWebapp

created the config file in YAML to deploy 2048 game pod into the cluster

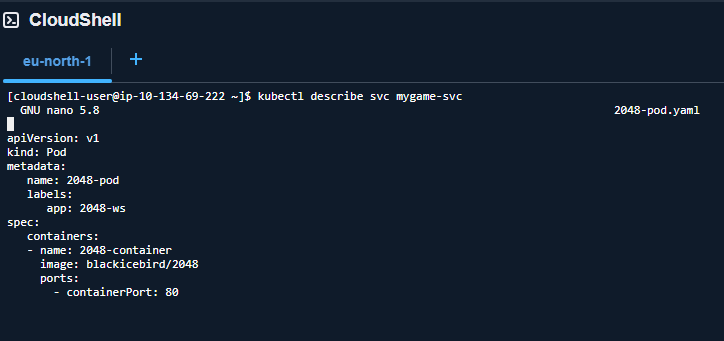
* nano 2048-pod.yaml

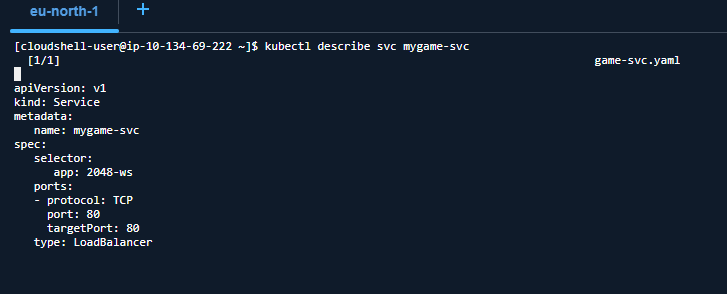
kubectl apply -f 2048-pod.yaml #pod/2048-pod created

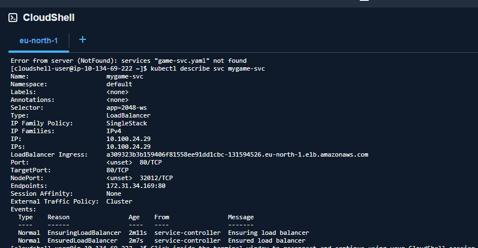
**Task 4: Setup Load Balancer Service**

Created the Service file in YAML to deploy 2048 game pod into the cluster

* nano mygame-svc.yaml
* kubectl apply -f mygame-svc.yaml
* kubectl describe svc mygame-svc # view details of the modified service







In EC2 console we get the DNS name of ELB and paste the DNS into address bar of the browser

It will show the 2048 game

Link - <http://a309323b3b159406f81558ee91dd1cbc-131594526.eu-north-1.elb.amazonaws.com/>

**Task – 5**

Monitoring solution configuration

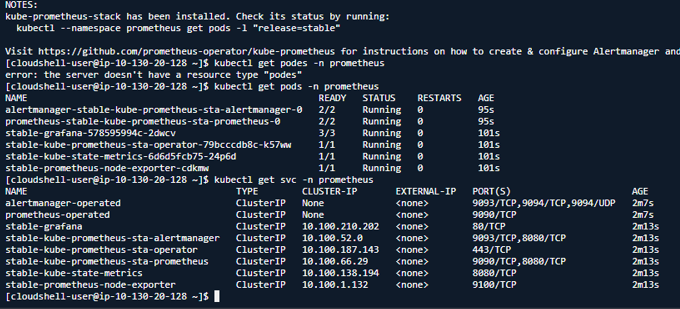
* 1 .Setup monitoring on EKS Cluster using Prometheus and Grafana .Through Helm I installed Prometheus and grafana

# Added prometheus Helm repo

* helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>
* helm install stable prometheus-community/kube-prometheus-stack -n Prometheus

# check if prometheus and grafana pods are running already

* kubectl get pods -n Prometheus
* kubectl get svc -n prometheus



prometheus and grafana have been installed successfully using Helm.

**Task – 6**

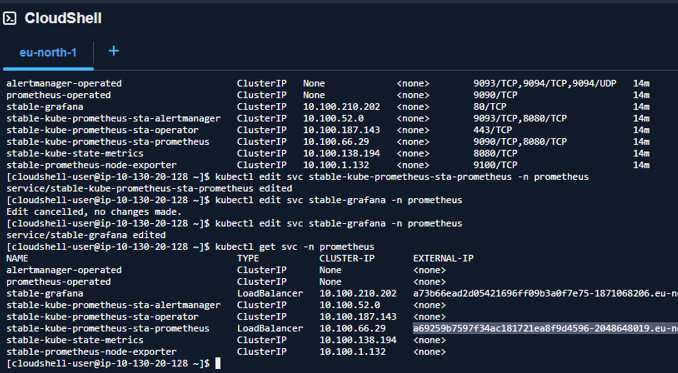
In order to make prometheus and grafana available outside the cluster, use LoadBalancer or NodePort instead of ClusterIP.

# Edit Prometheus Service

kubectl edit svc stable-kube-prometheus-sta-prometheus -n prometheus

# Edit Grafana Service

kubectl edit svc stable-grafana -n Prometheus



service is changed to LoadBalancer and we will also to get the Load Balancer URL to Access Prometheus ang grapana.