Deploying 2048-GAME on AWS EKS using YAML

(Service Type: Load Balancer)

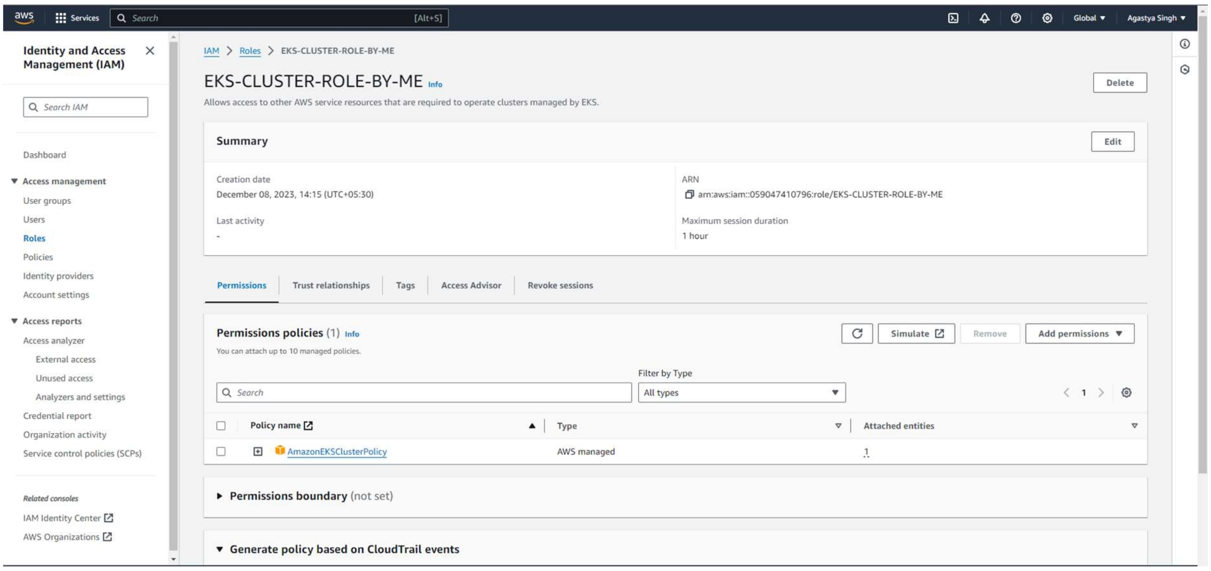
PROJECT OBJECTIVES:

- Deploy 2048-Game using AWS Elastic Kubernetes Service
- Make use of YAML files to deploy Pod and Service
- Establish Service type as LoadBalancer.

STEPS TAKEN TO ACHIEVE OBJECTIVES:

---CONTROL PLANE SET-UP---

- Creating Role for the cluster with ClusterPolicy attached.



EKS-CLUSTER-ROLE-BY-ME Info

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

Summary

Creation date
December 08, 2023, 14:15 (UTC+05:30)

ARN
arn:aws:iam::059047410796:role/EKS-CLUSTER-ROLE-BY-ME

Last activity
-

Maximum session duration
1 hour

Permissions policies (1) Info

You can attach up to 10 managed policies.

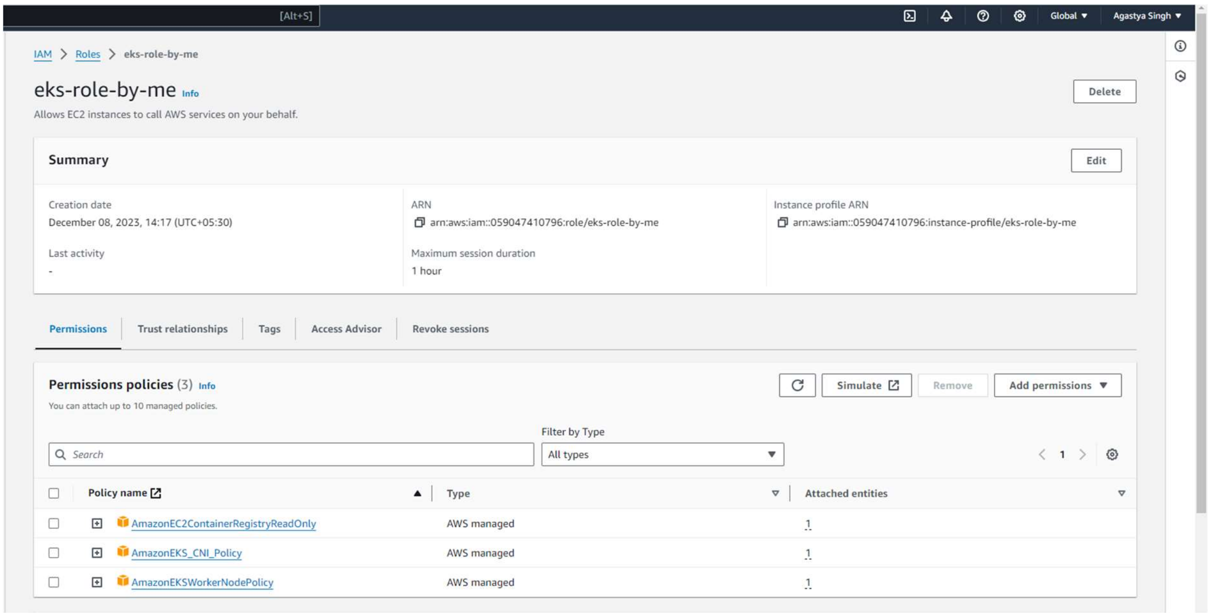
Filter by Type: All types

Policy name	Type	Attached entities
AmazonEKSClusterPolicy	AWS managed	1

Permissions boundary (not set)

Generate policy based on CloudTrail events

- Creating Role for Node Group with AmazonEKSWorkerNodePolicy, AmazonEC2ContainerRegistryReadOnly, AmazonEKS_CNI_Policy policies attached.



eks-role-by-me

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
December 08, 2023, 14:17 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::059047410796:role/eks-role-by-me

Maximum session duration
1 hour

Instance profile ARN
arn:aws:iam::059047410796:instance-profile/eks-role-by-me

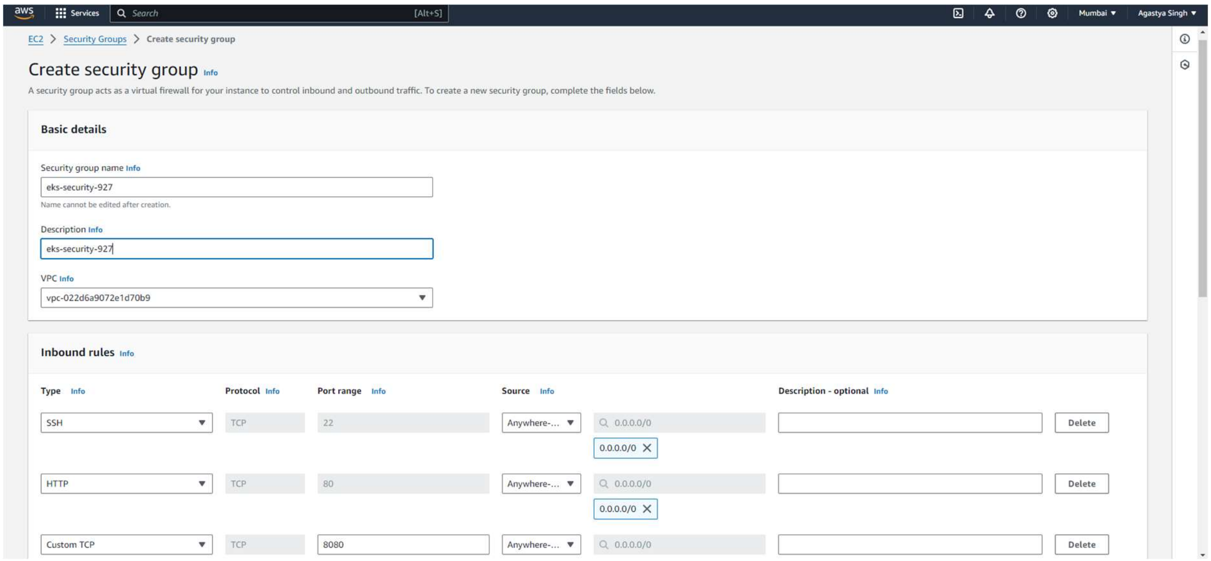
Permissions policies (3)

You can attach up to 10 managed policies.

Filter by Type: All types

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

- Creating Security groups to allow inbound traffic for ports 22(SSH), 80(HTTP) and 8080(TCP)



Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name
eks-security-927
Name cannot be edited after creation.

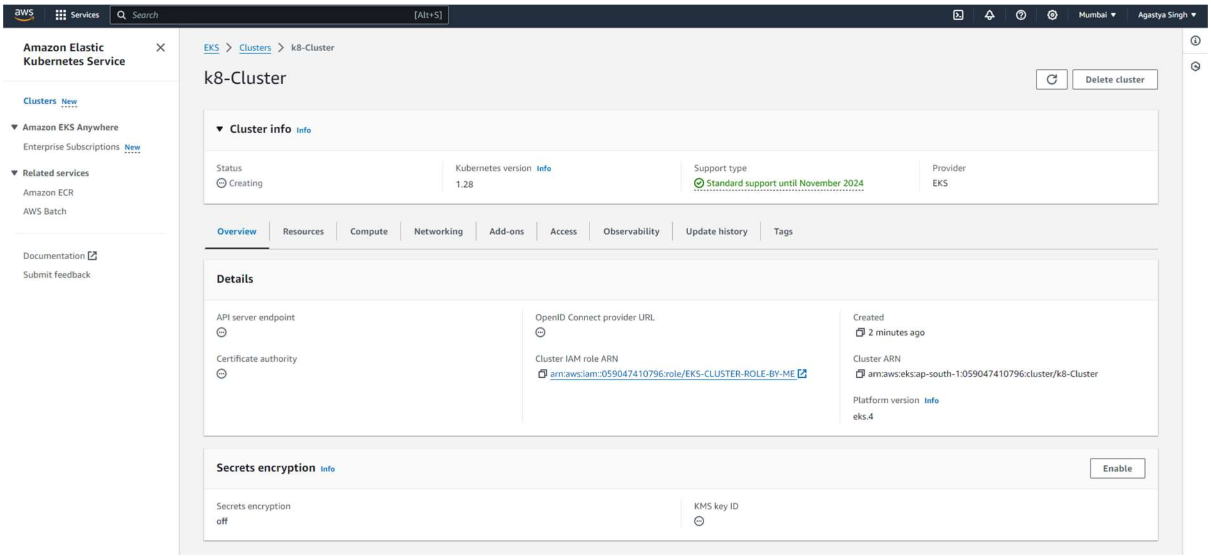
Description
eks-security-927

VPC
vpc-022d6a9072e1d70b9

Inbound rules

Type	Protocol	Port range	Source		Description - optional
SSH	TCP	22	Anywhere-...	0.0.0.0/0 0.0.0.0/0	
HTTP	TCP	80	Anywhere-...	0.0.0.0/0 0.0.0.0/0	
Custom TCP	TCP	8080	Anywhere-...	0.0.0.0/0	

- Now creating the EKS Cluster with roles attached as created above.



EKS > Clusters > k8-Cluster

k8-Cluster

Cluster info Info

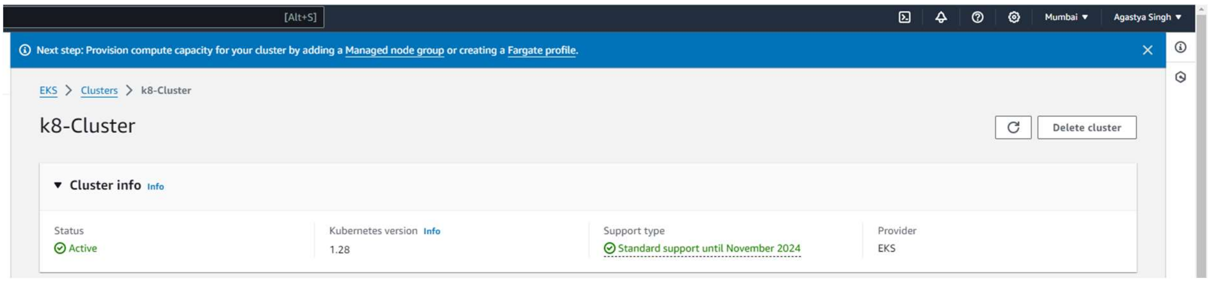
Status: Creating
Kubernetes version Info: 1.28
Support type: Standard support until November 2024
Provider: EKS

Details

API server endpoint: –
Certificate authority: –
OpenID Connect provider URL: –
Cluster IAM role ARN: arn:aws:iam::059047410796:role/EKS-CLUSTER-ROLE-BY-ME
Created: 2 minutes ago
Cluster ARN: arn:aws:eks:ap-south-1:059047410796:cluster/k8-Cluster
Platform version Info: eks.4

Secrets encryption Info

Secrets encryption: off
KMS key ID: –



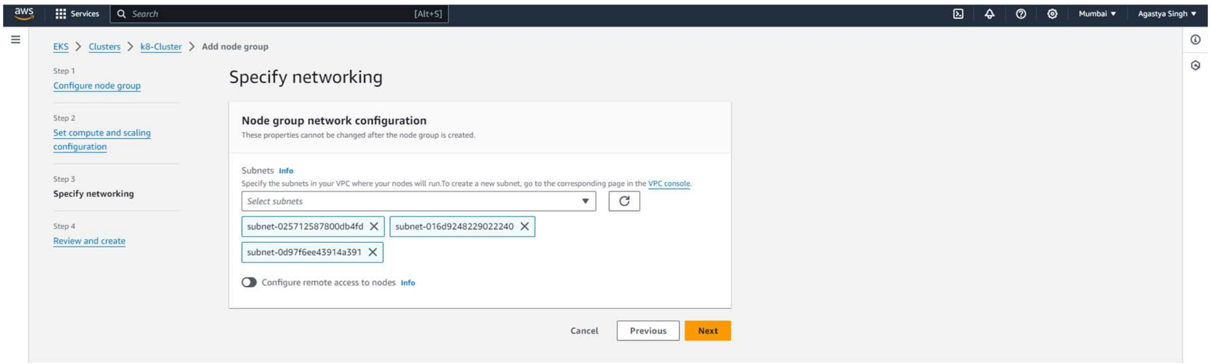
Next step: Provision compute capacity for your cluster by adding a Managed node group or creating a Fargate profile.

EKS > Clusters > k8-Cluster

k8-Cluster

Cluster info Info

Status: Active
Kubernetes version Info: 1.28
Support type: Standard support until November 2024
Provider: EKS



EKS > Clusters > k8-Cluster > Add node group

Step 1 Configure node group
Step 2 Set compute and scaling configuration
Step 3 Specify networking
Step 4 Review and create

Specify networking

Node group network configuration

These properties cannot be changed after the node group is created.

Subnets Info

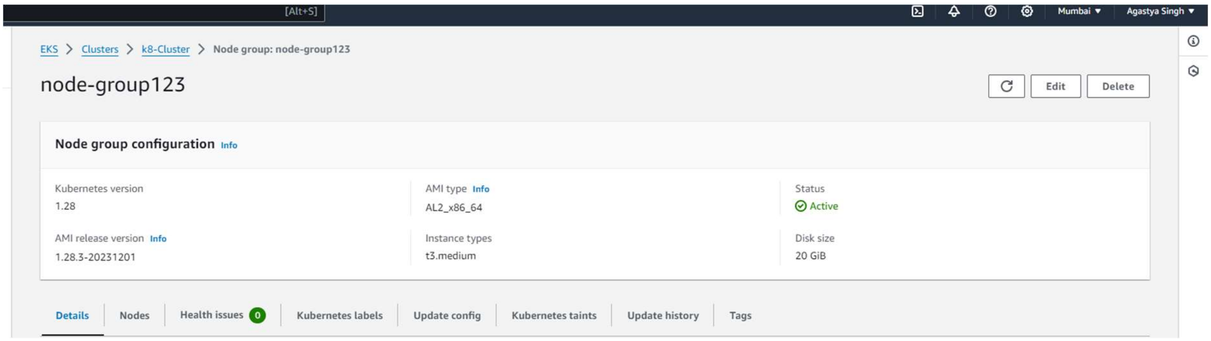
Specify the subnets in your VPC where your nodes will run.To create a new subnet, go to the corresponding page in the VPC console.

Select subnets

subnet-025712587800db4fd
subnet-016d9248229022240
subnet-0d97f6ee43914a391

○ Configure remote access to nodes Info

- Then adding Node Group to the cluster from Compute Section.



node-group123

Node group configuration

Kubernetes version
1.28

AMI type
AL2_x86_64

Status
Active

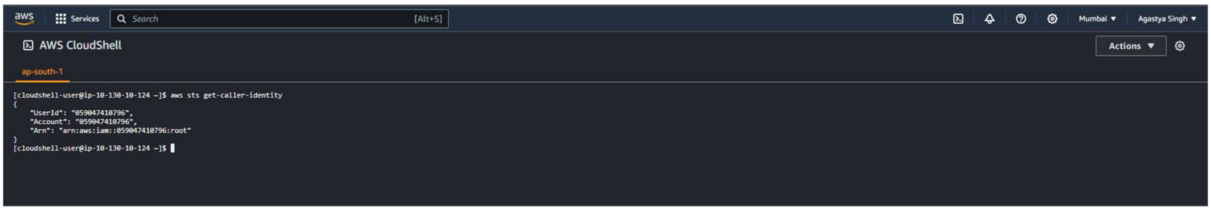
AMI release version
1.28.3-20231201

Instance types
t3.medium

Disk size
20 GiB

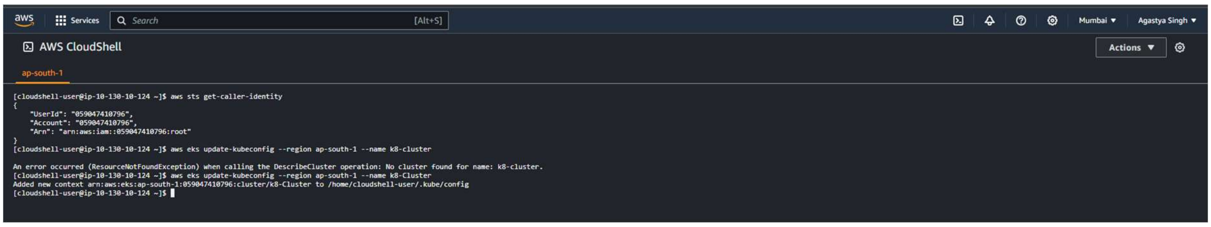
--DATA PLANE SET-UP---

- Authenticating and checking credentials

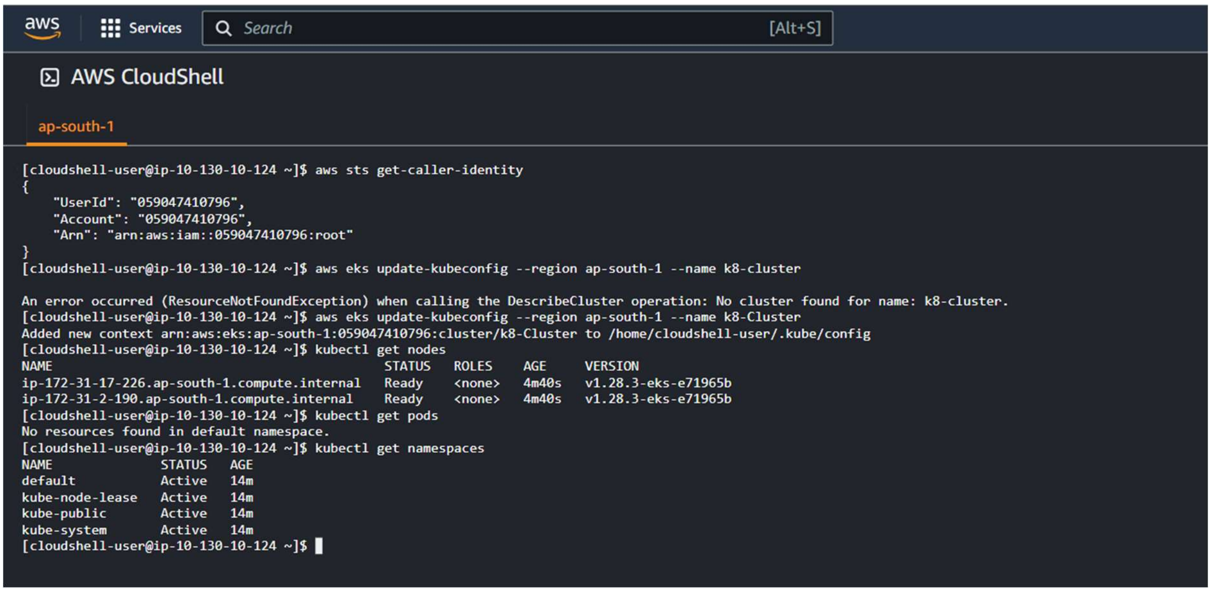


```
[cloudshell-user@ip-10-130-10-124 ~]$ aws sts get-caller-identity
{
    "UserId": "059047410796",
    "Account": "059047410796",
    "Arn": "arn:aws:iam::059047410796:root"
}
[cloudshell-user@ip-10-130-10-124 ~]$ 
```

- Updating the kube config file to set region and cluster name to be able to connect to the cluster created in above steps.
(command: aws eks update-kubeconfig --region region-code --name my-cluster)

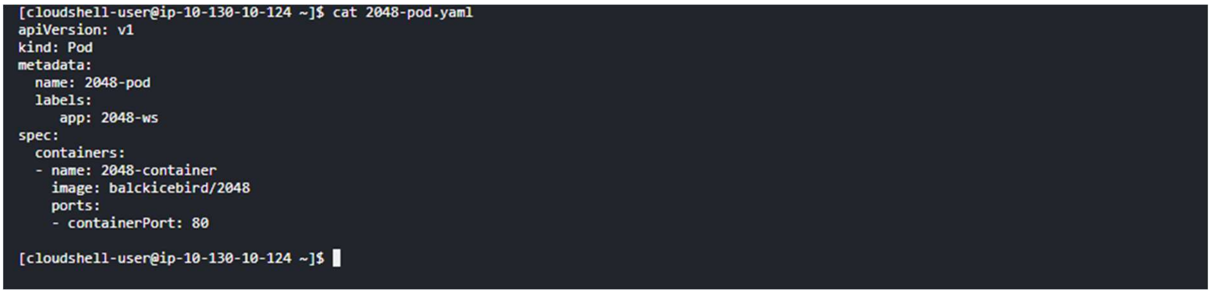


```
[cloudshell-user@ip-10-130-10-124 ~]$ aws sts get-caller-identity
{
    "UserId": "059047410796",
    "Account": "059047410796",
    "Arn": "arn:aws:iam::059047410796:root"
}
[cloudshell-user@ip-10-130-10-124 ~]$ aws eks update-kubeconfig --region ap-south-1 --name k8-cluster

An error occurred (ResourceNotFoundException) when calling the DescribeCluster operation: No cluster found for name: k8-cluster.
[cloudshell-user@ip-10-130-10-124 ~]$ aws eks update-kubeconfig --region ap-south-1 --name k8-Cluster
Added new context arn:aws:eks:ap-south-1:059047410796:cluster/k8-Cluster to /home/cloudshell-user/.kube/config
[cloudshell-user@ip-10-130-10-124 ~]$ 
```



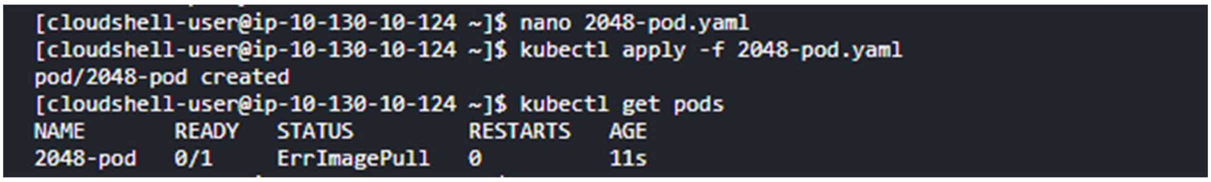
```
[cloudshell-user@ip-10-130-10-124 ~]$ aws sts get-caller-identity
{
    "UserId": "059047410796",
    "Account": "059047410796",
    "Arn": "arn:aws:iam::059047410796:root"
}
[cloudshell-user@ip-10-130-10-124 ~]$ aws eks update-kubeconfig --region ap-south-1 --name k8-cluster

An error occurred (ResourceNotFoundException) when calling the DescribeCluster operation: No cluster found for name: k8-cluster.
[cloudshell-user@ip-10-130-10-124 ~]$ aws eks update-kubeconfig --region ap-south-1 --name k8-Cluster
Added new context arn:aws:eks:ap-south-1:059047410796:cluster/k8-Cluster to /home/cloudshell-user/.kube/config
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl get nodes
NAME                                       STATUS   ROLES    AGE     VERSION
ip-172-31-17-226.ap-south-1.compute.internal   Ready    <none>   4m40s   v1.28.3-eks-e71965b
ip-172-31-2-190.ap-south-1.compute.internal    Ready    <none>   4m40s   v1.28.3-eks-e71965b
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl get pods
No resources found in default namespace.
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   14m
kube-node-lease   Active   14m
kube-public       Active   14m
kube-system       Active   14m
[cloudshell-user@ip-10-130-10-124 ~]$ 
```

- Creating YAML File for Pods. (typo: blackicebird/2048, corrected it later :)



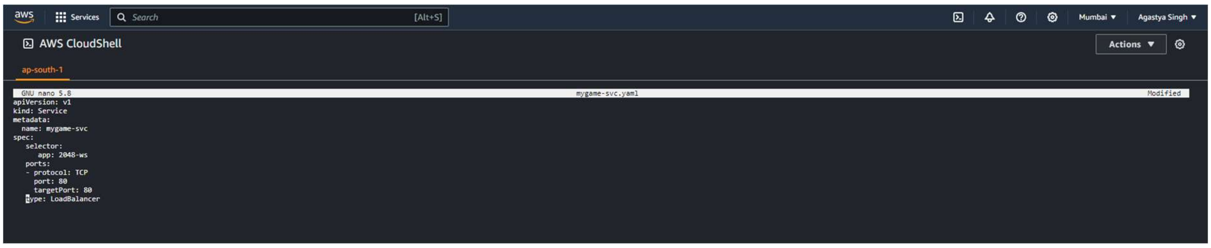
```
[cloudshell-user@ip-10-130-10-124 ~]$ cat 2048-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: 2048-pod
  labels:
     app: 2048-ws
spec:
  containers:
  - name: 2048-container
    image: balckicebird/2048
    ports:
    - containerPort: 80

[cloudshell-user@ip-10-130-10-124 ~]$ 
```

- Applying YAML FILE



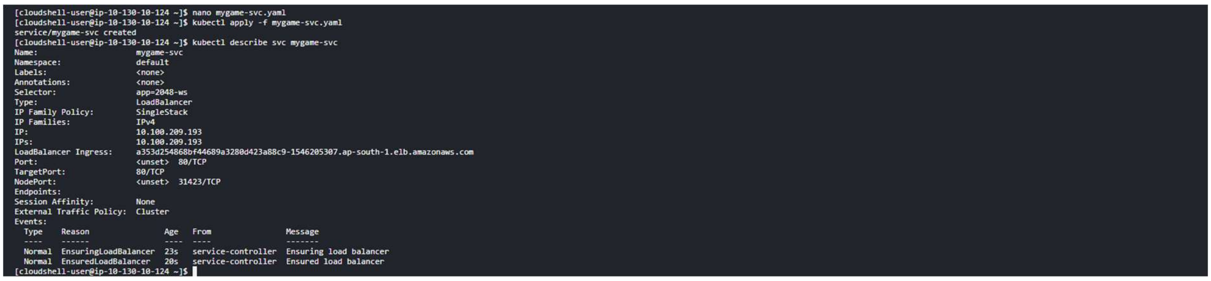
```
[cloudshell-user@ip-10-130-10-124 ~]$ nano 2048-pod.yaml
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl apply -f 2048-pod.yaml
pod/2048-pod created
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl get pods
NAME       READY   STATUS         RESTARTS   AGE
2048-pod   0/1     ErrImagePull   0          11s
```

- Now creating YAML File for applying Service type as Load Balancer.



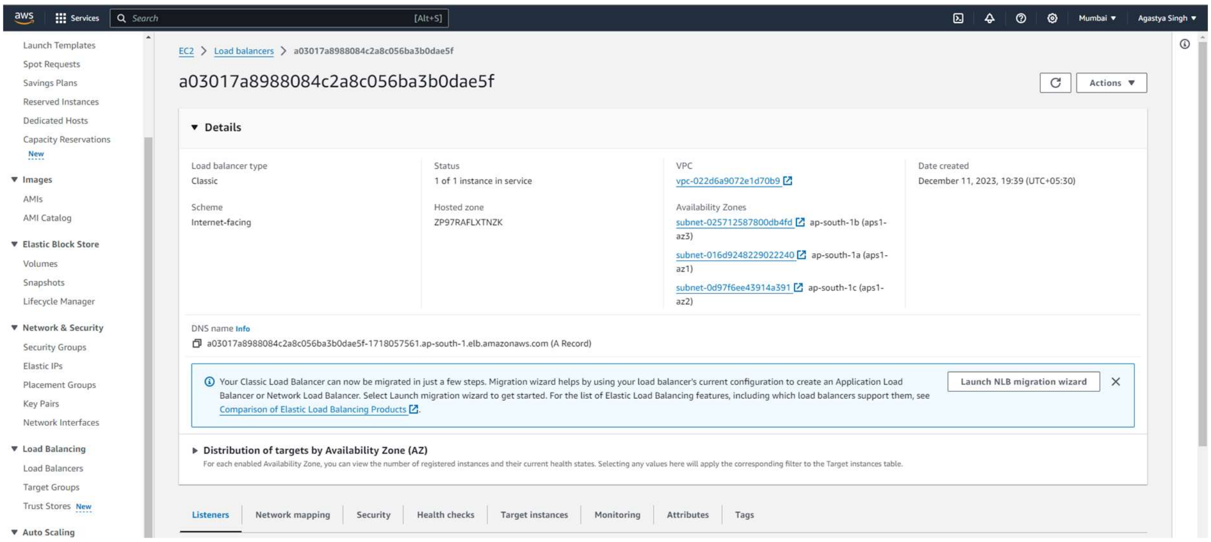
```
  GNU nano 5.8                                  mygame-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: mygame-svc
spec:
   selector:
      app: 2048-ws
   ports:
   - protocol: TCP
     port: 80
     targetPort: 80
   type: LoadBalancer
```

- Service is up and running



```
[cloudshell-user@ip-10-130-10-124 ~]$ nano mygame-svc.yaml
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl apply -f mygame-svc.yaml
service/mygame-svc created
[cloudshell-user@ip-10-130-10-124 ~]$ kubectl describe svc mygame-svc
Name:                     mygame-svc
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=2048-ws
Type:                     LoadBalancer
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.100.209.193
IPs:                      10.100.209.193
LoadBalancer Ingress:     a353d254868bf44689a3280d423a88c9-1546205307.ap-south-1.elb.amazonaws.com
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31423/TCP
Endpoints:
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age   From                Message
  ----    ------                ----  ----                -------
  Normal  EnsuringLoadBalancer  23s   service-controller  Ensuring load balancer
  Normal  EnsuredLoadBalancer   20s   service-controller  Ensured load balancer
[cloudshell-user@ip-10-130-10-124 ~]$
```

- Checking Load Balancer address to try and connect through a browser window.



a03017a8988084c2a8c056ba3b0dae5f

Load balancer type: Classic

Scheme: Internet-facing

Status: 1 of 1 instance in service

Hosted zone: ZP97RAFLXTNZK

VPC: vpc-022d6a9072e1d70b9

Availability Zones: subnet-025712587800db4fd ap-south-1b (aps1-az3); subnet-016d9248229022240 ap-south-1a (aps1-az1); subnet-0d97f6ee43914a391 ap-south-1c (aps1-az2)

Date created: December 11, 2023, 19:39 (UTC+05:30)

DNS name Info: a03017a8988084c2a8c056ba3b0dae5f-1718057561.ap-south-1.elb.amazonaws.com (A Record)

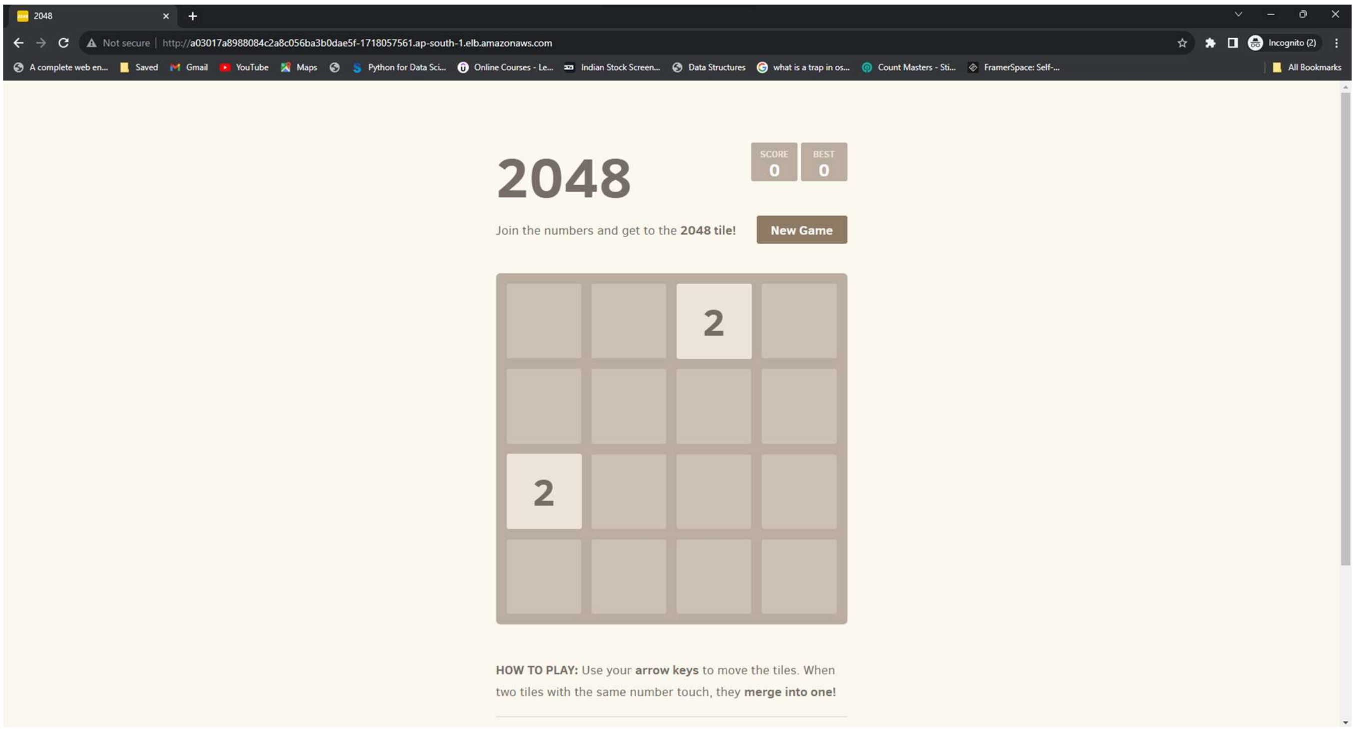
Your Classic Load Balancer can now be migrated in just a few steps. Migration wizard helps by using your load balancer's current configuration to create an Application Load Balancer or Network Load Balancer. Select Launch migration wizard to get started. For the list of Elastic Load Balancing features, including which load balancers support them, see Comparison of Elastic Load Balancing Products.

▶ Distribution of targets by Availability Zone (AZ)

For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

OUTPUT:

- **Game is successfully deployed and accessible through DNS address of Load Balancer through local browser.**



http://a03017a8988084c2a8c056ba3b0dae5f-1718057561.ap-south-1.elb.amazonaws.com

2048

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!**