#KubernetesIn30Days challenge :-

Demo-4

#Day 20 :-

Deploy 2048 game on Managed K8s cluster - EKS on AWS cloud using ALB Ingress controller.

EKS - Managed control Plane

Fargate/EC2 - Managed Data plane (Nodes/container hosting service)

Illustrative steps :-

① create Access keys for IAM user having Admin Access and store it confidentially.

② Have an EC2/local VM/local Machine which having awscli, kubectl, eksctl installed. Follow official documentation to install on respective OS requirements

③ configure aws access keys in AWS CLI to use aws resources.

④ confirm installations using below cmds :

```
# aws --version
# kubectl version
# eksctl version
```

⑤ Use below cmd to start eks cluster using Fargate service.

```
#eksctl create cluster --name cluster-game
--region us-east-1 --fargate
```

It'll take approximately 10-15 mins depending upon your connection speed.

⑥ Meanwhile we can check the status of cluster in AWS Management console at EKS service and this cluster will be created using cloudformation stacks.


Flow of Execution.

⑦ once the cluster is created successfully, Explore the cluster details from AWS Mgmt console. and update your kubeconfig file with below cmd :

```
#aws eks update-kubeconfig --name
cluster-game --region us-east-1
```

(8) check the pods and Nodes, service in the cluster using kubectl cmds.

```
#kubectl get nodes
#kubectl get pods
#kubectl get svc.
```

(9) we have defawt fargate profile but here we are creating a new fargate profile to isolate deployment using below cmd:

```
#eksctl create fargateprofile \
    -- cluster cluster-game
    --region us-east-1
    -- name alb-sample-app
    -- namespace game-2048
```

(10) check the Management consolc of Eks and cluster-game and Fargate profile. alb-sample-app fargate profile will be created. It should be in Active state.

(11) Now we need to create namespace game-2048, deployment deployment-2048, service service-2048 and Ingress Ingress-2048 resources using the file given in the caption. and use below

cmd to apply changes.

```
#kubectl apply -f <URL>
```

All Resources should be created. check and confirm the resources creation.

(12) we need to provide access to cluster using IAM as Open ID connector using below cmd.

```
# eksctl utils associate-iam-oidc-provider
--cluster cluster-game --approve
```

(13) Download the IAM-policy.json file using the URL given. Now we need to create this policy.

```
# aws iam create-policy \
    --policy-name AWSLoadBalancerControllerIAM-
                  -Policy \
    --policy-document file://iam_policy.json.
```

you'll be prompted back with the details of IAM policy.

(14) Now we need to create IAM service Account for the cluster and also attach the policy to the service account using below cmd:

```
#eksctl create iamserviceaccount \
  --cluster = cluster-game \
  -- namespace = kube-system \
  -- name = aws-load-balancer-controller \
  -- role-name AmazonEKSLoadBalancerControllerRole \
  -- attach-policy-arn = arn:aws:iam: AWS A/c id :-
        - policy/AWSLoadBalancerControllerIAMPolicy \
  -- approve.
```

* policy we created using policy document

(15) Make sure helm installed on your VM | EC2. We are using helm charts to install Ingress-controller for ALB. Follow the below command :

```
# helm repo add eks https://aws.github.io/
        eks-charts

# helm repo update

#helm install aws-load-balancer-controller eks/aws-load-balancer-controller \
 -n kube system \
--set clusterName = cluster-game \
--set serviceAccount.create = false \
-- set serviceAccount.name = aws-load-balancer-controller \
-- set region = us-east-1 \
-- set vpcid = <your cluster VPC Id>
```

Your ALB Ingress controller will be installed using helm charts.

check the deploy in kube-system namespace and you'll be seen two replicas of aws-load-balancer-controller.

(16) check the ALB servic in the AWS Mgmt Console and you'll seen a provisioned ALB and target groups, autoscaling Groups etc and there will be an LB Endpoint to access our game hit that End point in browser.

Alias Record of LB → URL to Application.

(Here default listner is 80 and if you want to use HTTPS you should have a domain purchased to use https protocol)

Tada! We are able to access the Application i.e game and you can also play it.

optional :-

(17) Here I had purchased domain so here I wanted to access this app at my subdomain such as :

vprofile.rajasekhardevopsproject.in

↓ Host/sub-domain

↓ Actual domain I hold.

(18) Now, I added the A Record in my domain DNS i.e LB endpoint and given name as vprofile. This makes my app is accessible at http://vprofile.rajasekhardevopsproject.in

↓ protocol is still the http.

(19) you need a certificate to authenticate yowself for using https site. so, Request a certificate at ACM @ AWS. and verify that by adding CNAME Records to Domain DNS both name and value.

(20) within short time after adding yow certificate will be issued and Now add HTTPS Listener at ALB and link the certificate that is issued already.

It is important to change the appropriate security group rules of the instances accordingly.

(21) Now you can access your Encrypted 2048 game at our very own domain i.e.

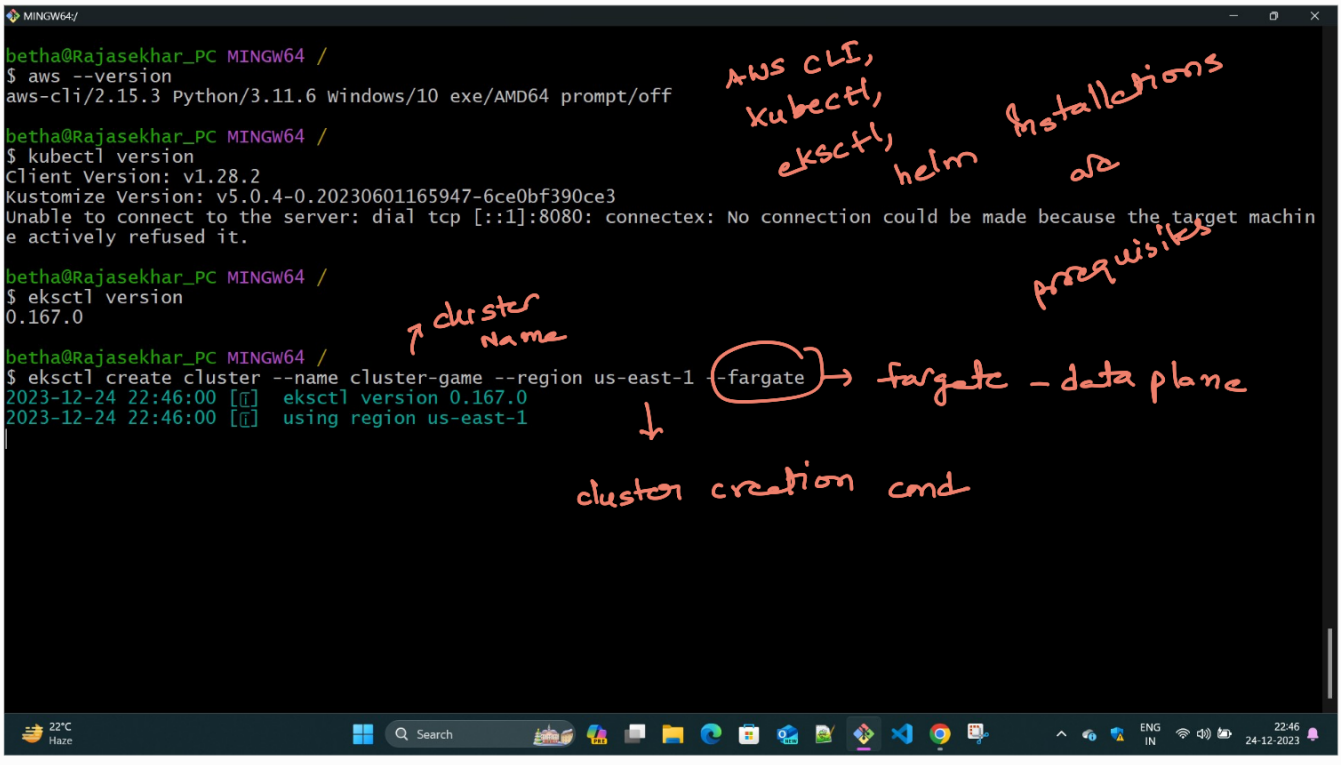
https://vprofile.rajasekhasdevopsproject.in/

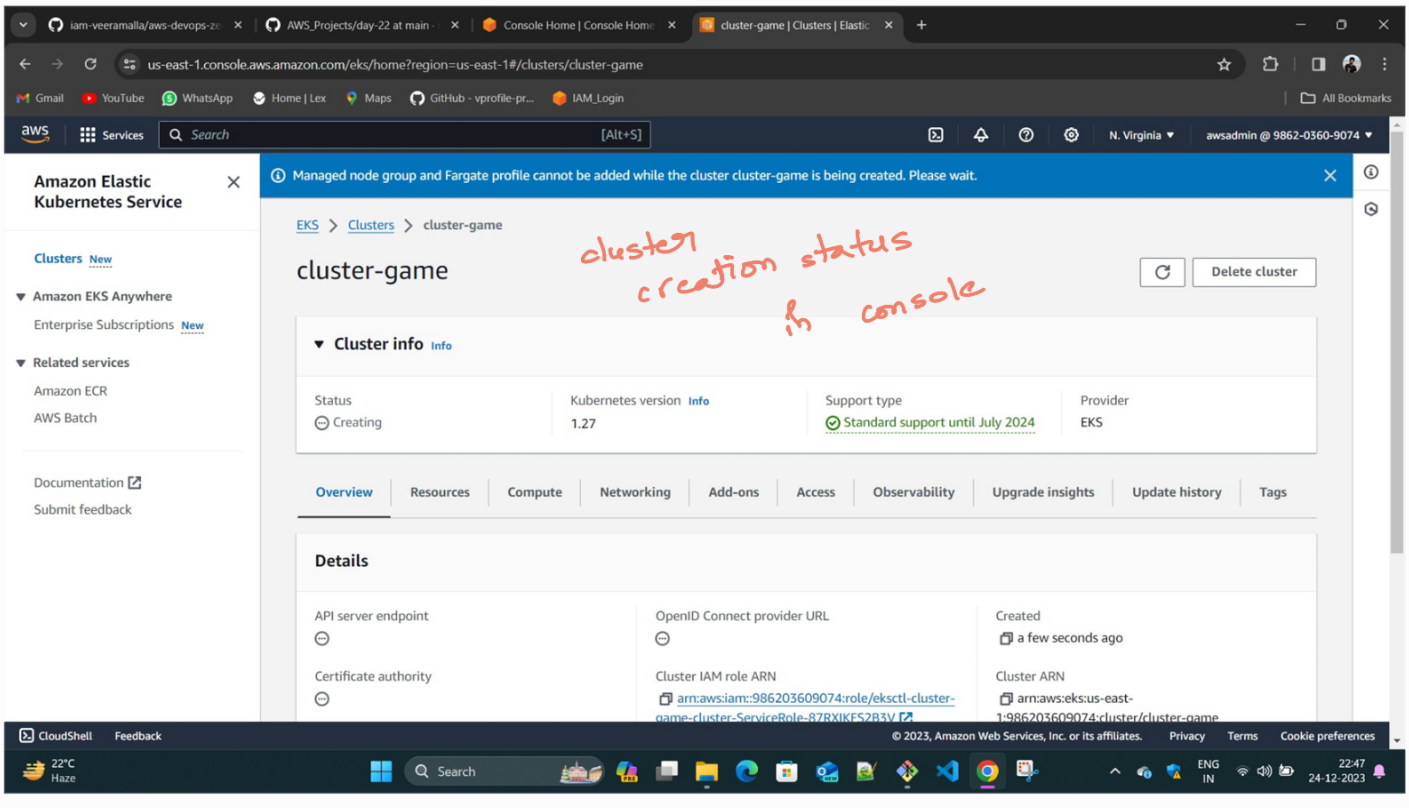
You can Explore the public certificate attached to the site.

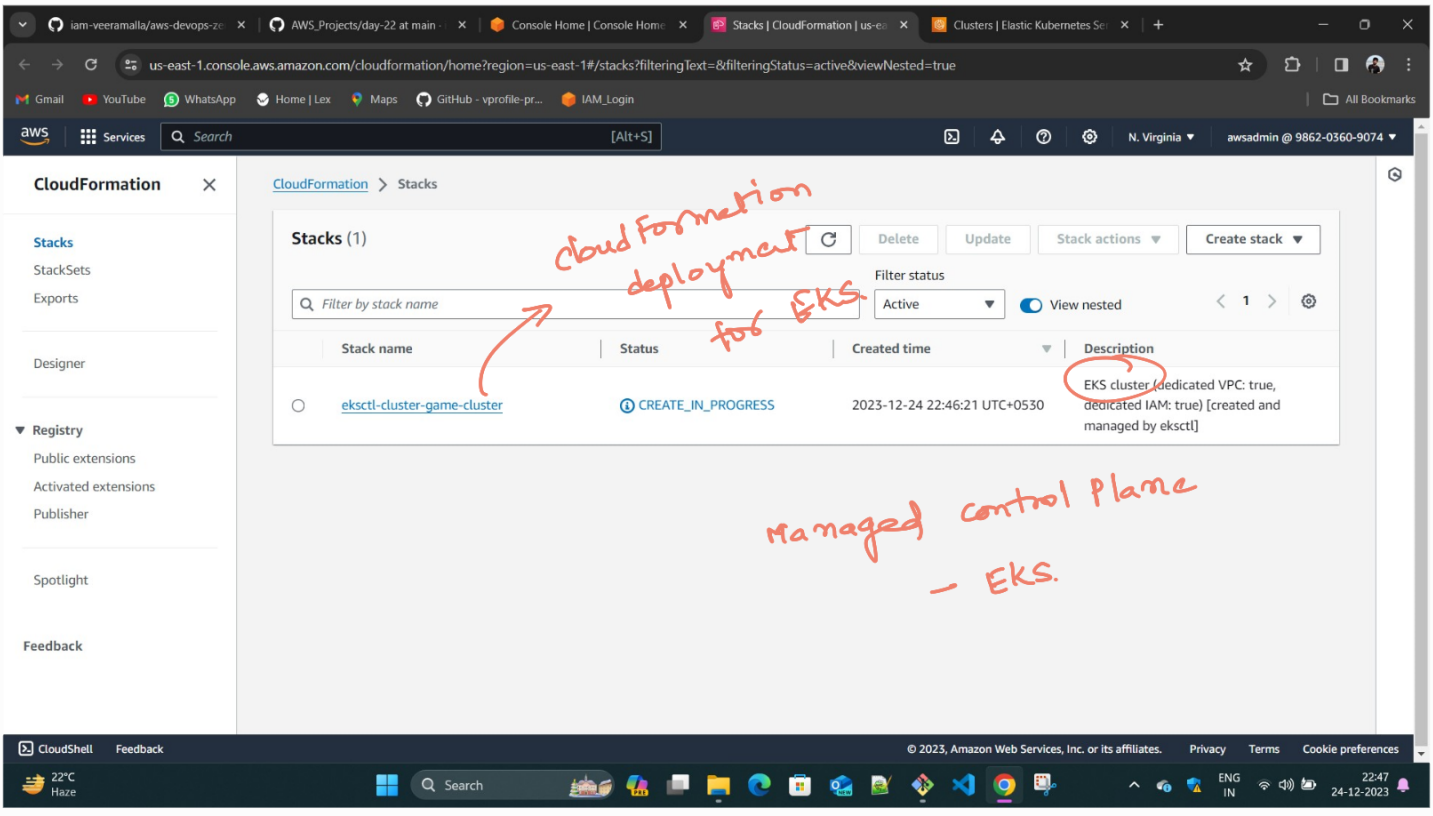
This is a Bit complex project and having a lot to understand. A Big Thanks to Abhishek Veeramalla for this demo @ YT.

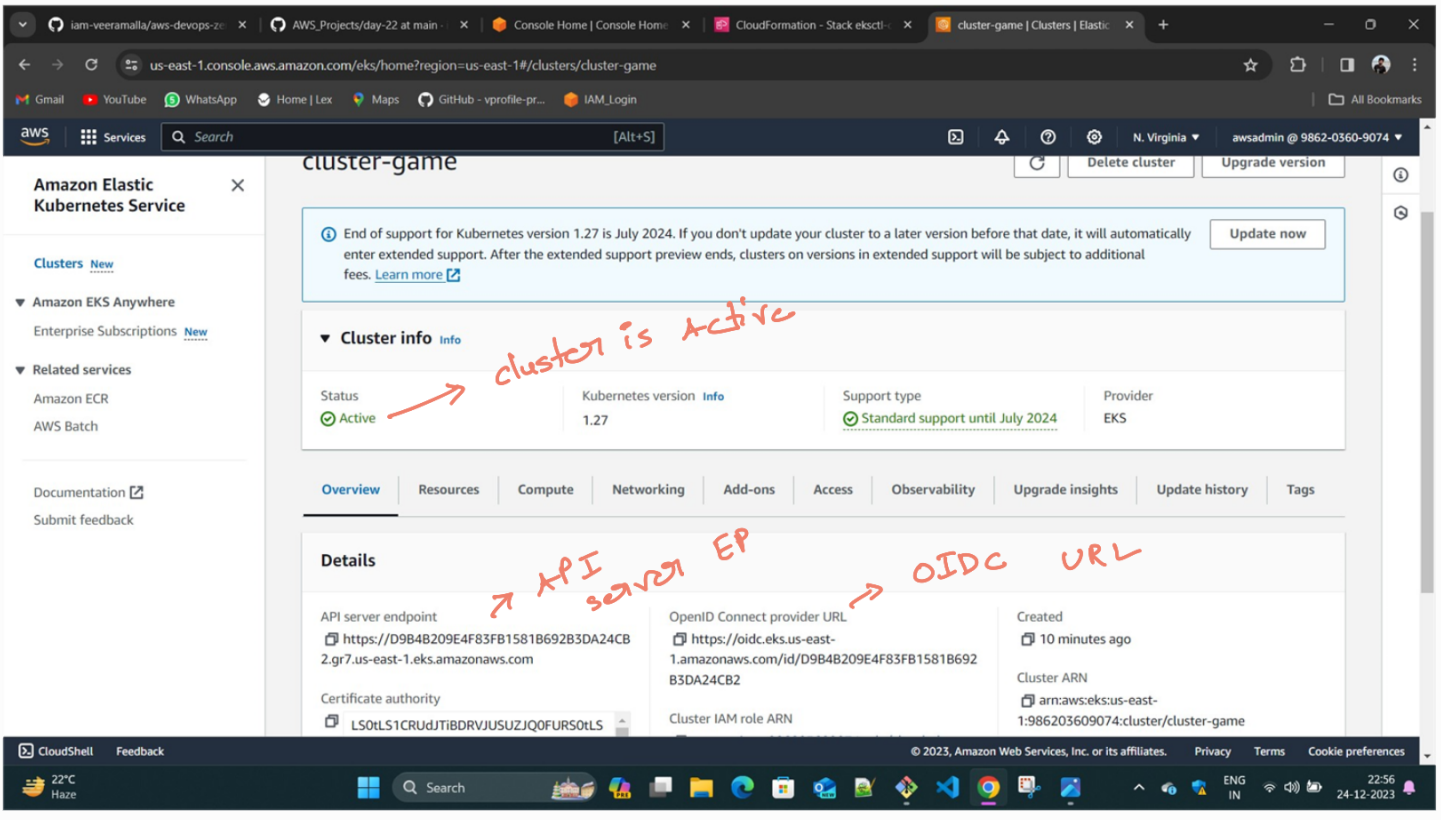
I Just added a toppings to his lecture by putting optional portional to encrpt your website using https.

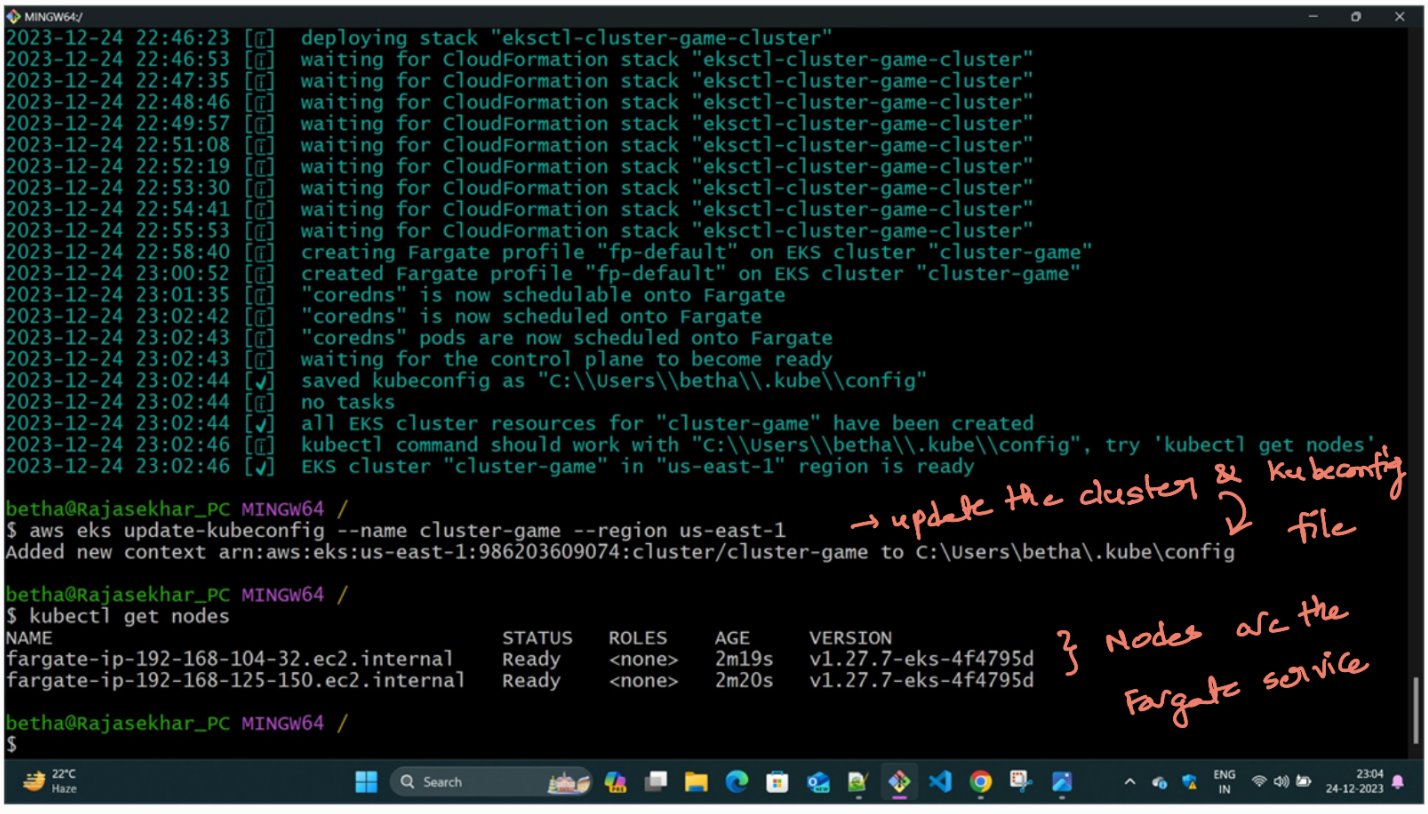
You can Explore my work samples and as well as steps reference screenshots below. It'll be very useful if you stuck check the screenshots. Thanks for Reading.

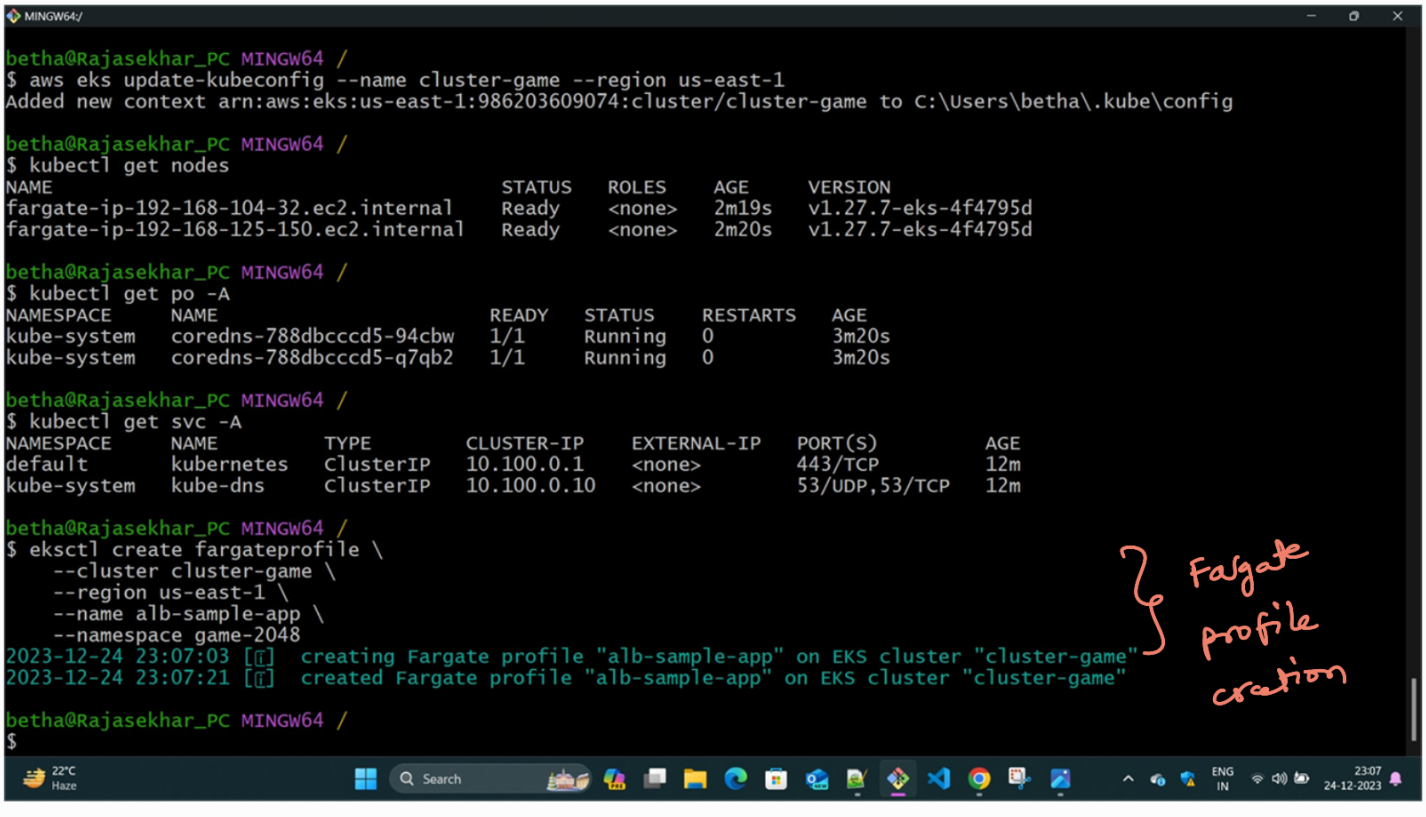

MINGW64:/
betha@Rajasekhar_PC MINGW64 /
$ aws --version
aws-cli/2.15.3 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
betha@Rajasekhar_PC MINGW64 /
$ kubectl version
Client Version: v1.28.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Unable to connect to the server: dial tcp [::1]:8080: connectex: No connection could be made because the target machine actively refused it.
betha@Rajasekhar_PC MINGW64 /
$ eksctl version
0.167.0
betha@Rajasekhar_PC MINGW64 /
$ eksctl create cluster --name cluster-game --region us-east-1 --fargate
2023-12-24 22:46:00 [ℹ] eksctl version 0.167.0
2023-12-24 22:46:00 [ℹ] using region us-east-1
AWS CLI, Kubectl, eksctl, helm installations as prerequisites
cluster Name
fargate - data plane
cluster creation cmd

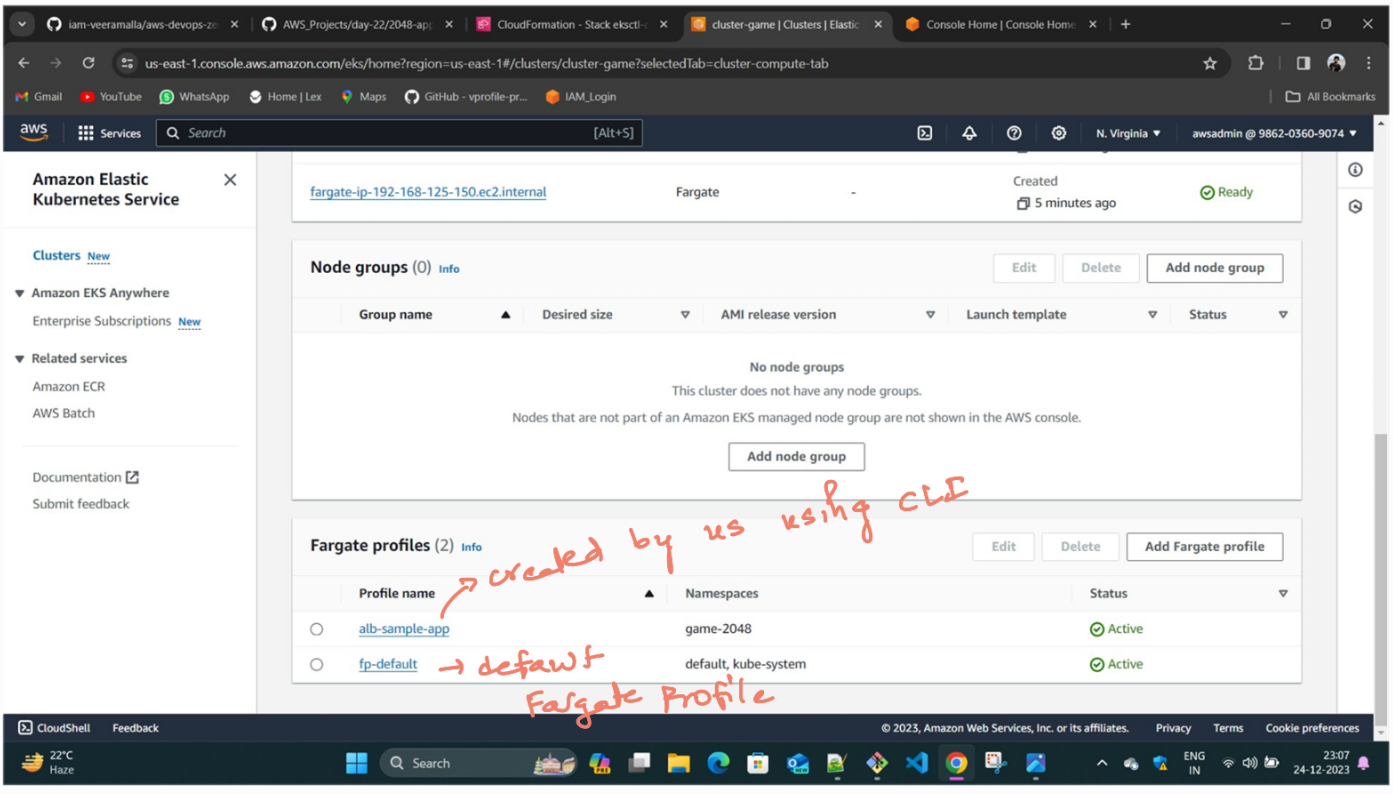

Amazon Elastic Kubernetes Service
Managed node group and Fargate profile cannot be added while the cluster cluster-game is being created. Please wait.
EKS > Clusters > cluster-game
cluster-game
cluster creation status in console
Delete cluster
Cluster info
Status
Creating
Kubernetes version
1.27
Support type
Standard support until July 2024
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Access
Observability
Upgrade insights
Update history
Tags
Details
API server endpoint
OpenID Connect provider URL
Created
a few seconds ago
Certificate authority
Cluster IAM role ARN
Cluster ARN


us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringText=&filteringStatus=active&viewNested=true
CloudFormation
Stacks
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Spotlight
Feedback
CloudFormation > Stacks
Stacks (1)
cloudformation deployment for EKS
Delete
Update
Stack actions
Create stack
Filter by stack name
Filter status
Active
View nested
Stack name
Status
Created time
Description
eksctl-cluster-game-cluster
CREATE_IN_PROGRESS
2023-12-24 22:46:21 UTC+0530
EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]
managed control plane - EKS.
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters/cluster-game
Amazon Elastic Kubernetes Service
Clusters New
Amazon EKS Anywhere
Enterprise Subscriptions New
Related services
Amazon ECR
AWS Batch
Documentation
Submit feedback
cluster-game
Delete cluster
Upgrade version
End of support for Kubernetes version 1.27 is July 2024. If you don't update your cluster to a later version before that date, it will automatically enter extended support. After the extended support preview ends, clusters on versions in extended support will be subject to additional fees. Learn more
Update now
Cluster info Info
cluster is Active
Status
Active
Kubernetes version Info
1.27
Support type
Standard support until July 2024
Provider
EKS
Overview
Resources
Compute
Networking
Add-ons
Access
Observability
Upgrade insights
Update history
Tags
Details
API server EP
OIDC URL
API server endpoint
https://D9B4B209E4F83FB1581B692B3DA24CB2.gr7.us-east-1.eks.amazonaws.com
OpenID Connect provider URL
https://oidc.eks.us-east-1.amazonaws.com/id/D9B4B209E4F83FB1581B692B3DA24CB2
Created
10 minutes ago
Certificate authority
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS
Cluster IAM role ARN
Cluster ARN
arn:aws:eks:us-east-1:986203609074:cluster/cluster-game
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

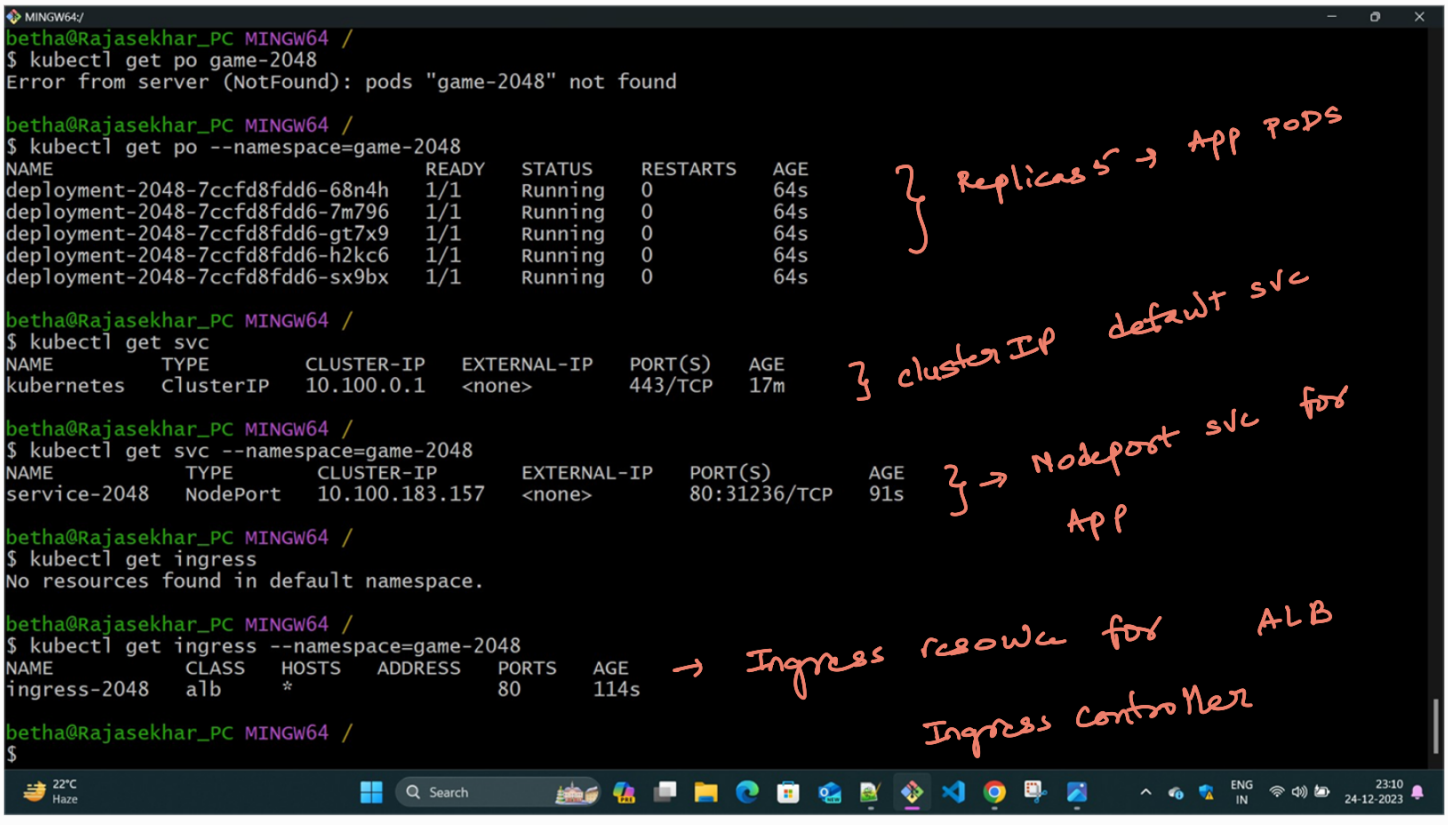

MINGW64:/
2023-12-24 22:46:23 [ℹ] deploying stack "eksctl-cluster-game-cluster"
2023-12-24 22:46:53 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:47:35 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:48:46 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:49:57 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:51:08 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:52:19 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:53:30 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:54:41 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:55:53 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-cluster"
2023-12-24 22:58:40 [ℹ] creating Fargate profile "fp-default" on EKS cluster "cluster-game"
2023-12-24 23:00:52 [ℹ] created Fargate profile "fp-default" on EKS cluster "cluster-game"
2023-12-24 23:01:35 [ℹ] "coredns" is now schedulable onto Fargate
2023-12-24 23:02:42 [ℹ] "coredns" is now scheduled onto Fargate
2023-12-24 23:02:43 [ℹ] "coredns" pods are now scheduled onto Fargate
2023-12-24 23:02:43 [ℹ] waiting for the control plane to become ready
2023-12-24 23:02:44 [✓] saved kubeconfig as "C:\\Users\\betha\\.kube\\config"
2023-12-24 23:02:44 [ℹ] no tasks
2023-12-24 23:02:44 [✓] all EKS cluster resources for "cluster-game" have been created
2023-12-24 23:02:46 [ℹ] kubectl command should work with "C:\\Users\\betha\\.kube\\config", try 'kubectl get nodes'
2023-12-24 23:02:46 [✓] EKS cluster "cluster-game" in "us-east-1" region is ready
betha@Rajasekhar_PC MINGW64 /
$ aws eks update-kubeconfig --name cluster-game --region us-east-1
Added new context arn:aws:eks:us-east-1:986203609074:cluster/cluster-game to C:\Users\betha\.kube\config
→ update the cluster & kubeconfig file
betha@Rajasekhar_PC MINGW64 /
$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
fargate-ip-192-168-104-32.ec2.internal Ready <none> 2m19s v1.27.7-eks-4f4795d
fargate-ip-192-168-125-150.ec2.internal Ready <none> 2m20s v1.27.7-eks-4f4795d
} Nodes are the Fargate service
betha@Rajasekhar_PC MINGW64 /
$

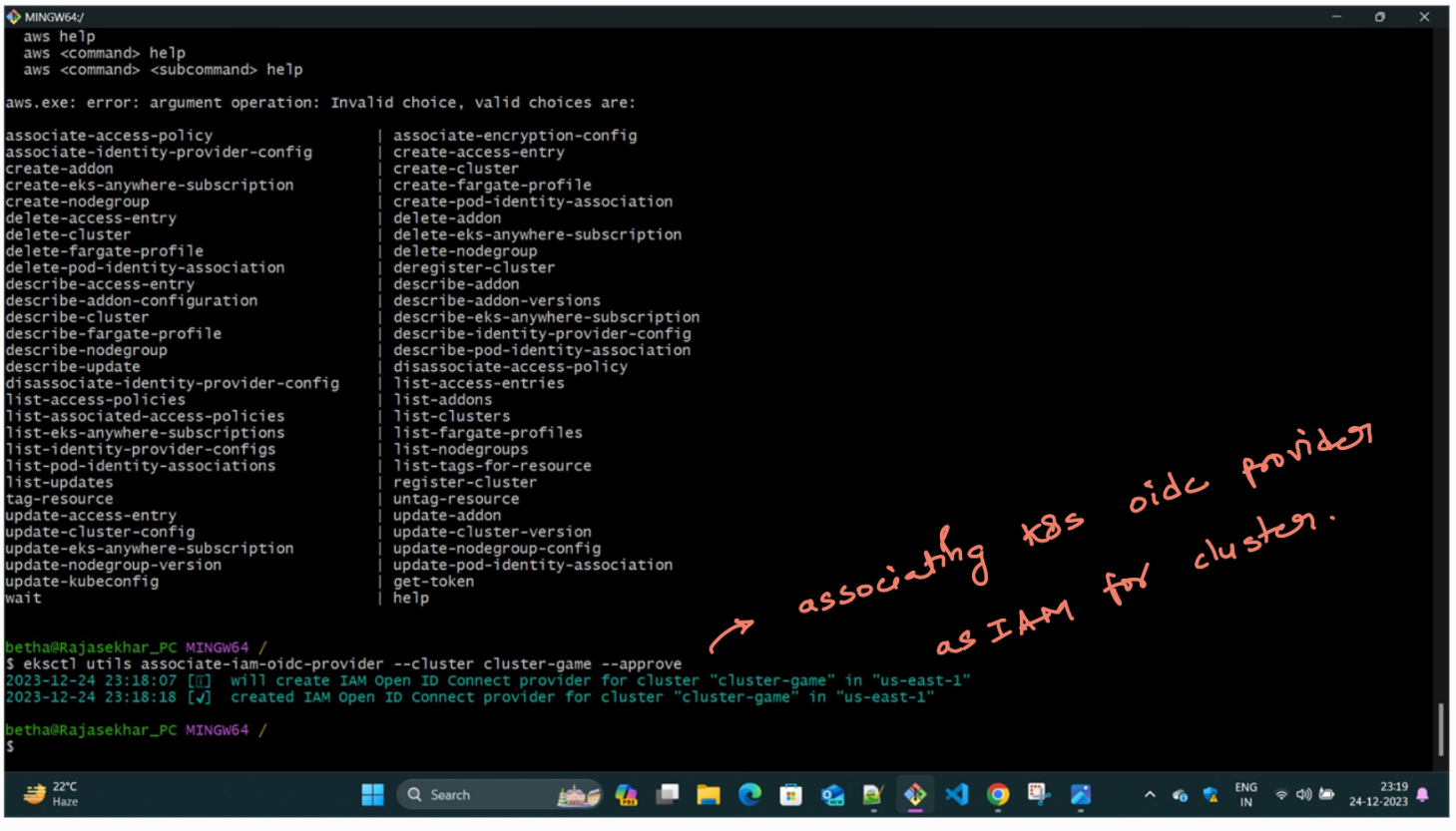

MINGW64:/
betha@Rajasekhar_PC MINGW64 /
$ aws eks update-kubeconfig --name cluster-game --region us-east-1
Added new context arn:aws:eks:us-east-1:986203609074:cluster/cluster-game to C:\Users\betha\.kube\config
betha@Rajasekhar_PC MINGW64 /
$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
fargate-ip-192-168-104-32.ec2.internal Ready <none> 2m19s v1.27.7-eks-4f4795d
fargate-ip-192-168-125-150.ec2.internal Ready <none> 2m20s v1.27.7-eks-4f4795d
betha@Rajasekhar_PC MINGW64 /
$ kubectl get po -A
NAMESPACE NAME READY STATUS RESTARTS AGE
kube-system coredns-788dbcccd5-94cbw 1/1 Running 0 3m20s
kube-system coredns-788dbcccd5-q7qb2 1/1 Running 0 3m20s
betha@Rajasekhar_PC MINGW64 /
$ kubectl get svc -A
NAMESPACE NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
default kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 12m
kube-system kube-dns ClusterIP 10.100.0.10 <none> 53/UDP,53/TCP 12m
betha@Rajasekhar_PC MINGW64 /
$ eksctl create fargateprofile \
--cluster cluster-game \
--region us-east-1 \
--name alb-sample-app \
--namespace game-2048
2023-12-24 23:07:03 [ℹ] creating Fargate profile "alb-sample-app" on EKS cluster "cluster-game"
2023-12-24 23:07:21 [ℹ] created Fargate profile "alb-sample-app" on EKS cluster "cluster-game"
} Fargate profile creation
betha@Rajasekhar_PC MINGW64 /
$

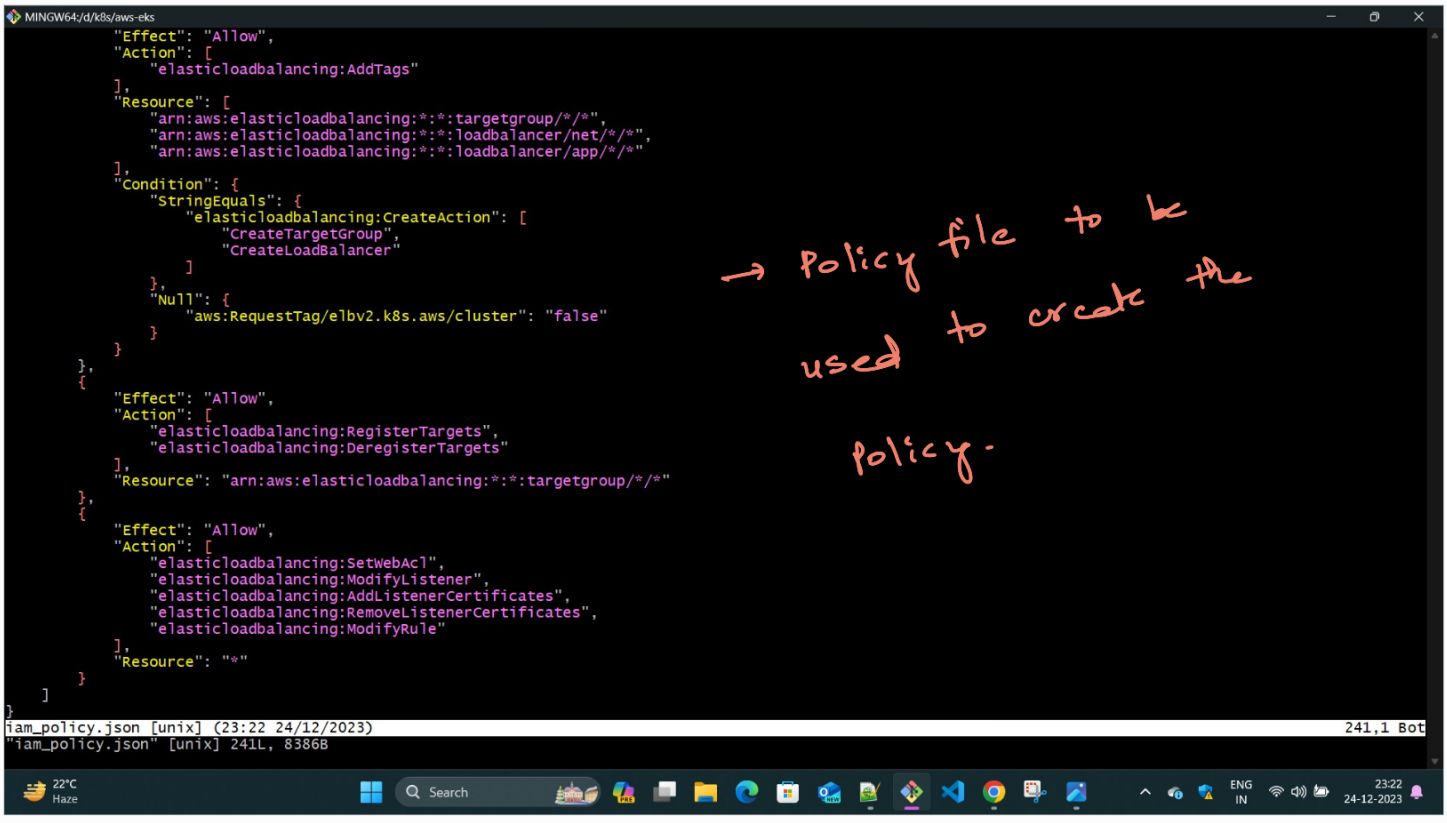


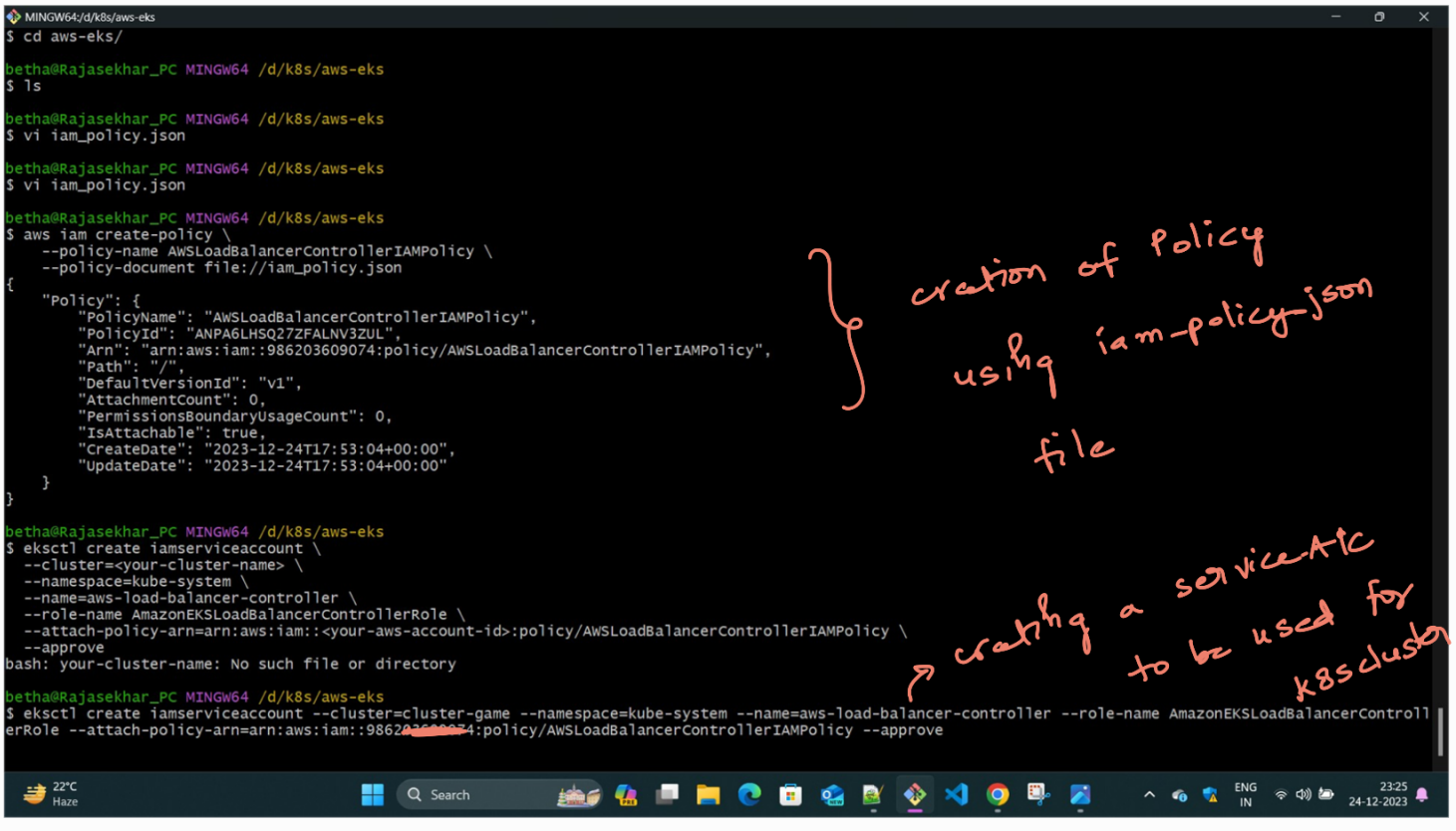

```
MINGW64:/

betha@Rajasekhar_PC MINGW64 /
$ kubectl get svc -A
NAMESPACE     NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)         AGE
default       kubernetes   ClusterIP   10.100.0.1     <none>        443/TCP         12m
kube-system   kube-dns     ClusterIP   10.100.0.10    <none>        53/UDP,53/TCP   12m

betha@Rajasekhar_PC MINGW64 /
$ eksctl create fargateprofile \
    --cluster cluster-game \
    --region us-east-1 \
    --name alb-sample-app \
    --namespace game-2048
2023-12-24 23:07:03 [ℹ]  creating Fargate profile "alb-sample-app" on EKS cluster "cluster-game"
2023-12-24 23:07:21 [ℹ]  created Fargate profile "alb-sample-app" on EKS cluster "cluster-game"

betha@Rajasekhar_PC MINGW64 /
$ kubectl get ns
NAME              STATUS   AGE
default           Active   15m
kube-node-lease   Active   15m
kube-public       Active   15m
kube-system       Active   15m

betha@Rajasekhar_PC MINGW64 /
$ 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

betha@Rajasekhar_PC MINGW64 /
$
```

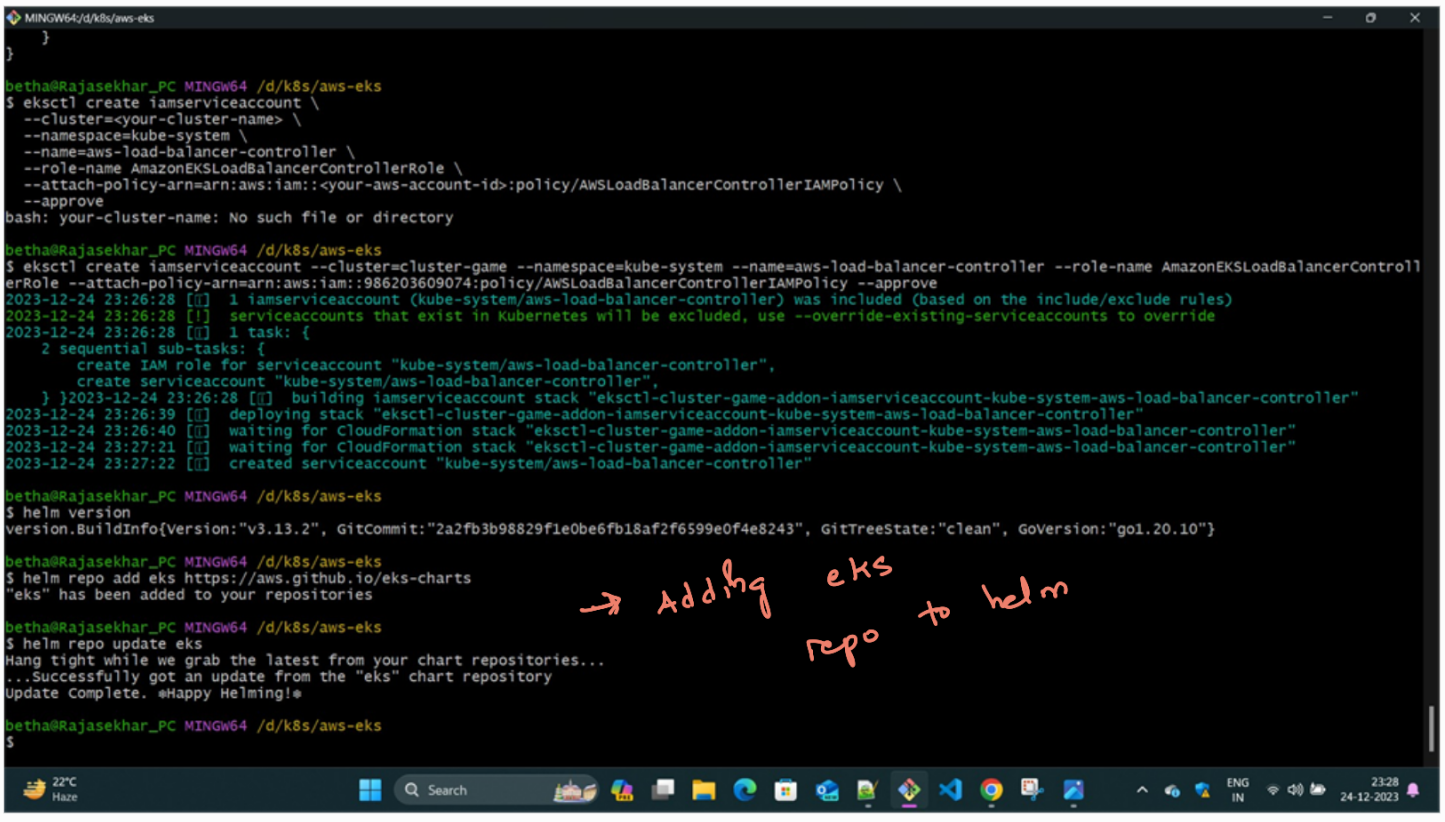
K8s Manifest file

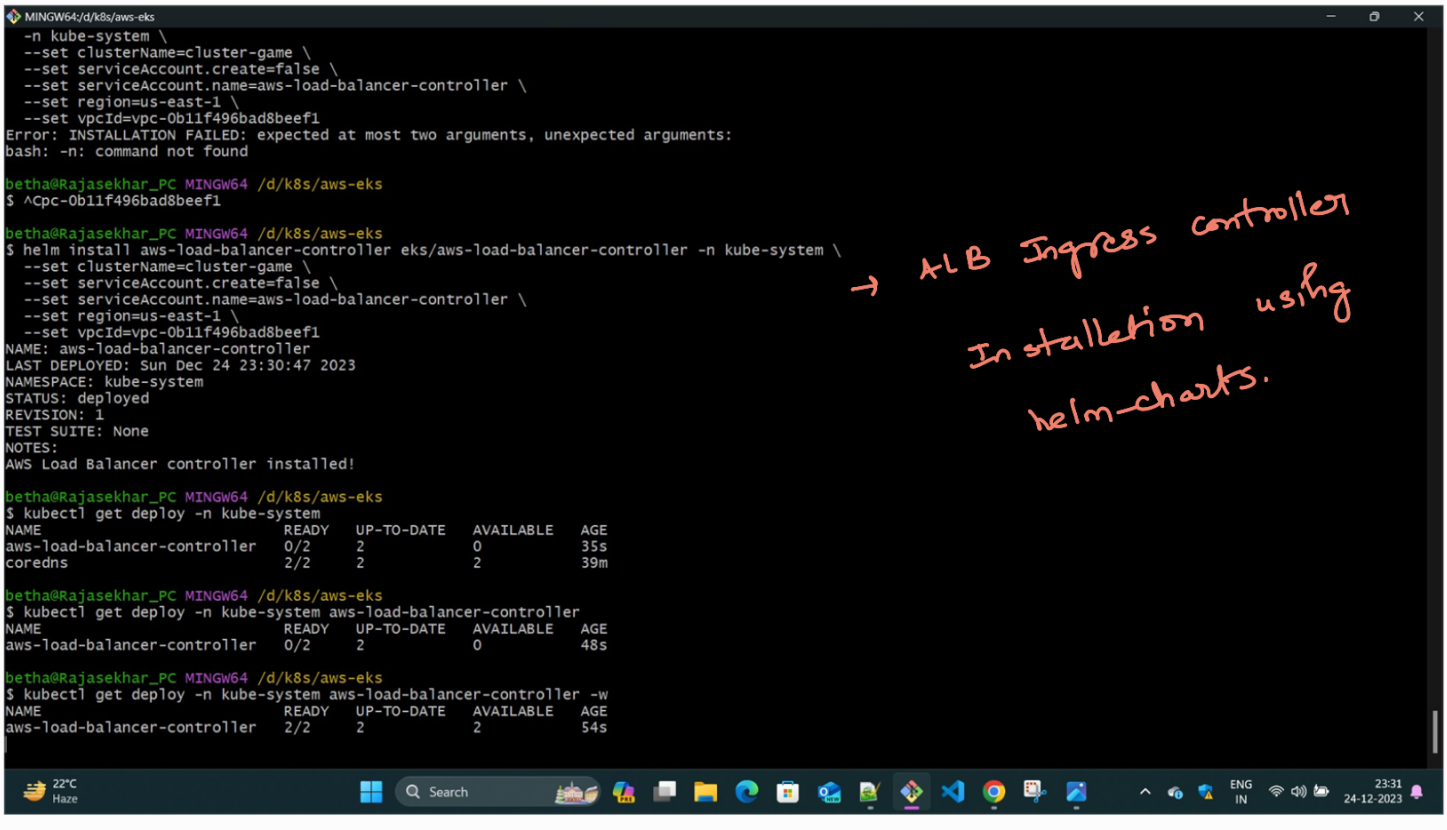

MINGW64:/
betha@Rajasekhar_PC MINGW64 /
$ kubectl get po game-2048
Error from server (NotFound): pods "game-2048" not found
betha@Rajasekhar_PC MINGW64 /
$ kubectl get po --namespace=game-2048
NAME READY STATUS RESTARTS AGE
deployment-2048-7ccfd8fdd6-68n4h 1/1 Running 0 64s
deployment-2048-7ccfd8fdd6-7m796 1/1 Running 0 64s
deployment-2048-7ccfd8fdd6-gt7x9 1/1 Running 0 64s
deployment-2048-7ccfd8fdd6-h2kc6 1/1 Running 0 64s
deployment-2048-7ccfd8fdd6-sx9bx 1/1 Running 0 64s
Replicaset → App PODS
betha@Rajasekhar_PC MINGW64 /
$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 17m
clusterIP default svc
betha@Rajasekhar_PC MINGW64 /
$ kubectl get svc --namespace=game-2048
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service-2048 NodePort 10.100.183.157 <none> 80:31236/TCP 91s
→ Nodeport svc for App
betha@Rajasekhar_PC MINGW64 /
$ kubectl get ingress
No resources found in default namespace.
betha@Rajasekhar_PC MINGW64 /
$ kubectl get ingress --namespace=game-2048
NAME CLASS HOSTS ADDRESS PORTS AGE
ingress-2048 alb * 80 114s
→ Ingress resource for ALB Ingress Controller
betha@Rajasekhar_PC MINGW64 /
$
22°C Haze
Search
ENG IN
23:10 24-12-2023

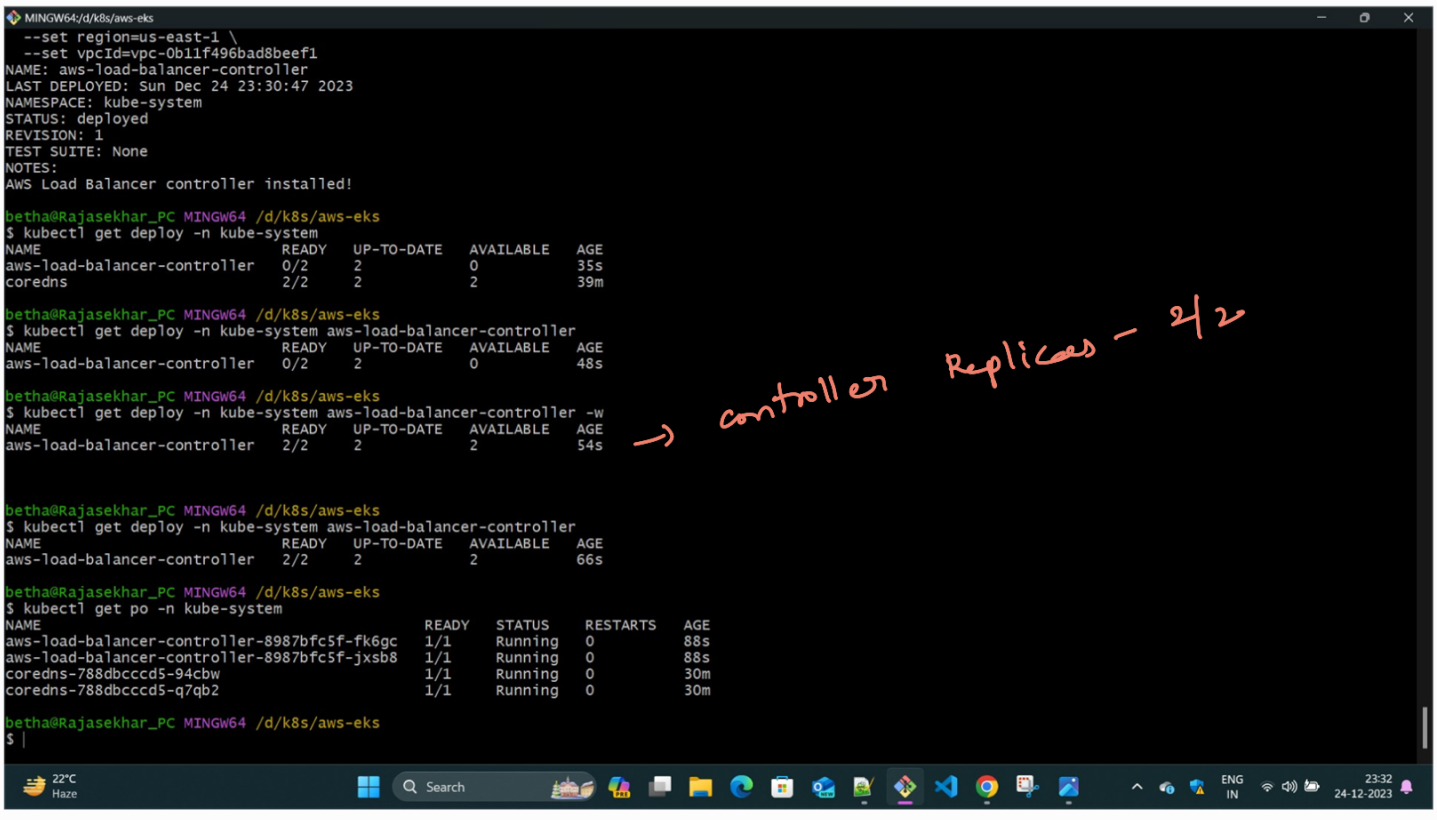

MINGW64:/
aws help
aws <command> help
aws <command> <subcommand> help
aws.exe: error: argument operation: Invalid choice, valid choices are:
associate-access-policy | associate-encryption-config
associate-identity-provider-config | create-access-entry
create-addon | create-cluster
create-eks-anywhere-subscription | create-fargate-profile
create-nodegroup | create-pod-identity-association
delete-access-entry | delete-addon
delete-cluster | delete-eks-anywhere-subscription
delete-fargate-profile | delete-nodegroup
delete-pod-identity-association | deregister-cluster
describe-access-entry | describe-addon
describe-addon-configuration | describe-addon-versions
describe-cluster | describe-eks-anywhere-subscription
describe-fargate-profile | describe-identity-provider-config
describe-nodegroup | describe-pod-identity-association
describe-update | disassociate-access-policy
disassociate-identity-provider-config | list-access-entries
list-access-policies | list-addons
list-associated-access-policies | list-clusters
list-eks-anywhere-subscriptions | list-fargate-profiles
list-identity-provider-configs | list-nodegroups
list-pod-identity-associations | list-tags-for-resource
list-updates | register-cluster
tag-resource | untag-resource
update-access-entry | update-addon
update-cluster-config | update-cluster-version
update-eks-anywhere-subscription | update-nodegroup-config
update-nodegroup-version | update-pod-identity-association
update-kubeconfig | get-token
wait | help
betha@Rajasekhar_PC MINGW64 /
$ eksctl utils associate-iam-oidc-provider --cluster cluster-game --approve
2023-12-24 23:18:07 [ℹ] will create IAM Open ID Connect provider for cluster "cluster-game" in "us-east-1"
2023-12-24 23:18:18 [✓] created IAM Open ID Connect provider for cluster "cluster-game" in "us-east-1"
→ associating k8s oidc provider as IAM for cluster.
betha@Rajasekhar_PC MINGW64 /
$
22°C Haze
Search
ENG IN
23:19 24-12-2023

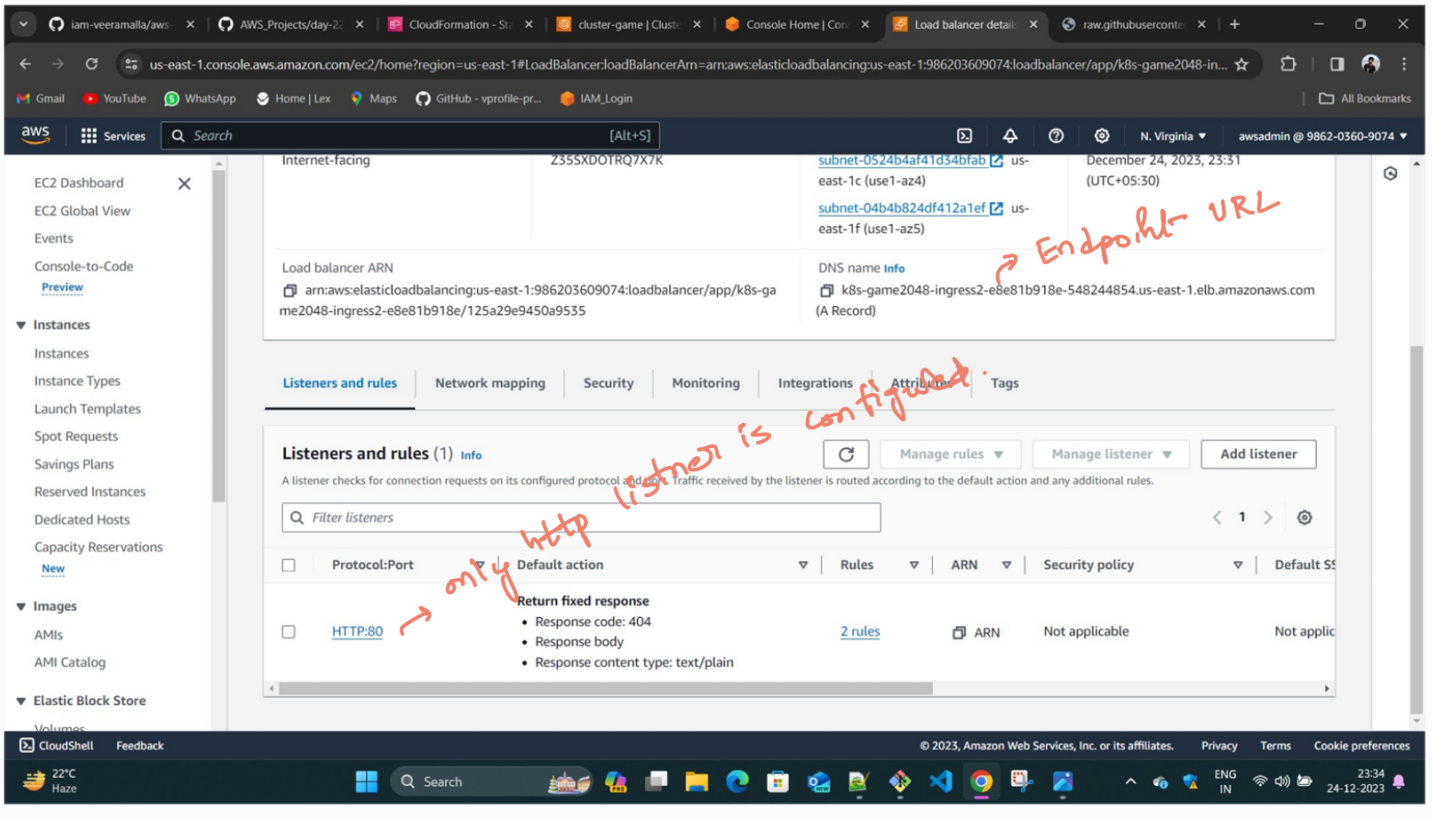

MINGW64:/d/k8s/aws-eks
"Effect": "Allow",
"Action": [
"elasticloadbalancing:AddTags"
],
"Resource": [
"arn:aws:elasticloadbalancing:*:*:targetgroup/*/*",
"arn:aws:elasticloadbalancing:*:*:loadbalancer/net/*/*",
"arn:aws:elasticloadbalancing:*:*:loadbalancer/app/*/*"
],
"Condition": {
"StringEquals": {
"elasticloadbalancing:CreateAction": [
"CreateTargetGroup",
"CreateLoadBalancer"
]
},
"Null": {
"aws:RequestTag/elbv2.k8s.aws/cluster": "false"
}
}
},
{
"Effect": "Allow",
"Action": [
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DeregisterTargets"
],
"Resource": "arn:aws:elasticloadbalancing:*:*:targetgroup/*/*"
},
{
"Effect": "Allow",
"Action": [
"elasticloadbalancing:SetWebAcl",
"elasticloadbalancing:ModifyListener",
"elasticloadbalancing:AddListenerCertificates",
"elasticloadbalancing:RemoveListenerCertificates",
"elasticloadbalancing:ModifyRule"
],
"Resource": "*"
}
]
}
iam_policy.json [unix] (23:22 24/12/2023) 241,1 Bot
"iam_policy.json" [unix] 241L, 8386B
→ Policy file to be used to create the policy.
22°C Haze
Search
ENG IN
23:22 24-12-2023

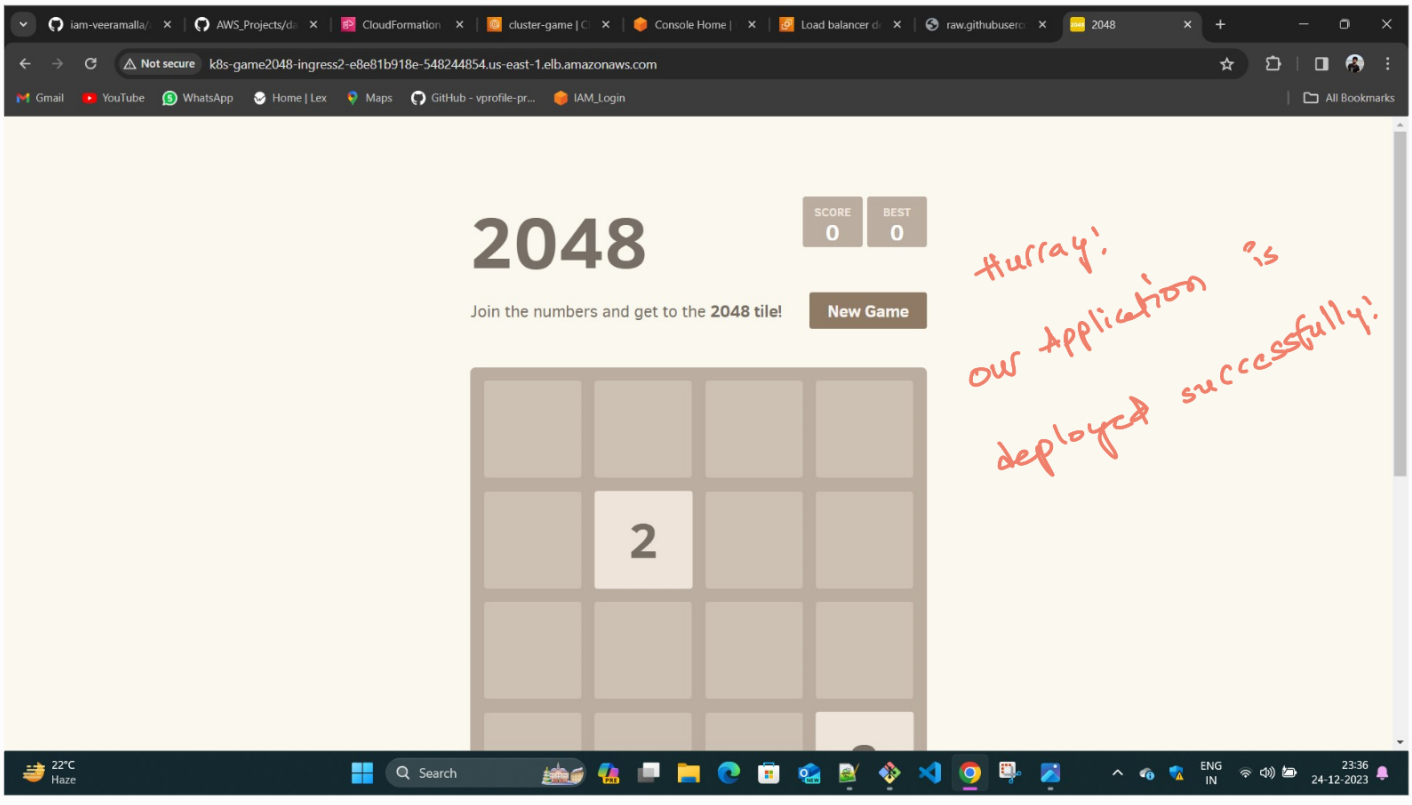

MINGW64:/d/k8s/aws-eks
$ cd aws-eks/
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ ls
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ vi iam_policy.json
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ vi iam_policy.json
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ aws iam create-policy \
--policy-name AWSLoadBalancerControllerIAMPolicy \
--policy-document file://iam_policy.json
{
"Policy": {
"PolicyName": "AWSLoadBalancerControllerIAMPolicy",
"PolicyId": "ANPA6LHSQ27ZFALNV3ZUL",
"Arn": "arn:aws:iam::986203609074:policy/AWSLoadBalancerControllerIAMPolicy",
"Path": "/",
"DefaultVersionId": "v1",
"AttachmentCount": 0,
"PermissionsBoundaryUsageCount": 0,
"IsAttachable": true,
"CreateDate": "2023-12-24T17:53:04+00:00",
"UpdateDate": "2023-12-24T17:53:04+00:00"
}
}
creation of Policy using iam-policy-json file
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ eksctl create iamserviceaccount \
--cluster=<your-cluster-name> \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--role-name AmazonEKSLoadBalancerControllerRole \
--attach-policy-arn=arn:aws:iam::<your-aws-account-id>:policy/AWSLoadBalancerControllerIAMPolicy \
--approve
bash: your-cluster-name: No such file or directory
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ eksctl create iamserviceaccount --cluster=cluster-game --namespace=kube-system --name=aws-load-balancer-controller --role-name AmazonEKSLoadBalancerControll
erRole --attach-policy-arn=arn:aws:iam::986203609074:policy/AWSLoadBalancerControllerIAMPolicy --approve
creating a serviceAtc to be used for k8scluster
22°C Haze
Search
ENG IN
23:25 24-12-2023

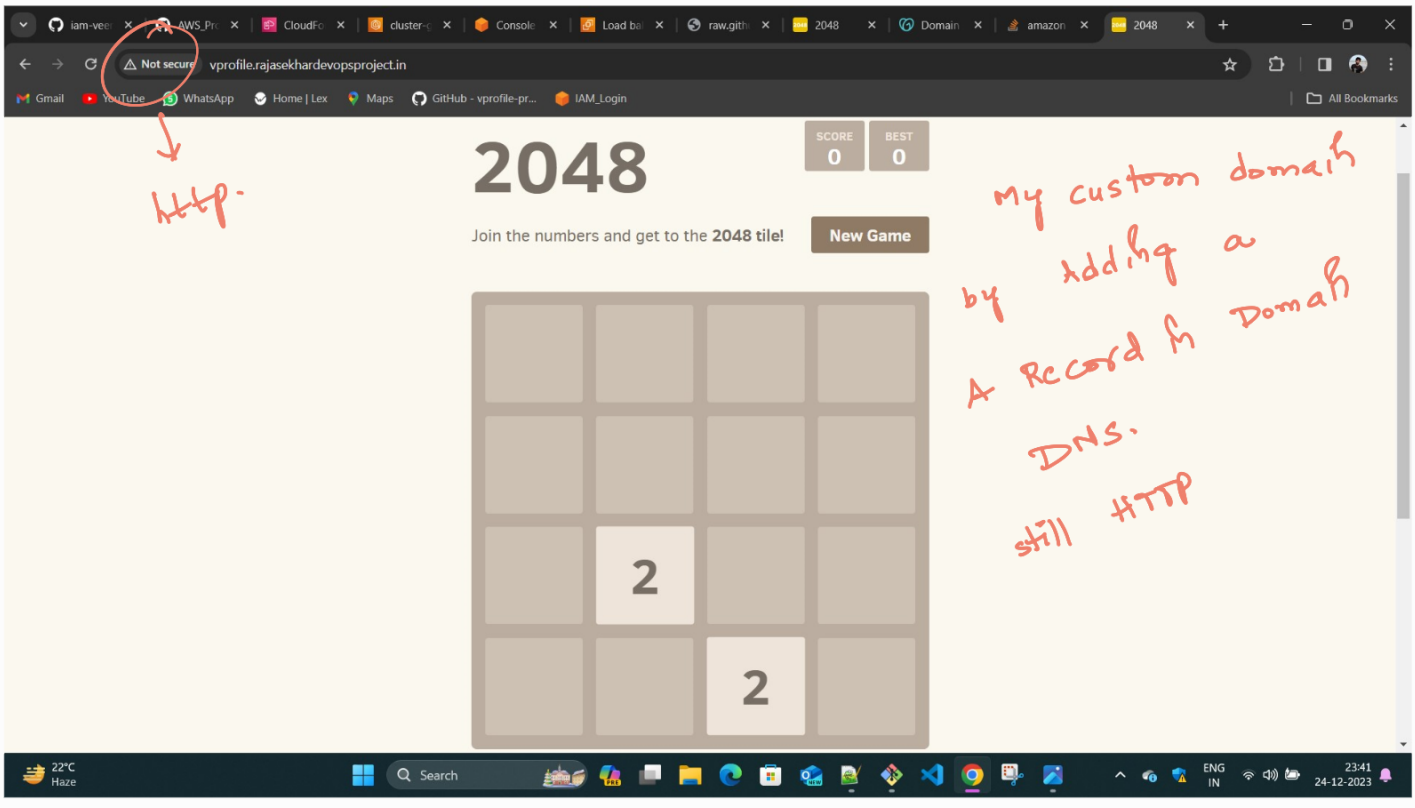

MINGW64:/d/k8s/aws-eks
}
}
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ eksctl create iamserviceaccount \
--cluster=<your-cluster-name> \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--role-name AmazonEKSLoadBalancerControllerRole \
--attach-policy-arn=arn:aws:iam::<your-aws-account-id>:policy/AWSLoadBalancerControllerIAMPolicy \
--approve
bash: your-cluster-name: No such file or directory
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ eksctl create iamserviceaccount --cluster=cluster-game --namespace=kube-system --name=aws-load-balancer-controller --role-name AmazonEKSLoadBalancerControll
erRole --attach-policy-arn=arn:aws:iam::986203609074:policy/AWSLoadBalancerControllerIAMPolicy --approve
2023-12-24 23:26:28 [ℹ] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2023-12-24 23:26:28 [!] serviceaccounts that exist in Kubernetes will be excluded, use --override-existing-serviceaccounts to override
2023-12-24 23:26:28 [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",
} }2023-12-24 23:26:28 [ℹ] building iamserviceaccount stack "eksctl-cluster-game-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-24 23:26:39 [ℹ] deploying stack "eksctl-cluster-game-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-24 23:26:40 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-24 23:27:21 [ℹ] waiting for CloudFormation stack "eksctl-cluster-game-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-24 23:27:22 [ℹ] created serviceaccount "kube-system/aws-load-balancer-controller"
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ helm version
version.BuildInfo{Version:"v3.13.2", GitCommit:"2a2fb3b98829f1e0be6fb18af2f6599e0f4e8243", GitTreeState:"clean", GoVersion:"go1.20.10"}
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ helm repo update eks
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "eks" chart repository
Update Complete. ⎈Happy Helming!⎈
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$
→ Adding eks repo to helm
22°C Haze
Search
ENG IN
23:28 24-12-2023

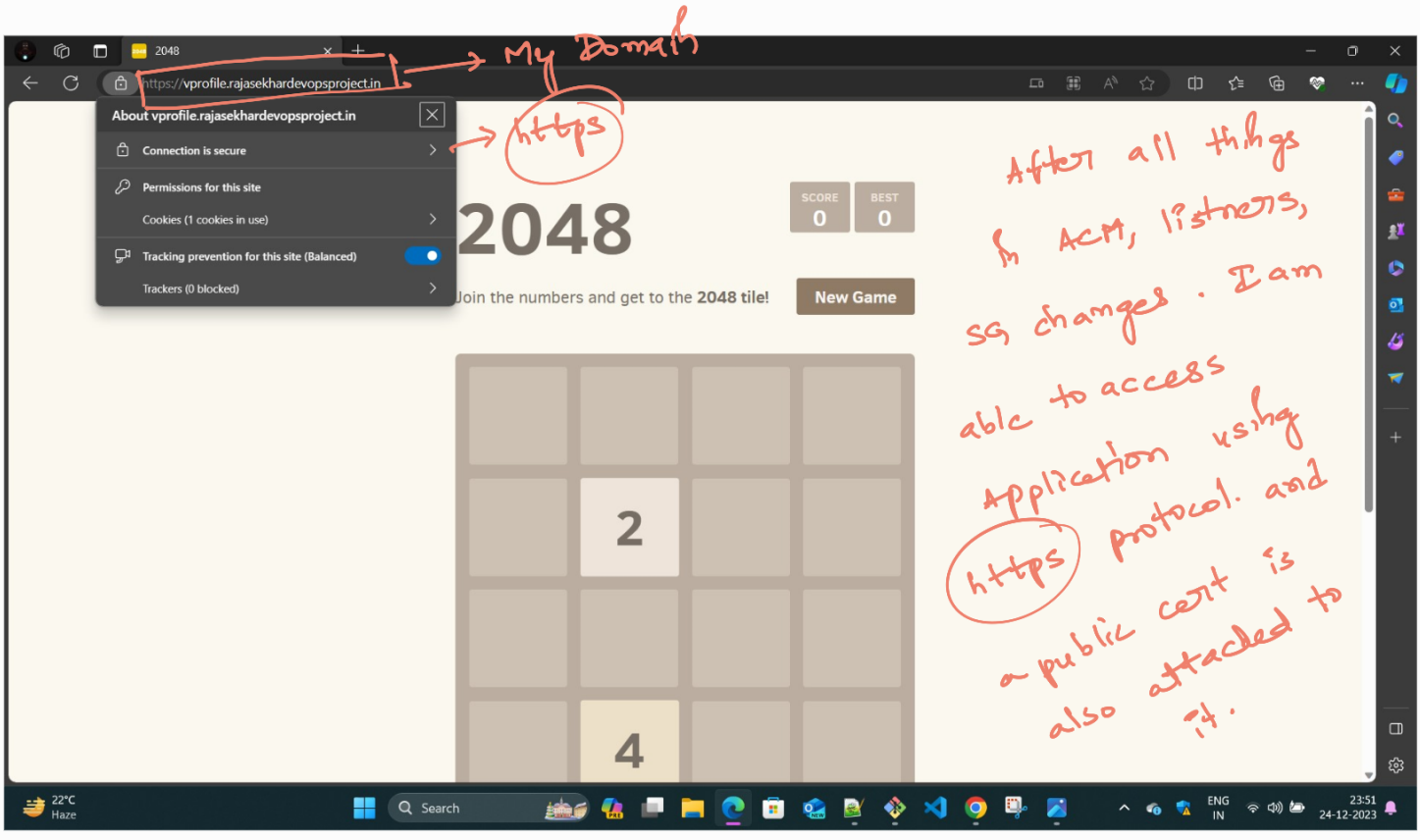

MINGW64:/d/k8s/aws-eks
-n kube-system \
--set clusterName=cluster-game \
--set serviceAccount.create=false \
--set serviceAccount.name=aws-load-balancer-controller \
--set region=us-east-1 \
--set vpcId=vpc-0b11f496bad8beef1
Error: INSTALLATION FAILED: expected at most two arguments, unexpected arguments:
bash: -n: command not found
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ ^Cpc-0b11f496bad8beef1
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ helm install aws-load-balancer-controller eks/aws-load-balancer-controller -n kube-system \
--set clusterName=cluster-game \
--set serviceAccount.create=false \
--set serviceAccount.name=aws-load-balancer-controller \
--set region=us-east-1 \
--set vpcId=vpc-0b11f496bad8beef1
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sun Dec 24 23:30:47 2023
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 0/2 2 0 35s
coredns 2/2 2 2 39m
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system aws-load-balancer-controller
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 0/2 2 0 48s
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system aws-load-balancer-controller -w
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 2/2 2 2 54s
→ ALB Ingress Controller Installation using helm_charts.
22°C Haze
Search
ENG IN
23:31 24-12-2023

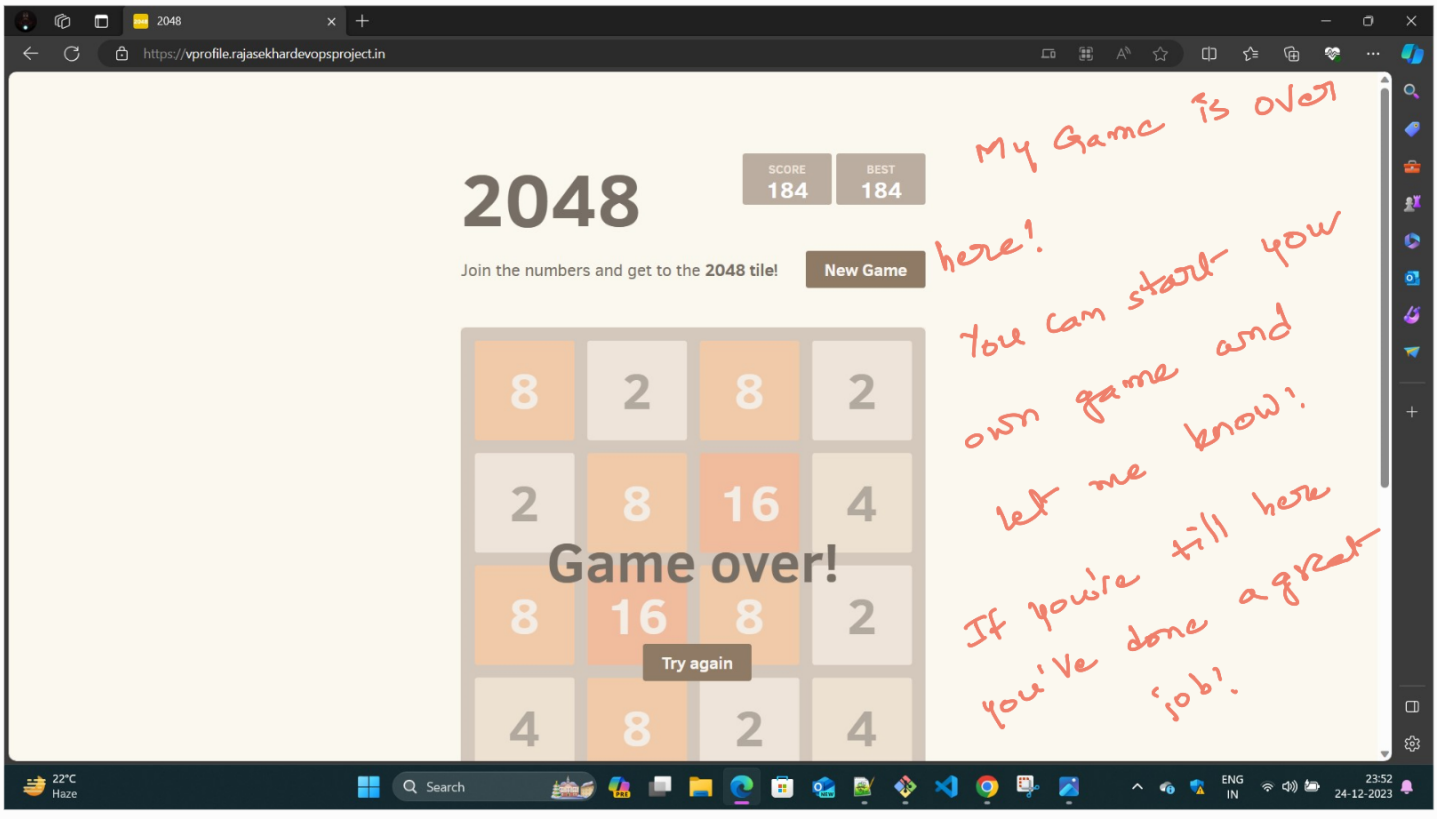

MINGW64:/d/k8s/aws-eks
--set region=us-east-1 \
--set vpcId=vpc-0b11f496bad8beef1
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sun Dec 24 23:30:47 2023
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 0/2 2 0 35s
coredns 2/2 2 2 39m
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system aws-load-balancer-controller
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 0/2 2 0 48s
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system aws-load-balancer-controller -w
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 2/2 2 2 54s
→ controller Replicas - 2/2
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get deploy -n kube-system aws-load-balancer-controller
NAME READY UP-TO-DATE AVAILABLE AGE
aws-load-balancer-controller 2/2 2 2 66s
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$ kubectl get po -n kube-system
NAME READY STATUS RESTARTS AGE
aws-load-balancer-controller-8987bfc5f-fk6gc 1/1 Running 0 88s
aws-load-balancer-controller-8987bfc5f-jxsb8 1/1 Running 0 88s
coredns-788dbcccd5-94cbw 1/1 Running 0 30m
coredns-788dbcccd5-q7qb2 1/1 Running 0 30m
betha@Rajasekhar_PC MINGW64 /d/k8s/aws-eks
$

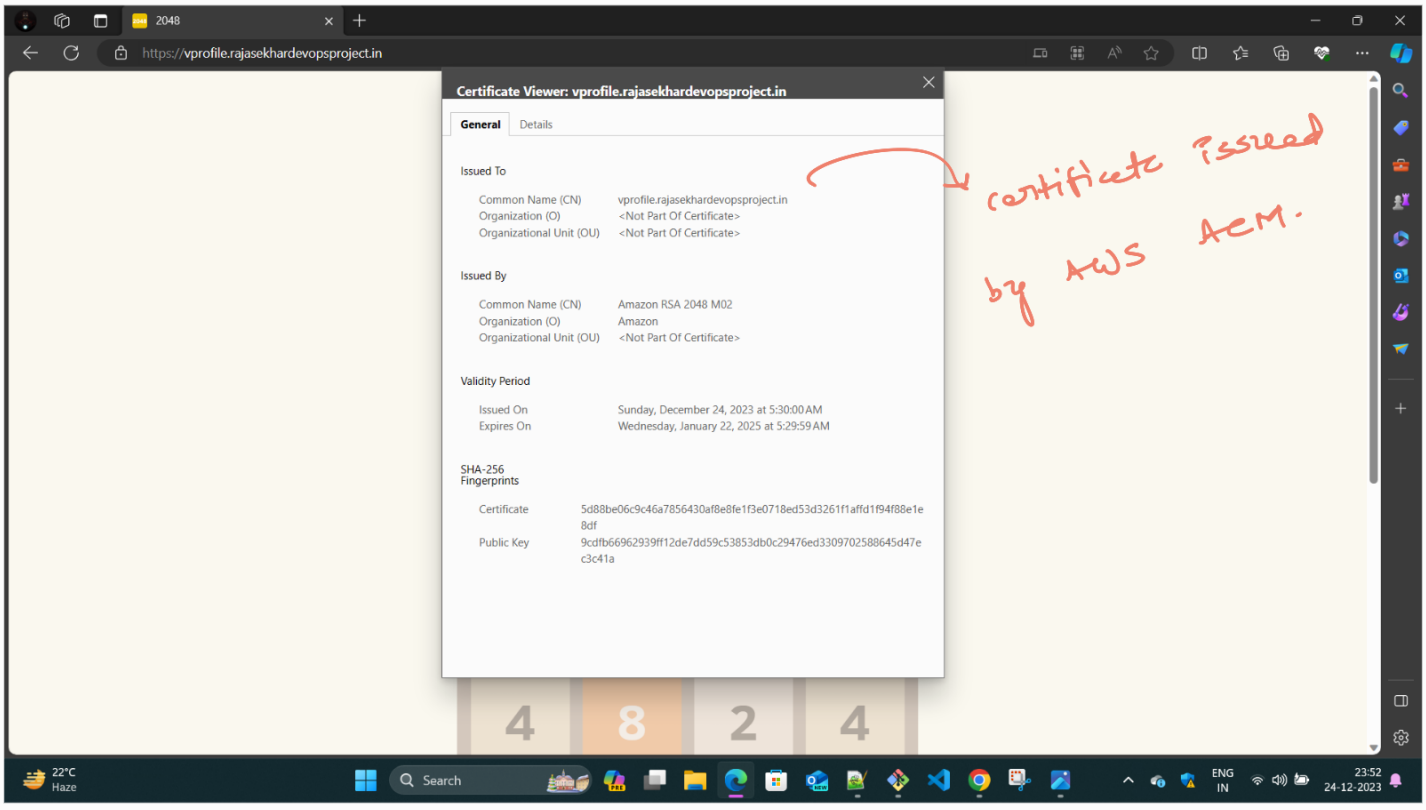

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-east-1:986203609074:loadbalancer/app/k8s-game2048-in...
Internet-facing
subnet-0524b4af41d34bfab us-east-1c (use1-az4)
subnet-04b4b824df412a1ef us-east-1f (use1-az5)
December 24, 2023, 23:31 (UTC+05:30)
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:986203609074:loadbalancer/app/k8s-game2048-ingress2-e8e81b918e/125a29e9450a9535
DNS name Info
k8s-game2048-ingress2-e8e81b918e-548244854.us-east-1.elb.amazonaws.com (A Record)
→ Endpoint URL
Listeners and rules
Network mapping
Security
Monitoring
Integrations
Tags
Listeners and rules (1) Info
Manage rules
Manage listener
Add listener
Filter listeners
Protocol:Port
Default action
Rules
ARN
Security policy
HTTP:80
Return fixed response
Response code: 404
Response body
Response content type: text/plain
2 rules
Not applicable
→ only http listner is configured

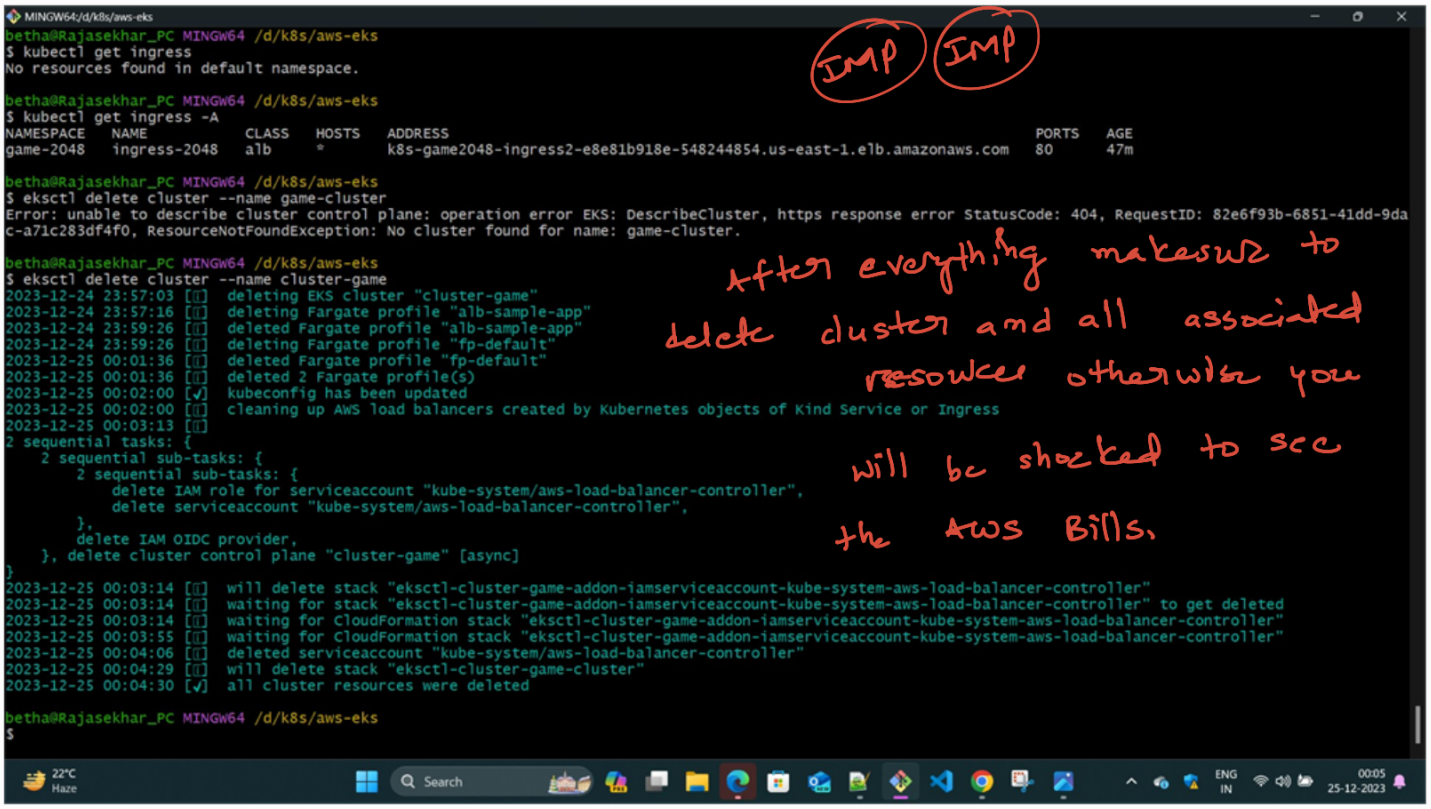

Not secure
k8s-game2048-ingress2-e8e81b918e-548244854.us-east-1.elb.amazonaws.com
2048
SCORE
0
BEST
0
Join the numbers and get to the 2048 tile!
New Game
2
Hurray!
our application is
deployed successfully!


Not secure
vprofile.rajasekhardevopsproject.in
http.
2048
SCORE
0
BEST
0
Join the numbers and get to the 2048 tile!
New Game
2
2
My custom domain
by adding a
A Record in Domain
DNS.
still HTTP


2048
https://vprofile.rajasekhardevopsproject.in
My Domain
About vprofile.rajasekhardevopsproject.in
Connection is secure
Permissions for this site
Cookies (1 cookies in use)
Tracking prevention for this site (Balanced)
Trackers (0 blocked)
https
2048
SCORE 0
BEST 0
Join the numbers and get to the 2048 tile!
New Game
2
4
After all things & ACM, listners, SG changes. I am able to access application using https protocol. and a public cert is also attached to it.
22°C Haze
Search
ENG IN
23:51 24-12-2023


2048
https://vprofile.rajasekhardevopsproject.in
2048
SCORE 184
BEST 184
Join the numbers and get to the 2048 tile!
New Game
Game over!
Try again
My Game is over here!
You can start your own game and let me know!
If you're till here you've done a great job!
22°C Haze
Search
ENG IN
23:52 24-12-2023




Thats All Guys for Today. Thank You so much for reading all the way down.

- Rajasekhar.