AWS-Eks

Today we are going to deploy a very interesting game with the help of Amazon EKS.

All commands and files are present on my GitHub.

Please follow the steps.

1. We need to install eksctl.

2. Next, install kubectl.

3. Then, install the AWS CLI.

4. Finally, install Docker.

1. Fist we have to install eksctl

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

Than we have move that in bin (becose all the exicuteble file is store)

sudo mv /tmp/eksctl /usr/local/bin

# Verify installation

eksctl version > /dev/null 2>&1; then

echo "eksctl installation completed successfully."

else

echo "eksctl installation failed. Please try again."

exit 1

# Clean up /tmp

rm -rf /tmp/eksctl

1)Than we have to install docker

# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

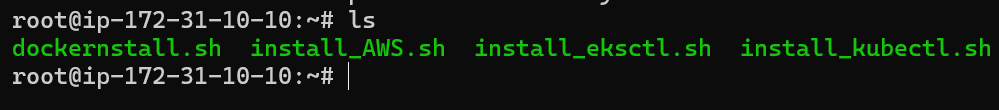
$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

1. Than this command

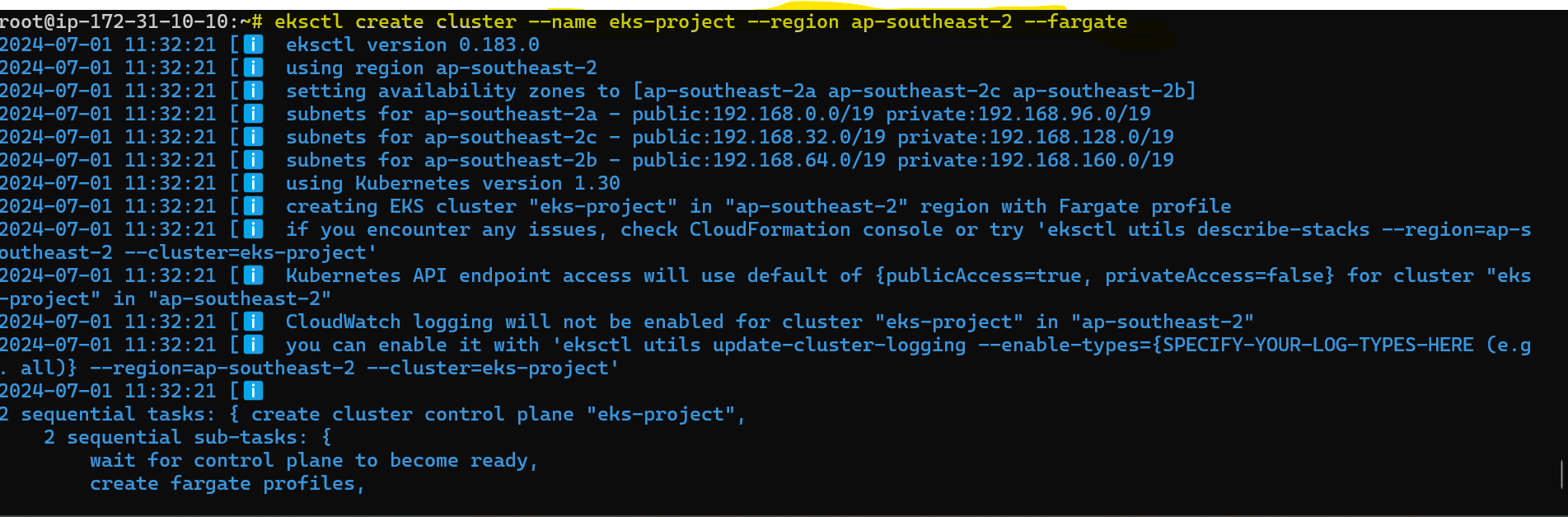
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

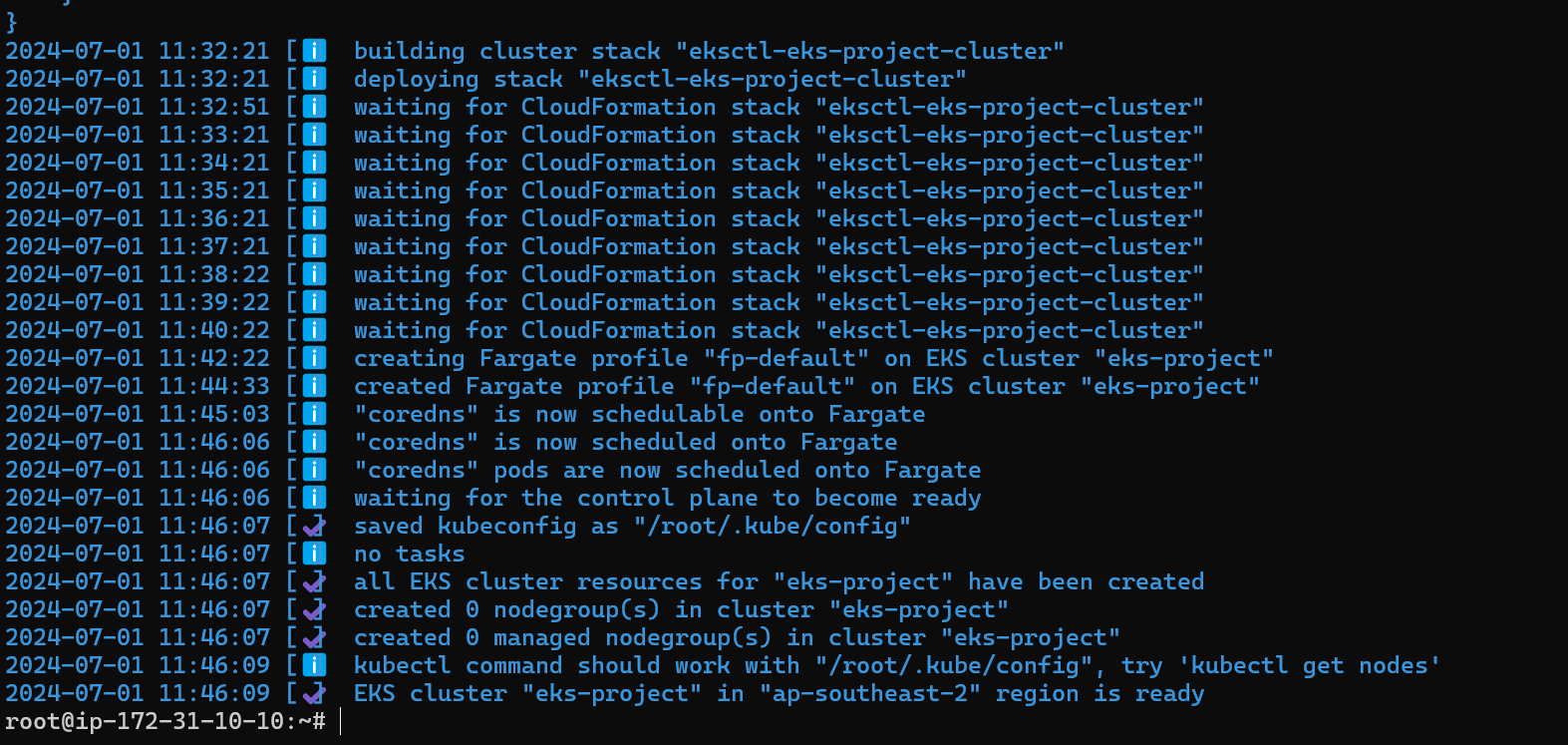


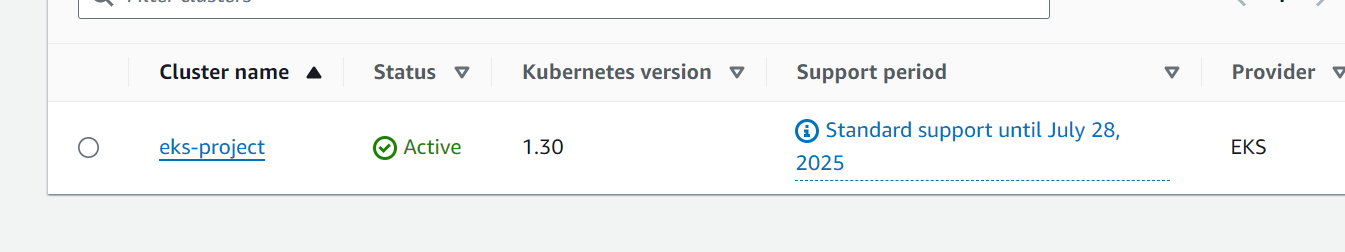
Than we have to **configure aws** add the keys



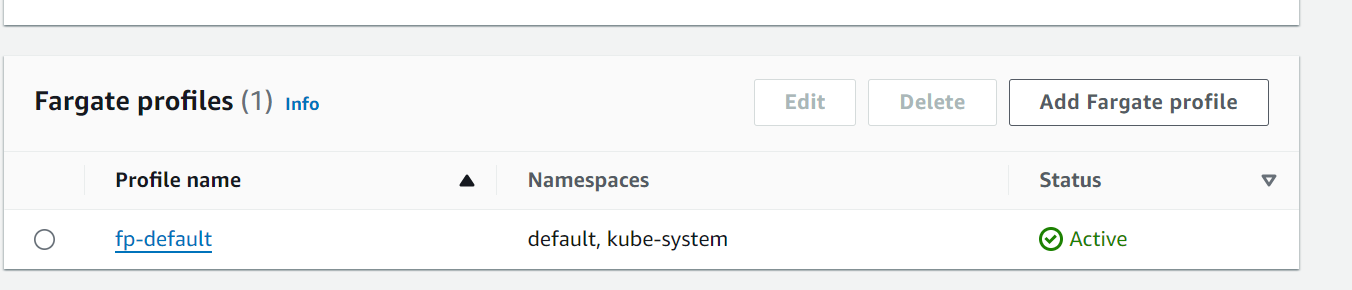
Then, we need to create an Amazon EKS cluster with the project name and region. This process typically takes around 10 to 15 minutes.





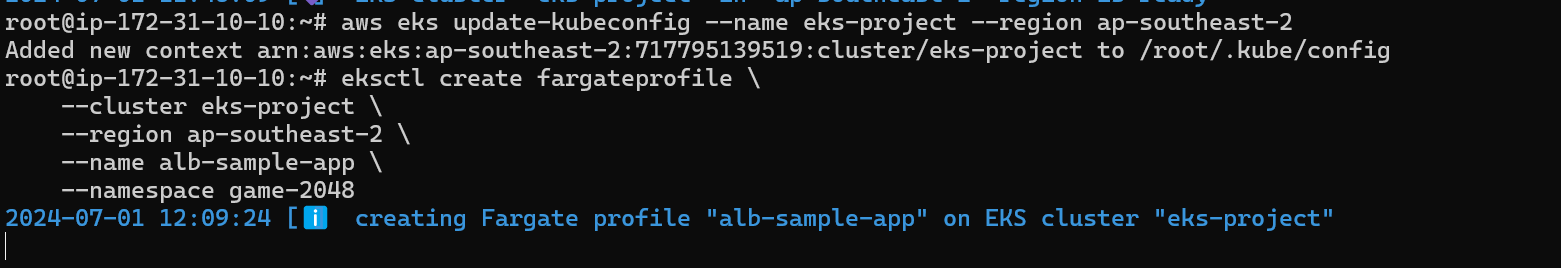
Here, we can check if the cluster has been created or not.

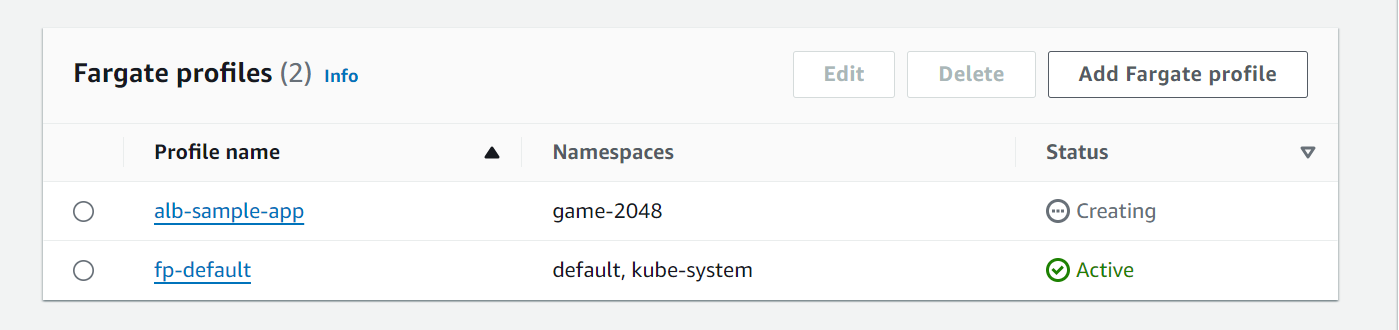
Here, we need to deploy our pod or service under this namespace.





Here, we need to deploy a Fargate profile for an Application Load Balancer (ALB) in the namespace game-2048

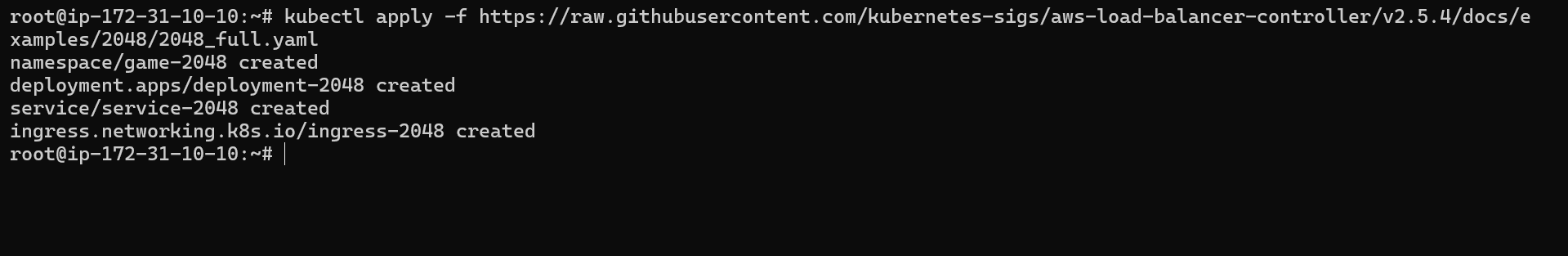


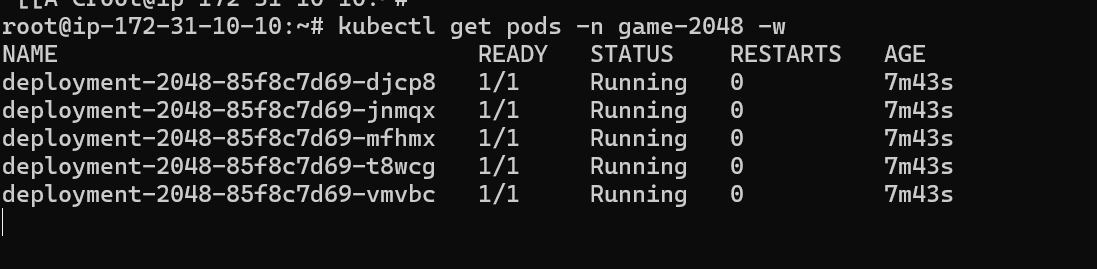
Here, we create an ALB (Application Load Balancer ) the game-2048 namespace

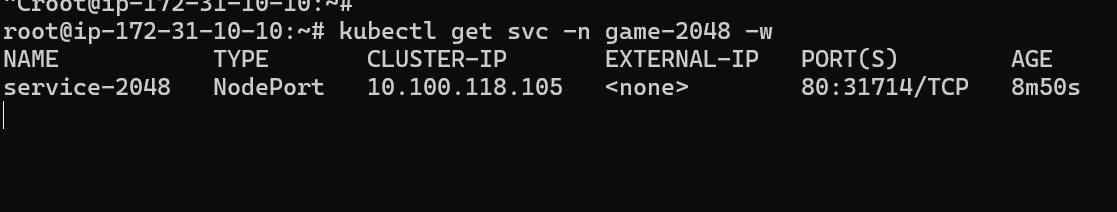
Then, we need to create a pod using a YAML file. This file will allow us to create multiple components:

1. Deployment

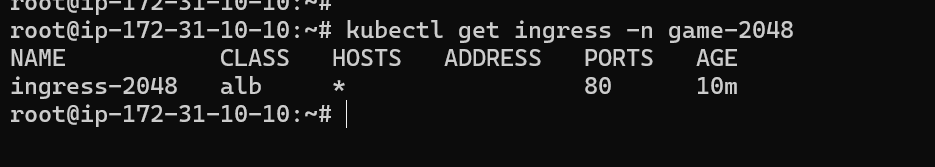
2. Service

3.Ingress

Then all pods are created by our previous command.

Check the service for the pod. 

.Ingress is created



We just deployed this YAML file.

---

apiVersion: v1

kind: Namespace

metadata:

name: game-2048

---

apiVersion: apps/v1

kind: Deployment

metadata:

namespace: game-2048

name: deployment-2048

spec:

selector:

matchLabels:

app.kubernetes.io/name: app-2048

replicas: 5

template:

metadata:

labels:

app.kubernetes.io/name: app-2048

spec:

containers:

- image: public.ecr.aws/l6m2t8p7/docker-2048:latest

imagePullPolicy: Always

name: app-2048

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

namespace: game-2048

name: service-2048

spec:

ports:

- port: 80

targetPort: 80

protocol: TCP

type: NodePort

selector:

app.kubernetes.io/name: app-2048

---

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

namespace: game-2048

name: ingress-2048

annotations:

alb.ingress.kubernetes.io/scheme: internet-facing

alb.ingress.kubernetes.io/target-type: ip

spec:

ingressClassName: alb

rules:

- http:

paths:

- path: /

pathType: Prefix

backend:

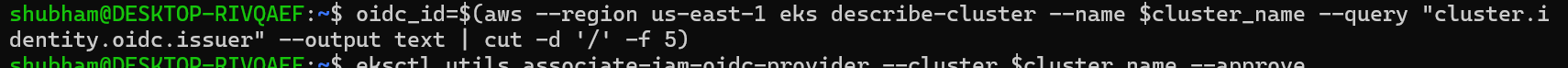
service:

name: service-2048

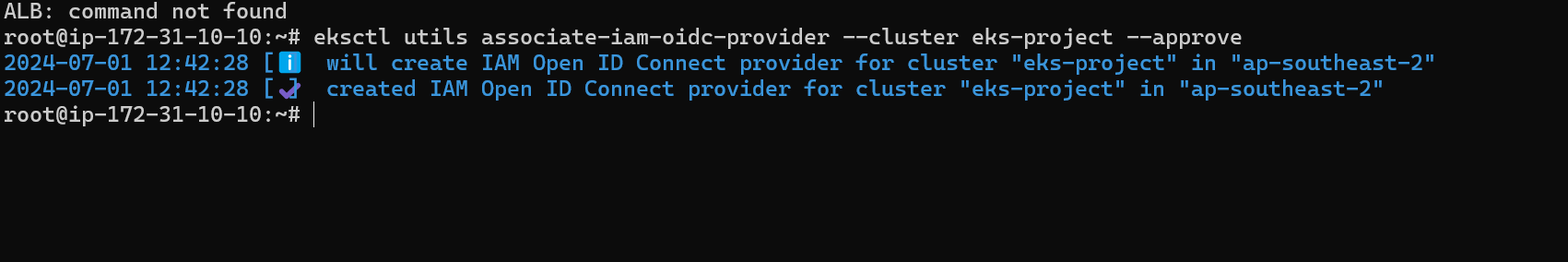
port:

number: 80

Here is where you add your cluster name to the connector.



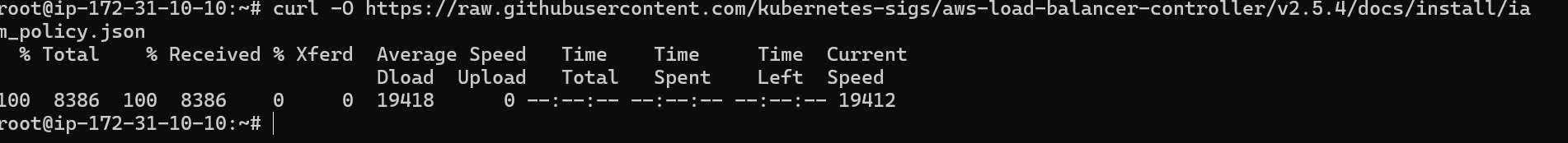
To set up OIDC (OpenID Connect) provider, simply add your cluster name.





For ALB (APP LAOD BALANCER )POD WE HAVE TO CREATE A POLLICE AND ROLL

ITS IN JSON FORMATE. We just have to dwonload this





POLLICE JSONE FORNMTE IS LIKE

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"iam:CreateServiceLinkedRole"

],

"Resource": "\*",

"Condition": {

"StringEquals": {

"iam:AWSServiceName": "elasticloadbalancing.amazonaws.com"

}

}

},

{

"Effect": "Allow",

"Action": [

"ec2:DescribeAccountAttributes",

"ec2:DescribeAddresses",

"ec2:DescribeAvailabilityZones",

"ec2:DescribeInternetGateways",

"ec2:DescribeVpcs",

"ec2:DescribeVpcPeeringConnections",

"ec2:DescribeSubnets",

"ec2:DescribeSecurityGroups",

"ec2:DescribeInstances",

"ec2:DescribeNetworkInterfaces",

"ec2:DescribeTags",

"ec2:GetCoipPoolUsage",

"ec2:DescribeCoipPools",

"elasticloadbalancing:DescribeLoadBalancers",

"elasticloadbalancing:DescribeLoadBalancerAttributes",

"elasticloadbalancing:DescribeListeners",

"elasticloadbalancing:DescribeListenerCertificates",

"elasticloadbalancing:DescribeSSLPolicies",

"elasticloadbalancing:DescribeRules",

"elasticloadbalancing:DescribeTargetGroups",

"elasticloadbalancing:DescribeTargetGroupAttributes",

"elasticloadbalancing:DescribeTargetHealth",

"elasticloadbalancing:DescribeTags"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"cognito-idp:DescribeUserPoolClient",

"acm:ListCertificates",

"acm:DescribeCertificate",

"iam:ListServerCertificates",

"iam:GetServerCertificate",

"waf-regional:GetWebACL",

"waf-regional:GetWebACLForResource",

"waf-regional:AssociateWebACL",

"waf-regional:DisassociateWebACL",

"wafv2:GetWebACL",

"wafv2:GetWebACLForResource",

"wafv2:AssociateWebACL",

"wafv2:DisassociateWebACL",

"shield:GetSubscriptionState",

"shield:DescribeProtection",

"shield:CreateProtection",

"shield:DeleteProtection"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"ec2:AuthorizeSecurityGroupIngress",

"ec2:RevokeSecurityGroupIngress"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"ec2:CreateSecurityGroup"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"ec2:CreateTags"

],

"Resource": "arn:aws:ec2:\*:\*:security-group/\*",

"Condition": {

"StringEquals": {

"ec2:CreateAction": "CreateSecurityGroup"

},

"Null": {

"aws:RequestTag/elbv2.k8s.aws/cluster": "false"

}

}

},

{

"Effect": "Allow",

"Action": [

"ec2:CreateTags",

"ec2:DeleteTags"

],

"Resource": "arn:aws:ec2:\*:\*:security-group/\*",

"Condition": {

"Null": {

"aws:RequestTag/elbv2.k8s.aws/cluster": "true",

"aws:ResourceTag/elbv2.k8s.aws/cluster": "false"

}

}

},

{

"Effect": "Allow",

"Action": [

"ec2:AuthorizeSecurityGroupIngress",

"ec2:RevokeSecurityGroupIngress",

"ec2:DeleteSecurityGroup"

],

"Resource": "\*",

"Condition": {

"Null": {

"aws:ResourceTag/elbv2.k8s.aws/cluster": "false"

}

}

},

{

"Effect": "Allow",

"Action": [

"elasticloadbalancing:CreateLoadBalancer",

"elasticloadbalancing:CreateTargetGroup"

],

"Resource": "\*",

"Condition": {

"Null": {

"aws:RequestTag/elbv2.k8s.aws/cluster": "false"

}

}

},

{

"Effect": "Allow",

"Action": [

"elasticloadbalancing:CreateListener",

"elasticloadbalancing:DeleteListener",

"elasticloadbalancing:CreateRule",

"elasticloadbalancing:DeleteRule"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"elasticloadbalancing:AddTags",

"elasticloadbalancing:RemoveTags"

],

"Resource": [

"arn:aws:elasticloadbalancing:\*:\*:targetgroup/\*/\*",

"arn:aws:elasticloadbalancing:\*:\*:loadbalancer/net/\*/\*",

"arn:aws:elasticloadbalancing:\*:\*:loadbalancer/app/\*/\*"

],

"Condition": {

"Null": {

"aws:RequestTag/elbv2.k8s.aws/cluster": "true",

"aws:ResourceTag/elbv2.k8s.aws/cluster": "false"

}

}

},

{

"Effect": "Allow",

"Action": [

"elasticloadbalancing:AddTags",

"elasticloadbalancing:RemoveTags"

],

"Resource": [

"arn:aws:elasticloadbalancing:\*:\*:listener/net/\*/\*/\*",

"arn:aws:elasticloadbalancing:\*:\*:listener/app/\*/\*/\*",

"arn:aws:elasticloadbalancing:\*:\*:listener-rule/net/\*/\*/\*",

"arn:aws:elasticloadbalancing:\*:\*:listener-rule/app/\*/\*/\*"

]

},

{

"Effect": "Allow",

"Action": [

"elasticloadbalancing:ModifyLoadBalancerAttributes",

"elasticloadbalancing:SetIpAddressType",

"elasticloadbalancing:SetSecurityGroups",

"elasticloadbalancing:SetSubnets",

"elasticloadbalancing:DeleteLoadBalancer",

"elasticloadbalancing:ModifyTargetGroup",

"elasticloadbalancing:ModifyTargetGroupAttributes",

"elasticloadbalancing:DeleteTargetGroup"

],

"Resource": "\*",

"Condition": {

"Null": {

"aws:ResourceTag/elbv2.k8s.aws/cluster": "false"

}

}

},

{

"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",

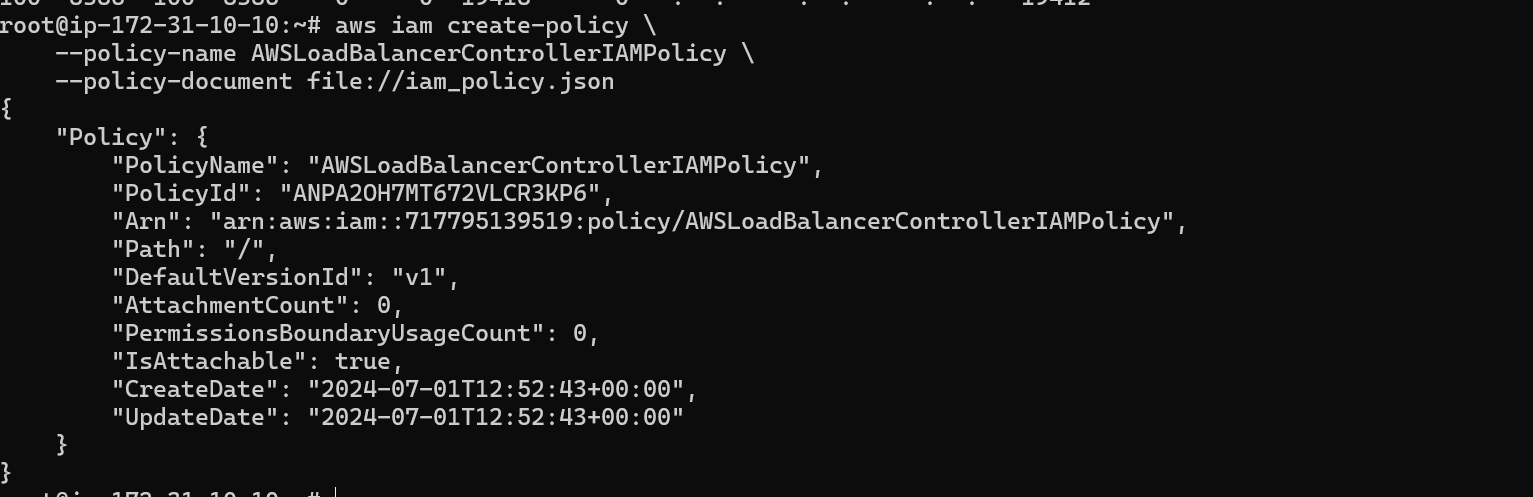
"elasticloadbalancing:ModifyRule"

],

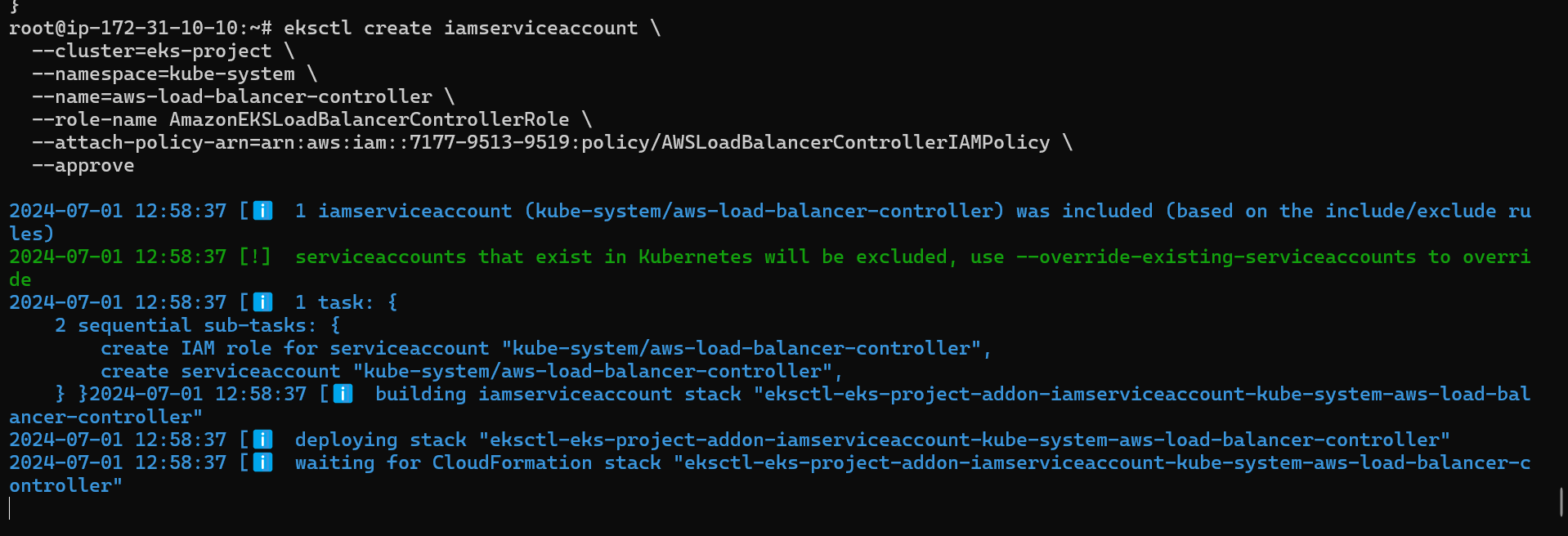
"Resource": "\*"

}

This is a iam policy in a namespace of kube-system

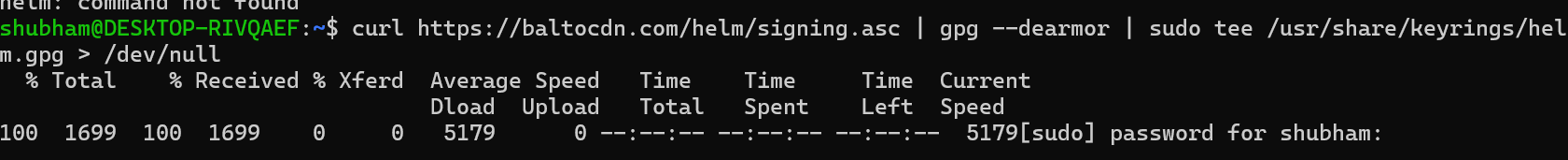




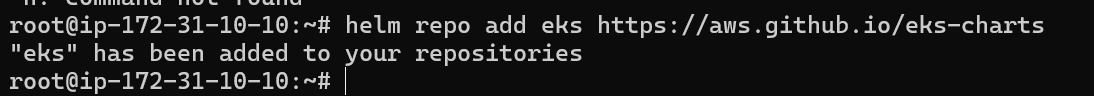
we create an IAM service account. Make sure to update it with your cluster name, namespace, and AWS account number as needed . 

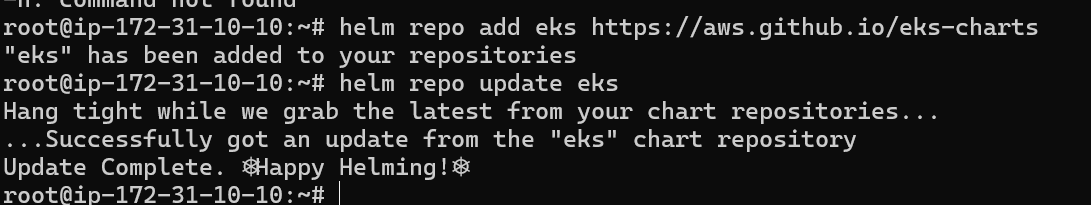


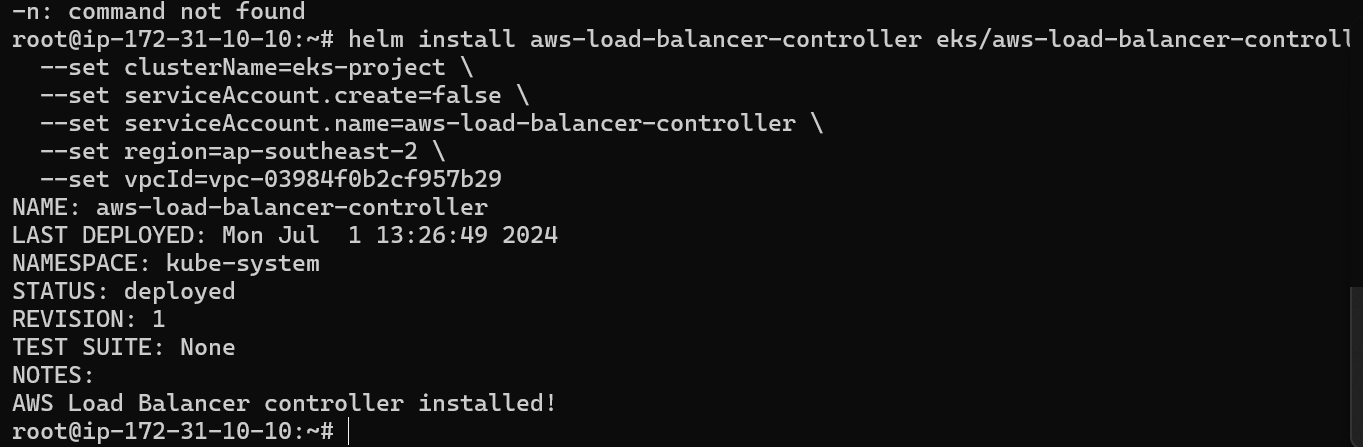
Here we have to install helm for that we have to run by below command

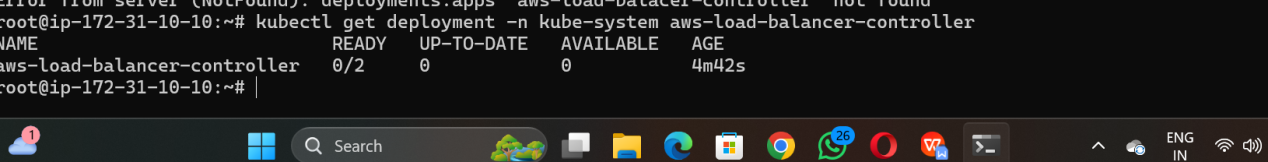


Here we have run this command

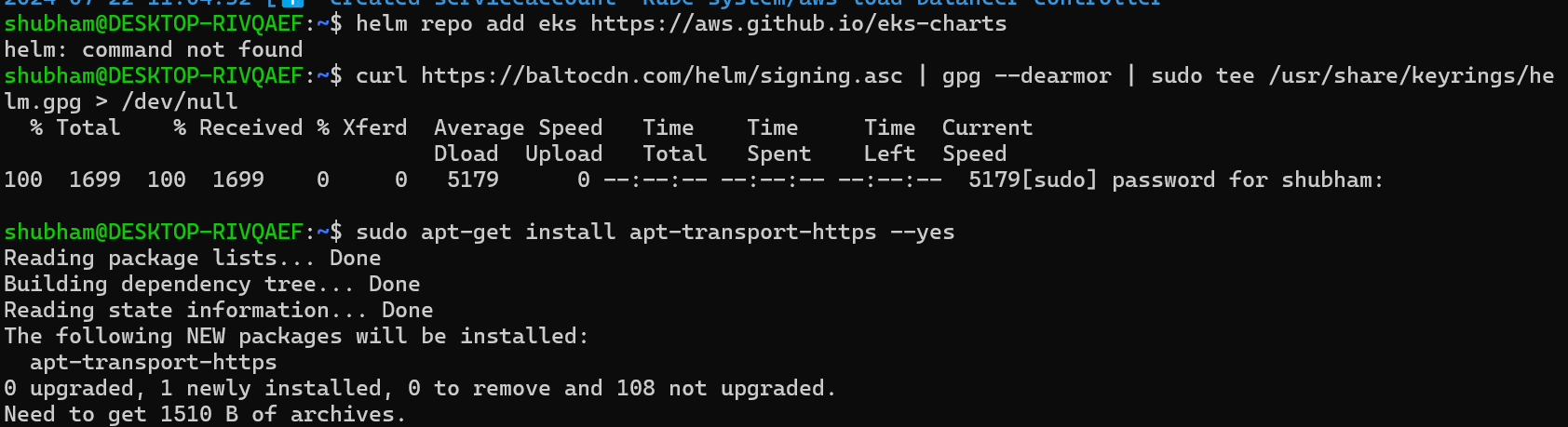


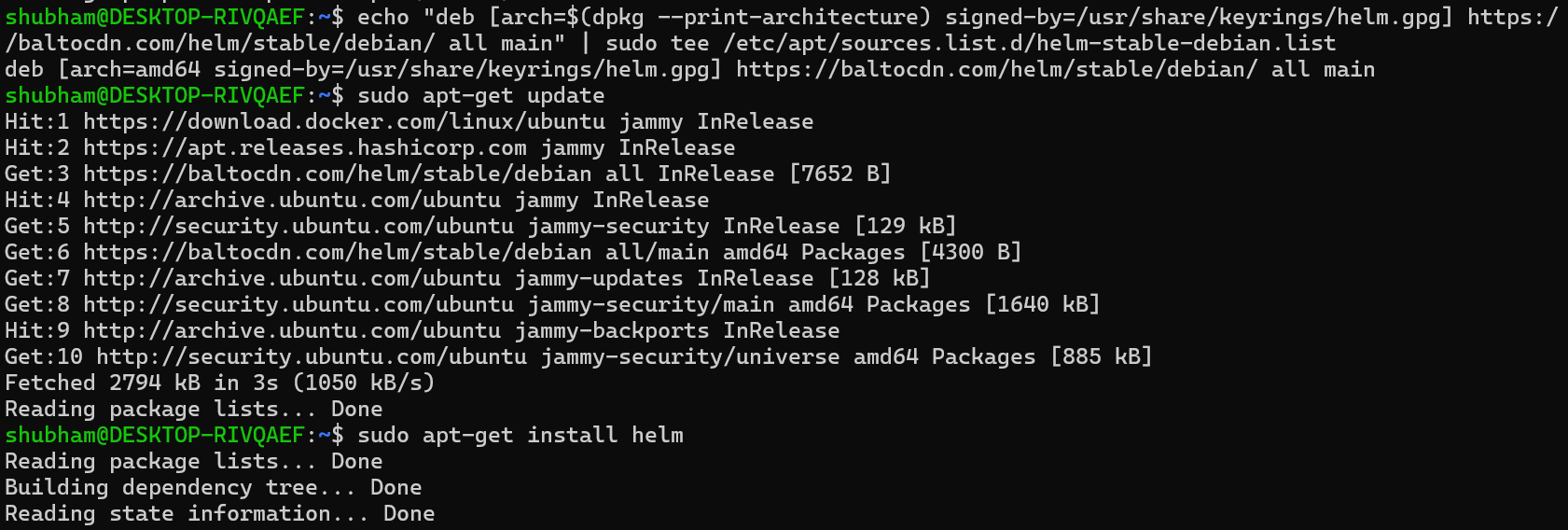


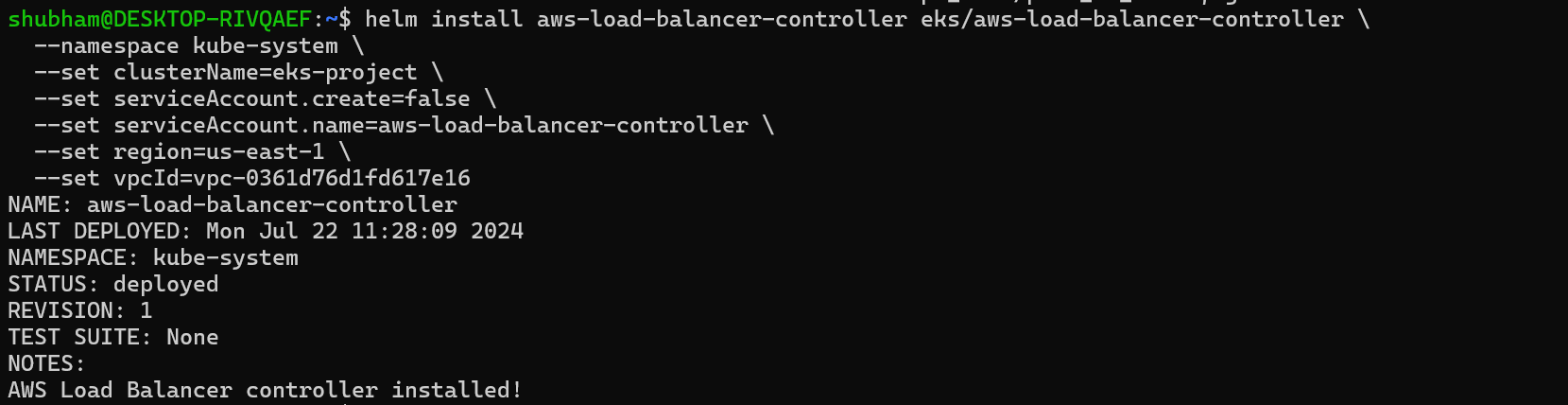




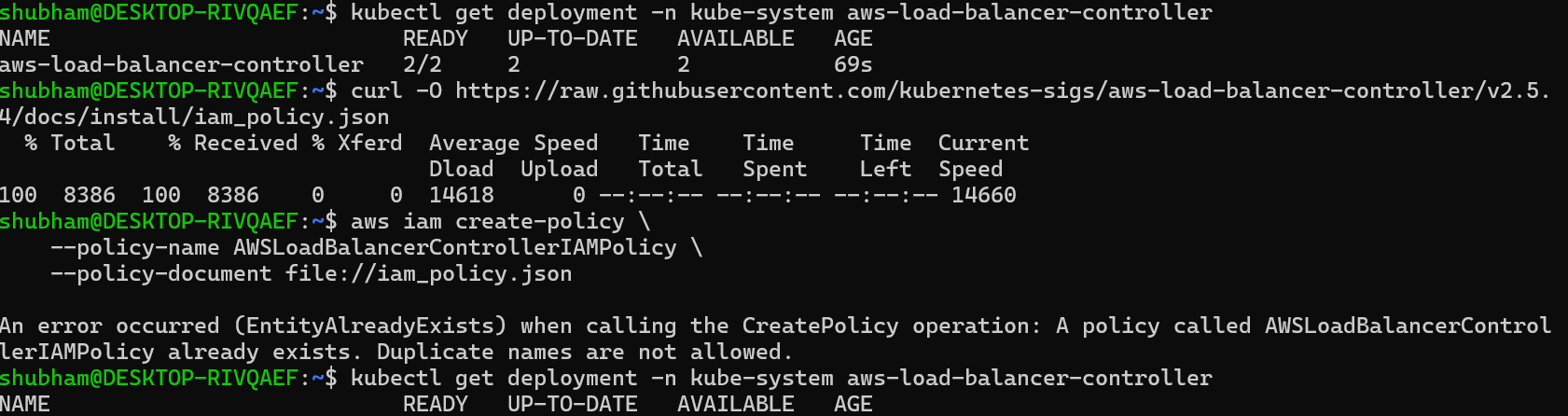
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/nullsudo apt-get install apt-transport-https --yesecho "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.listsudo apt-get updatesudo apt-get install helm



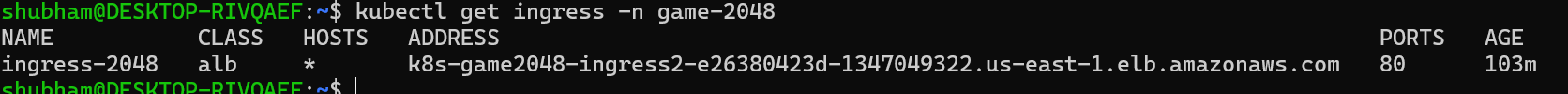




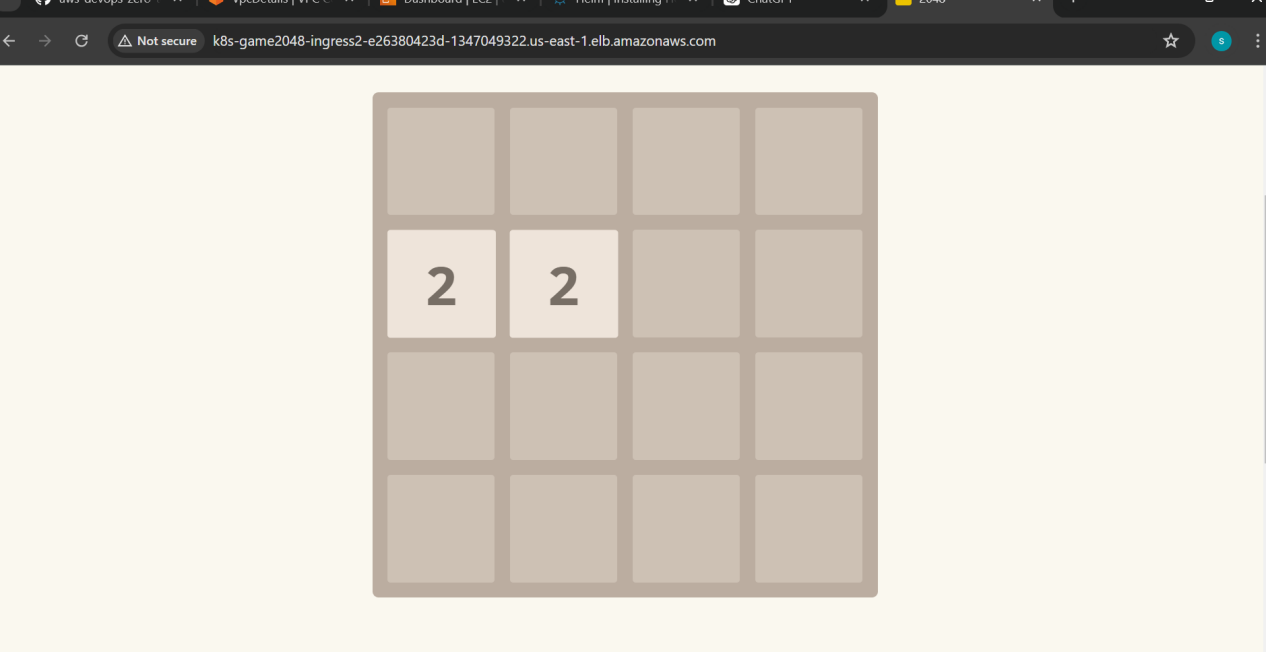
Check if all pods are working properly or not.



Then, you need to copy this link and paste it into your browser.



Here is over game we just have to copy the ingress and run the game



This document was created by Shubham shinde