

```
winget install Amazon.AWSCLI
```

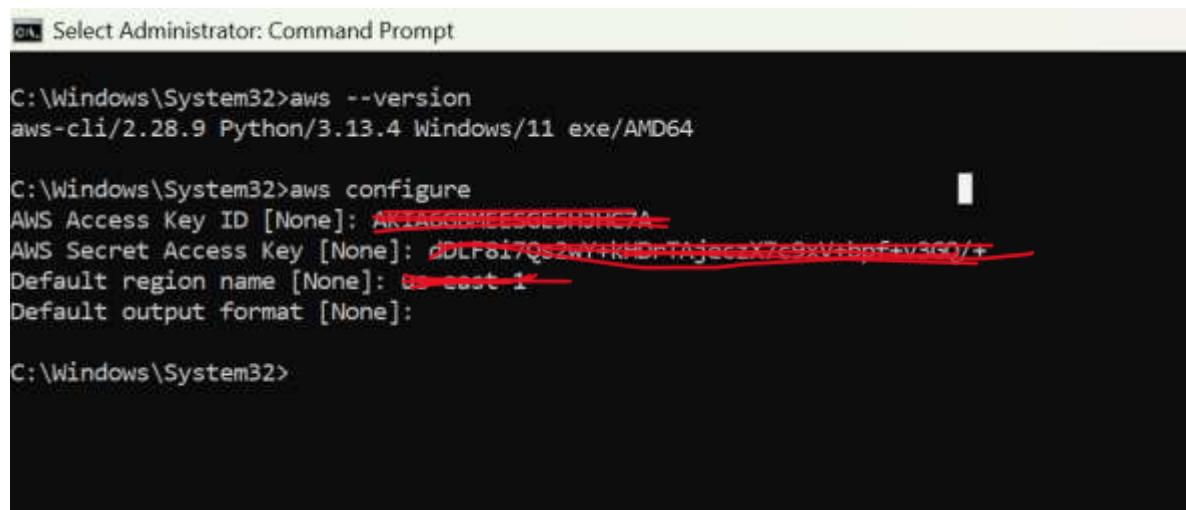
if winget not installed

```
https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi
```

```
#check version
```

```
aws --version
```

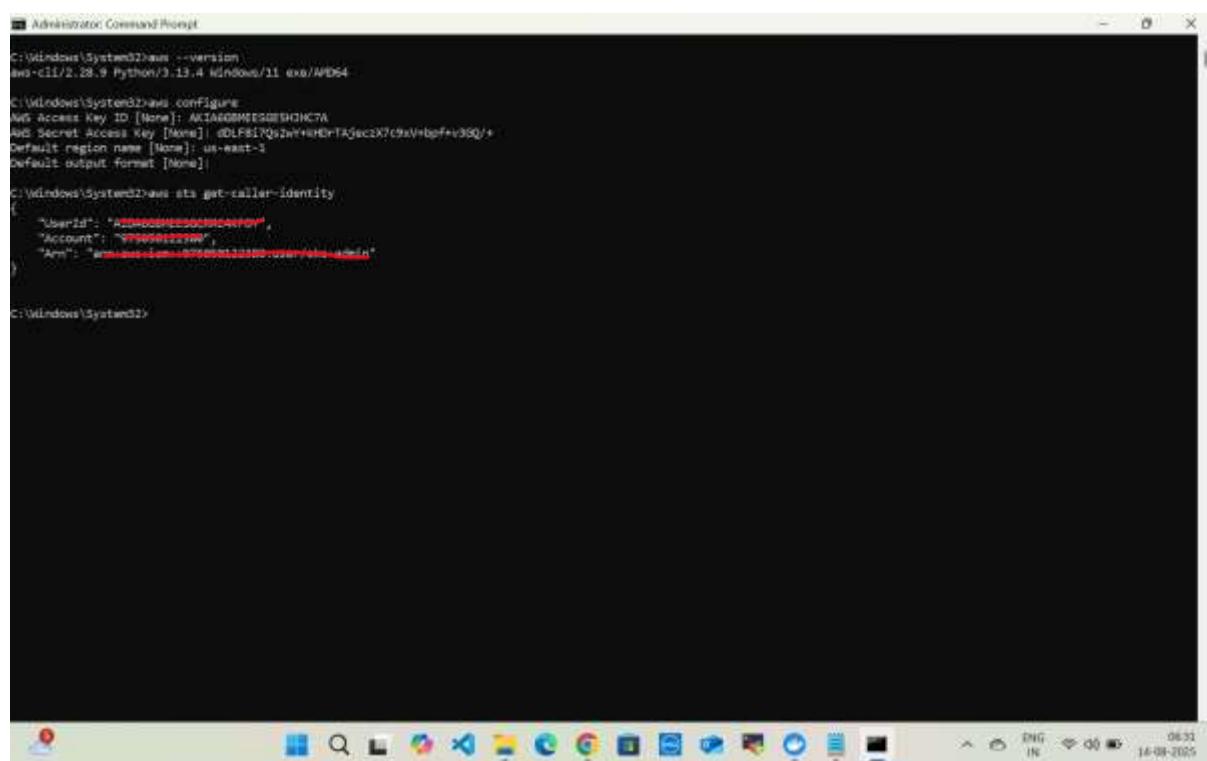
```
aws configure
```



```
C:\Windows\System32>aws --version
aws-cli/2.28.9 Python/3.13.4 Windows/11 exe/AMD64

C:\Windows\System32>aws configure
AWS Access Key ID [None]: AKIAJGGMEE5G5HJC7A
AWS Secret Access Key [None]: dDLF8i7QzWYikHDfTAjczX7c9xVtbpFtv3gQ/+
Default region name [None]: us-east-1
Default output format [None]:
```

```
aws sts get-caller-identity
```



```
C:\Windows\System32>aws --version
aws-cli/2.28.9 Python/3.13.4 Windows/11 exe/AMD64

C:\Windows\System32>aws configure
AWS Access Key ID [None]: AKIAJGGMEE5G5HJC7A
AWS Secret Access Key [None]: dDLF8i7QzWYikHDfTAjczX7c9xVtbpFtv3gQ/+  
Default region name [None]: us-east-1
Default output format [None]
```

```
C:\Windows\System32>aws sts get-caller-identity
{
    "UserId": "AWSUSERID",
    "Account": "ACCOUNTID",
    "Arn": "ARN"
}
```

Add permission to IAM

Add permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
Copy of group credentials, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1388)

Filter by Type All types 1 match

Policy name	Type	Attached entities
AmazonEC2FullAccess	AWS managed	0

[Cancel](#) [Next](#)

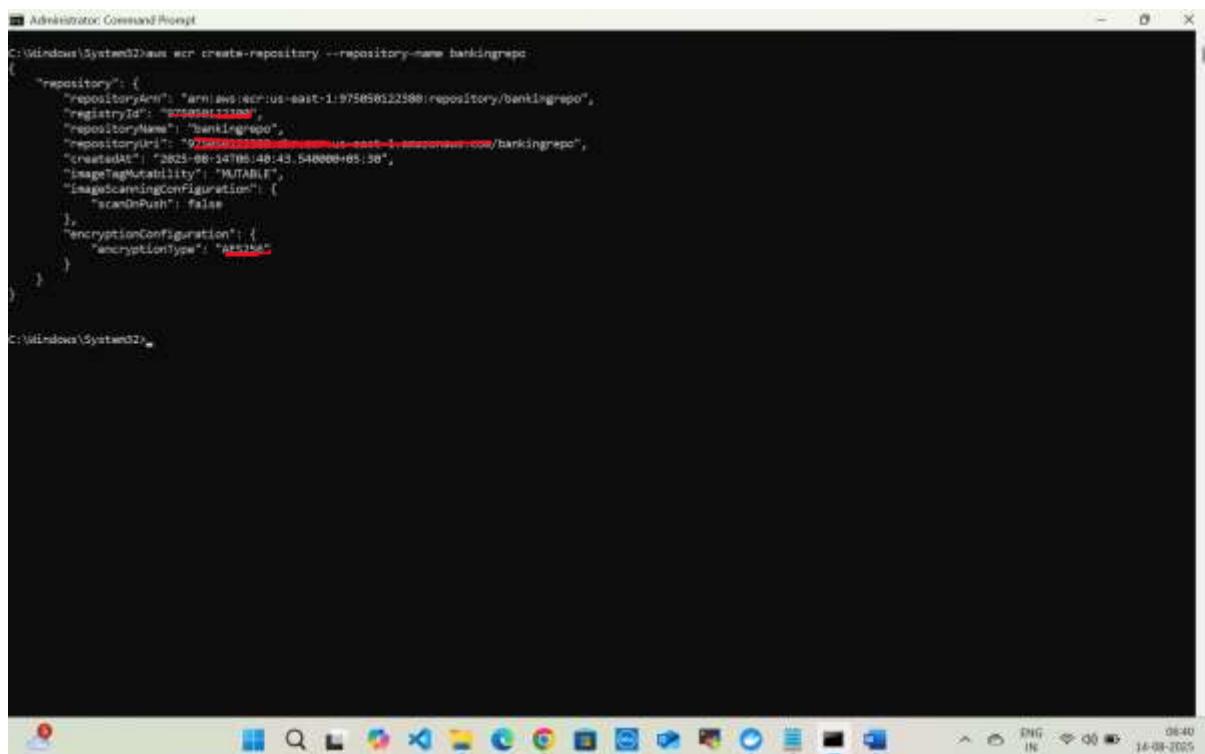
Permissions policies (1/1387)

Filter by Type All types 8 matches

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryFullAccess	AWS managed	1
<input type="checkbox"/> AmazonEC2ContainerRegistryPowerUser	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryPullOnly	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryReadOnly	AWS managed	1
<input type="checkbox"/> AmazonEC2ContainerServiceAutoscaleRole	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceEventsRole	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceforEC2Role	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceRole	AWS managed	0

Create container repo

```
aws ecr create-repository --repository-name bankingrepo
```



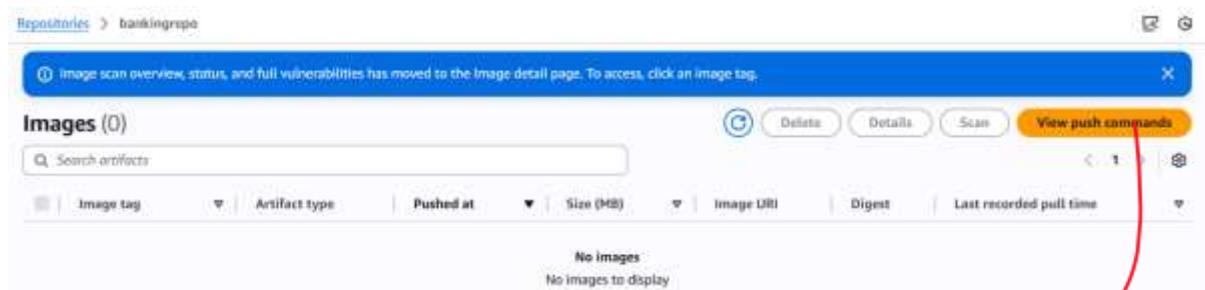
```
C:\Windows\System32\cmd.exeAdministrator: Command Prompt

C:\Windows\System32>aws ecr create-repository --repository-name bankingrepo
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-1:975050122388:repository/bankingrepo",
        "registryId": "975050122388",
        "repositoryName": "bankingrepo",
        "repositoryUri": "https://975050122388.dkr.ecr.us-east-1.amazonaws.com/bankingrepo",
        "createdAt": "2023-08-14T05:48:43.548000+05:30",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AWS_KMS"
        }
    }
}

C:\Windows\System32>
```

#get docker login credential

Get Screen shot from



Push commands for bankingrepo

X

macOS / Linux

Windows

Make sure that you have the latest version of the AWS TOOLS for PowerShell and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS TOOLS for PowerShell:

```
 (Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin 975050122380.dkr.ecr.us-east-1.amazonaws.com
```

Note: If you receive an error using the AWS TOOLS for PowerShell, make sure that you have the latest version of the AWS TOOLS for PowerShell and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
 docker build -t bankingrepo .
```

3. After the build completes, tag your image so you can push the image to this repository:

```
 docker tag bankingrepo:latest 975050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo:latest
```

4. Run the following command to push this image to your newly created AWS repository:

```
 docker push 975050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo:latest
```

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 975050122380.dkr.ecr.us-east-1.amazonaws.com



```
C:\Windows\System32\cmd.exe
C:\Windows\System32>aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 975050122380.dkr.ecr.us-east-1.amazonaws.com
Login Succeeded
C:\Windows\System32>
```

```
Administrator: Command Prompt  
C:\Windows\System32>cd E:\MLT\PRSAU02025\customerapi  
C:\Windows\System32>  
E:\MLT\PRSAU02025\customerapi>docker build -f dockerfile -t customerapi .  
[+] Building 2.0s (14/14) FINISHED  
=> [internal] load build definition from dockerfile  
=> [internal] load metadata for docker://library/openjdk:11-jdk  
=> [internal] load metadata for docker://library/openjdk:11-jdk  
=> [auth] jelastic/venn pull token for registry-1.docker.io  
=> [auth] library/openjdk:pull token for registry-1.docker.io  
=> [internal] load dockerignore  
=> transferring context: 2B  
=> [stage 1/2] FROM docker://library/openjdk:11-jdk@sha256:a98a795d70439e01724155ca9ba115798baef2c50aef7845cf70404a51ac  
=> [internal] resolve docker://library/openjdk:11-jdk@sha256:a98a795d70439e01724155ca9ba115798baef2c50aef7845cf70404a51ac  
=> [internal] load build context  
=> transferring context: 3.4MB  
=> [build 1/9] FROM docker://library/openjdk:11-jdk@sha256:a98a795d70439e01724155ca9ba115798baef2c50aef7845cf70404a51ac  
=> [internal] resolve docker://library/openjdk:11-jdk@sha256:a98a795d70439e01724155ca9ba115798baef2c50aef7845cf70404a51ac  
> CACHED [build 2/9] COPY .src/. .  
> CACHED [build 3/9] COPY pom.xml pom.xml  
> CACHED [build 4/9] RUN mvn clean install -Dmaven.test.skip=true  
> CACHED [stage-1 2/2] COPY --from=build target/customerapi-0.0.1-SNAPSHOT.jar customerapi-0.0.1-SNAPSHOT.jar  
=> exporting layers  
=> exporting manifest: sha256:0d88fbef72a21ed0db577eaed9118ed9a520d315a10a8a0303184e4d4d0815  
=> exporting config: sha256:4cd85a7788a92238159402161001ab49ff138814c58669229739a6cc11c1  
=> exporting attachment manifest: sha256:3add7948a719f3c454124ef92a1c1303161178041cb900254fae499b02ef  
=> exporting manifest list: sha256:f43872ae7f8e169248c79287a2b1a6fd1a7e22374a9513c119ec6e845f3  
>> moving to docker://library/customerapi:latest  
>> inspecting to docker://library/customerapi:latest  
view build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/mad4lk5szz194bly8ekj:  
1 warning found (use docker --debug to expand).  
- 2504argsRecommended: 2504 arguments recommended for ENTRYPOINT to prevent unintended behavior related to OS signals (line 13)  
E:\MLT\PRSAU02025\customerapi>
```

docker tag customerapi:latest 97050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo:customerapiv1

docker push 97050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo:customerapiv1

```
C:\Windows\System32>docker tag customerapi:latest 97050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo:customerapiv1  
C:\Windows\System32>docker push 97050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo:customerapiv1  
The push refers to repository [97050122380.dkr.ecr.us-east-1.amazonaws.com/bankingrepo]  
74dfdf4f10b8f1: Pushed  
5262579a8e45: Pushed  
6d239e44642e: Pushed  
8eab4e2287a5: Pushed  
7c802a8f6062: Pushed  
customerapiv1: digest: sha256:fa3872ecffab6c802a0194c778287a2016e8bd1a7e323f4d4913c119ec6e845f3 size: 856
```

```
C:\Windows\System32>
```

<input type="checkbox"/> Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
<input type="checkbox"/>	customerapi:v1	Image Index	August 14, 2025, 08:17:32 (UTC+05:5)	315.17	<input type="button"/> Copy URI	<input type="button"/> sha256:fa3072ecfffa6c002...
<input type="checkbox"/>	-	Image	August 14, 2025, 08:17:31 (UTC+05:5)	0.00	<input type="button"/> Copy URI	<input type="button"/> sha256:5addf7048a7fbfb...
<input type="checkbox"/>	-	Image	August 14, 2025, 08:17:31 (UTC+05:5)	315.17	<input type="button"/> Copy URI	<input type="button"/> sha256:0688bbfe72a2d4...

OIDCPolicy

{

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

 "Statement": [

 {

 "Effect": "Allow",

 "Action": [

 "iam:GetOpenIDConnectProvider",

 "iam>CreateOpenIDConnectProvider",

 "iam>DeleteOpenIDConnectProvider",

 "iam:UpdateOpenIDConnectProviderThumbprint",

 "iam:TagOpenIDConnectProvider",

 "iam>CreatePolicy",

 "iam:GetPolicy",

 "iam:AttachRolePolicy",

 "iam>CreateRole",

 "iam:PassRole",

 "iam:TagRole",

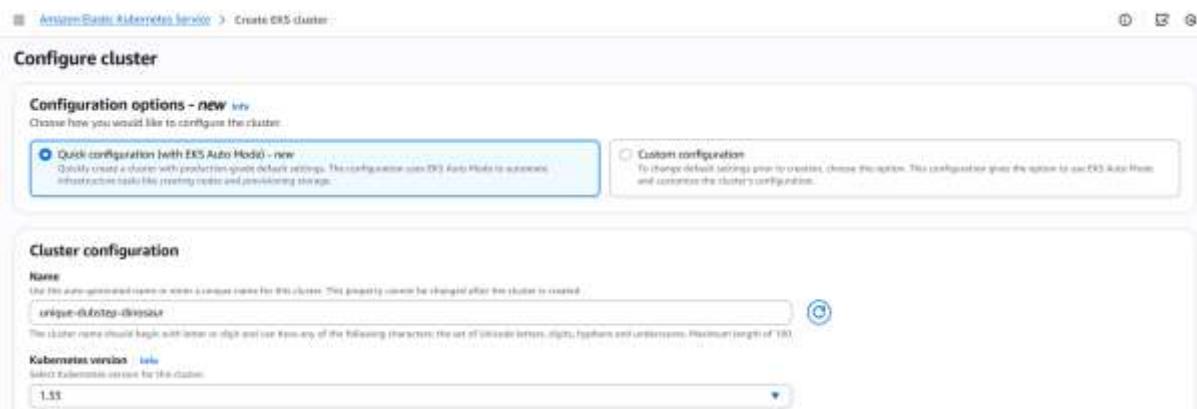
 "iam:GetRole"

],

 "Resource": "*"

```
    }  
]  
}
```

Create an EKS cluster (with eksctl)



EKS cluster Policy

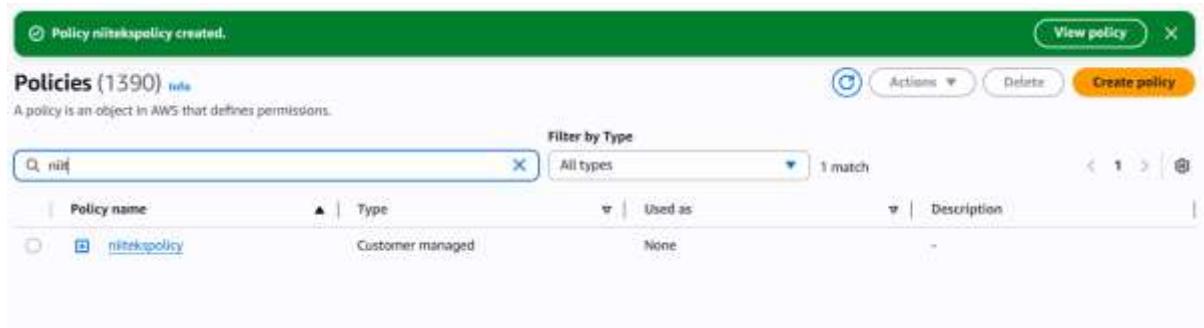
```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AmazonEKSClusterPolicy",  
      "Effect": "Allow",  
      "Action": [  
        "autoscaling:DescribeAutoScalingGroups",
```

"autoscaling:UpdateAutoScalingGroup",
"ec2:AttachVolume",
"ec2:AuthorizeSecurityGroupIngress",
"ec2>CreateRoute",
"ec2>CreateSecurityGroup",
"ec2>CreateTags",
"ec2>CreateVolume",
"ec2>DeleteRoute",
"ec2>DeleteSecurityGroup",
"ec2>DeleteVolume",
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeVolumes",
"ec2:DescribeVolumesModifications",
"ec2:DescribeVpcs",
"ec2:DescribeDhcpOptions",
"ec2:DescribeNetworkInterfaces",
"ec2:DescribeAvailabilityZones",
"ec2:DetachVolume",
"ec2:ModifyInstanceAttribute",
"ec2:ModifyVolume",
"ec2:RevokeSecurityGroupIngress",
"ec2:DescribeAccountAttributes",
"ec2:DescribeAddresses",
"ec2:DescribeInternetGateways",
"ec2:DescribeInstanceTopology",

"elasticloadbalancing:AddTags",
"elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
"elasticloadbalancing:AttachLoadBalancerToSubnets",
"elasticloadbalancing:ConfigureHealthCheck",
"elasticloadbalancing:CreateListener",
"elasticloadbalancing:CreateLoadBalancer",
"elasticloadbalancing:CreateLoadBalancerListeners",
"elasticloadbalancing:CreateLoadBalancerPolicy",
"elasticloadbalancing:CreateTargetGroup",
"elasticloadbalancing:DeleteListener",
"elasticloadbalancing:DeleteLoadBalancer",
"elasticloadbalancing:DeleteLoadBalancerListeners",
"elasticloadbalancing:DeleteTargetGroup",
"elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
"elasticloadbalancing:DeregisterTargets",
"elasticloadbalancing:DescribeListeners",
"elasticloadbalancing:DescribeLoadBalancerAttributes",
"elasticloadbalancing:DescribeLoadBalancerPolicies",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeTargetGroupAttributes",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:DescribeTargetHealth",
"elasticloadbalancing:DetachLoadBalancerFromSubnets",
"elasticloadbalancing:ModifyListener",
"elasticloadbalancing:ModifyLoadBalancerAttributes",
"elasticloadbalancing:ModifyTargetGroup",
"elasticloadbalancing:ModifyTargetGroupAttributes",
"elasticloadbalancing:RegisterInstancesWithLoadBalancer",

```
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:SetLoadBalancerPoliciesForBackendServer",
        "elasticloadbalancing:SetLoadBalancerPoliciesOfListener",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
{
    "Sid": "AmazonEKSClusterPolicySLRCREATE",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:AWSServiceName": "elasticloadbalancing.amazonaws.com"
        }
    }
},
{
    "Sid": "AmazonEKSClusterPolicyENIDelete",
    "Effect": "Allow",
    "Action": "ec2:DeleteNetworkInterface",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/eks:eni:owner": "amazon-vpc-cni"
        }
    }
}
```

}



Secret Manager read write policy

```
        "kms:DescribeKey",
        "kms>ListAliases",
        "kms>ListKeys",
        "lambda>ListFunctions",
        "rds>DescribeDBClusters",
        "rds>DescribeDBInstances",
        "redshift>DescribeClusters",
        "redshift-serverless>ListWorkgroups",
        "redshift-serverless>GetNamespace",
        "tag>GetResources"
    ],
    "Resource": "*"
},
{
    "Sid": "LambdaPermissions",
    "Effect": "Allow",
    "Action": [
        "lambda>AddPermission",
        "lambda>CreateFunction",
        "lambda>GetFunction",
        "lambda>InvokeFunction",
        "lambda>UpdateFunctionConfiguration"
    ],
    "Resource": "arn:aws:lambda:*:*:function:SecretsManager*"
},
{
    "Sid": "SARPermissions",
    "Effect": "Allow",

```

```

    "Action": [
        "serverlessrepo>CreateCloudFormationChangeSet",
        "serverlessrepo:GetApplication"
    ],
    "Resource": "arn:aws:serverlessrepo:*:*:applications/SecretsManager*"
},
{
    "Sid": "S3Permissions",
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::awsserverlessrepo-changesets*",
        "arn:aws:s3:::secrets-manager-rotation-apps-*/*"
    ]
}
]
}

```

The screenshot shows the AWS IAM Policies list. A green banner at the top indicates that the policy 'n1tsecretmanagerpolicy' has been created. The main table lists two policies:

Policy name	Type	Used as	Description
n1tsecretmanagerpolicy	Customer managed	None	-
n1tsecretmanagerpolicy	Customer managed	None	-

Noderole Policy (Ec2 usecase)

Permissions policies (5) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryFullAccess	AWS managed	2
AmazonEC2ContainerRegistryReadOnly	AWS managed	1
AmazonEKS_CNI_Policy	AWS managed	1
AmazonEKSWorkerNodePolicy	AWS managed	1
SecretsManagerReadWrite	AWS managed	2

Create role

Role niiitnodepolicy created.

niiitnodepolicy Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date: August 14, 2023, 08:46 (UTC+05:30)

Last activity:

ARN: arn:aws:iam::975050122580:role/niiitnodepolicy

Instance profile ARN: arn:aws:iam::975050122580:instance-profile/niiitnodepolicy

Maximum session duration: 1 hour

Permissions **Trust relationships** **Tags** **Last Accessed** **Revoke sessions**

Permissions policies (4) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryFullAccess	AWS managed	3
AmazonEKS_CNI_Policy	AWS managed	2
AmazonEKSWorkerNodePolicy	AWS managed	2
SecretsManagerReadWrite	AWS managed	3

Niit cluster admin role (EKS usecase)

Add permissions

Permissions policies (2/1076) Info

Choose one or more policies to attach to your new role.

Filter by Type: All types

Policy name	Type	Description
<input checked="" type="checkbox"/> niiitnodepolicy	Customer managed	
<input checked="" type="checkbox"/> niiitclusteradminpolicy	Customer managed	

The screenshot shows the AWS IAM Roles page. At the top, a green banner indicates that the role 'niiitclusteradmin' has been created. Below this, the 'Roles (21)' section is shown, with a search bar containing 'niiit'. A table lists three roles: 'niiitclusteradmin' (selected), 'AWS Lambda', and 'awslogs-policy'. The 'niiitclusteradmin' row shows it was last updated on August 14, 2025, at 09:07 UTC+05:30, with an entity ARN of arn:aws:iam::975050122580:role/niiitnoderole. It has a maximum session duration of 1 hour and was last active 7 minutes ago. Below this, tabs for 'Permissions', 'Trust relationships', 'Tags', 'Last Accessed', and 'Revoke sessions' are visible.

Permissions

Permissions policies (4)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryFullAccess	AWS managed	3
AmazonEKS_CNI_Policy	AWS managed	2
AmazonEKSWorkerNodePolicy	AWS managed	2
SecretsManagerReadWrite	AWS managed	5

Niiit cluster admin role

Add

Cluster role missing recommended managed policies

The cluster role must have the following managed policies or equivalent permissions to use EKS Auto Mode:

- AmazonEKSBlockStoragePolicy
- AmazonEKSCo~~Compute~~Policy
- AmazonEKSLoadBalancingPolicy
- AmazonEKSNetworkingPolicy

niitclusteradmin Info

Moves the cluster administrator control plane to manage ARN resources on your behalf.

Delete

Summary

Creation date: August 14, 2025, 06:53 (UTC+05:30)

Last activity: -

ARN: arn:aws:eks:ap-south-1:111111111111:cluster/niitclusteradmin

Max session duration: 1 hour

Permissions Trust relationships Tags Last Accessed Revoked sessions

Permissions policies (6) Info

You can attach up to 10 managed policies.

Search Filter by Type: All type Attached entities

Policy name	Type	Attached entities
AmazonECSFargatePolicy	AWS managed	1
AmazonECCUserPolicy	AWS managed	1
AmazonECAccessPolicy	AWS managed	1
AmazonECAuditAccessPolicy	AWS managed	1
AmazonECCNetworkingPolicy	AWS managed	1
AmazonECAccessPolicy	AWS managed	1

Unlink Delete Add permissions

Edit Trust policy of cluster admin

≡ [IAM](#) > [Roles](#) > [niitclusteradmin](#) > Edit trust policy

Edit trust policy

```
1 ▼ {  
2     "Version": "2012-10-17",  
3     "Statement": [{  
4         "Effect": "Allow",  
5         "Principal": { "Service": "eks.amazonaws.com" },  
6         "Action": ["sts:AssumeRole", "sts:TagSession"]  
7     }]  
8 }  
9 |
```

[Amazon EKS Kubernetes Service](#) > Create EKS cluster

Cluster configuration

Name
Use the auto-generated name or enter a unique name for this cluster. This property cannot be changed after the cluster is created.

Kubernetes version [Info](#)
Select Kubernetes version for this cluster:

1.31

Cluster IAM role [Info](#)
Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

arn:aws:iam::975050122380:role/mitchusteradmin

Node IAM role [Info](#)
Select the Node IAM role to launch and register with a cluster. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

arn:aws:iam::975050122380:role/mitchusternode

VPC [Info](#)
Select a VPC to use for your EKS cluster resources.

vpc-0f9bfe500050015a | Default

Subnets [Info](#)
Choose the subnets in your VPC where the control plane may place static network interfaces (IPs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the VPC console.

Select subnets

- subnet-0d001e565425400cb
az-1: 172.31.16.0/20 - Type: Public
- subnet-0320556c735f01090
az-1: 172.31.80.0/20 - Type: Public
- subnet-0619a860d23446cb8
az-1: 172.31.31.0/20 - Type: Public
- subnet-098fc92192a2200d
az-1: 172.31.6.0/20 - Type: Public

[Create VPC](#)

[Create recommended role](#)

[Create recommended role](#)

[Clear selected subnets](#)

bankingcluster

The Amazon EKS console does not show Kubernetes resources until the cluster has an ACTIVE or UPDATING state. Please try again in a few minutes.

Cluster info [Info](#)

Status: Creating	Kubernetes version: Info 1.31	Support period: Standard support until November 26, 2025	Provider: EKS
Cluster health: 0	Upgrade insights: 0	Node health issues: 0	

[Overview](#) [Resources](#) [Compute](#) [Networking](#) [Add-ons](#) [Access](#) [Observability](#) [Update history](#) [Tags](#)

Details

API server endpoint: https://2397C20A59B1EA80E759023B3C5A0F0F.gr7.us-east-1.eks.amazonaws.com	OpenID Connect provider URL: -	Created: a few seconds ago
Certificate authority: -	Cluster IAM role ARN: arn:aws:iam::975050122380:role/mitchusteradmin	Cluster ARN: arn:aws:eks:us-east-1:975050122380:cluster/bankingcluster

Or

#eks tool

curl --silent --location

```
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

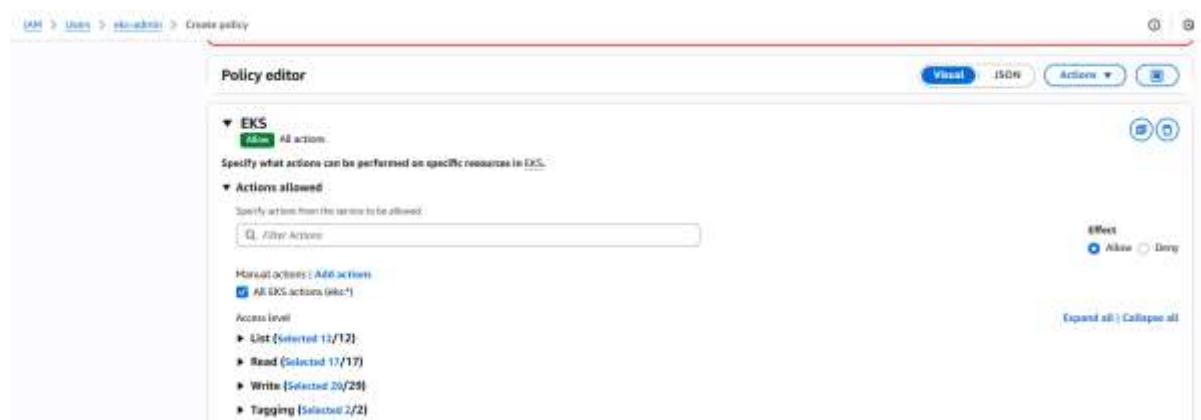
sudo mv /tmp/eksctl /usr/local/bin

```
eksctl version
```

```
#create kube cluster using us-east-2
```

```
eksctl create cluster --name my-cluster --region us-east-2 --nodegroup-name  
standard-workers --node-type t3.medium --nodes 2 --nodes-min 1 --nodes-max 3 --  
managed
```

Add EKS permission to eksadmin



```
#configure aws kube in local
```

Use the AWS CLI update-kubeconfig command to create or update your kubeconfig for your cluster.

```
aws eks update-kubeconfig --name bankingcluster
```

```
C:\Windows\System32\cmd.exe
C:\Windows\System32>aws eks update-kubeconfig --name bankingcluster
Cluster status is CREATING
C:\Windows\System32>kubectl version
Client Version: v1.32.2
Kustomize Version: v5.5.8
Server Version: v1.31.4
C:\Windows\System32>
```

```
aws eks update-kubeconfig --name bankingcluster --region us-east-1
```

```
[Administrator: Command Prompt]
An error occurred (ResourceNotFoundException) when calling the DescribeCluster operation: No cluster found for name: my-cluster.
C:\Windows\System32>aws eks update-kubeconfig --name bankingcluster --region us-east-1
Cluster status is: CREATING
C:\Windows\System32>
```

#check status

```
aws eks describe-cluster --name bankingcluster --region us-east-1 --query
"cluster.status" --output text
```

update config file

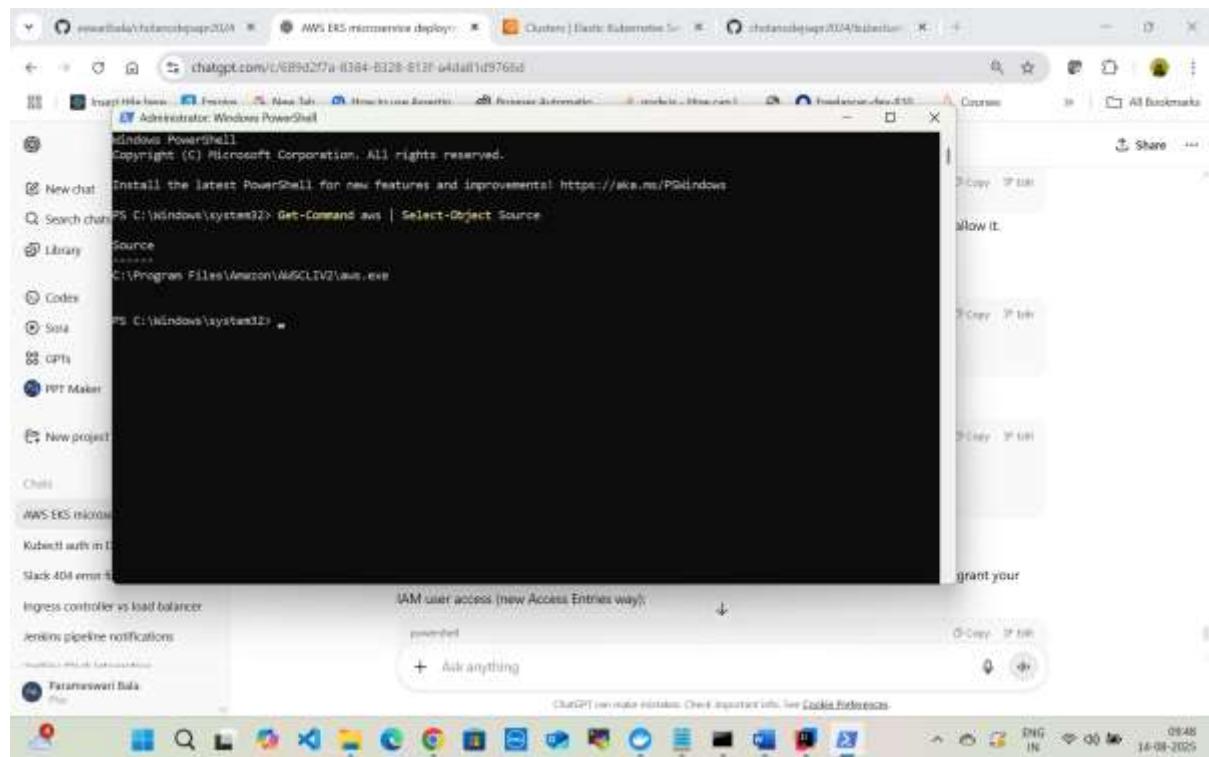
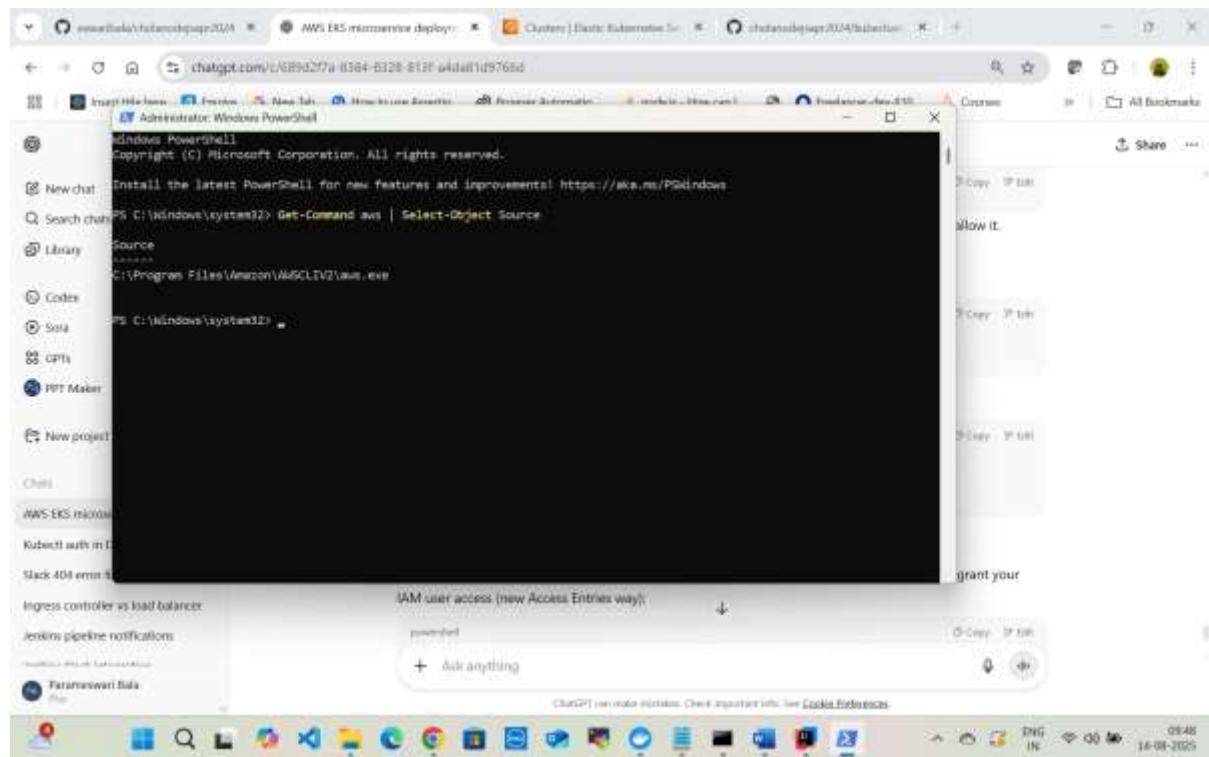
```
aws eks update-kubeconfig --name bankingcluster --region us-east-1 --kubeconfig
"C:\Users\param\.kube\config"
```

kubectl config get-contexts

```
[Administrator: Command Prompt]
C:\Windows\System32>aws sts get-token --cluster-name bankingcluster --region us-east-1 > null
C:\Windows\System32>kubectl config get-contexts
CURRENT      NAME          CLUSTER          AUTHINFO
*           aws:eks:us-east-1:975858122388:cluster/bankingcluster   aws:eks:us-east-1:975858122388:cluster/bankingcluster
/cluster/bankingcluster
  docker-desktop          docker-desktop          docker-desktop
  default                 default                 kind-niit-kube    kind-niit-kube
C:\Windows\System32>
```

In powershell

```
Get-Command aws | Select-Object Source
```



Add access policy to eks admin

The screenshot shows the AWS Elastic Kubernetes Service Cluster details page. In the 'Access policies' section, there are two policies listed: 'AmazonEKSAdminPolicy' and 'AmazonEKSCusterAdminPolicy'. The 'AmazonEKSCusterAdminPolicy' is selected.

```
C:\Windows\System32>kubectl cluster-info
The Kubernetes control plane is running at https://2397C9A5981EA88E759D2380C540F0F.gr7.us-east-1.eks.amazonaws.com
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
C:\Windows\System32>
```

```
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
C:\Windows\System32>kubectl get no
NAME      STATUS   ROLES   AGE   VERSION
kubelet-1  Ready    master  9m   v1.33.9-eks-cc4d9b
C:\Windows\System32>kubectl get pods
No resources found in default namespace.
C:\Windows\System32>kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.3.0.1       <none>        443/TCP     70m
C:\Windows\System32>kubectl get deploy
No resources found in default namespace.
C:\Windows\System32>
```

```
[Administrator: Command Prompt]
2 DRives) 774,138,494,976 bytes free

E:\Docker\postgresdeployment\kubectl apply -f postgres-pv.yaml
persistentvolume/postgres-pv-volume created
persistentvolumeclaim/postgres-pv-claim created

E:\Docker\postgresdeployment\kubectl apply -f postgres-deployment.yaml
service/postgres created
deployment.apps/postgres created

E:\Docker\postgresdeployment\kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
postgres-c0fc89c6-mirkp   0/1   Pending   0        18s

E:\Docker\postgresdeployment\kubectl get pv
NAME           CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
postgres-pv-volume   1Gi       RWO            Retain        Bound   default/postgres-pv-claim   manual   <unset>   37s

E:\Docker\postgresdeployment\kubectl get pvc
NAME           STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
postgres-pv-claim   Bound   postgres-pv-volume   1Gi       RWO            manual   <unset>   41s

E:\Docker\postgresdeployment\kubectl get pods -w
NAME          READY   STATUS    RESTARTS   AGE
postgres-c0fc89c6-mirkp   0/1   Pending   0        31s
postgres-c0fc89c6-mirkp   0/1   Pending   0        42s
postgres-c0fc89c6-mirkp   0/1   ContainerCreating   0        42s
postgres-c0fc89c6-mirkp   1/1   Running   0        47s

E:\Docker\postgresdeployment>
```

```
[Administrator: Command Prompt]
customerservice LoadBalancer 10.100.214.126 pending 7874:38197/TCP 21s
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 77s
postgres ClusterIP None <none> 5432/TCP 569s

E:\Docker\postgresdeployment\kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
customerservice LoadBalancer 10.100.214.126 <pending> 7874:38197/TCP 71s
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 78s
postgres ClusterIP None <none> 5432/TCP 569s

E:\Docker\postgresdeployment\kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
customerservice LoadBalancer 10.100.214.126 <pending> 7874:38197/TCP 186s
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 79s
postgres ClusterIP None <none> 5432/TCP 663s

E:\Docker\postgresdeployment\kubectl delete deploy customerapi
deployment.apps "customerapi" deleted

E:\Docker\postgresdeployment\kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
postgres-c0fc89c6-mirkp   1/1   Running   0        9d99s

E:\Docker\postgresdeployment\kubectl apply -f deployment-v28.yaml
deployment.apps/customersvc created
service/customerservice configured

E:\Docker\postgresdeployment\kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
customersvc-64885cf9b-kw8ds   1/1   Running   0        4s
postgres-c0fc89c6-mirkp   1/1   Running   0        10s

E:\Docker\postgresdeployment\kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
customerservice NodePort 10.100.214.126 <none> 7874:38197/TCP 4m0s
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 89s
postgres ClusterIP None <none> 5432/TCP 10s

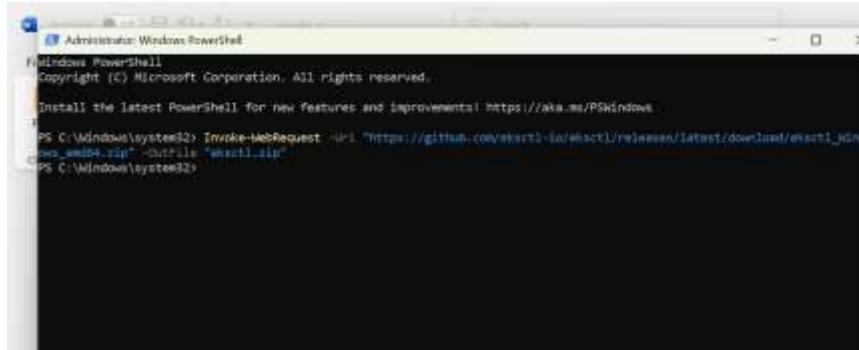
E:\Docker\postgresdeployment>
```

Create load balancer controller

Create EKS Tool

Open powershell

```
Invoke-WebRequest -Uri "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_Windows_amd64.zip" -OutFile "eksctl.zip"
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> Invoke-WebRequest -Uri "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_Windows_amd64.zip" -OutFile "eksctl.zip"
PS C:\Windows\system32>
```

```
Expand-Archive -Path "eksctl.zip" -DestinationPath "C:\eksctl"
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

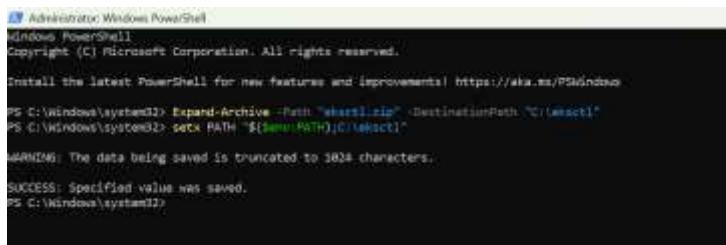
PS C:\Windows\system32> Expand-Archive -Path "eksctl.zip" -DestinationPath "C:\eksctl"
PS C:\Windows\system32>
```

Add eksctl to your PATH

In PowerShell (run as Administrator):

```
setx PATH "$($env:PATH);C:\eksctl"
```

Close and reopen PowerShell so the PATH refreshes.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

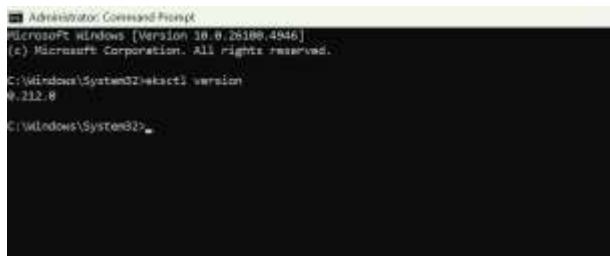
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> Expand-Archive -Path "eksctl.zip" -DestinationPath "C:\eksctl"
PS C:\Windows\system32> setx PATH "$($env:PATH);C:\eksctl"

WARNING: The data being saved is truncated to 1024 characters.

SUCCESS: Specified value was saved.
PS C:\Windows\system32>
```

Set env variable for eksctl.exe



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.26386.4946]
(c) Microsoft Corporation. All Rights Reserved.

C:\Windows\system32>eksctl version
v.232.8

C:\Windows\System32>
```

Go to user eksadmin add policy json

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetOpenIDConnectProvider",  
                "iam>CreateOpenIDConnectProvider",  
                "iam>DeleteOpenIDConnectProvider",  
                "iam:UpdateOpenIDConnectProviderThumbprint",  
                "iam:TagOpenIDConnectProvider",  
                "iam>CreatePolicy",  
                "iam:GetPolicy",  
                "iam:AttachRolePolicy",  
                "iam>CreateRole",  
                "iam:PassRole",  
                "iam:TagRole",  
                "iam:GetRole"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Policy oidcpolicyv1 created.			
Filter by Type			
<input type="checkbox"/> Policy name	Type	<input type="checkbox"/> Attached via	
<input type="checkbox"/> AmazonEC2ContainerRegistryFullAccess	AWS managed	<input type="checkbox"/> Directly	
<input type="checkbox"/> AmazonEC2FullAccess	AWS managed	<input type="checkbox"/> Directly	
<input type="checkbox"/> eks-policy	Customer inline	<input type="checkbox"/> Inline	
<input type="checkbox"/> IAMUserChangePassword	AWS managed	<input type="checkbox"/> Directly	
<input type="checkbox"/> oidcpolicyv1	Customer inline	<input type="checkbox"/> Inline	

```
eksctl utils associate-iam-oidc-provider --cluster bankingcluster --region us-east-1 --approve
```

```
C:\Windows\system32\cmd.exe /s /k & eksctl utils associate-iam-oidc-provider --cluster bankingcluster --region us-east-1 because no identity-based policy allows the iam:DeleteOpenIDConnectProvider action
C:\Windows\system32\cmd.exe /s /k & eksctl utils associate-iam-oidc-provider --cluster bankingcluster --region us-east-1 --approve
2025-08-14 11:09:48 [+] will create IAM Open ID Connect provider for cluster "bankingcluster" in "us-east-1"
2025-08-14 11:09:49 [+] created IAM Open ID Connect provider for cluster "bankingcluster" in "us-east-1"
C:\Windows\system32>
```

Create IAM policy

```
curl -o iam-policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/main/docs/install/iam\_policy.json
```

```
aws iam create-policy --policy-name AWSLoadBalancerControllerIAMPolicy --policy-document file://iam-policy.json
```

```

Administrator: Command Prompt
C:\Windows\System32>aws iam create-policy --policy-name AdLoadBalancerControllerIAMPolicy --policy-document file://iam-policy.json
{
    "Policy": {
        "PolicyName": "AdLoadBalancerControllerIAMPolicy",
        "PolicyId": "AMP46008E15087F7D95B8",
        "Arn": "arn:aws:iam::97958121388:policy/AdLoadBalancerControllerIAMPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2025-08-14T05:39:28+00:00",
        "UpdateDate": "2025-08-14T05:39:28+00:00"
    }
}

C:\Windows\System32>

```

Permissions policies (6)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type			
<input type="text"/> Search	All types	Type	Attached via
<input type="checkbox"/> AmazonEC2ContainerRegistryFullAccess	AWS managed		Directly
<input type="checkbox"/> AmazonEC2FullAccess	AWS managed		Directly
<input type="checkbox"/> AWSCloudFormationFullAccess	AWS managed		Directly
<input type="checkbox"/> ekspolicy	Customer inline		Inline
<input type="checkbox"/> IAMUserChangePassword	AWS managed		Directly
<input type="checkbox"/> nidpolicy1	Customer inline		Inline

2 policies added to eks-admin

Filter by Type			
<input type="text"/> Search	All types	Type	Attached via
<input type="checkbox"/> AmazonEC2ContainerRegistryFullAccess	AWS managed		Directly
<input type="checkbox"/> AmazonEC2FullAccess	AWS managed		Directly
<input type="checkbox"/> AmazonEKSLoadBalancingPolicy	AWS managed		Directly
<input type="checkbox"/> AWSCloudFormationFullAccess	AWS managed		Directly
<input type="checkbox"/> ekspolicy	Customer inline		Inline
<input type="checkbox"/> ElasticLoadBalancingFullAccess	AWS managed		Directly
<input type="checkbox"/> IAMUserChangePassword	AWS managed		Directly
<input type="checkbox"/> nidpolicy1	Customer inline		Inline

```
{  
    "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",
    "ec2:GetSecurityGroupsForVpc",
    "ec2:DescribelpamPools",
    "ec2:DescribeRouteTables",
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DescribeLoadBalancerAttributes",
    "elasticloadbalancing:DescribeListeners",
    "elasticloadbalancing:DescribeListenerCertificates",
    "elasticloadbalancing:DescribeSSLPolicies",
    "elasticloadbalancing:DescribeRules",
    "elasticloadbalancing:DescribeTargetGroups",
    "elasticloadbalancing:DescribeTargetGroupAttributes",
    "elasticloadbalancing:DescribeTargetHealth",
    "elasticloadbalancing:DescribeTags",
    "elasticloadbalancing:DescribeTrustStores",
    "elasticloadbalancing:DescribeListenerAttributes",
    "elasticloadbalancing:DescribeCapacityReservation"
],
{
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "cognito-idp:DescribeUserPoolClient",
    "cognito-idp:ListUserPoolClients",
    "cognito-idp:DescribeUserPool"
  ]
}
```

```
        "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",
        "elasticloadbalancing:ModifyListenerAttributes",
        "elasticloadbalancing:ModifyCapacityReservation",
        "elasticloadbalancing:ModifyIpPools"
    ],
    "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",
        "elasticloadbalancing:SetRulePriorities"
      ],
      "Resource": "*"
    }
]
```

}

AWSLoadBalancerControllerIAMPolicyV1 [Info](#)

[Edit](#) [Delete](#)

Policy details

Type Customer managed	Creation time August 14, 2025, 11:35 (UTC+05:30)	Edited time August 14, 2025, 11:35 (UTC+05:30)	ARN arn:aws:iam::975050122380:policy/AWSLoadBalancerControllerIAMPolicyV1
--------------------------	---	---	--

[Permissions](#) [Entities attached](#) [Tags](#) [Policy versions](#) [Last Accessed](#)

Permissions defined in this policy [Info](#)

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Search

[Edit](#) [Summary](#) [JSON](#)

Allow (9 of 449 services)

Service	Access level	Resource	Request condition
Certificate Manager	Full: List Limited: Read	All resources	None
Cognito User Pools	Limited: Read	All resources	None
ECS	Full: Tagging Limited: List, Read, Write	Multiple	Multiple

[Show remaining 440 services](#)

Create Load Balancer

```
eksctl create iamserviceaccount --cluster bankingcluster --region us-east-1 --namespace kube-system --name aws-load-balancer-controller --role-name AmazonEKSLoadBalancerControllerRole --attach-policy-arn arn:aws:iam::975050122380:policy/AWSLoadBalancerControllerIAMPolicy --override-existing-serviceaccounts --approve
```

```
[Administrator: Command Prompt]
C:\Windows\System32>kubectl create serviceaccount --cluster bankingcluster --region us-east-1 --namespace kube-system --name aws-load-balancer-controller --role Arn:AmazonEKSLoadBalancerControllerRole --attach-policy-arm Arn:iam:iam::975050122588:policy/AmazonEKSLoadBalancerControllerIAMPolicy --override-existing-serviceaccounts --approve
2025-08-14 12:10:08 [+] 1 ServiceAccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2025-08-14 12:10:08 [+] metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set.
2025-08-14 12:10:08 [+] 3 tasks:
  2 sequential sub-tasks:
    - create IAM role for serviceaccount "kube-system/aws-load-balancer-controller",
      create serviceaccount "kube-system/aws-load-balancer-controller",
    - 2025-08-14 12:10:08 [+] Building iamServiceAccount stack "eksctl-bankingcluster-addon-iamServiceAccount-kube-system-aws-load-balancer-controller"
2025-08-14 12:10:08 [+] deploying stack "eksctl-bankingcluster-addon-iamServiceAccount-kube-system-aws-load-balancer-controller"
2025-08-14 12:10:08 [+] waiting for CloudFormation stack "eksctl-bankingcluster-addon-iamServiceAccount-kube-system-aws-load-balancer-controller"
2025-08-14 12:10:08 [+] waiting for CloudFormation stack "eksctl-bankingcluster-addon-iamServiceAccount-kube-system-aws-load-balancer-controller"
2025-08-14 12:10:12 [+] created serviceaccount: "kube-system/aws-load-balancer-controller"

C:\Windows\System32>
```

Confirm

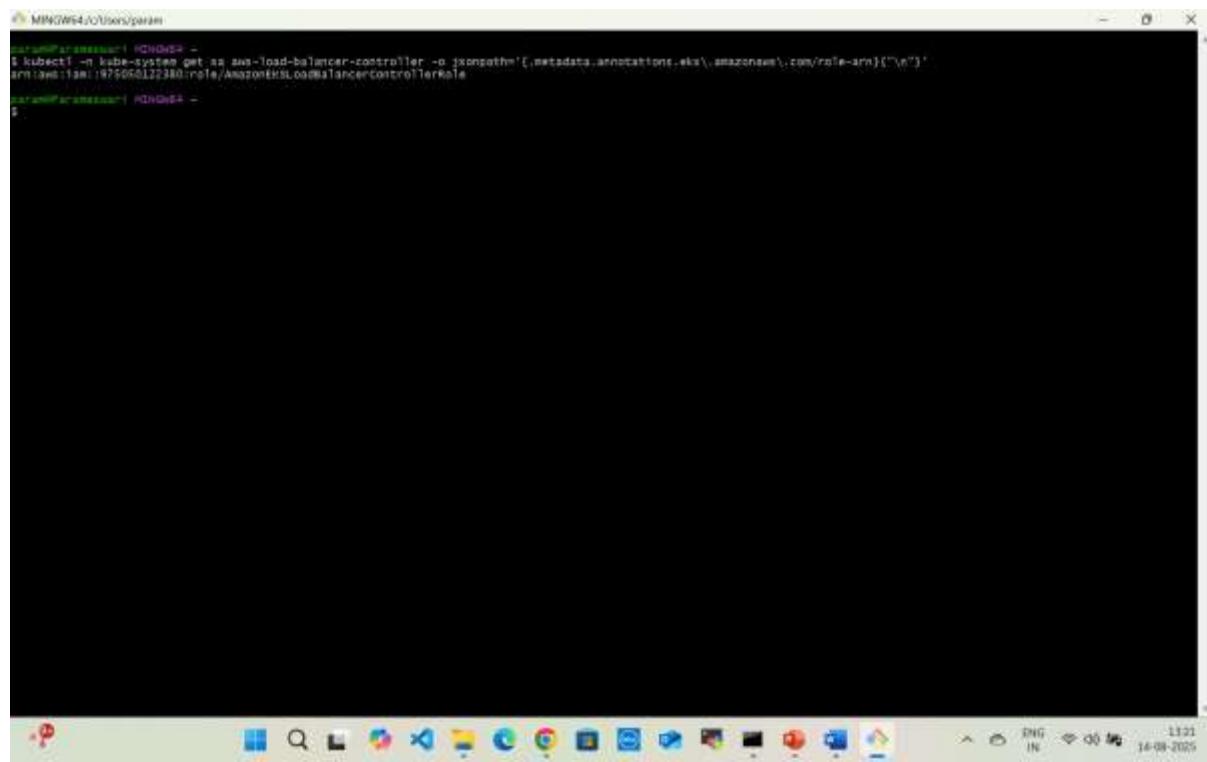
```
kubectl -n kube-system get sa aws-load-balancer-controller -o yaml
```

```
[Administrator: Command Prompt]
C:\Windows\System32>kubectl -n kube-system get sa aws-load-balancer-controller -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  annotations:
    kia.amazonaws.com/role-arm: Arn:iam:iam::975050122588:role/AmazonEKSLoadBalancerControllerRole
  creationTimestamp: "2025-08-14T06:48:13Z"
  labels:
    app.kubernetes.io/managed-by: eksctl
    name: aws-load-balancer-controller
  namespace: kube-system
  resourceVersion: "37256"
  uid: f91618fe-5e02-4d13-95e6-8223b7f0e049
C:\Windows\System32>
```

```
kubectl -n kube-system get sa aws-load-balancer-controller
```

```
kubectl -n kube-system get sa aws-load-balancer-controller -o  
jsonpath='{.metadata.annotations.eks\\.amazonaws\\.com/role-arn}{"\n"}'
```

open bash script



A screenshot of a Windows terminal window titled "MINGW64 / C:\Users\parikh". The window contains a single line of command output: "arn:aws:iam::175956122380:role/AmazonEKSLoadBalancerControllerRole". The terminal window has a black background and white text. The system tray at the bottom right shows the date as 16-09-2025.

```
arn:aws:iam::175956122380:role/AmazonEKSLoadBalancerControllerRole
```

C) Install the controller (Helm)

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

```
helm repo update
```

```
helm upgrade -i aws-load-balancer-controller eks/aws-load-balancer-controller -n  
kube-system --set clusterName=bankingcluster --set region=us-east-1 --set  
serviceAccount.create=false --set serviceAccount.name=aws-load-balancer-controller
```

Verify it's up:

```
kubectl -n kube-system get deploy aws-load-balancer-controller
```

```
kubectl -n kube-system get pods -l app.kubernetes.io/name=aws-load-balancer-  
controller -o wide
```

```
kubectl -n kube-system logs deploy/aws-load-balancer-controller | tail -n 100
```

D) Create the IngressClass (fixes your earlier error)

```
cat <<'YAML' | kubectl apply -f -  
apiVersion: networking.k8s.io/v1  
kind: IngressClass  
metadata:  
  name: alb  
spec:  
  controller: ingress.k8s.aws/alb
```

YAML

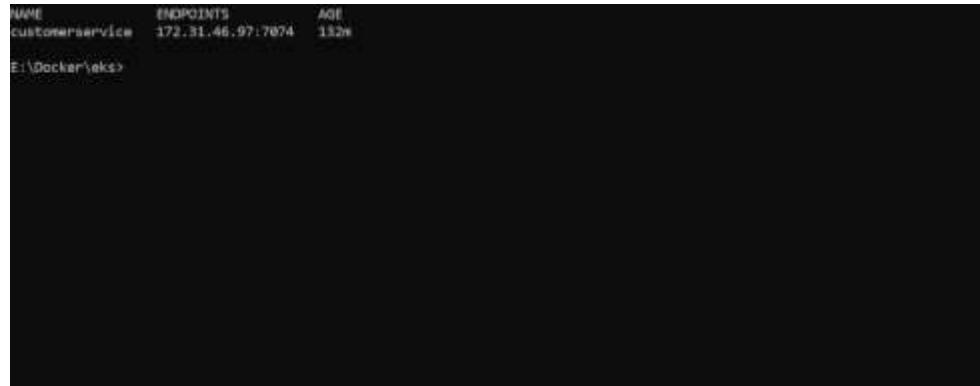
kubectl get ingressclass

Goto docker/eks

>kubectl apply -f ingress-customer.yaml

kubectl get ingress customerservice -w

kubectl get endpoints customerservice



A screenshot of a terminal window showing the output of a Kubernetes command. The command is 'kubectl get endpoints customerservice'. The output shows a single endpoint entry for the 'customerservice' service, with an IP address of 172.31.46.97 and a port of 7074. The service is labeled 'customerservice' and is 132h old. The terminal prompt is 'E:\docker\eks>'. The rest of the terminal window is blacked out.

Name	Endpoints	Age
customerservice	172.31.46.97:7074	132h

Do these 5 commands (safe to copy-paste)

```
# 1) Tell Kubernetes this Ingress should be handled by the ALB controller
kubectl patch ingress customerservice --type='json' \
-p='[{"op":"add","path":"/spec/ingressClassName","value":"alb"}]'

# (alternative to step 1 if you prefer annotations)
kubectl annotate ingress customerservice kubernetes.io/ingress.class=alb --overwrite

# 2) Make it public (or use "internal" for VPC-only)
kubectl annotate ingress customerservice alb.ingress.kubernetes.io/scheme=internet-
facing --overwrite

# 3) Choose the target type:
#   - If your Service is ClusterIP -> use ip
#   - If your Service is NodePort -> use instance
kubectl annotate ingress customerservice alb.ingress.kubernetes.io/target-type=ip --
overwrite

# If your Service is NodePort, instead run:
# kubectl annotate ingress customerservice alb.ingress.kubernetes.io/target-
type=instance --overwrite

# 4) (Recommended) set a correct healthcheck path for your app
kubectl annotate ingress customerservice alb.ingress.kubernetes.io/healthcheck-
path=/actuator/health --overwrite

# 5) Watch it reconcile
kubectl get ing customerservice -w

If things are wired correctly, ADDRESS will populate with an ALB DNS name within 1–3
minutes.
```

If ADDRESS is still empty, check these quickly

A) Ingress events (tells you the exact blocker)

```
kubectl describe ingress customerservice | sed -n '/Events/, $p'
```

B) Controller health & logs

```
kubectl -n kube-system get deploy aws-load-balancer-controller
```

```
kubectl -n kube-system logs deploy/aws-load-balancer-controller | egrep -i 'error|denied|forbid|subnet|iam|quota'
```

C) Service & endpoints exist

```
kubectl get svc customerservice -o wide
```

```
kubectl get endpoints customerservice
```

- Using target-type: ip? Service should be **ClusterIP**.
- Using target-type: instance? Service can be **NodePort**.

D) Subnet tags (most common cause)

- On all EKS subnets: kubernetes.io/cluster/<cluster-name>=shared (or owned)
- Public ALB: kubernetes.io/role/elb=1 on public subnets
- Internal ALB: kubernetes.io/role/internal-elb=1 on private subnets

Minimal known-good Ingress (HTTP, public, IP targets)

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
  name: customerservice
```

```
  annotations:
```

```
    kubernetes.io/ingress.class: alb
```

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

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

```
    alb.ingress.kubernetes.io/healthcheck-path: /actuator/health # change if needed
```

```
spec:
```

```
rules:  
- http:  
  paths:  
  - path: /  
    pathType: Prefix  
  backend:  
    service:  
      name: customerservice  
    port:  
      number: 7074
```

Apply it:

```
# If you use target-type: ip, ensure Service is ClusterIP  
kubectl patch svc customerservice -p '{"spec":{"type":"ClusterIP"}}'
```

```
kubectl apply -f ingress-customerservice.yaml
```

```
kubectl get ing customerservice -w
```