

## **KUBERNETES:**

### **PREREQUISITES:**

- a) CREATE VPC
- b) ATTACH INTERNET GATEWAY TO THAT VPC
- c) CREATE 2 PUBLIC SUBNETS
- d) CREATE ROUTE TABLE
- e) IN ROUTES ATTACH INTERNET GATEWAY
- f) IN SUBNET ASSOCIATION, SELECT PUBLIC SUBNET AND CLICK ON SAVE.

### **GO TO IAM IN AWS CONSOLE :**

#### **CREATE ROLE FOR CLUSTER & NODE GROUP**

- 1) **ROLE FOR EKS CLUSTER:** use case type eks & select eks cluster , save.

[AmazonEKSClusterPolicy](#)

#### **ROLE FOR NODE-GROUP: use case, select ec2 & click on next**

- 2) [AmazonEC2ContainerRegistryReadOnly](#)

- 3) [AmazonEKS\\_CNI\\_Policy](#)

- 4) [AmazonEKSWorkerNodePolicy](#)

- 5) [AmazonEBSCSIDriverPolicy](#)

- 6) [AmazonEFSCSIDriverPolicy](#)

- 7) CREATE KEY PAIR

- 8) CREATE SECURITY GROUP: ALLOW ALL TRAFFIC

- 9) **AWS-CLI LINK:**

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

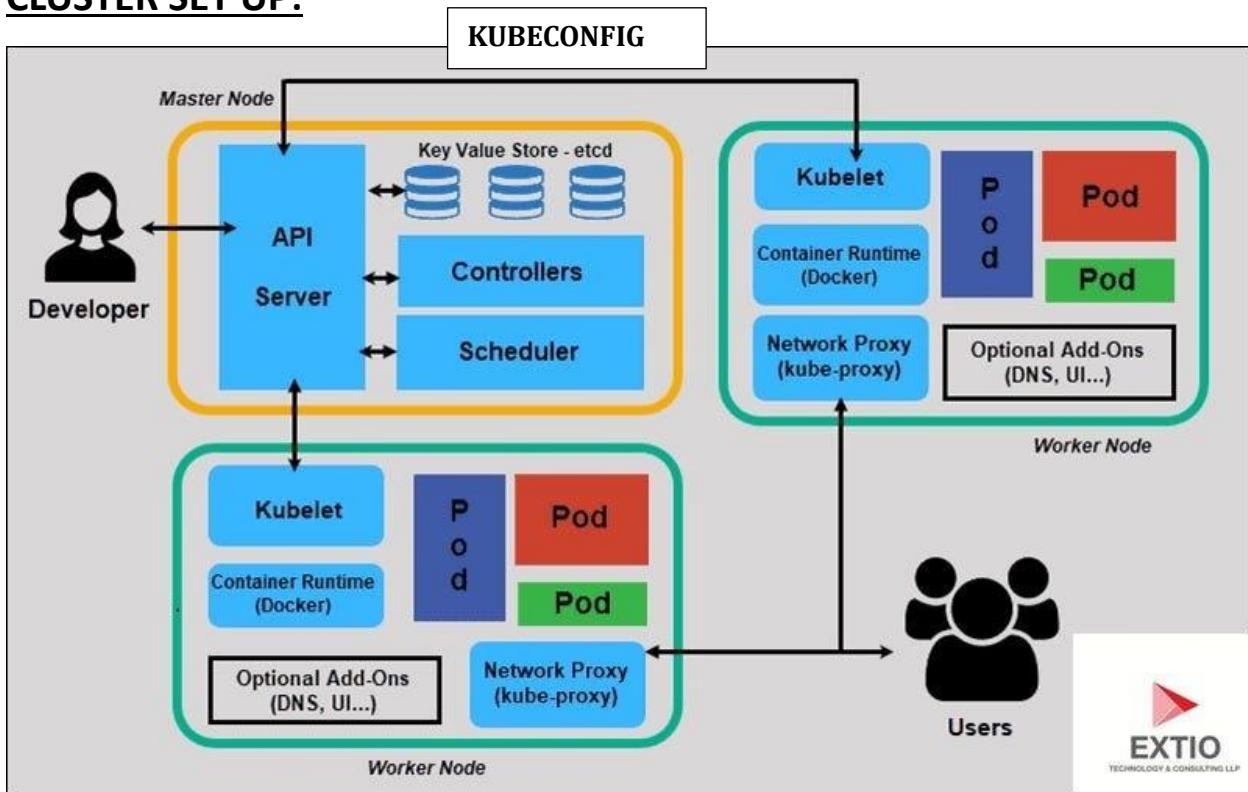
- 10) AFTER DOWNLOADING AWS CI , CONFIGURE IN COMMAND PROMPT USING **aws configure** COMMAND

- 11) **KUBECTL LINK** :RUN BELOW COMMAND IN COMMAND PROMPT

curl.exe -O <https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.0/2024-01-04/bin/windows/amd64/kubectl.exe>

- 12) SET PATH IN ENVIRONMENT VARIABLE

## CLUSTER SET UP:



## CREATE ROLE FOR CLUSTER IN IAM

The screenshot shows the AWS IAM Roles page with 44 roles listed. The roles include various AWS services like EC2, Lambda, CodePipeline, EKS, and AutoScaling. Below this, the 'Create role' wizard is displayed, specifically Step 1: Select trusted entity. It shows five options: AWS service (selected), AWS account, Web identity, SAML 2.0 federation, and Custom trust policy. A note indicates that the selected use case allows actions from EC2, Lambda, or others.

Role name	Trusted entities	Last activity
a-access	AWS Service: ec2	4 days ago
Admin-access-ec2	AWS Service: ec2	21 days ago
AWSCodePipelineServiceRole-us-east-1-1-click-pipeline	AWS Service: codepipeline	6 days ago
AWSServiceRoleForAmazonEKS	AWS Service: eks (Service-Linked Role)	2 hours ago
AWSServiceRoleForAmazonEKSNodegroup	AWS Service: eks-nodegroup (Service-Linked Role)	1 hour ago
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked Role)	18 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	1 hour ago
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked Role)	Yesterday
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	2 hours ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	-

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Service or use case**  
EKS

**Choose a use case for the specified service.**

**Use case**

- EKS Allows EKS to manage clusters on your behalf.
- EKS - Cluster Allows access to other AWS service resources that are required to operate clusters managed by EKS.
- EKS - Nodegroup Allows EKS to manage nodegroups on your behalf.
- EKS - Fargate pod Allows access to other AWS service resources that are required to run Amazon EKS pods on AWS Fargate.
- EKS - Fargate profile Allows EKS to run Fargate tasks.
- EKS - Connector Allows access to other AWS service resources that are required to connect to external clusters.
- EKS Local - Outpost Allows Amazon EKS Local to call AWS services on your behalf.
- EKS - Pod Identity Allows pods running in Amazon EKS cluster to access AWS resources.

**Add permissions**

**Permissions policies (1)** Info  
The type of role that you selected requires the following policy.

Policy name	Type
AmazonEKSClusterPolicy	AWS managed

**Set permissions boundary - optional**

Cancel Previous Next

**Step 1: Select trusted entities**

**Trust policy**

```

1 = [{}]
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": "*",
7       "Service": [
8         "eks.amazonaws.com"
9       ],
10      }
11    ],
12    "Action": "sts:AssumeRole"
13  ]
14 ]

```

**Step 2: Add permissions**

**Permissions policy summary**

Policy name	Type	Attached as
AmazonEKSClusterPolicy	AWS managed	Permissions policy

**Step 3: Add tags**

**Add tags - optional** Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

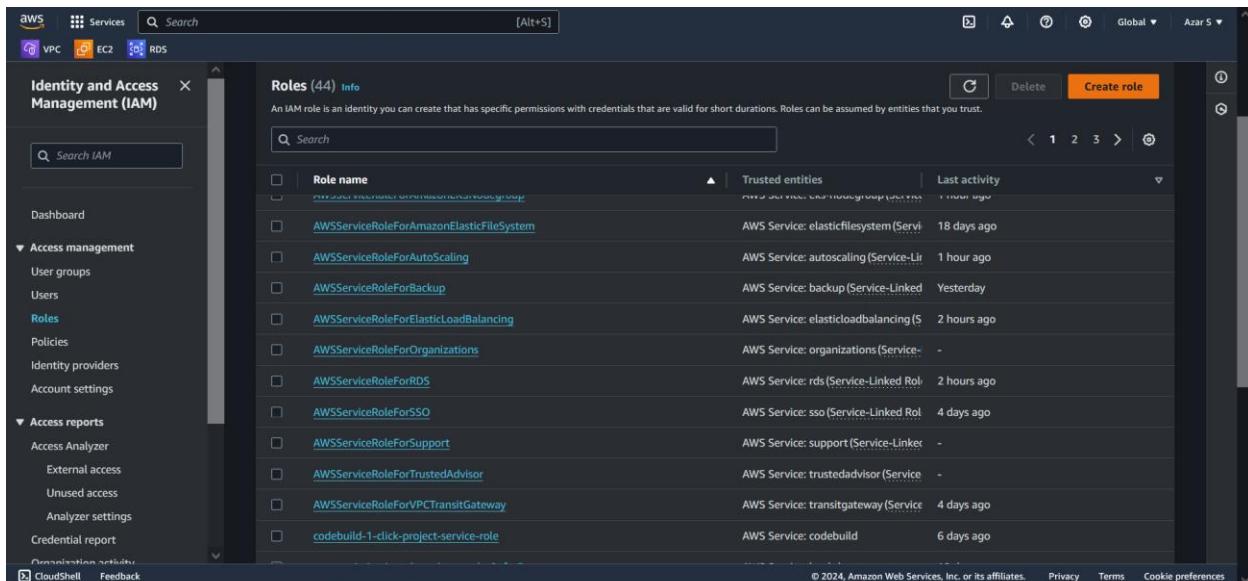
No tags associated with the resource.

**Add new tag**

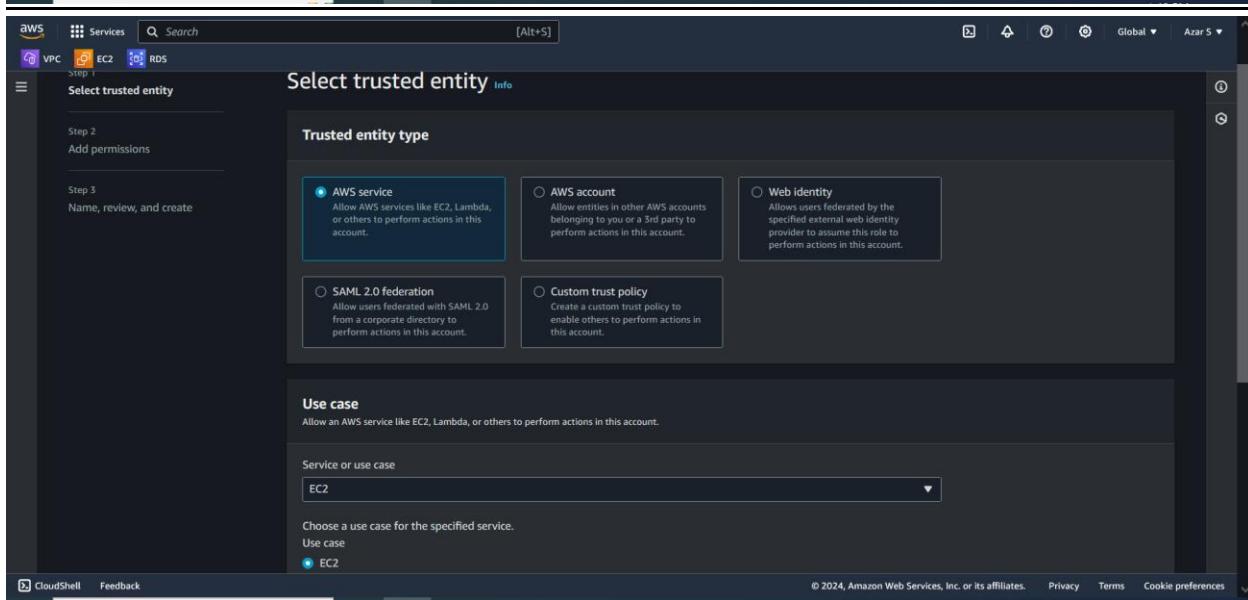
You can add up to 50 more tags.

**Create role**

## CREATE ROLE FOR NODE-GROUP IN IAM



The screenshot shows the AWS IAM Roles page with a list of 44 roles. The columns are Role name, Trusted entities, and Last activity. Some visible role names include `codebuild-1-click-project-service-role`, `AWSServiceRoleForAmazonElasticFileSystem`, `AWSServiceRoleForAutoScaling`, `AWSServiceRoleForBackup`, `AWSServiceRoleForElasticLoadBalancing`, `AWSServiceRoleForOrganizations`, `AWSServiceRoleForRDS`, `AWSServiceRoleForSSO`, `AWSServiceRoleForSupport`, `AWSServiceRoleForTrustedAdvisor`, `AWSServiceRoleForVPCTransitGateway`, and `codebuild-1-click-project-service-role`. The last activity column shows various times from yesterday to 6 days ago.

The screenshot shows the 'Select trusted entity' step of the IAM Role creation wizard. It has three tabs: Step 1 (selected), Step 2 (Add permissions), and Step 3 (Name, review, and create). Under Step 1, the 'Trusted entity type' section shows four options: AWS service (selected), AWS account, SAML 2.0 federation, and Custom trust policy. Below this, the 'Use case' section shows 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' A dropdown for 'Service or use case' is set to 'EC2'. At the bottom, it says 'Choose a use case for the specified service.' and 'Use case' with 'EC2' selected.

**Screenshot 1: Permissions policies search for 'NODE'**

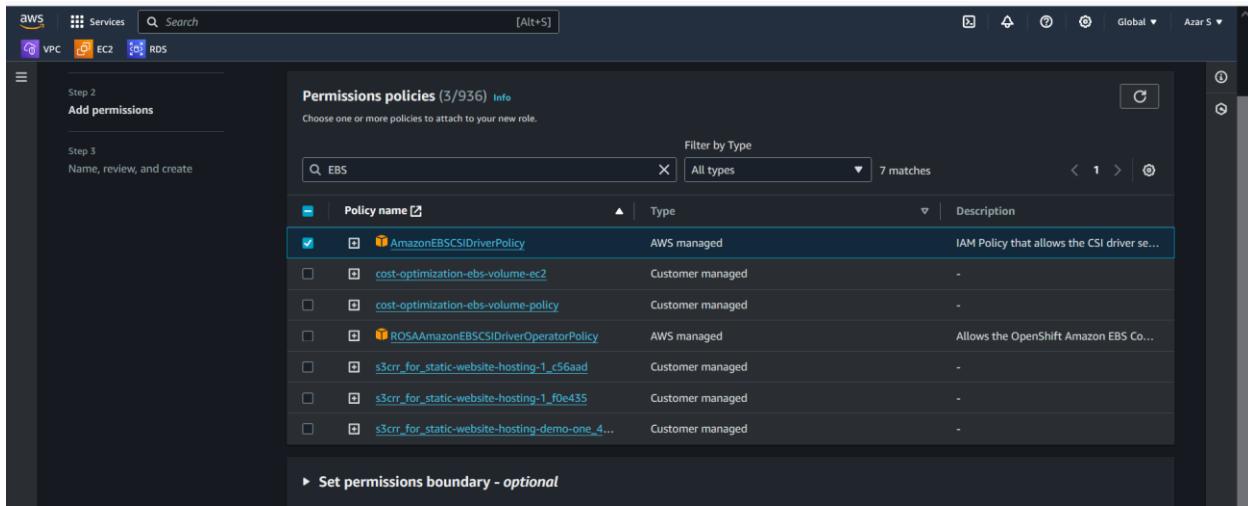
Policy name	Type	Description
AmazonEKSWorkerNodePolicy	AWS managed	This policy allows Amazon EKS worker no...
EKS-NODE-GROUP-POLICY	Customer managed	-
ROSA_NodePoolManagementPolicy	AWS managed	Allows Red Hat OpenShift Service on AWS...

**Screenshot 2: Permissions policies search for 'CNI'**

Policy name	Type	Description
AmazonEKS_CNI_Policy	AWS managed	This policy provides the Amazon VPC CNI ...

**Screenshot 3: Permissions policies search for 'CONTAINER'**

Policy name	Type	Description
AmazonEC2ContainerRegistryFullAccess	AWS managed	Provides administrative access to Ama...
AmazonEC2ContainerRegistryPowerUser	AWS managed	Provides full access to Amazon EC2 Co...
AmazonEC2ContainerRegistryReadOnly	AWS managed	Provides read-only access to Amazon E...
AmazonEC2ContainerServiceAutoscaleRole	AWS managed	Policy to enable Task AutoScaling for A...
AmazonEC2ContainerServiceEventsRole	AWS managed	Policy to enable CloudWatch Events fo...
AmazonEC2ContainerServiceforEC2Role	AWS managed	Default policy for the Amazon EC2 Rol...
AmazonEC2ContainerServiceRole	AWS managed	Default policy for Amazon ECS service ...
AmazonElasticContainerRegistryPublicFullA...	AWS managed	Provides administrative access to Ama...
AmazonElasticContainerRegistryPublicPow...	AWS managed	Provides full access to Amazon ECR Pu...
AmazonElasticContainerRegistryPublicRead...	AWS managed	Provides read-only access to Amazon E...
AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicon...
EC2InstanceProfileForImageBuilderECRCont...	AWS managed	EC2 Instance profile for building conta...

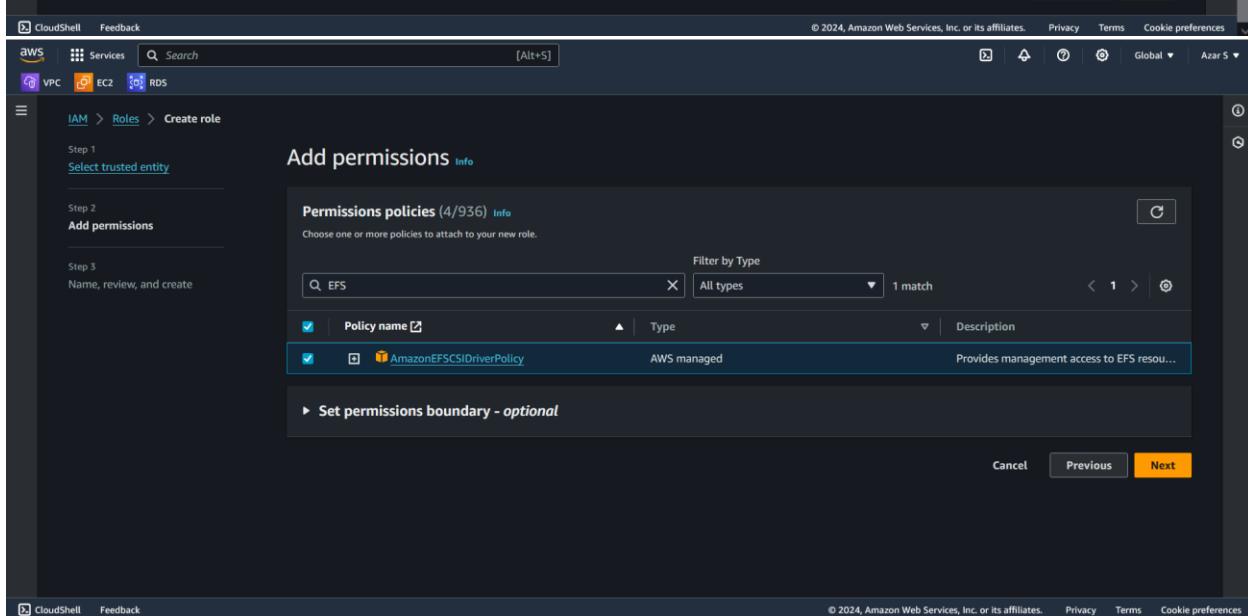


The screenshot shows the 'Add permissions' step of creating a new IAM role. The search bar at the top is set to 'EBS'. A table lists several policies, with 'AmazonEBSCSIDriverPolicy' selected. Below the table is a note about setting a permissions boundary.

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonEBSCSIDriverPolicy	AWS managed	IAM Policy that allows the CSI driver se...
<input type="checkbox"/> cost-optimization-ebs-volume-ec2	Customer managed	-
<input type="checkbox"/> cost-optimization-ebs-volume-policy	Customer managed	-
<input type="checkbox"/> ROSAAmazonEBSCSIDriverOperatorPolicy	AWS managed	Allows the OpenShift Amazon EBS Co...
<input type="checkbox"/> s3crr_for_static-website-hosting-1_c56aad	Customer managed	-
<input type="checkbox"/> s3crr_for_static-website-hosting-1_f0e435	Customer managed	-
<input type="checkbox"/> s3crr_for_static-website-hosting-demo-one_4...	Customer managed	-

**Set permissions boundary - optional**

Cancel Previous Next

The screenshot shows the 'Add permissions' step of creating a new IAM role. The search bar at the top is set to 'EFS'. A table lists several policies, with 'AmazonEFSCSIDriverPolicy' selected. Below the table is a note about setting a permissions boundary.

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonEFSCSIDriverPolicy	AWS managed	Provides management access to EFS resou...

**Set permissions boundary - optional**

Cancel Previous Next

**Step 2: Add permissions**

**Role details**

**Role name:** EKS-NODE-ROLE

**Description:** Allows EC2 instances to call AWS services on your behalf.

**Step 1: Select trusted entities**

**Trust policy:**

```

1: [
2:   {
3:     "Version": "2012-10-17",
4:     "Statement": [
5:       {
6:         "Effect": "Allow",
7:         "Action": [
8:           "sts:AssumeRole"
9:         ],
10:        "Principal": [

```

**Step 3: Add tags**

**Add tags - optional** Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

**Add new tag**

You can add up to 50 more tags.

**Create role**

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is collapsed, and the main area displays the 'Permissions' tab of the 'Permissions policies' section. There are five managed policies listed:

Policy name	Type	Attached entities
AmazonEBSCSI Driver Policy	AWS managed	1
AmazonEC2 Container Registry Read Only	AWS managed	9
AmazonEFSCSI Driver Policy	AWS managed	1
AmazonEKS_CNI Policy	AWS managed	8
AmazonEKS Worker Node Policy	AWS managed	9

At the bottom of the page, there is a footer with links to 'Privacy', 'Terms', and 'Cookie preferences'.

## CREATE CLUSTER IN EKS SERVICE:

**Cluster configuration** Info

**Name**  
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.

ⓘ Kubernetes version 1.29 reaches the end of standard support on March 23, 2025. If you don't update your cluster to a later version before that date, it will automatically enter extended support. After the extended support period ends, the cluster will no longer receive security patches or updates.

**Cluster access** Info

Control how IAM principals can access this cluster.

**Bootstrap cluster administrator access** Info  
Choose whether the IAM principal creating the cluster has Kubernetes cluster administrator access.

**Allow cluster administrator access**  
Allow cluster administrator access for your IAM principal.

**Disallow cluster administrator access**  
Disallow cluster administrator access for your IAM principal.

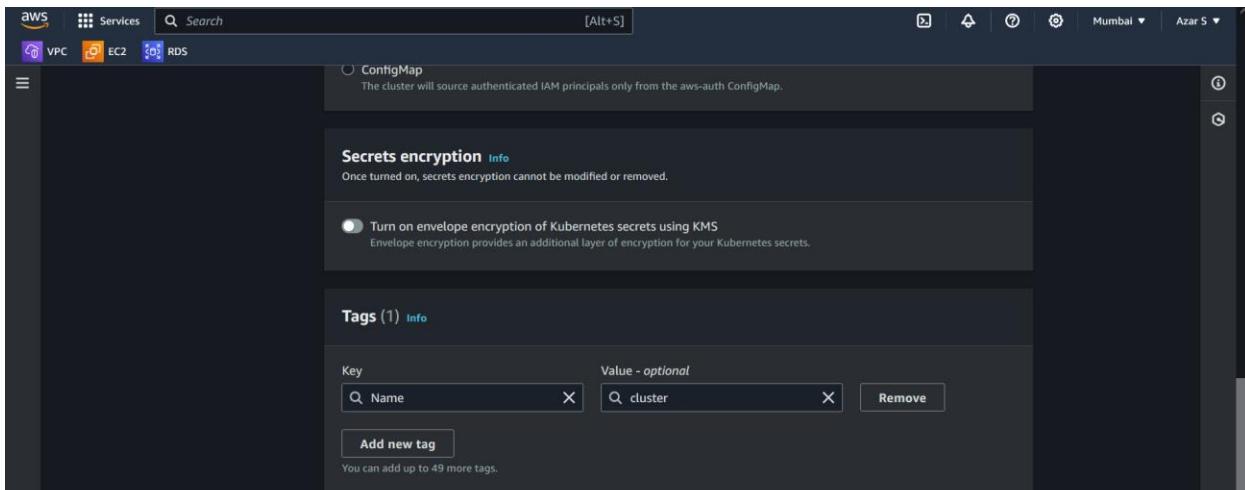
**Cluster authentication mode** Info  
Configure which source the cluster will use for authenticated IAM principals.

**EKS API**  
The cluster will source authenticated IAM principals only from EKS access entry APIs.

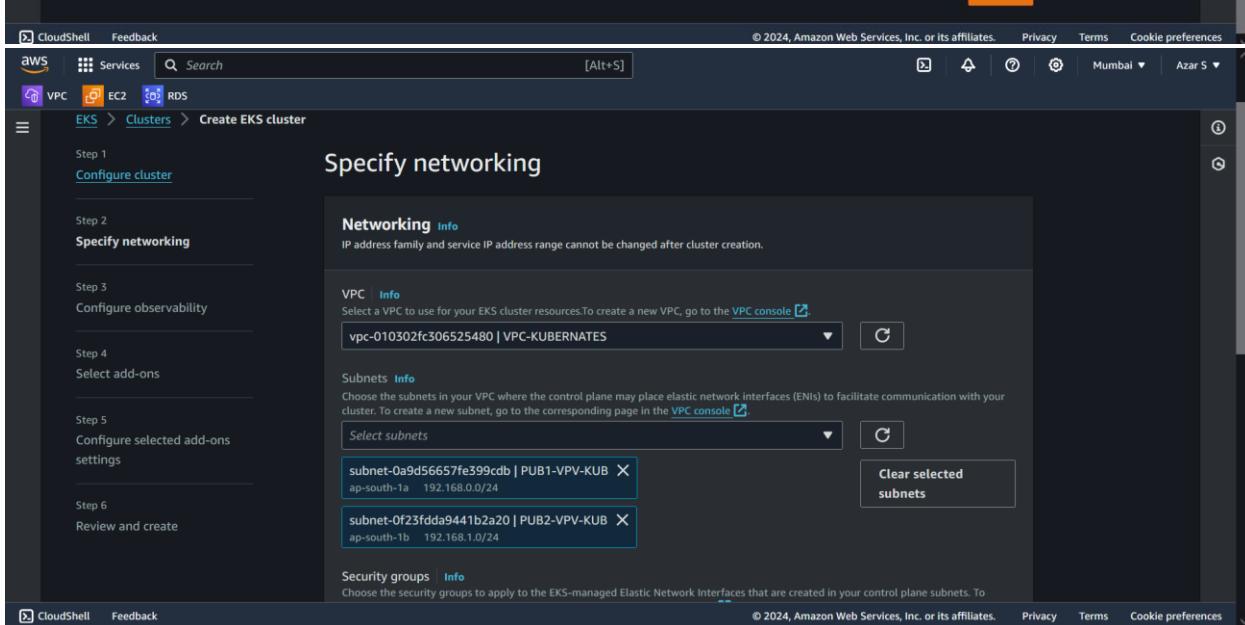
**EKS API and ConfigMap**  
The cluster will source authenticated IAM principals from both EKS access entry APIs and the aws-auth ConfigMap.

**ConfigMap**  
The cluster will source authenticated IAM principals only from the aws-auth ConfigMap.

**Secrets encryption** Info  
Once turned on, secrets encryption cannot be modified or removed.



The screenshot shows the AWS EKS Cluster Configuration page. It includes sections for 'ConfigMap' (describing IAM principals from aws-auth ConfigMap), 'Secrets encryption' (with an option to turn on envelope encryption using KMS), and 'Tags (1)'. A single tag named 'cluster' is listed with a value of 'cluster'. Buttons for 'Cancel' and 'Next' are at the bottom.

The screenshot shows the 'Specify networking' step in the EKS cluster creation wizard. It includes a 'Networking' section (info: IP address family and service IP address range cannot be changed after cluster creation), a 'VPC' section (info: select a VPC for resources, with 'vpc-010302fc306525480 | VPC-KUBERNATES' selected), and a 'Subnets' section (info: choose subnets for control plane ENIs). Two subnets are selected: 'subnet-0a9d56657fe399cddb | PUB1-VPV-KUB' (ap-south-1a, 192.168.0.0/24) and 'subnet-0f23fdada9441b2a20 | PUB2-VPV-KUB' (ap-south-1b, 192.168.1.0/24). A 'Clear selected subnets' button is also present. The sidebar on the left lists steps 1 through 6: Configure cluster, Specify networking, Configure observability, Select add-ons, Configure selected add-ons settings, and Review and create.

The screenshot shows the 'Edit inbound rules' page for a security group named 'sg-0630ab87d305ba464 - EKS-CLUSTER-SECURITY-GROUP'. It lists two rules:

- sgr-00b88ba0352e56f39**: Type 'All traffic', Protocol 'All', Port range 'All', Source 'Cust...', Description '0.0.0.0/0'.
- sgr-0bfae0f1f3b641367**: Type 'SSH', Protocol 'TCP', Port range '22', Source 'Cust...', Description '0.0.0.0/0'.

An 'Add rule' button is visible at the bottom left.

The screenshot shows the 'Review and create' step for an EKS cluster endpoint. It includes the following configuration:

- Security groups**: sg-0630ab87d305ba464 | EKS-CLUSTER-SECURITY-GROUP
- Choose cluster IP address family**: IPv4 selected.
- Configure Kubernetes service IP address range**: IP range 192.168.1.0/24.
- Cluster endpoint access**: Public selected.

The screenshot shows two consecutive steps of the AWS EKS Cluster Configuration wizard.

**Step 1: Choose cluster IP address family**

Configure the IP address type for pods and services in your cluster.

- IPv4
- IPv6

Configure Kubernetes service IP address range [Info](#)  
Specify the range from which cluster services will receive IP addresses.

**Step 2: Cluster endpoint access**

Configure access to the Kubernetes API server endpoint.

- Public  
The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.
- Public and private  
The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.
- Private  
The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

[Advanced settings](#)

**Step 3: Control plane logging**

Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

- API server  
Logs pertaining to API requests to the cluster.
- Audit  
Logs pertaining to cluster access via the Kubernetes API.
- Authenticator  
Logs pertaining to authentication requests into the cluster.
- Controller manager  
Logs pertaining to state of cluster controllers.
- Scheduler  
Logs pertaining to scheduling decisions.

Cancel Previous Next

The screenshot shows the AWS EKS Add-ons configuration interface. It displays two tabs: Step 4 (Select add-ons) and Step 5 (Configure selected add-ons settings). Under Step 4, three add-ons are listed: Amazon VPC CNI, kube-proxy, and CoreDNS. All three are marked as "Installed by default". Under Step 5, two add-ons are listed: Amazon GuardDuty EKS Runtime Monitoring and Amazon EKS Pod Identity Agent. Both are marked as "Ready to install". The interface includes navigation buttons for Cancel, Previous, and Next.

**Step 4: Select add-ons**

- Amazon VPC CNI Info  
Enable pod networking within your cluster.  
Category: networking  
Status: Installed by default
- kube-proxy Info  
Enable service networking within your cluster.  
Category: networking  
Status: Installed by default
- CoreDNS Info  
Enable service discovery within your cluster.  
Category: networking  
Status: Installed by default

**Step 5: Configure selected add-ons settings**

- Amazon GuardDuty EKS Runtime Monitoring Info  
Install EKS Runtime Monitoring add-on within your cluster. Ensure to enable EKS Runtime Monitoring within Amazon GuardDuty.  
Category: security
- Amazon EKS Pod Identity Agent Info  
Install EKS Pod Identity Agent to use EKS Pod Identity to grant AWS IAM permissions to pods through Kubernetes service accounts.  
Category: security

Cancel Previous Next

**Step 5: Configure selected add-ons settings**

- CoreDNS Info  
Category: networking Status: Installed by default
- Amazon EKS Pod Identity Agent Info  
Remove add-on  
Category: security Status: Ready to install

Cancel Previous Next

Step 5: Versions

Selected add-ons version	
Add-on name vpc-cni	Version v1.16.0-eksbuild.1
Add-on name kube-proxy	Version v1.29.0-eksbuild.1
Add-on name coredns	Version v1.11.1-eksbuild.4
Add-on name eks-pod-identity-agent	Version v1.2.0-eksbuild.1

Cancel Previous Create

## INSTALL AWS CLI ON WINDOWS:

LINK: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

**Install and update requirements**

- We support the AWS CLI on Microsoft-supported versions of 64-bit Windows.
- Admin rights to install software

**Install or update the AWS CLI**

To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the [AWS CLI version 2 Changelog](#) on GitHub.

- Download and run the AWS CLI MSI installer for Windows (64-bit):  
<https://awscli.amazonaws.com/AWSCLIV2.msi>

Alternatively, you can run the `msiexec` command to run the MSI installer.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

## INSTALL KUBECTL ON WINDOWS:

LINK: <https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

The screenshot shows a web browser displaying the AWS Documentation for Amazon EKS. The URL is https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html. The page is titled "Install or update kubectl on macOS, Linux, and Windows operating systems." The "Windows" tab is selected. The content includes instructions for opening a PowerShell terminal and downloading the kubectl binary from Amazon S3. It lists two Kubernetes versions: 1.29 and 1.28. Below each version is a "curl.exe" command. On the right side of the page, there are social sharing icons for LinkedIn, GitHub, and a hexagonal icon.

### RUN COMMAND IN COMMAND PROMPT

Screenshot showing the Windows File Explorer interface with a file named 'kubectl' selected. Below it is a terminal window showing command-line output for AWS CLI and kubectl.

```

C:\Windows>aws --version
aws-cli/2.15.40 Python/3.11.8 Windows/10 exe/AMD64 prompt/off

C:\Tools>curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.0/2024-01-04/bin/windows/amd64/kubectl.exe
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Spent  Left Speed
  33 48.6M  33 16.5M  0     0  894k      0 0:00:55  48.6M  40 19.7M  0     0 1014k      53 48.6M
100 48.6M 100 48.6M  0     0 1693k      0 0:00:29  0:00:29 --:--:-- 2898k114k

C:\Tools>kubectl version
'kubectl' is not recognized as an internal or external command,
operable program or batch file.

C:\Tools>kubectl version
Client Version: v1.29.0-eks-5e0fdde
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Unable to connect to the server: dial tcp: lookup FFA91C65B9034FA261C57550ADB10904.ap-south-1.eks.amazonaws.com: no such host

C:\Tools>
  
```

Below the terminal is a screenshot of the AWS Management Console showing the Amazon Elastic Kubernetes Service (EKS) Clusters page. It displays one cluster named 'cluster1'.

Cluster name	Status	Kubernetes version	Support period	Provider
cluster1	Active	1.29	Standard support until March 23, 2025	EKS

The screenshot shows the AWS EKS Compute tab. On the left sidebar, there are sections for Clusters (New), Related services (Amazon ECR, AWS Batch), Documentation, and Submit feedback. The main content area has tabs for Overview, Resources, Compute (selected), Networking, Add-ons, Access, Observability, and Upgrade insights. Under Compute, there are two sections: Nodes (0) and Node groups (0). Both sections show a table header with columns like Node name, Instance type, Node group, Created, and Status. Below the headers, it says "No Nodes" and "This cluster does not have any Nodes, or you don't have permission to view them." and "No node groups". There are "Edit", "Delete", and "Add node group" buttons.

The screenshot shows the "Configure node group" step of a wizard. The left sidebar shows steps 1 through 4: Step 1 (Configure node group), Step 2 (Set compute and scaling configuration), Step 3 (Specify networking), and Step 4 (Review and create). The main content area is titled "Configure node group". It includes a "Node group configuration" section with a note that properties cannot be changed after creation. A "Name" field is set to "WEB". A "Node IAM role" dropdown is set to "EKS-NODE-ROLE". A warning message states: "The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.".

The screenshot shows the AWS CloudFormation console interface. A modal window is open for creating a new AWS Lambda function. The 'Lambda function' tab is selected. In the 'Code' section, there is a large blue button labeled 'Upload a file'. Below it, there are sections for 'Handler' (selected as 'lambda.lambda\_handler'), 'Runtime' (selected as 'Node.js 14.x'), and 'Role' (selected as 'Lambda execution role'). The 'Configure triggers' section is collapsed. At the bottom of the modal, there are 'Next Step' and 'Cancel' buttons.

**Node group compute configuration**

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

**AMI type:** [Info](#) Select the EKS-optimized Amazon Machine Image for nodes. **Amazon Linux 2 (AL2\_x86\_64)**

**Capacity type:** [Info](#) Select the capacity purchase option for this node group. **On-Demand**

**Instance types:** [Info](#) Select instance types you prefer for this node group. **t3.medium**

vCPU: 2 vCPUs Memory: 4 GiB Network: Up to 5 Gigabit Max ENI: 5 Max IPs: 18

**Disk size:** Select the size of the attached EBS volume for each node. **20 GiB**

**CloudShell Feedback** © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Node group scaling configuration**

**Desired size:** Set the desired number of nodes that the group should launch with initially. **2 nodes**  
Desired node size must be greater than or equal to 0

**Minimum size:** Set the minimum number of nodes that the group can scale in to. **2 nodes**  
Minimum node size must be greater than or equal to 0

**Maximum size:** Set the maximum number of nodes that the group can scale out to. **2 nodes**  
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size

**Node group update configuration:** [Info](#)

**CloudShell Feedback** © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows two sequential configuration steps for an EKS node group.

**Node group update configuration:**

- Maximum size:** Set to 2 nodes.
- Maximum unavailable:** Set to 1 node.
- Value:** Set to 1 node.

**Specify networking:**

- Node group network configuration:** Subnets selected: subnet-0a9d56657fe399cdb | PUB1-VPV-KUB and subnet-0f23fdda9441b2a20 | PUB2-VPV-KUB.
- Configure remote access to nodes:** Option is available but not selected.

Both steps include "Cancel", "Previous", and "Next" buttons at the bottom.

The screenshot shows the 'Node group update configuration' screen in the AWS EKS console. It is on 'Step 3: Networking'. The 'Node group network configuration' section shows two subnets: 'subnet-0a9d56657fe399cdb' and 'subnet-0f23fdde9441b2a20'. The 'Configure remote access to nodes' option is set to 'off'. Buttons at the bottom include 'Cancel', 'Previous', and a prominent orange 'Create' button.

The screenshot shows the 'Node group configuration' screen in the AWS EKS console. It is on 'Step 2'. The 'Name' field is filled with 'DB'. The 'Node IAM role' dropdown is set to 'EKS-NODE-ROLE'. A note states: 'The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.' Buttons at the bottom include 'Cancel', 'Previous', and a prominent orange 'Create' button.

**Launch template**

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

Use launch template  
Configure this node group using an EC2 launch template.

**Kubernetes labels**

Key	Value
application	database

**Add label**  
Remaining labels available to add: 49

**Kubernetes taints**

This node group does not have any taints.

**Add taint**  
Remaining taints available to add: 50

**Tags**

Key	Value - optional
Name	DB-NODES

**Add new tag**  
You can add up to 49 more tags.

**Node group compute configuration**

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

**AMI type** Info  
Select the EKS-optimized Amazon Machine Image for nodes.  
Amazon Linux 2 (AL2\_x86\_64)

**Capacity type**  
Select the capacity purchase option for this node group.  
On-Demand

**Instance types** Info  
Select instance types you prefer for this node group.  
 Enter an instance type  
t3.medium  
vCPU: 2 vCPUs | Memory: 4 GiB | Network: Up to 5 Gigabit | Max ENI: 3 | Max IPs: 18

**Disk size**  
Select the size of the attached EBS volume for each node.  
20 GiB

The screenshots show the configuration steps for a Node Group:

- Node group scaling configuration:** Set Desired size, Minimum size, and Maximum size all to 2 nodes.
- Node group update configuration:** Set Maximum size to 2 nodes. Under Maximum unavailable, choose Number (1 node).
- Specify networking:** Select subnets: subnet-0a9d56657fe399cdb | PUB1-VPV-KUB and subnet-0f23fd9a9441b2a20 | PUB2-VPV-KUB. Configure remote access to nodes.

The screenshot shows two related pages from the AWS CloudShell interface.

**Node group update configuration:**

- Step 3: Networking
- Maximum unavailable: 1 node
- Subnets: subnet-0a9d56657fe399cdb, subnet-0f23fdda9441b2a20
- Configure remote access to nodes: off

**Amazon Elastic Kubernetes Service - Nodes (4) Info:**

Node name	Instance type	Node group	Created	Status
ip-192-168-0-183.ap-south-1.compute.internal	t3.medium	WEB	Created 4 minutes ago	Ready
ip-192-168-0-69.ap-south-1.compute.internal	t3.medium	DB	Created a minute ago	Ready
ip-192-168-1-31.ap-south-1.compute.internal	t3.medium	WEB	Created 4 minutes ago	Ready
ip-192-168-1-79.ap-south-1.compute.internal	t3.medium	DB	Created a minute ago	Ready

The screenshot displays two separate AWS Cloud Console sessions.

**Top Session (Amazon Elastic Kubernetes Service):**

- Left Sidebar:** Shows 'Clusters' (New), 'Amazon EKS Anywhere' (New), 'Related services' (Amazon ECR, AWS Batch), 'Documentation' (New), and 'Submit feedback'.
- Table View:** Lists four node groups:
 

Node name	Instance type	Node group	Created	Status
ip-192-168-0-183.ap-south-1.compute.internal	t3.medium	WEB	Created 4 minutes ago	Ready
ip-192-168-0-69.ap-south-1.compute.internal	t3.medium	DB	Created a minute ago	Ready
ip-192-168-1-31.ap-south-1.compute.internal	t3.medium	WEB	Created 4 minutes ago	Ready
ip-192-168-1-79.ap-south-1.compute.internal	t3.medium	DB	Created a minute ago	Ready
- Bottom Section:** Shows 'Node groups (2) Info' with two entries:
 

Group name	Desired size	AMI release version	Launch template	Status
DB	2	1.29.0-20240415	-	Active
WEB	2	1.29.0-20240415	-	Active

**Bottom Session (EC2 Instances):**

- Left Sidebar:** Shows 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations' (New), 'Images', 'Elastic Block Store' (Volumes, Snapshots, Lifecycle Manager), 'Network & Security' (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and 'Load Balancing'.
- Table View:** Shows a list of running EC2 instances:
 

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
	i-0f86f59f1f2005ebf	Running	t3.medium	2/2 checks passed	<a href="#">View alarms +</a>
	i-0074c2f5c18424d84	Running	t3.medium	2/2 checks passed	<a href="#">View alarms +</a>
	i-0be9f0cca53a72819	Running	t3.medium	2/2 checks passed	<a href="#">View alarms +</a>
	i-03609a81a7e7d174a	Running	t3.medium	2/2 checks passed	<a href="#">View alarms +</a>
- Bottom Section:** Shows a modal dialog titled 'Select an instance'.

### TO UPDATE THE KUBE CONFIG FILE:

```
aws eks update-kubeconfig --region ap-south-1 --name cluster1
```

### TO SEE THE CLUSTER INFORMATION:

```
kubectl cluster-info
```

```
C:\Tools>aws eks update-kubeconfig --region ap-south-1 --name cluster1
Added new context arn:aws:eks:ap-south-1:709398145454:cluster/cluster1 to C:\Users\DELL\.kube\config

C:\Tools>kubectl cluster info
error: unknown command "cluster" for "kubectl"

Did you mean this?
  cluster-info

C:\Tools>kubectl cluster-info
Kubernetes control plane is running at https://12B5C0285E81EE2BF5C403AFD454206F.gr7.ap-south-1.eks.amazonaws.com
CoreDNS is running at https://12B5C0285E81EE2BF5C403AFD454206F.gr7.ap-south-1.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

C:\Tools>
```

```
C:\Tools>kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-0-183.ap-south-1.compute.internal  Ready   <none>  20m   v1.29.0-eks-5e0fdde
ip-192-168-0-69.ap-south-1.compute.internal  Ready   <none>  17m   v1.29.0-eks-5e0fdde
ip-192-168-1-31.ap-south-1.compute.internal  Ready   <none>  20m   v1.29.0-eks-5e0fdde
ip-192-168-1-79.ap-south-1.compute.internal  Ready   <none>  17m   v1.29.0-eks-5e0fdde
```

```
C:\Tools>dir
Volume in drive C has no label.
Volume Serial Number is 0EBC-E8DD
```

```
Directory of C:\Tools

04/23/2024  01:34 PM    <DIR>          .
04/23/2024  01:34 PM    <DIR>          ..
04/23/2024  01:30 PM           40,767,488 AWSCLIV2.msi
04/23/2024  01:35 PM           51,922,848 kubectl.exe
               2 File(s)        91,790,336 bytes
               2 Dir(s)   184,739,332,096 bytes free
```

```
C:\Tools>type pod.yaml.txt
The system cannot find the file specified.
```

```
C:\Tools>kubectl get pods -n kube-system
NAME          READY  STATUS  RESTARTS  AGE
aws-node-624mw  2/2   Running  0         20m
aws-node-c4rkb  2/2   Running  0         20m
aws-node-mtbrp  2/2   Running  0         23m
aws-node-rzjrb  2/2   Running  0         23m
coredns-5d5f56f475-c85xw  1/1  Running  0         38m
coredns-5d5f56f475-xkz9q  1/1  Running  0         38m
eks-pod-identity-agent-b6w2f  1/1  Running  0         20m
eks-pod-identity-agent-kb4zk  1/1  Running  0         23m
eks-pod-identity-agent-lhg2  1/1  Running  0         20m
eks-pod-identity-agent-sgtr  1/1  Running  0         23m
kube-proxy-gbx78  1/1  Running  0         23m
kube-proxy-mmv6s  1/1  Running  0         20m
kube-proxy-pbshm  1/1  Running  0         20m
kube-proxy-rrr75  1/1  Running  0         23m
```

```
C:\Tools>kubectl apply -f pod.yaml.txt
error: the path "pod.yaml.txt" does not exist
```

```
C:\Tools>type pod.yaml.txt
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 0
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
C:\Tools>kubectl apply -f pod.yaml.txt
deployment.apps/nginx-deployment created
```

```
C:\Tools>kubectl apply -f pod1.yaml.txt
pod/nginx created

C:\Tools>kubectl get pods -o wide
NAME      READY   STATUS    RESTARTS   AGE     IP           NODE          NOMINATED NODE   READINESS GATES
nginx    1/1     Running   0          24s    192.168.0.14 ip-192-168-0-183.ap-south-1.compute.internal <none>        <none>

C:\Tools>
```

```
C:\Tools>kubectl apply -f service.yaml.txt
service/nginx-service created

C:\Tools>kubectl apply -f pod1.yaml.txt
deployment.apps/nginx-deployment configured

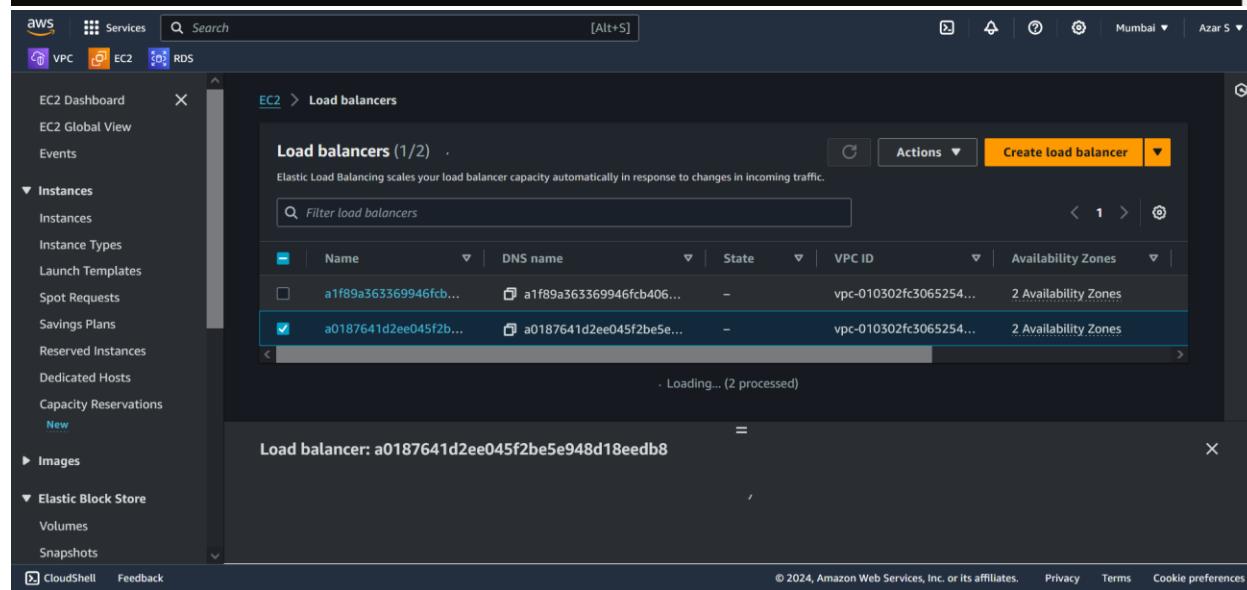
C:\Tools>kubectl get svc
NAME            TYPE      CLUSTER-IP      EXTERNAL-IP
kubernetes      ClusterIP   10.100.0.1    <none>
nginx-service   LoadBalancer 10.100.92.54  a0187641d2ee045f2be5e948d18eedb8-136229412.ap-south-1.elb.amazonaws.com  80:30749/TCP  59s

C:\Tools>kubectl get deployment
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 1/1     1           1           25m

C:\Tools>kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
nginx           1/1     Running   0          12m
nginx-deployment-7c79c4bf97-s86fw 1/1     Running   0          100s

C:\Tools>kubectl get node
NAME                  STATUS   ROLES   AGE   VERSION
ip-192-168-0-183.ap-south-1.compute.internal Ready   <none>  54m   v1.29.0-eks-5e0fdde
ip-192-168-1-31.ap-south-1.compute.internal   Ready   <none>  53m   v1.29.0-eks-5e0fdde

C:\Tools>
```



**EC2 > Load balancers > a0187641d2ee045f2be5e948d18eedb8**

**Details**

Load balancer type Classic	Status 2 of 2 instances in service	VPC <a href="#">vpc-010302fc306525480</a>	Date created April 23, 2024, 14:40 (UTC+05:30)
Scheme Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones	
		subnet-0a9d56657fe399cdb	ap-south-1a (aps1-az1)
		subnet-0f23fd9441b2a20	a
		p-south-1b (aps1-az3)	

**DNS name copied**

[a0187641d2ee045f2be5e948d18eedb8-136229412.ap-south-1.elb.amazonaws.com \(A Record\)](#)

This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. [Learn more](#)

[Launch migration wizard](#)

**CloudShell Feedback**

**EC2 > Security Groups > sg-0c497c0d159cdc85a - k8s-elb-a0187641d2ee045f2be5e948d18eedb8 > Edit inbound rules**

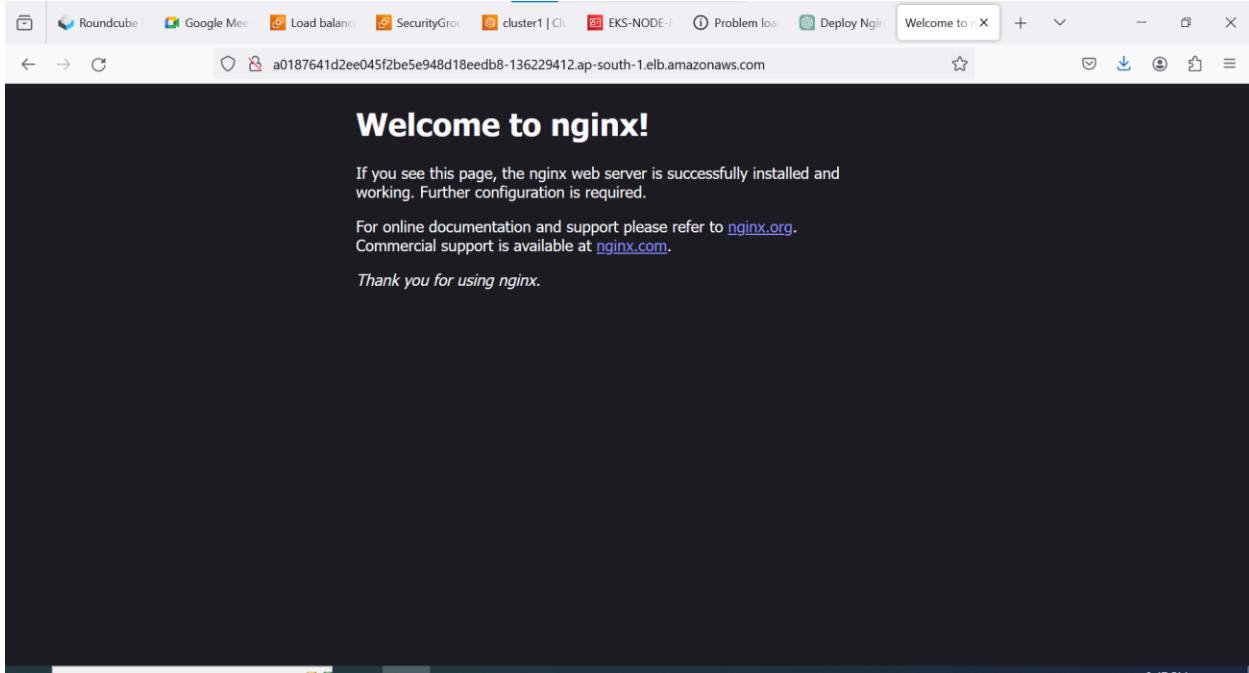
### Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
sgr-0ad30cae3bcd93f7a	Custom ICMP - IPv4	Dest... ▾	frag... ▾	Cust... ▾	<input type="text"/> 0.0.0.0 X
sgr-0aa9eb771a242a6e1	HTTP	TCP	80	Cust... ▾	<input type="text"/> 0.0.0.0 X

[Add rule](#)

**CloudShell Feedback**



### MANIFEST FILE:

#### Pod.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
```

```
- name: nginx  
  image: nginx:latest  
  ports:  
    - containerPort: 80
```

### Service.yaml

```
apiVersion: v1  
kind: Service  
metadata:  
  name: nginx-service  
spec:  
  selector:  
    app: nginx  
  ports:  
    - protocol: TCP  
      port: 80  
      targetPort: 80  
type: LoadBalancer
```

