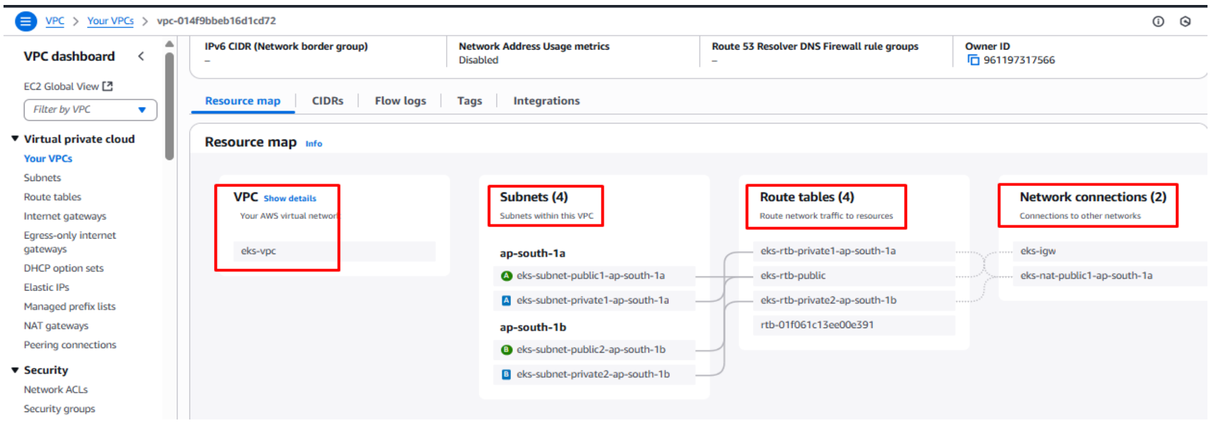
By Arun Baidya

EKS Cluster Provisioning from AWS Console

1. Create a VPC with 2 Public Subnet and 2 Private Subnets



VPC > Your VPCs > vpc-014f9bbeb16d1cd72

IPv6 CIDR (Network border group): –
Network Address Usage metrics: Disabled
Route 53 Resolver DNS Firewall rule groups: –
Owner ID: 961197317566

Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map Info

VPC Show details
Your AWS virtual networ
eks-vpc

Subnets (4)
Subnets within this VPC
ap-south-1a
eks-subnet-public1-ap-south-1a
eks-subnet-private1-ap-south-1a
ap-south-1b
eks-subnet-public2-ap-south-1b
eks-subnet-private2-ap-south-1b

Route tables (4)
Route network traffic to resources
eks-rtb-private1-ap-south-1a
eks-rtb-public
eks-rtb-private2-ap-south-1b
rtb-01f061c13ee00e391

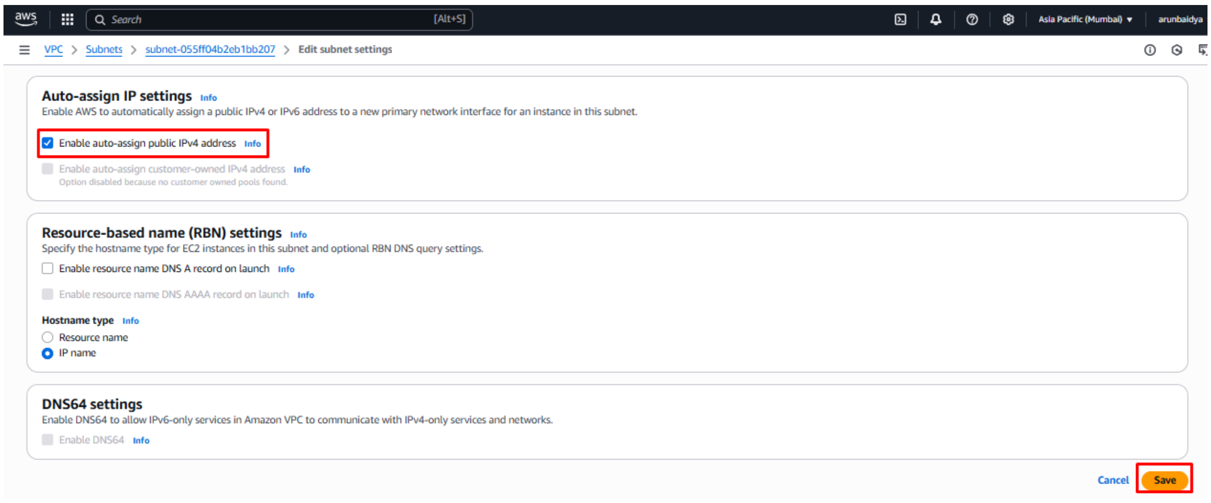
Network connections (2)
Connections to other networks
eks-igw
eks-nat-public1-ap-south-1a

2. Check All Subnets, Route Table, IGW, NAT GW, EIP, Security Group

a. There must have 2 Private Subnets and 2 Public Subnets in Available State
b. There must be created 2 Private Route Tables and 1 Public Route Table for 2 subnets. Another default Rout Table will get created as a "Main=Yes"
c. 1 IGW will get created which must be Attached with the VPC
d. 1 NAT GW (or 2) must be created in the Public Subnet and 1 Elastic IP Address should be attached with that NAT GW.
e. 1 default VPC security group will be created at the time of VPC Creation and will be attached with the VPC

3. Now Enable auto-assign public IPv4 address for both the Public Subnets

Go to Subnet => Select Public Subnet => Actions => Edit Subnet Settings => Tick on "Enable auto-assign public IPv4 address" => Save



VPC > Subnets > subnet-055ff04b2eb1bb207 > Edit subnet settings

Auto-assign IP settings Info
Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

- [x] Enable auto-assign public IPv4 address Info
- [ ] Enable auto-assign customer-owned IPv4 address Info
  Option disabled because no customer owned pools found.

Resource-based name (RBN) settings Info
Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

- [ ] Enable resource name DNS A record on launch Info
- [ ] Enable resource name DNS AAAA record on launch Info

Hostname type Info
Resource name [✓ IP name]

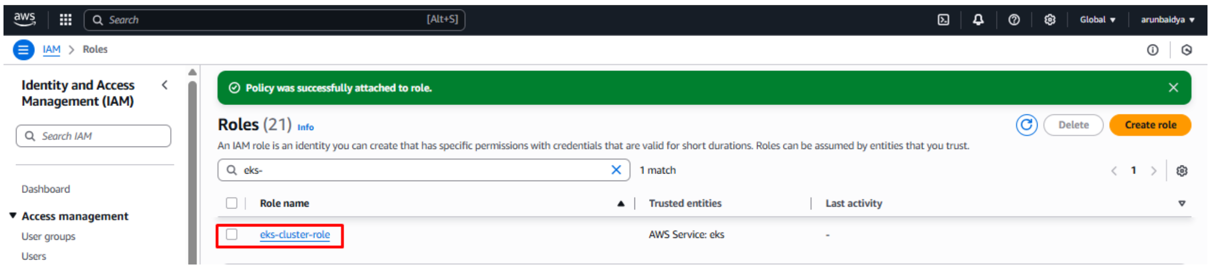
DNS64 settings
Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.

- [ ] Enable DNS64 Info

By Arun Baidya

4. Create IAM Role for EKS Cluster

a. *Go to IAM => Role => Create role => AWS service => Use case => Select "EKS" for Service or use case => Select "EKS-Cluster" for Use case => Next => Next => Set a Role name "eks-cluster-role" => Create role*



Policy was successfully attached to role.

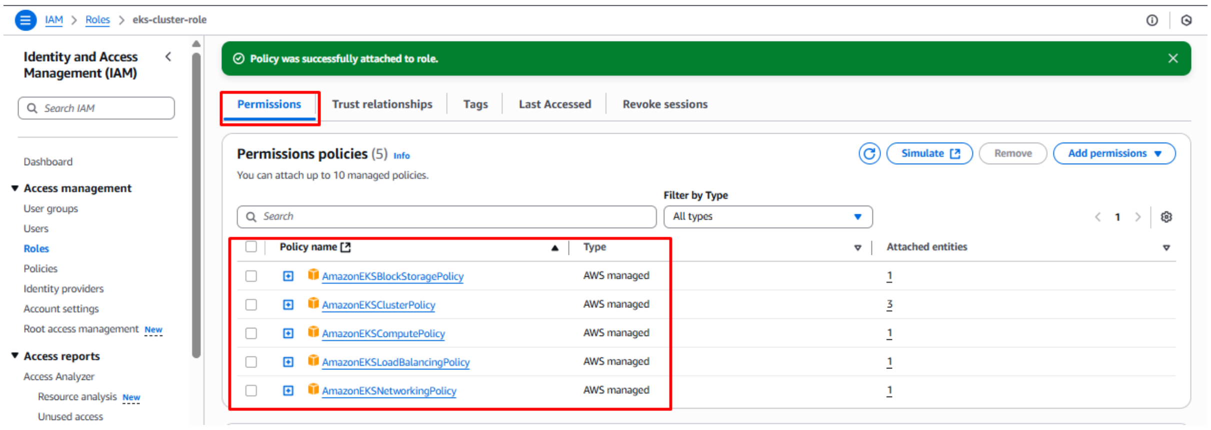
Roles (21) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

eks- — 1 match

Role name	Trusted entities	Last activity
eks-cluster-role	AWS Service: eks	-

b. *Click on the created Role "eks-cluster-role" => Permissions => Add Permissions => Attach policies => Add these Permissions => AmazonEKSBlockStoragePolicy, AmazonEKSComputePolicy, AmazonEKSLoadBalancingPolicy, AmazonEKSNetworkingPolicy => Add permissions*



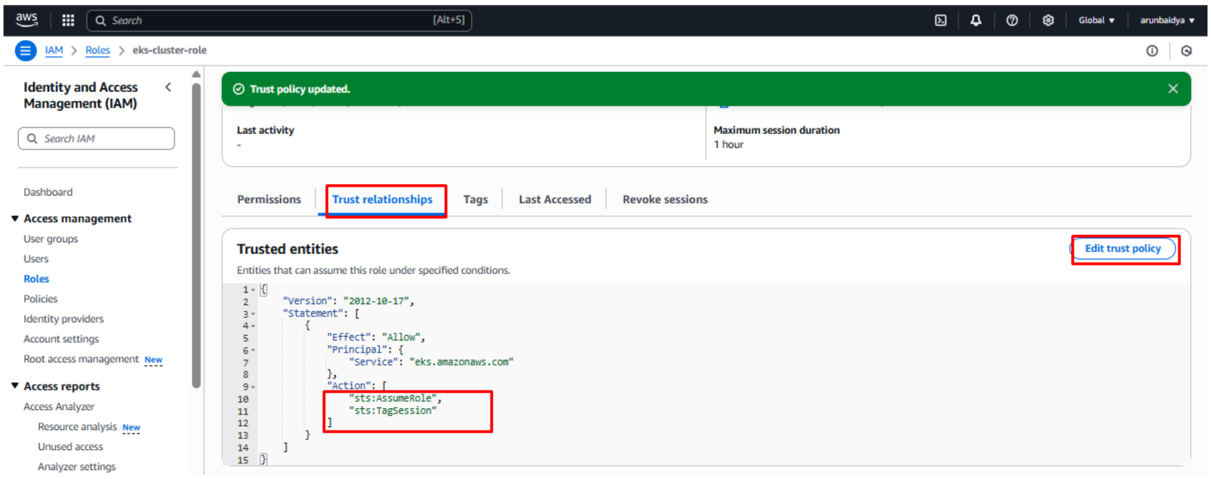
Policy was successfully attached to role.

Permissions policies (5) Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonEKSBlockStoragePolicy	AWS managed	1
AmazonEKSClusterPolicy	AWS managed	3
AmazonEKSComputePolicy	AWS managed	1
AmazonEKSLoadBalancingPolicy	AWS managed	1
AmazonEKSNetworkingPolicy	AWS managed	1

c. *Go to "Trust Relationship" => Edit trusted policies => Add the below trust policy "sts:TagSession" in json script => Update policy*



Trust policy updated.

Last activity: -

Maximum session duration: 1 hour

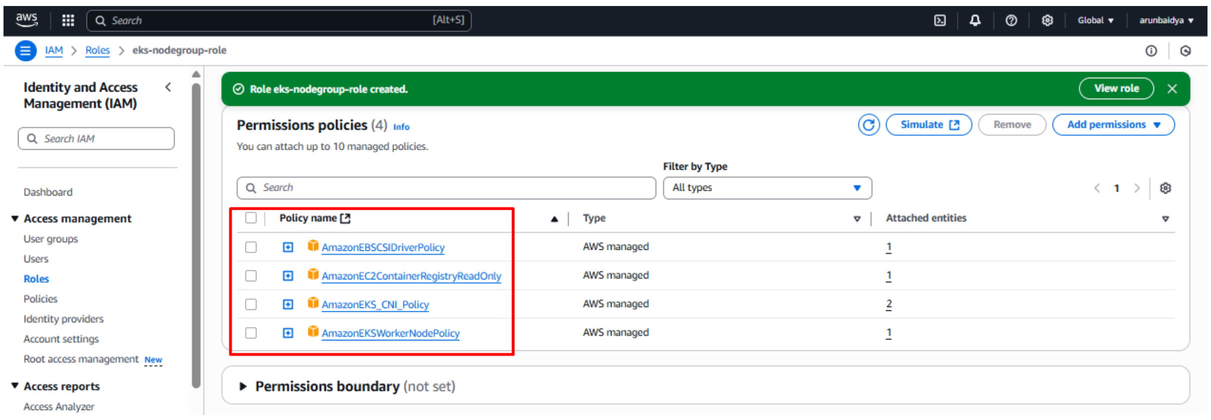
Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
```

5. Create IAM role for EKS Node Group

a. *Go to Role => Create role => AWS service => Use case => Select “EC2” for Service or use case => Select “EC2” for Use case => Next => Add Permissions => Select these Policies => AmazonEKSWorkerNodePolicy, AmazonEC2ContainerRegistryReadOnly, AmazonEKS_CNI_Policy, AmazonEBSCSIDriverPolicy => Next => Set a Role name “eks-nodegroup-role” => Create role*


IAM > Roles > eks-nodegroup-role

Role eks-nodegroup-role created.

Permissions policies (4)

You can attach up to 10 managed policies.

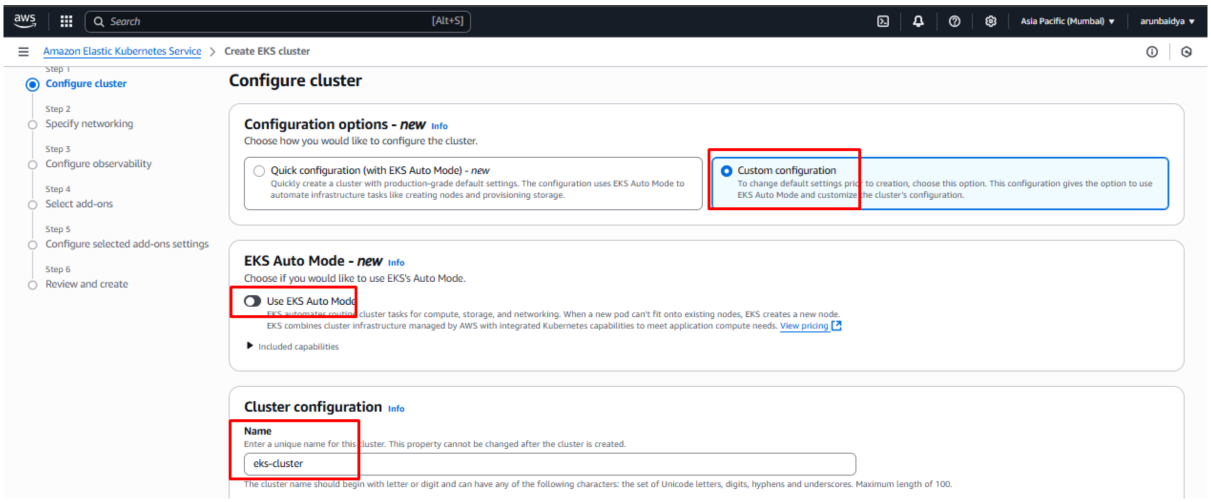
Filter by Type: All types

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

Permissions boundary (not set)

6. Create the EKS Cluster

a. *Go to AWS EKS => Create cluster => Custom configuration => “Disable” EKS Auto Mode => Set Cluster Name “eks-cluster” => Select Cluster IAM role “eks-cluster-role” => Select Kubernetes version “1.33” or latest version => Upgrade policy “Standard” => Cluster access “Allow cluster administrator access” => Cluster authentication mode “EKS API” => Next*


Amazon Elastic Kubernetes Service > Create EKS cluster

Configure cluster

Configuration options - *new*

Choose how you would like to configure the cluster.

Quick configuration (with EKS Auto Mode) - *new*: Quickly create a cluster with production-grade default settings. The configuration uses EKS Auto Mode to automate infrastructure tasks like creating nodes and provisioning storage.

[✓ Custom configuration]: To change default settings prior to creation, choose this option. This configuration gives the option to use EKS Auto Mode and customize the cluster's configuration.

EKS Auto Mode - *new*

Choose if you would like to use EKS's Auto Mode.

Use EKS Auto Mode: off

EKS automates routine cluster tasks for compute, storage, and networking. When a new pod can't fit onto existing nodes, EKS creates a new node. EKS combines cluster infrastructure managed by AWS with integrated Kubernetes capabilities to meet application compute needs. View pricing

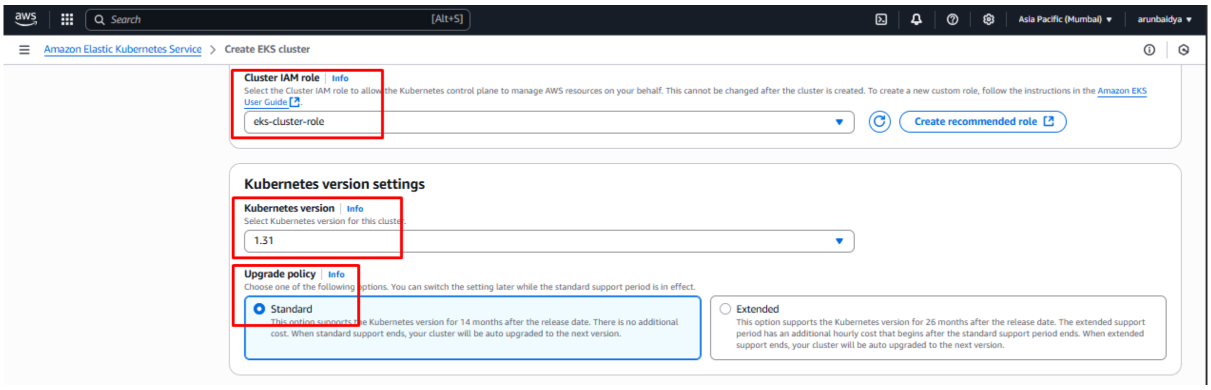
Included capabilities

Cluster configuration

Name: eks-cluster

Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.


Amazon Elastic Kubernetes Service > Create EKS cluster

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.
eks-cluster-role

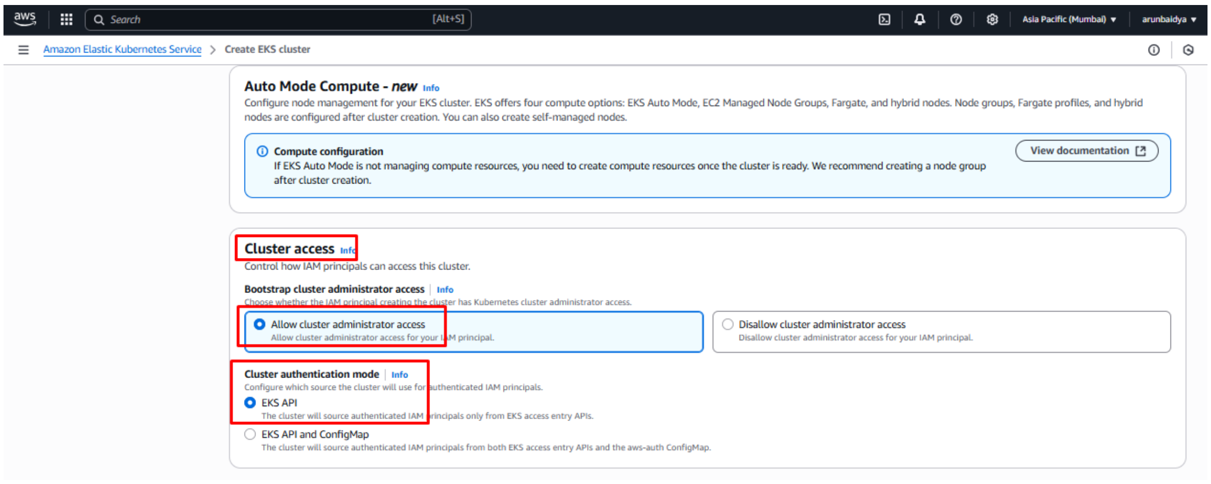
Kubernetes version settings

Kubernetes version Info
Select Kubernetes version for this cluster.
1.31

Upgrade policy Info
Choose one of the following options. You can switch the setting later while the standard support period is in effect.

[✓ Standard] This option supports the Kubernetes version for 14 months after the release date. There is no additional cost. When standard support ends, your cluster will be auto upgraded to the next version.

Extended: This option supports the Kubernetes version for 26 months after the release date. The extended support period has an additional hourly cost that begins after the standard support period ends. When extended support ends, your cluster will be auto upgraded to the next version.


Amazon Elastic Kubernetes Service > Create EKS cluster

Auto Mode Compute - *new* Info
Configure node management for your EKS cluster. EKS offers four compute options: EKS Auto Mode, EC2 Managed Node Groups, Fargate, and hybrid nodes. Node groups, Fargate profiles, and hybrid nodes are configured after cluster creation. You can also create self-managed nodes.

Compute configuration
If EKS Auto Mode is not managing compute resources, you need to create compute resources once the cluster is ready. We recommend creating a node group after cluster creation.

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.

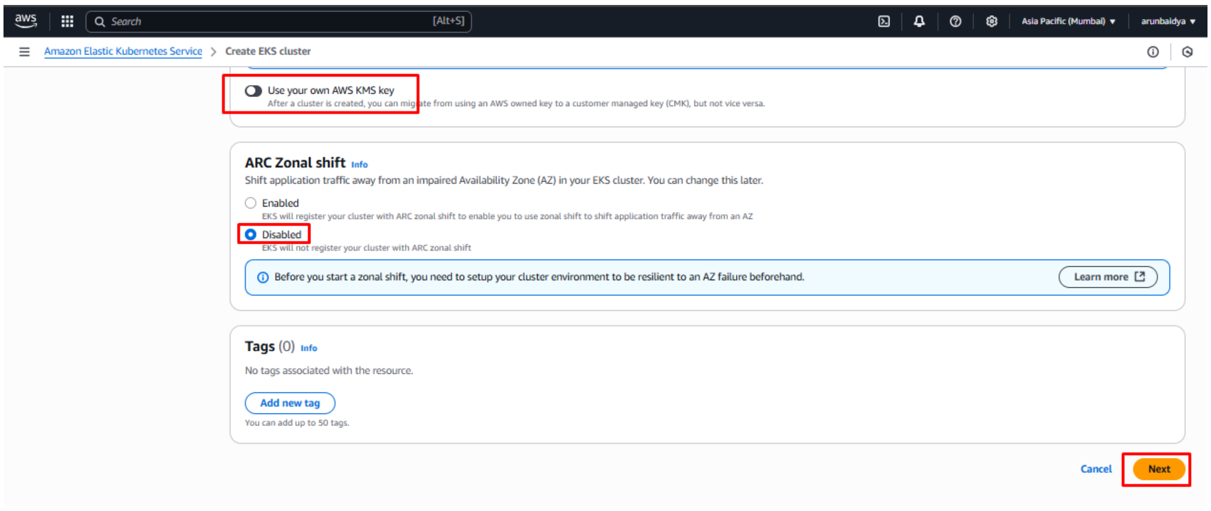
[✓ Allow cluster administrator access] Allow cluster administrator access for your IAM principal.

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.


Amazon Elastic Kubernetes Service > Create EKS cluster

Use your own AWS KMS key
After a cluster is created, you can migrate from using an AWS owned key to a customer managed key (CMK), but not vice versa.

ARC Zonal shift Info
Shift application traffic away from an impaired Availability Zone (AZ) in your EKS cluster. You can change this later.

Enabled: EKS will register your cluster with ARC zonal shift to enable you to use zonal shift to shift application traffic away from an AZ

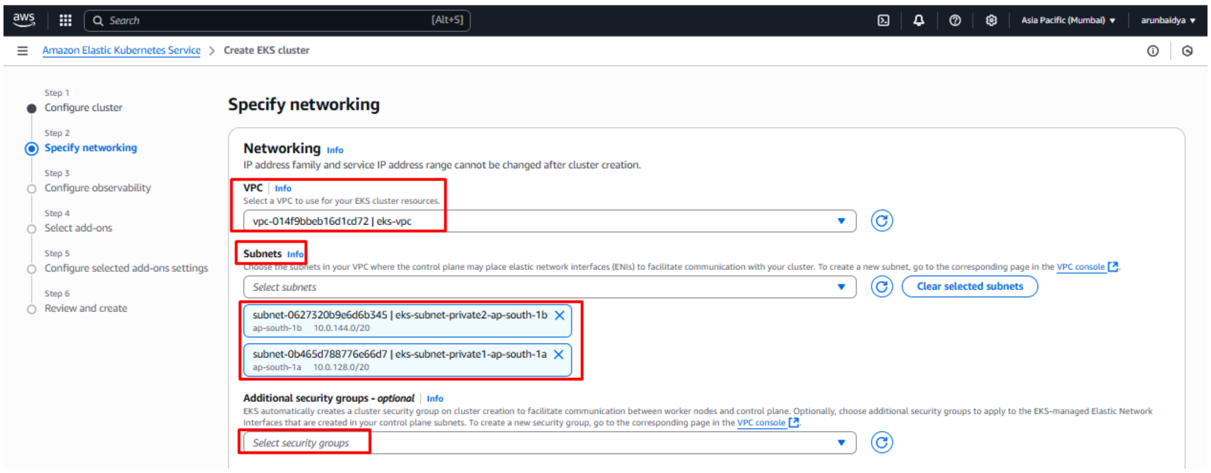
[✓ Disabled] EKS will not register your cluster with ARC zonal shift

Before you start a zonal shift, you need to setup your cluster environment to be resilient to an AZ failure beforehand.

Tags (0) Info
No tags associated with the resource.
You can add up to 50 tags.

b. *Go to Network => Select VPC “eks-vpc” => Select only “2 Private Subnets” => Additional security groups – Optional => Choose cluster IP address family “IPV4” => Cluster endpoint access “Public & private” => Advanced settings, set “CIDR block” – can set company’s public IP CIDR or your local machine’s CDIR => Next*

Note: Search “what is my public ip”



Amazon Elastic Kubernetes Service > Create EKS cluster

Step 1 Configure cluster
Step 2 Specify networking
Step 3 Configure observability
Step 4 Select add-ons
Step 5 Configure selected add-ons settings
Step 6 Review and create

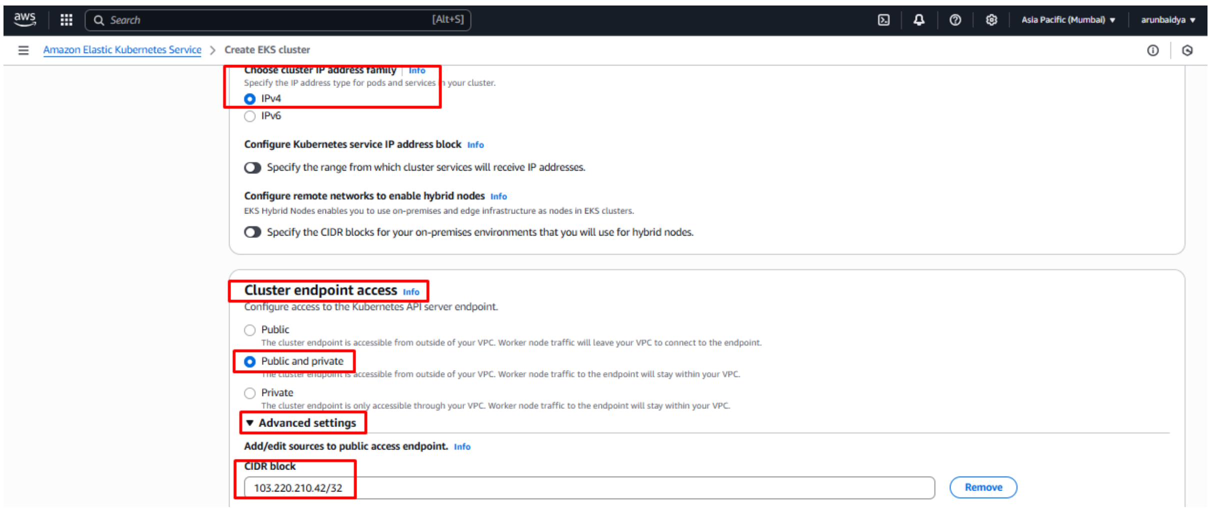
Specify networking

Networking Info
IP address family and service IP address range cannot be changed after cluster creation.

VPC Info
Select a VPC to use for your EKS cluster resources.
vpc-014f9bbeb16d1cd72 | eks-vpc

Subnets Info
Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the VPC console.
Select subnets
Clear selected subnets
subnet-0627320b9e6d6b345 | eks-subnet-private2-ap-south-1b
ap-south-1b 10.0.144.0/20
subnet-0b465d788776e66d7 | eks-subnet-private1-ap-south-1a
ap-south-1a 10.0.128.0/20

Additional security groups - *optional* Info
EKS automatically creates a cluster security group on cluster creation to facilitate communication between worker nodes and control plane. Optionally, choose additional security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your control plane subnets. To create a new security group, go to the corresponding page in the VPC console.
Select security groups



Amazon Elastic Kubernetes Service > Create EKS cluster

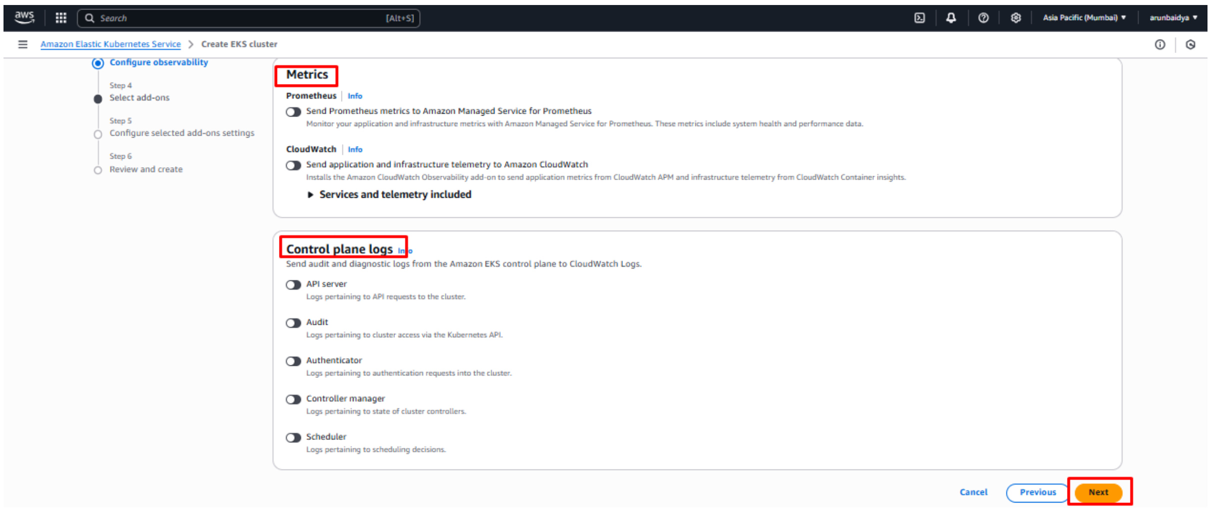
Choose cluster IP address family Info
Specify the IP address type for pods and services in your cluster.
[✓ IPv4] IPv6

Configure Kubernetes service IP address block Info
Specify the range from which cluster services will receive IP addresses.

Configure remote networks to enable hybrid nodes Info
EKS Hybrid Nodes enables you to use on-premises and edge infrastructure as nodes in EKS clusters.
Specify the CIDR blocks for your on-premises environments that you will use for hybrid nodes.

Cluster endpoint access Info
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
Add/edit sources to public access endpoint. Info
CIDR block
103.220.210.42/32
Remove

c. *Configure observability => Select Metrics Tools as per the plan => Enable Control plane logs as per the requirements => Next*



Amazon Elastic Kubernetes Service > Create EKS cluster

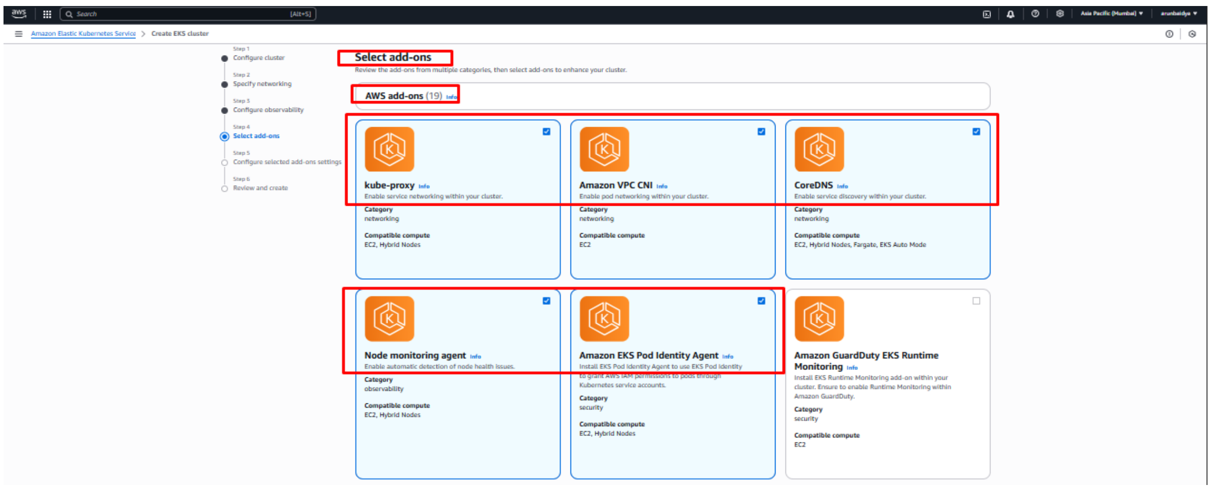
Configure observability
Step 4 Select add-ons
Step 5 Configure selected add-ons settings
Step 6 Review and create

Metrics
Prometheus Info
Send Prometheus metrics to Amazon Managed Service for Prometheus
Monitor your application and infrastructure metrics with Amazon Managed Service for Prometheus. These metrics include system health and performance data.
CloudWatch Info
Send application and infrastructure telemetry to Amazon CloudWatch
Installs the Amazon CloudWatch Observability add-on to send application metrics from CloudWatch APM and infrastructure telemetry from CloudWatch Container insights.
▸ Services and telemetry included

Control plane logs Info
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

d. *Select Add-ons => Keep the default selected Add-ons Plug-ins (kube-proxy, Amazon VPC CNI, CoreDNS, Node monitoring agent, AWS EKS Pod Identity Agent, External DNS, Metrics Server) => Next*



Amazon Elastic Kubernetes Service > Create EKS cluster

Step 1 Configure cluster
Step 2 Specify networking
Step 3 Configure observability
Step 4 Select add-ons
Step 5 Configure selected add-ons settings
Step 6 Review and create

Select add-ons
Review the add-ons from multiple categories, then select add-ons to enhance your cluster.

AWS add-ons (19) Info

kube-proxy Info ✓
Enable service networking within your cluster.
Category: networking
Compatible compute: EC2, Hybrid Nodes

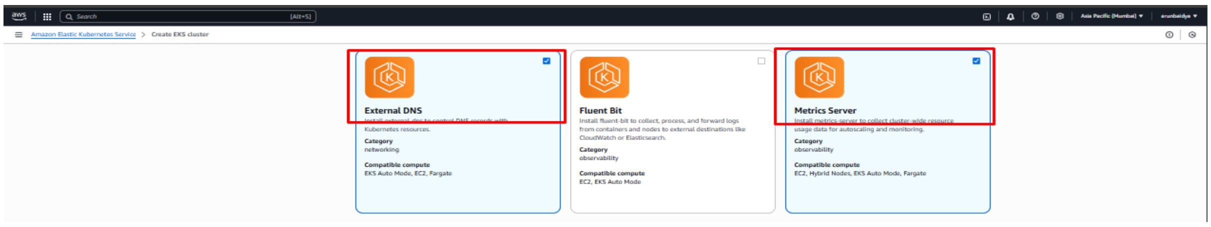
Amazon VPC CNI Info ✓
Enable pod networking within your cluster.
Category: networking
Compatible compute: EC2

CoreDNS Info ✓
Enable service discovery within your cluster.
Category: networking
Compatible compute: EC2, Hybrid Nodes, Fargate, EKS Auto Mode

Node monitoring agent Info ✓
Enable automatic detection of node health issues.
Category: observability
Compatible compute: EC2, Hybrid Nodes

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
Compatible compute: EC2, Hybrid Nodes

Amazon GuardDuty EKS Runtime Monitoring Info ☐
Install EKS Runtime Monitoring add-on within your cluster. Ensure to enable Runtime Monitoring within Amazon GuardDuty.
Category: security
Compatible compute: EC2



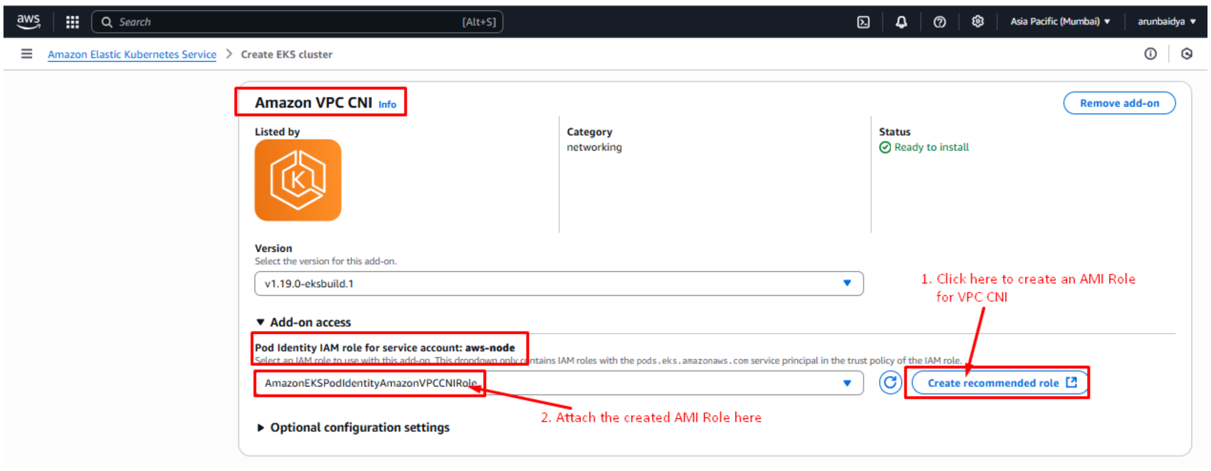
Amazon Elastic Kubernetes Service > Create EKS cluster

External DNS ✓
Install external-dns to control DNS records with Kubernetes resources.
Category: networking
Compatible compute: EKS Auto Mode, EC2, Fargate

Fluent Bit ☐
Install fluent-bit to collect, process, and forward logs from containers and nodes to external destinations like CloudWatch or Elasticsearch.
Category: observability
Compatible compute: EC2, EKS Auto Mode

Metrics Server ✓
Install metrics-server to collect cluster-wide resource usage data for autoscaling and monitoring.
Category: observability
Compatible compute: EC2, Hybrid Nodes, EKS Auto Mode, Fargate

e. *Configure selected add-ons settings => AWS VPC CNI => create an AMI Role => Attach the Role*



Amazon Elastic Kubernetes Service > Create EKS cluster

Amazon VPC CNI Info | Remove add-on

Listed by | Category: networking | Status: Ready to install

Version
Select the version for this add-on.
v1.19.0-eksbuild.1

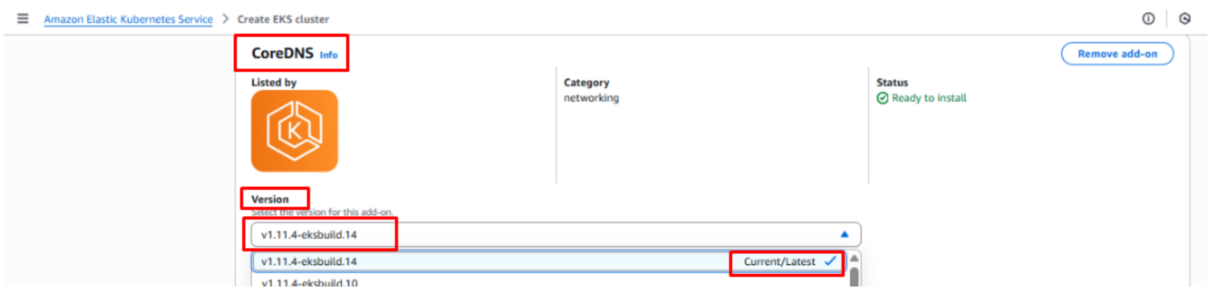
▼ **Add-on access**
Pod Identity IAM role for service account: aws-node
Select an IAM role to use with this add-on. This dropdown only contains IAM roles with the pods.eks.amazonaws.com service principal in the trust policy of the IAM role.
AmazonEKSPodIdentityAmazonVPCCNIRole | Create recommended role

1. Click here to create an AMI Role for VPC CNI

2. Attach the created AMI Role here

▶ **Optional configuration settings**

f. *CoreDNS => Change the Default Version to Current Version*



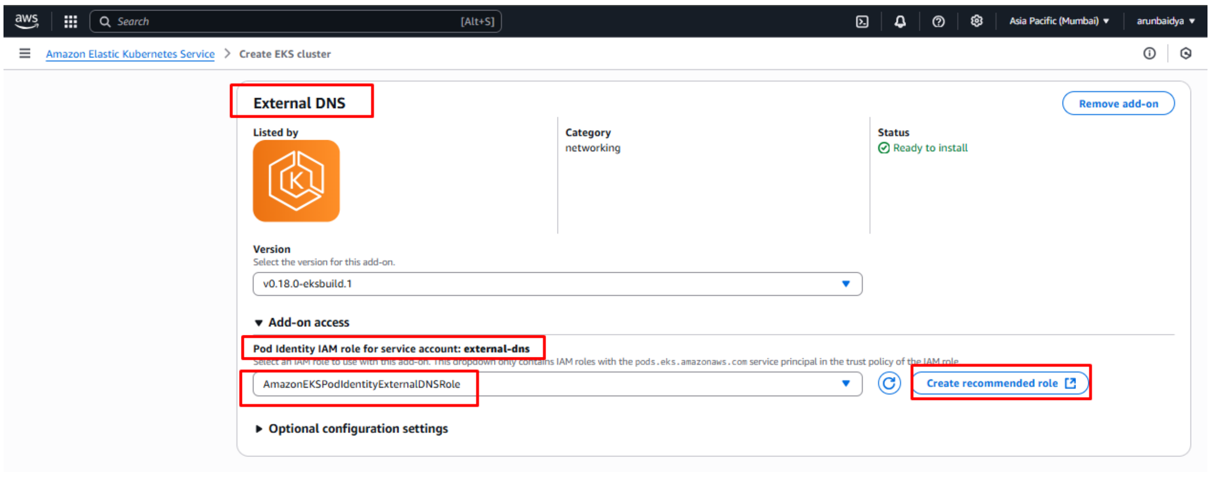
Amazon Elastic Kubernetes Service > Create EKS cluster

CoreDNS Info | Remove add-on

Listed by | Category: networking | Status: Ready to install

Version
Select the version for this add-on.
v1.11.4-eksbuild.14
v1.11.4-eksbuild.14 Current/Latest ✓
v1.11.4-eksbuild.10

g. *Configure selected add-ons settings => External DNS => create an AMI Role => Attach the Role => Next*



External DNS

Listed by

Category
networking

Status
Ready to install

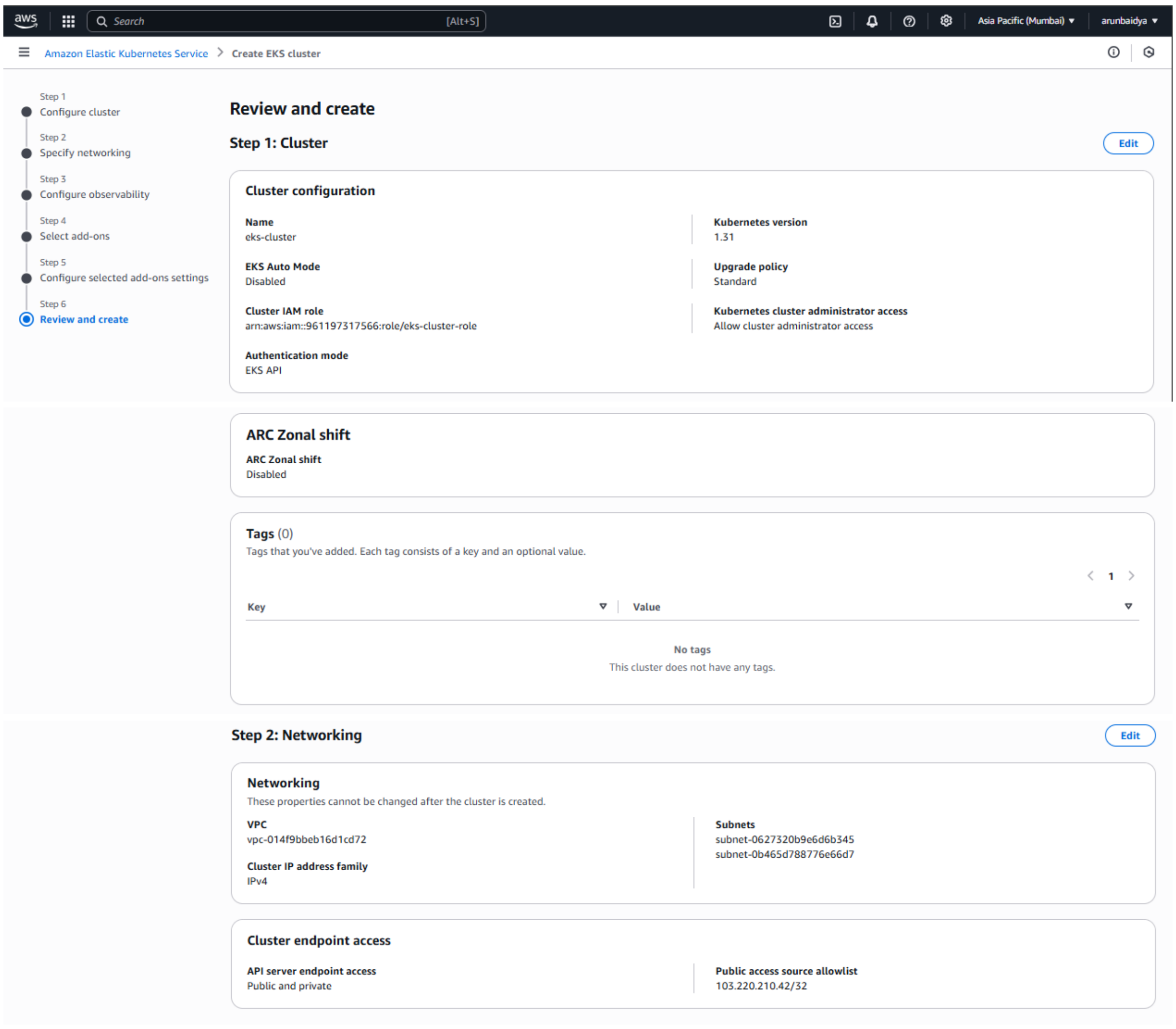
Version
Select the version for this add-on.
v0.18.0-eksbuild.1

▼ Add-on access

Pod Identity IAM role for service account: **external-dns**
Select an IAM role to use with this add-on. This dropdown only contains IAM roles with the `pods.eks.amazonaws.com` service principal in the trust policy of the IAM role.
AmazonEKSPodIdentityExternalDNSRole

▶ Optional configuration settings

7. Review => Click on "Create" => Wait for 10 to 15 minutes to get the EKS Cluster Created



Review and create

Step 1: Cluster

Cluster configuration

Name eks-cluster	Kubernetes version 1.31
EKS Auto Mode Disabled	Upgrade policy Standard
Cluster IAM role arn:aws:iam::961197317566:role/eks-cluster-role	Kubernetes cluster administrator access Allow cluster administrator access
Authentication mode EKS API	

ARC Zonal shift

ARC Zonal shift
Disabled

Tags (0)
Tags that you've added. Each tag consists of a key and an optional value.

Key	Value

No tags
This cluster does not have any tags.

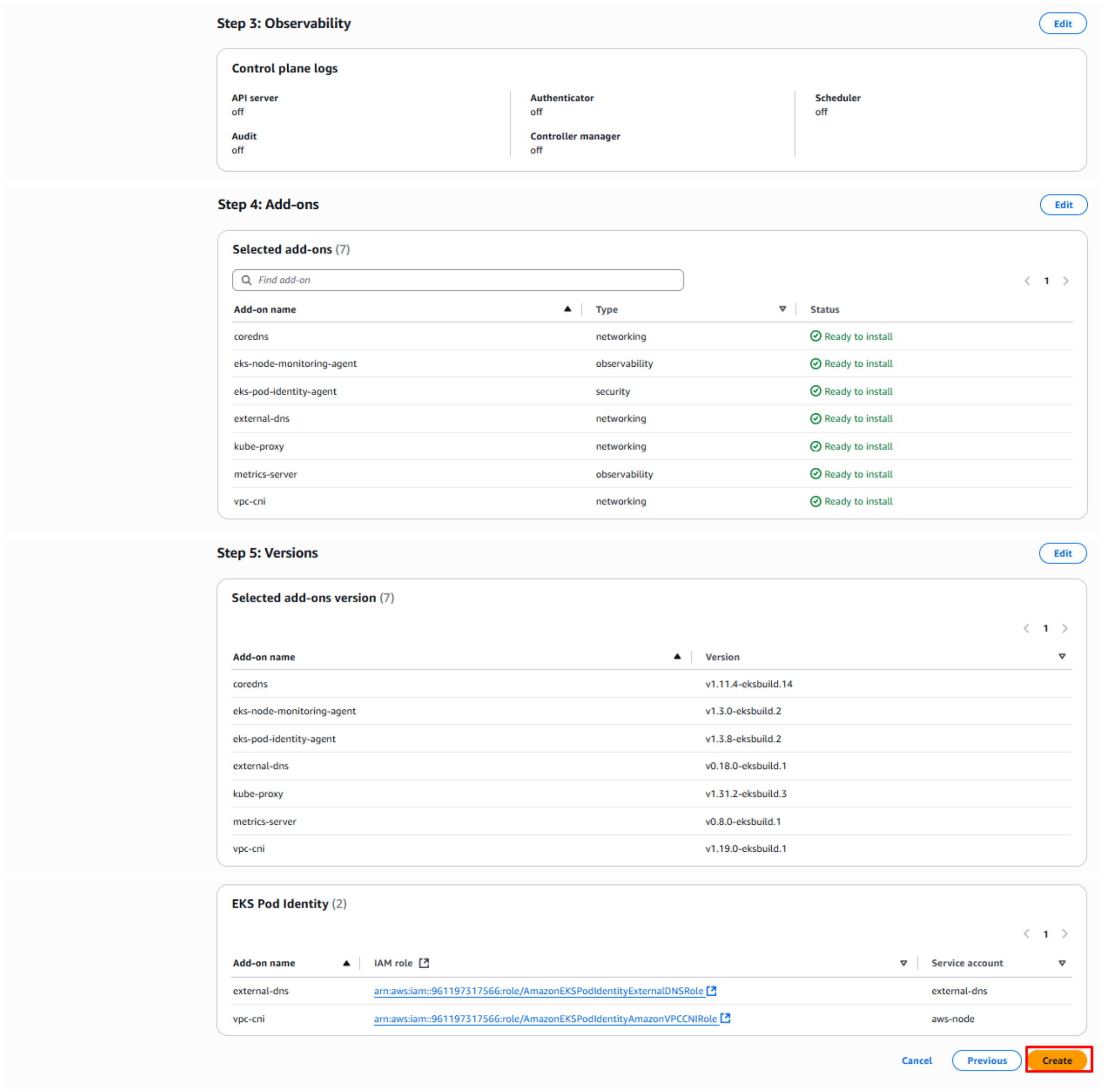
Step 2: Networking

Networking
These properties cannot be changed after the cluster is created.

VPC vpc-014f9bbeb16d1cd72	Subnets subnet-0627320b9e6d6b345 subnet-0b465d788776e66d7
Cluster IP address family IPv4	

Cluster endpoint access

API server endpoint access Public and private	Public access source allowlist 103.220.210.42/32



Step 3: Observability

Control plane logs

API server: off	Authenticator: off	Scheduler: off
Audit: off	Controller manager: off	

Step 4: Add-ons

Selected add-ons (7)

Add-on name	Type	Status
coredns	networking	Ready to install
eks-node-monitoring-agent	observability	Ready to install
eks-pod-identity-agent	security	Ready to install
external-dns	networking	Ready to install
kube-proxy	networking	Ready to install
metrics-server	observability	Ready to install
vpc-cni	networking	Ready to install

Step 5: Versions

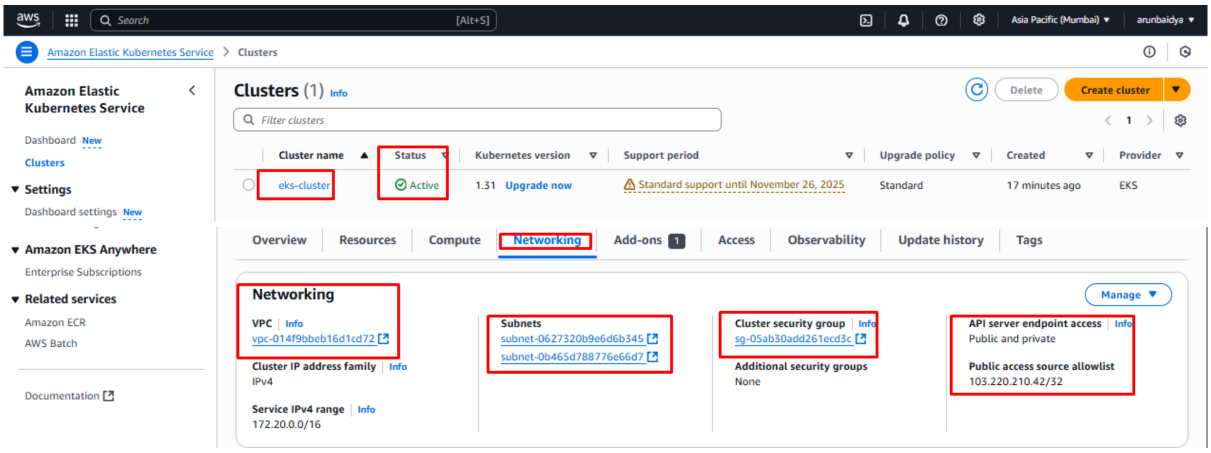
Selected add-ons version (7)

Add-on name	Version
coredns	v1.11.4-eksbuild.14
eks-node-monitoring-agent	v1.3.0-eksbuild.2
eks-pod-identity-agent	v1.3.8-eksbuild.2
external-dns	v0.18.0-eksbuild.1
kube-proxy	v1.31.2-eksbuild.3
metrics-server	v0.8.0-eksbuild.1
vpc-cni	v1.19.0-eksbuild.1

EKS Pod Identity (2)

Add-on name	IAM role	Service account
external-dns	arn:aws:iam::961197317566:role/AmazonEKSPodIdentityExternalDNSRole	external-dns
vpc-cni	arn:aws:iam::961197317566:role/AmazonEKSPodIdentityAmazonVPCCNIRole	aws-node

8. After 10 to 15 minutes, EKS Cluster will get Created



Clusters (1) Info

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
eks-cluster	Active	1.31 Upgrade now	Standard support until November 26, 2025	Standard	17 minutes ago	EKS

Networking

VPC | Info
vpc-014f9bbeb16d1cd72

Cluster IP address family | Info
IPv4

Service IPv4 range | Info
172.20.0.0/16

Subnets
subnet-0627320b9e6d6b345
subnet-0b465d788776e66d7

Cluster security group | Info
sg-05ab30add261ecd3c

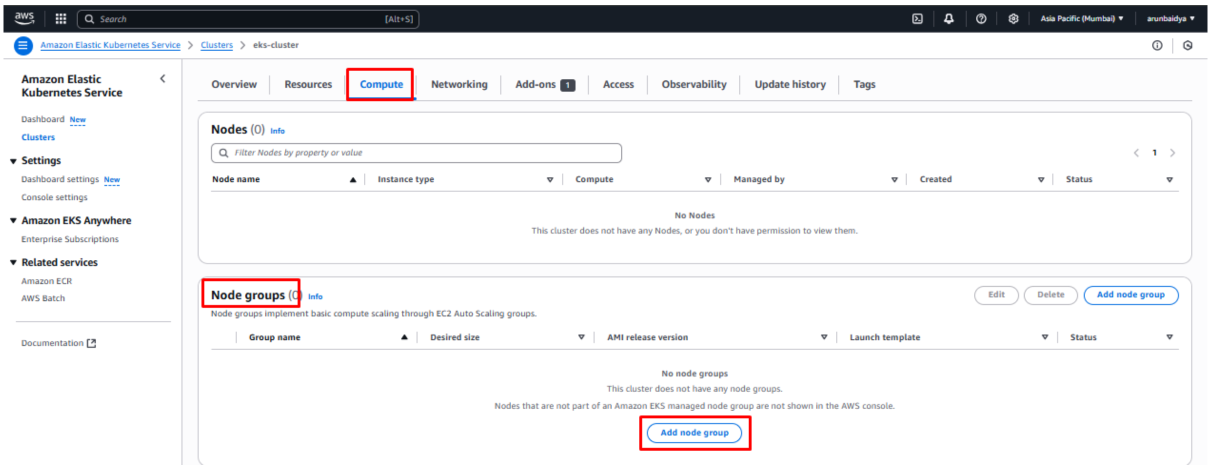
Additional security groups
None

API server endpoint access | Info
Public and private

Public access source allowlist
103.220.210.42/32

9. Now, create EKS Node Group

a. *Click on EKS Cluster => Compute => Node groups => Add node group*



Nodes (0) Info

Filter Nodes by property or value

Node name	Instance type	Compute	Managed by	Created	Status

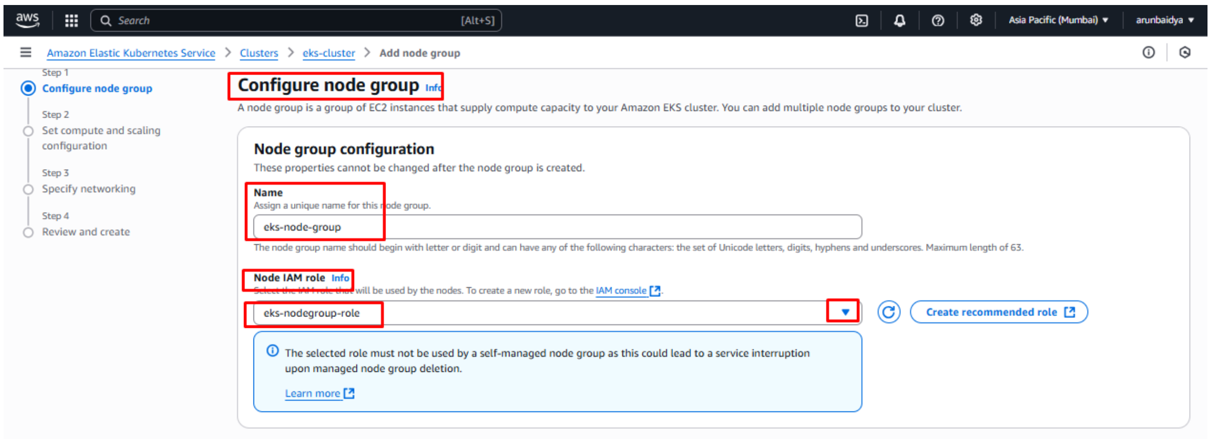
No Nodes
This cluster does not have any Nodes, or you don't have permission to view them.

Node groups (0) Info
Node groups implement basic compute scaling through EC2 Auto Scaling groups.

Group name	Desired size	AMI release version	Launch template	Status

No node groups
This cluster does not have any node groups.
Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.

b. *Configure node group => Set a Name "eks-node-group" => Select Node ALI role "eks-nodegroup-role" => If you have then, enable and select a "Launch Template" => Else Skip => Next*



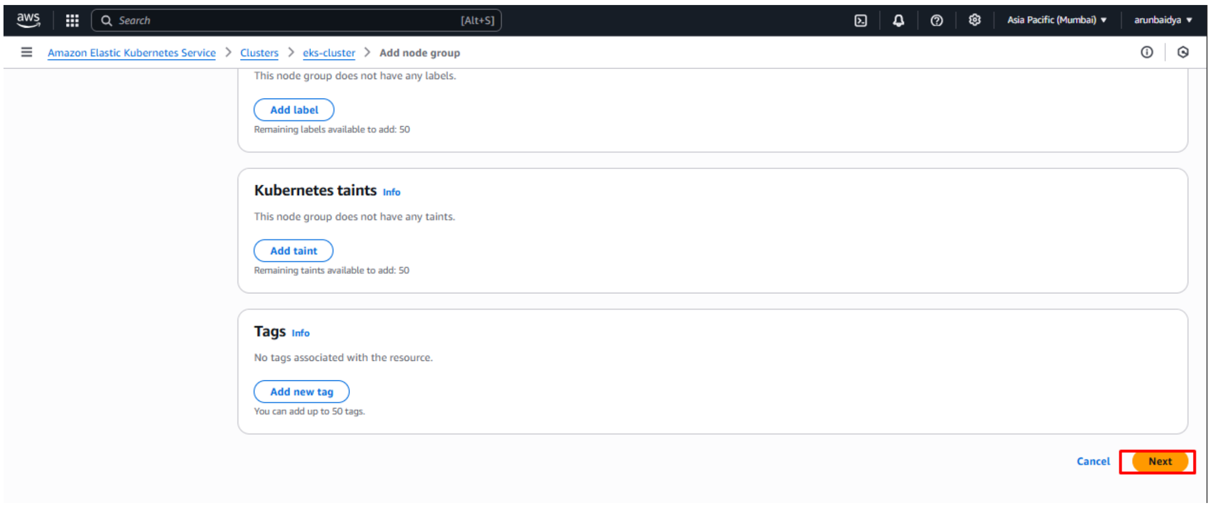
Configure node group Info
A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

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

Name
Assign a unique name for this node group.
eks-node-group
The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role Info
Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console
eks-nodegroup-role

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.
Learn more


Search [Alt+S]
Asia Pacific (Mumbai) ▾
arunbaidya ▾
Amazon Elastic Kubernetes Service > Clusters > eks-cluster > Add node group

This node group does not have any labels.

Remaining labels available to add: 50

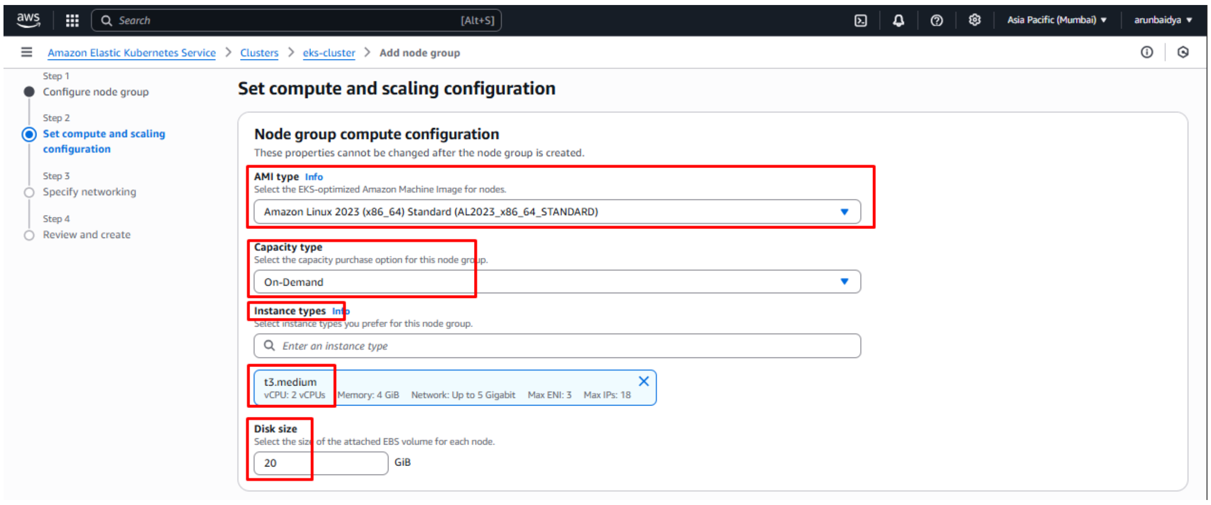
Kubernetes taints Info
This node group does not have any taints.

Remaining taints available to add: 50

Tags Info
No tags associated with the resource.

You can add up to 50 tags.

c. *Set compute and scaling configuration => Select AMI type => Select Capacity type => Select Instance types => Set Disk size*


Amazon Elastic Kubernetes Service > Clusters > eks-cluster > Add node group

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

Set compute and scaling configuration

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

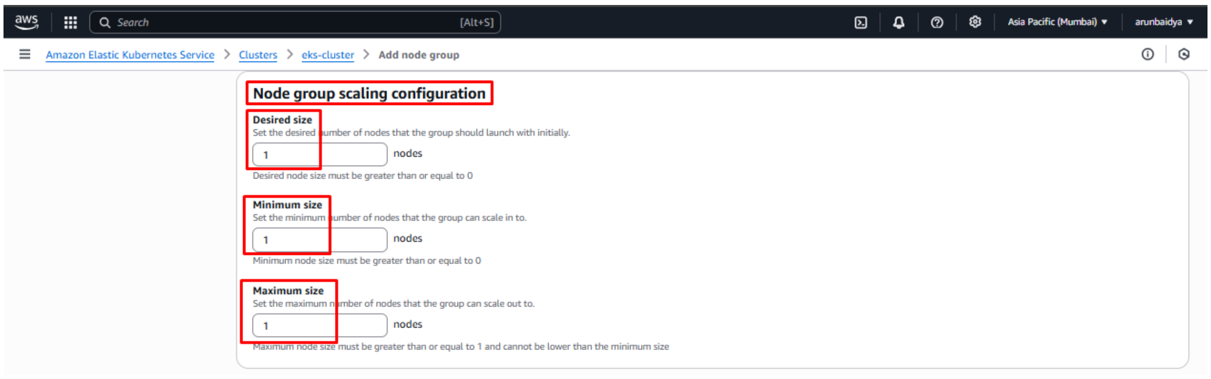
AMI type Info
Select the EKS-optimized Amazon Machine Image for nodes.
Amazon Linux 2023 (x86_64) Standard (AL2023_x86_64_STANDARD)

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

d. *Node group scaling configuration -> Set Desired size => Minimum size => Maximum size*


Amazon Elastic Kubernetes Service > Clusters > eks-cluster > Add node group

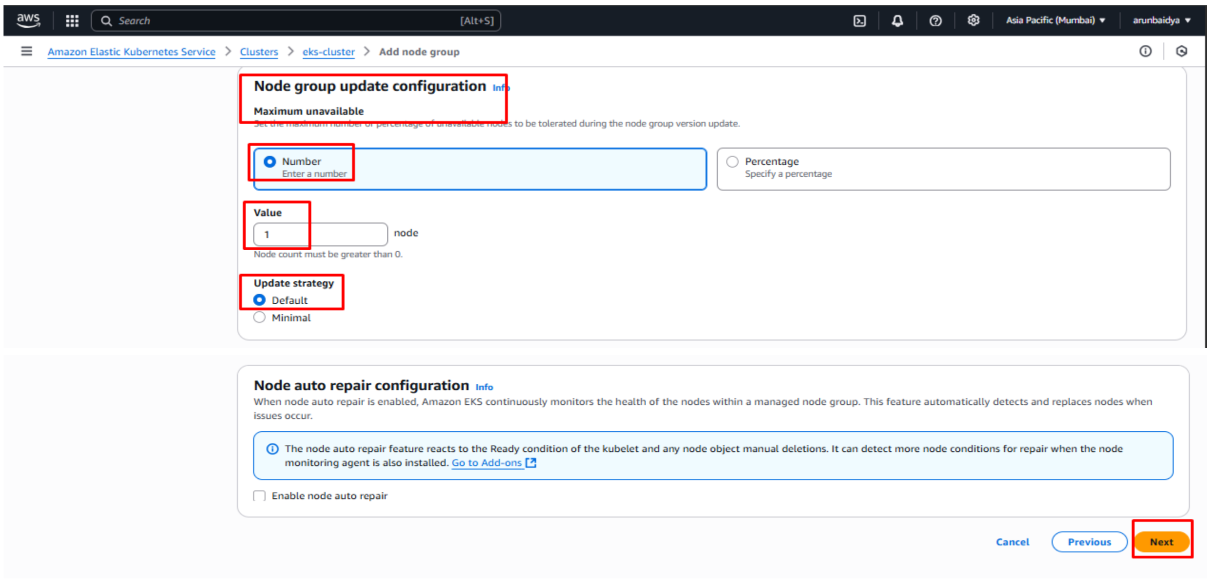
Node group scaling configuration

Desired size
Set the desired number of nodes that the group should launch with initially.
1 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.
1 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.
1 nodes
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size

e. *Node group update configuration -> Set either one and set the Value accordingly => Keep Update strategy "Default" => Next*


Amazon Elastic Kubernetes Service > Clusters > eks-cluster > Add node group

Node group update configuration Info
Maximum unavailable
Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.
[✓ Number] Enter a number
Percentage Specify a percentage

Value
1 node
Node count must be greater than 0.

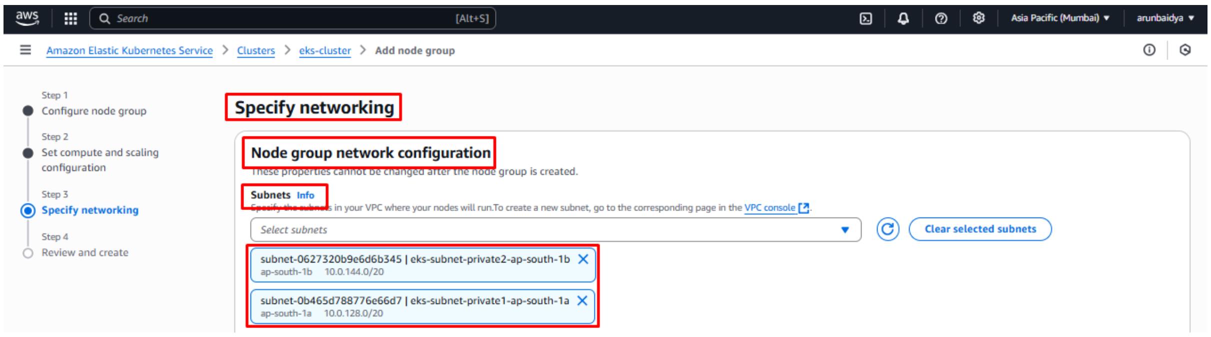
Update strategy
[✓ Default]
Minimal

Node auto repair configuration Info
When node auto repair is enabled, Amazon EKS continuously monitors the health of the nodes within a managed node group. This feature automatically detects and replaces nodes when issues occur.

The node auto repair feature reacts to the Ready condition of the kubelet and any node object manual deletions. It can detect more node conditions for repair when the node monitoring agent is also installed. Go to Add-ons

☐ Enable node auto repair

f. Specify networking => Node group network configuration -> Select only "2 Private Subne"



Specify networking

Node group network configuration

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

Subnets Info

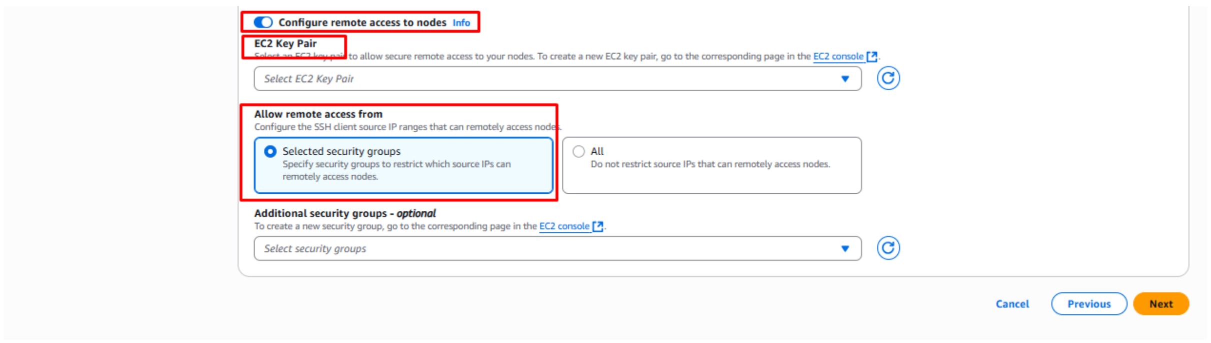
Specify the subnets in your VPC where your nodes will run.To create a new subnet, go to the corresponding page in the VPC console.

Select subnets

subnet-0627320b9e6d6b345 | eks-subnet-private2-ap-south-1b
ap-south-1b 10.0.144.0/20

subnet-0b465d788776e66d7 | eks-subnet-private1-ap-south-1a
ap-south-1a 10.0.128.0/20

g. You can enable "Configure remote access" => Create a EC2 Key=Pair and attach here => Set "Allow remote access from" to "Selected security groups" => Next



Configure remote access to nodes Info

EC2 Key Pair

Select an EC2 key pair to allow secure remote access to your nodes. To create a new EC2 key pair, go to the corresponding page in the EC2 console.

Select EC2 Key Pair

Allow remote access from

Configure the SSH client source IP ranges that can remotely access nodes.

[✓ Selected security groups] Specify security groups to restrict which source IPs can remotely access nodes.

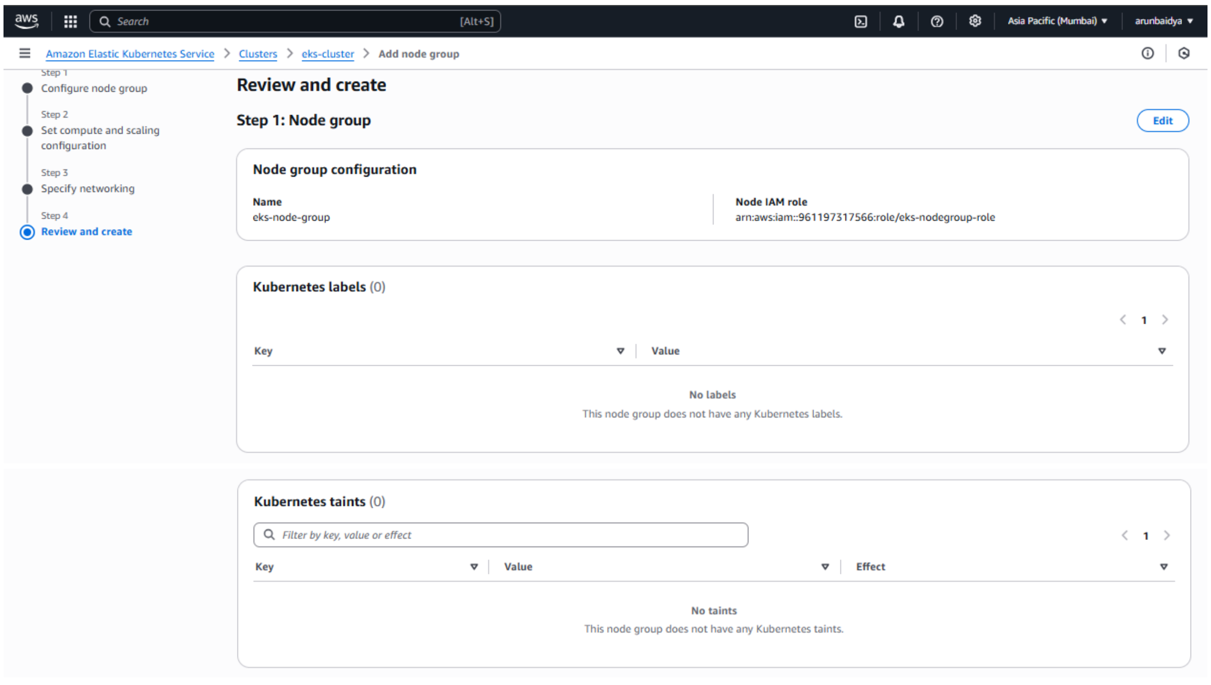
All — Do not restrict source IPs that can remotely access nodes.

Additional security groups - *optional*

To create a new security group, go to the corresponding page in the EC2 console.

Select security groups

10. Review => Click on "Create" => Wait for 5 to 10 minutes to get the EKS Nodegroup Created



Review and create

Step 1: Node group

Node group configuration

Name	Node IAM role
eks-node-group	arn:aws:iam::961197317566:role/eks-nodegroup-role

Kubernetes labels (0)

Key	Value

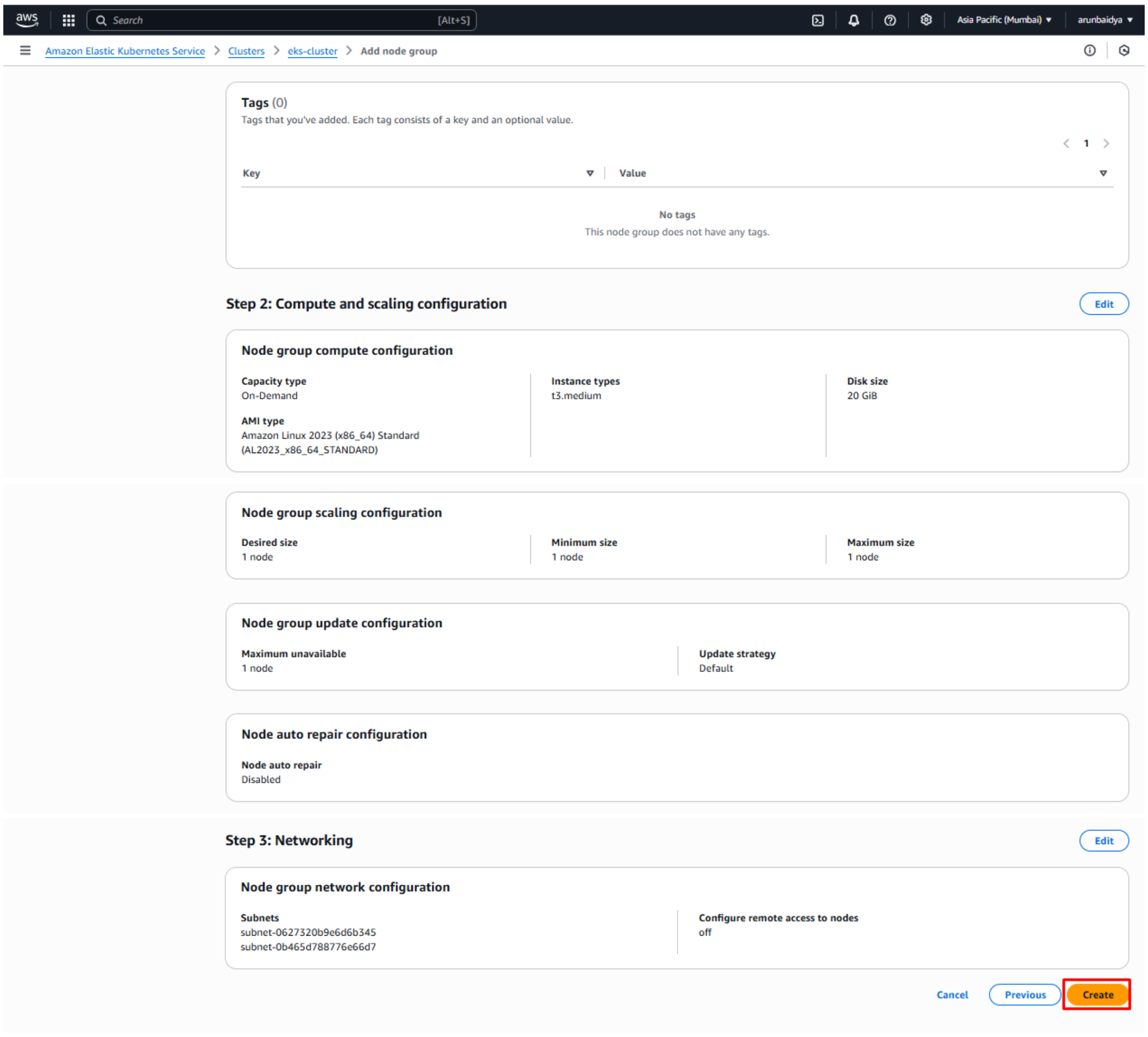
No labels
This node group does not have any Kubernetes labels.

Kubernetes taints (0)

Filter by key, value or effect

Key	Value	Effect

No taints
This node group does not have any Kubernetes taints.



Amazon Elastic Kubernetes Service > Clusters > eks-cluster > Add node group

Tags (0)
Tags that you've added. Each tag consists of a key and an optional value.

Key	Value

No tags
This node group does not have any tags.

Step 2: Compute and scaling configuration

Node group compute configuration

Capacity type	Instance types	Disk size
On-Demand	t3.medium	20 GiB

AMI type
Amazon Linux 2023 (x86_64) Standard (AL2023_x86_64_STANDARD)

Node group scaling configuration

Desired size	Minimum size	Maximum size
1 node	1 node	1 node

Node group update configuration

Maximum unavailable	Update strategy
1 node	Default

Node auto repair configuration

Node auto repair
Disabled

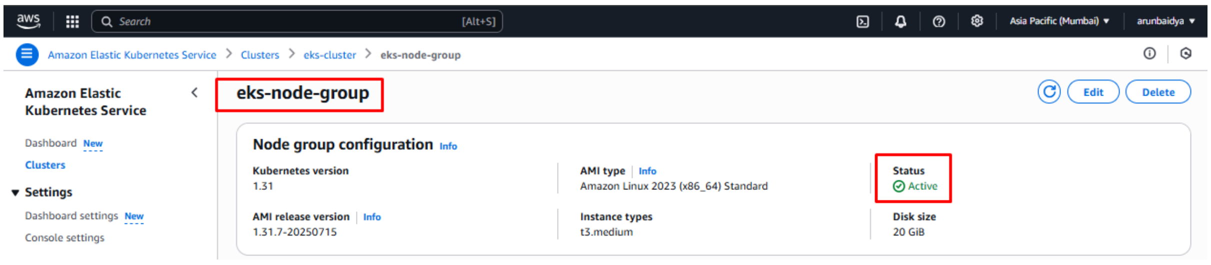
Step 3: Networking

Node group network configuration

Subnets
subnet-0627320b9e6d6b345
subnet-0b465d788776e66d7

Configure remote access to nodes
off

11. After 5 minutes, EKS Nodegroup got created



Amazon Elastic Kubernetes Service > Clusters > eks-cluster > eks-node-group

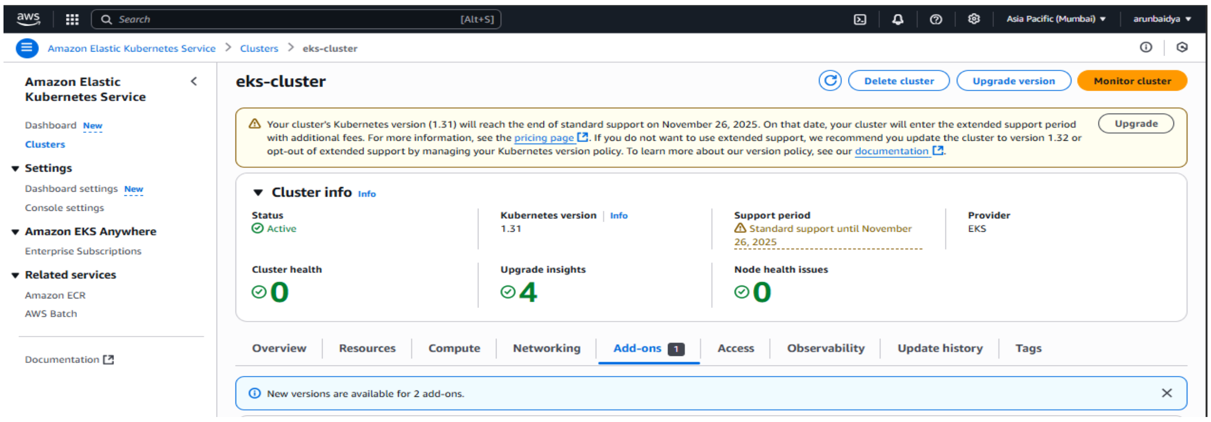
eks-node-group

Node group configuration

Kubernetes version	AMI type	Status
1.31	Amazon Linux 2023 (x86_64) Standard	Active

AMI release version	Instance types	Disk size
1.31.7-20250715	t3.medium	20 GiB

By Arun Baidya



Search [Alt+S] | Asia Pacific (Mumbai) ▾ | arunbaidya ▾

Amazon Elastic Kubernetes Service > Clusters > eks-cluster

Amazon Elastic Kubernetes Service
- Dashboard New
- Clusters
- ▼ Settings
  - Dashboard settings New
  - Console settings
- ▼ Amazon EKS Anywhere
  - Enterprise Subscriptions
- ▼ Related services
  - Amazon ECR
  - AWS Batch
- Documentation

eks-cluster — Delete cluster | Upgrade version | Monitor cluster

⚠ Your cluster's Kubernetes version (1.31) will reach the end of standard support on November 26, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the pricing page. If you do not want to use extended support, we recommend you update the cluster to version 1.32 or opt-out of extended support by managing your Kubernetes version policy. To learn more about our version policy, see our documentation. — Upgrade

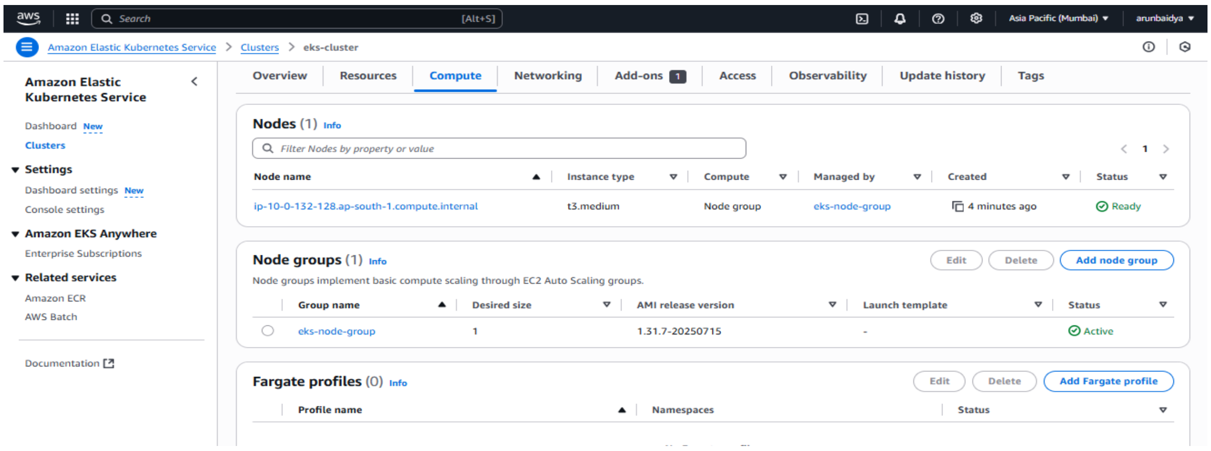
▼ **Cluster info** Info

Status	Kubernetes version Info	Support period	Provider
✓ Active	1.31	⚠ Standard support until November 26, 2025	EKS

Cluster health	Upgrade insights	Node health issues
✓ 0	✓ 4	✓ 0

Overview | Resources | Compute | Networking | Add-ons 1 | Access | Observability | Update history | Tags

ⓘ New versions are available for 2 add-ons.



Search [Alt+S] | Asia Pacific (Mumbai) ▾ | arunbaidya ▾

Amazon Elastic Kubernetes Service > Clusters > eks-cluster

Amazon Elastic Kubernetes Service
- Dashboard New
- Clusters
- ▼ Settings
  - Dashboard settings New
  - Console settings
- ▼ Amazon EKS Anywhere
  - Enterprise Subscriptions
- ▼ Related services
  - Amazon ECR
  - AWS Batch
- Documentation

Overview | Resources | Compute | Networking | Add-ons 1 | Access | Observability | Update history | Tags

Nodes (1) Info

Filter Nodes by property or value

Node name	Instance type	Compute	Managed by	Created	Status
ip-10-0-132-128.ap-south-1.compute.internal	t3.medium	Node group	eks-node-group	4 minutes ago	✓ Ready

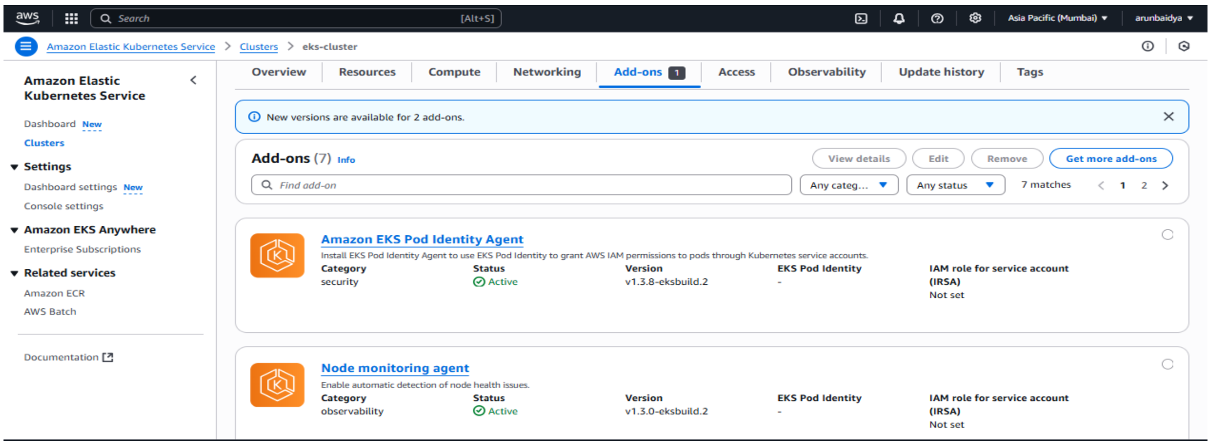
Node groups (1) Info — Edit | Delete | Add node group

Node groups implement basic compute scaling through EC2 Auto Scaling groups.

	Group name	Desired size	AMI release version	Launch template	Status
○	eks-node-group	1	1.31.7-20250715	-	✓ Active

Fargate profiles (0) Info — Edit | Delete | Add Fargate profile

Profile name	Namespaces	Status



Search [Alt+S] | Asia Pacific (Mumbai) ▾ | arunbaidya ▾

Amazon Elastic Kubernetes Service > Clusters > eks-cluster

Amazon Elastic Kubernetes Service
- Dashboard New
- Clusters
- ▼ Settings
  - Dashboard settings New
  - Console settings
- ▼ Amazon EKS Anywhere
  - Enterprise Subscriptions
- ▼ Related services
  - Amazon ECR
  - AWS Batch
- Documentation

Overview | Resources | Compute | Networking | Add-ons 1 | Access | Observability | Update history | Tags

ⓘ New versions are available for 2 add-ons.

Add-ons (7) Info — View details | Edit | Remove | Get more add-ons

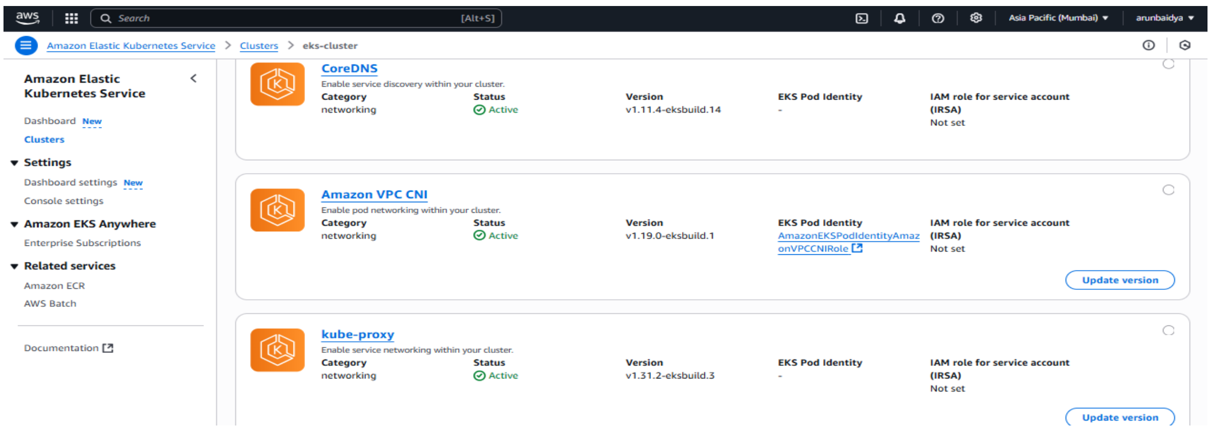
Find add-on | Any categ… ▾ | Any status ▾ | 7 matches | 1 2

Amazon EKS Pod Identity Agent
Install EKS Pod Identity Agent to use EKS Pod Identity to grant AWS IAM permissions to pods through Kubernetes service accounts.

Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
security	✓ Active	v1.3.8-eksbuild.2	-	Not set

Node monitoring agent
Enable automatic detection of node health issues.

Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
observability	✓ Active	v1.3.0-eksbuild.2	-	Not set



Search [Alt+S] | Asia Pacific (Mumbai) ▾ | arunbaidya ▾

Amazon Elastic Kubernetes Service > Clusters > eks-cluster

Amazon Elastic Kubernetes Service
- Dashboard New
- Clusters
- ▼ Settings
  - Dashboard settings New
  - Console settings
- ▼ Amazon EKS Anywhere
  - Enterprise Subscriptions
- ▼ Related services
  - Amazon ECR
  - AWS Batch
- Documentation

CoreDNS
Enable service discovery within your cluster.

Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
networking	✓ Active	v1.11.4-eksbuild.14	-	Not set

Amazon VPC CNI
Enable pod networking within your cluster.

Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
networking	✓ Active	v1.19.0-eksbuild.1	AmazonEKSPodIdentityAmazonVPCCNIRole	Not set

Update version

kube-proxy
Enable service networking within your cluster.

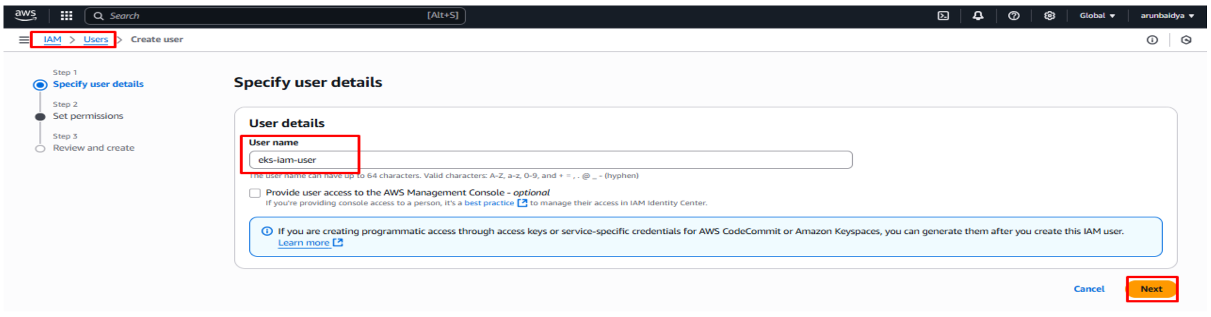
Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
networking	✓ Active	v1.31.2-eksbuild.3	-	Not set

Update version

12. Accessing the EKS Cluster

Step 12.1: Create an IAM User and Create Access Keys

a. *Go to IAM => Users => Create users => User name (eks-iam-user) => Next => Select Attach policies directly => Without adding any policy, do Next => Create user.*


IAM > Users > Create user

Step 1 Specify user details
Step 2 Set permissions
Step 3 Review and create

Specify user details

User details

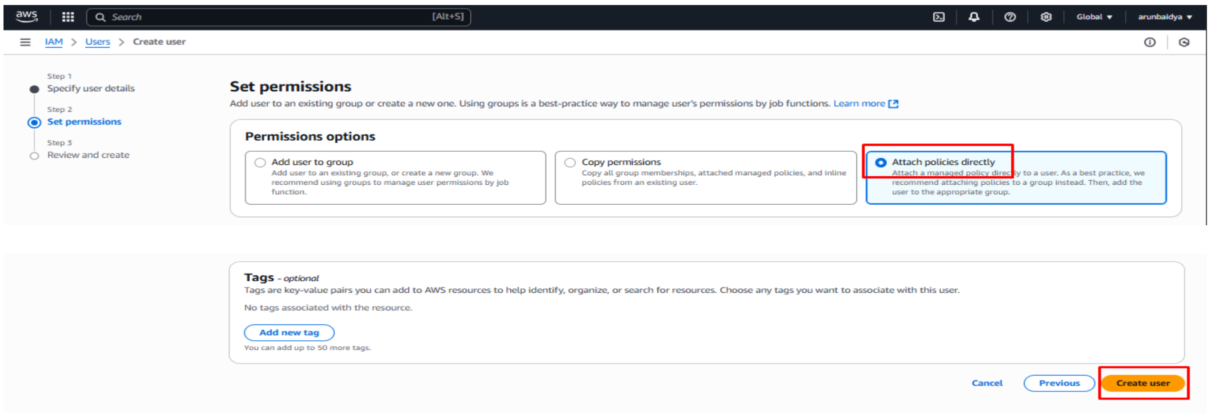
User name: eks-iam-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

Cancel | Next


IAM > Users > Create user

Step 1 Specify user details
Step 2 Set permissions
Step 3 Review and create

Set 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 all group memberships, attached managed policies, and inline policies 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.

Tags - *optional*

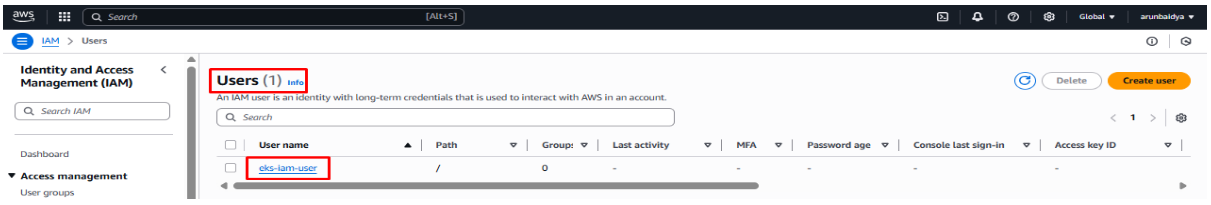
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel | Previous | Create user

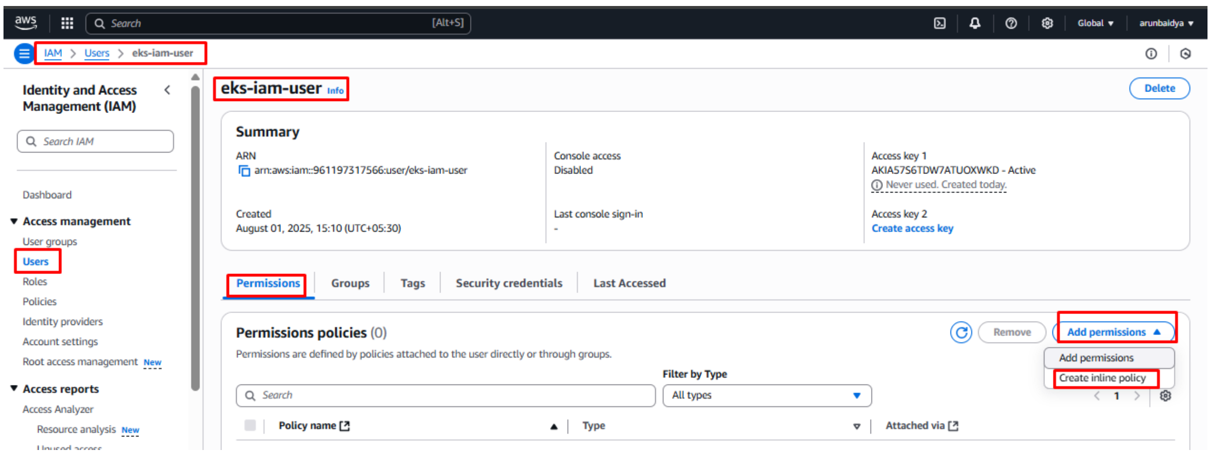

IAM > Users

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
eks-iam-user	/	0	-	-	-	-	-

b. *Create* ***“inline policy”*** *for the create IAM user. Click on IAM user => Permissions => Add permissions => Select Create inline policy*


IAM > Users > eks-iam-user

eks-iam-user Info

Summary

ARN arn:aws:iam::961197317566:user/eks-iam-user	Console access Disabled	Access key 1 AKIA57S6TDW7ATUOXWKD - Active Never used. Created today.
Created August 01, 2025, 15:10 (UTC+05:30)	Last console sign-in -	Access key 2 Create access key

Permissions | Groups | Tags | Security credentials | Last Accessed

Permissions policies (0)

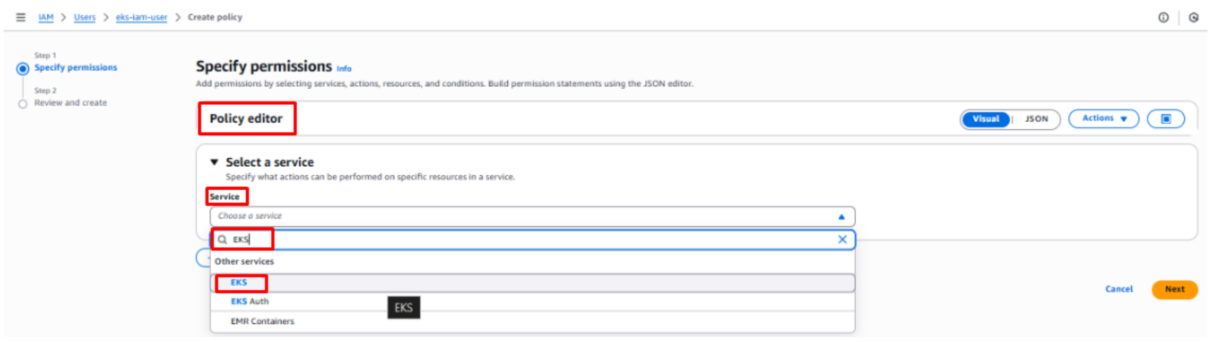
Permissions are defined by policies attached to the user directly or through groups.

Add permissions ▲
- Add permissions
- Create inline policy

Filter by Type: All types

Policy name	Type	Attached via

c. *Select Service "EKS" => Actions allowed => Access Level => Select "All read actions" => Select "All write actions" => Resources => Select "All" => Next => Give a policy name (eks-iam-user-policy) => Create policy*



IAM > Users > eks-iam-user > Create policy

Step 1 Specify permissions
Step 2 Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Select a service

Specify what actions can be performed on specific resources in a service.

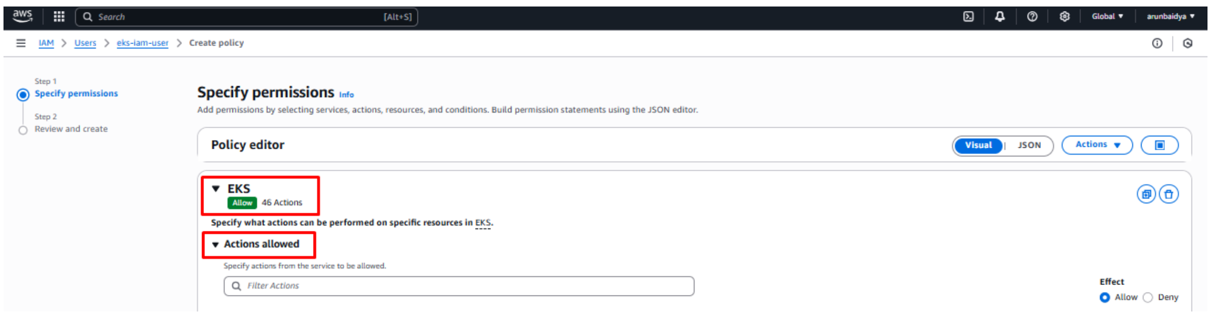
Service

Choose a service

EKS

Other services

EKS
EKS Auth
EMR Containers



IAM > Users > eks-iam-user > Create policy

Step 1 Specify permissions
Step 2 Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

EKS
Allow 46 Actions

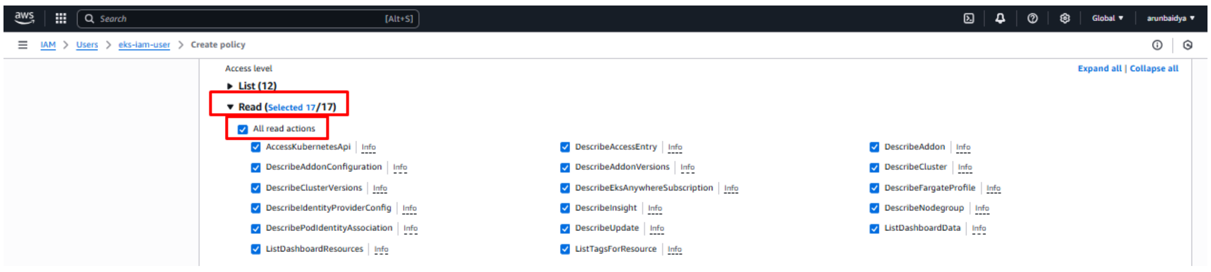
Specify what actions can be performed on specific resources in EKS.

Actions allowed

Specify actions from the service to be allowed.

Filter Actions

Effect: [✓ Allow] Deny



IAM > Users > eks-iam-user > Create policy

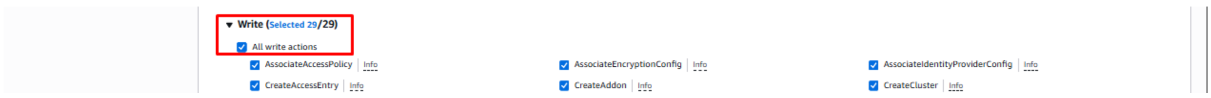
Access level

List (12)

Read (Selected 17/17)

- [x] All read actions

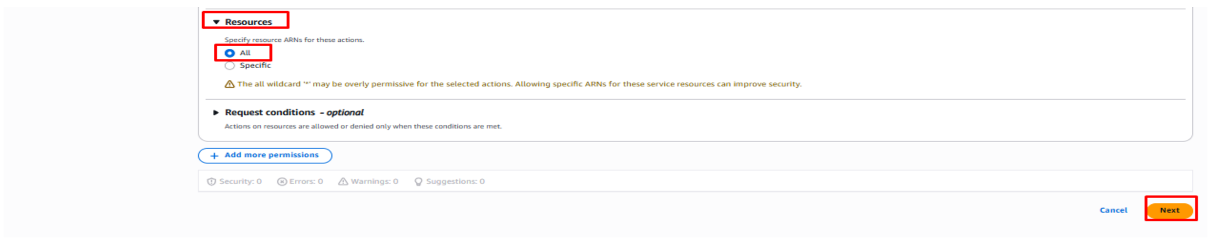
☑ AccessKubernetesApi	☑ DescribeAccessEntry	☑ DescribeAddon
☑ DescribeAddonConfiguration	☑ DescribeAddonVersions	☑ DescribeCluster
☑ DescribeClusterVersions	☑ DescribeEksAnywhereSubscription	☑ DescribeFargateProfile
☑ DescribeIdentityProviderConfig	☑ DescribeInsight	☑ DescribeNodegroup
☑ DescribePodIdentityAssociation	☑ DescribeUpdate	☑ ListDashboardData
☑ ListDashboardResources	☑ ListTagsForResource	



Write (Selected 29/29)

- [x] All write actions

☑ AssociateAccessPolicy	☑ AssociateEncryptionConfig	☑ AssociateIdentityProviderConfig
☑ CreateAccessEntry	☑ CreateAddon	☑ CreateCluster



Resources

Specify resource ARNs for these actions.

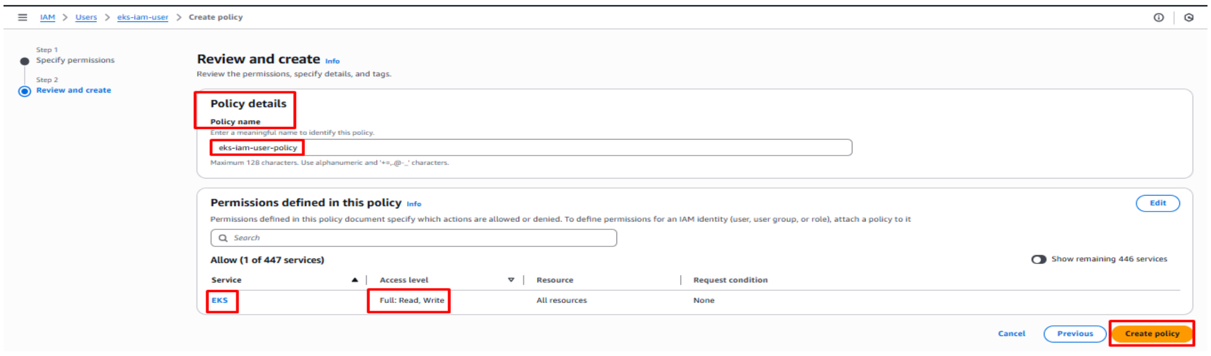
[✓ All] Specific

The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.

Request conditions - *optional*

Actions on resources are allowed or denied only when these conditions are met.

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0



IAM > Users > eks-iam-user > Create policy

Step 1 Specify permissions
Step 2 Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

eks-iam-user-policy

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

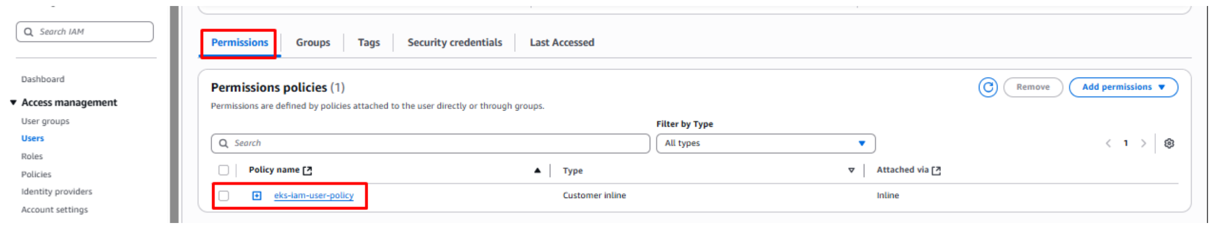
Permissions defined in this policy Info

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

Allow (1 of 447 services)

Show remaining 446 services

Service	Access level	Resource	Request condition
EKS	Full: Read, Write	All resources	None

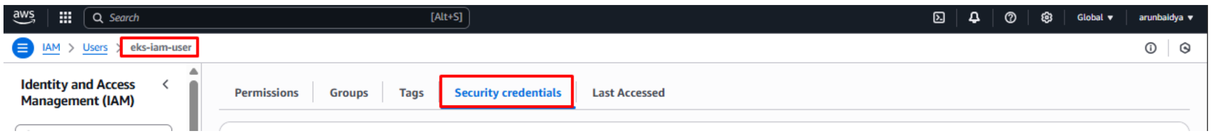


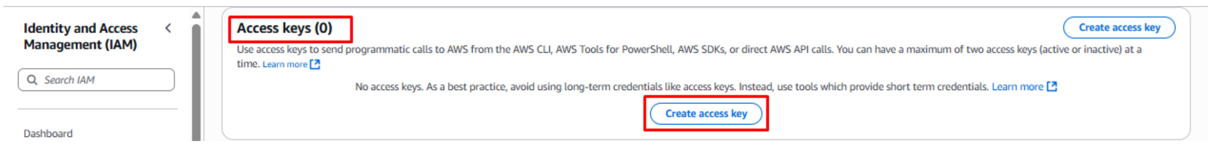
Permissions policies (1)

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

Policy name	Type	Attached via
eks-iam-user-policy	Customer inline	Inline

d. *Go to IAM user => Security Credentials => Access keys => Create access key => Use case => Select "Command Line Interface (CLI)" => Confirm => Next => Create access key => Download csv file => Done*

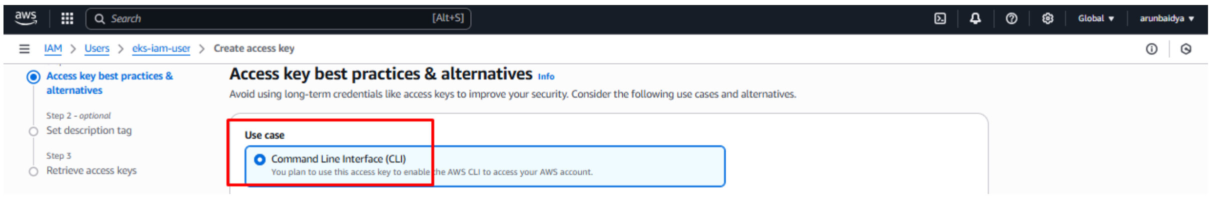




Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more



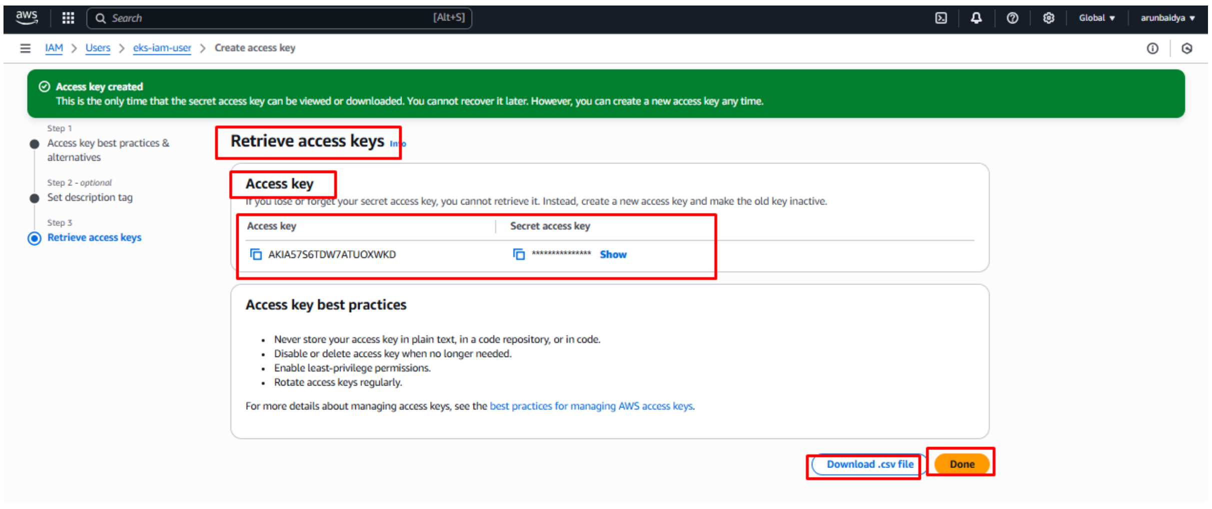
Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.



Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA57S6TDW7ATUOXWKD	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Step 12.2: Configure the AWS CLI and Access Key & Secret Key

a. *Now open Laptop command prompt or Git Bash Terminal => Go to Download folder => Configure the credentials*

```
$ cd Download
$ aws configure
```

- Access Key ID: AKIA57S6TDW7ATUOXWKD
- Secret Access Key: jJGIvJLDswITlboLbJMVV01X1RhqzvkBLxJ1hoNq
- Default region (e.g., us-east-1): ap-south-1
- Output format (e.g., json, text, or table): Press Enter for none

b. Validate the AWS CLI configuration and check the version

$ aws configure list

$ cat ~/.aws/credentials

$ cat ~/.aws/config

OR

$ aws s3 ls

$ aws --version

Step 12.3: Accessing the EKS Cluster from Local machine (Laptop)

a. To connect EKS Cluster from your Local Machine Terminal, we need to add kubeconfig for kubectl. So, to add and connect, run below command. As we did not configure the Access keys yet on the local machine, so, it will show unable to locate credentials.

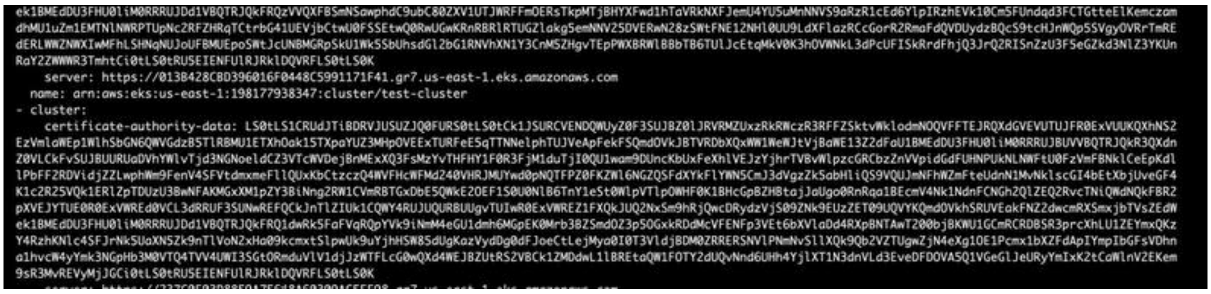
Syntax:

$ aws eks update-kubeconfig --region <region_name> --name <eks_cluster_name>

$ aws eks update-kubeconfig --region ap-south-1 --name eks-cluster

If, all configurations are okay then above command will show the user and path where ***config*** *file is available.*

$ cat /Users/<user_name>/.kube/config



```
ek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQzVVQXFBSmNSawphdC9ubC80ZXV1UTJWRFFmOERsTkpMTjBHYXFwd1hTaVRkNXFJemU4YU5uMnNNVS9aRzR1cEd6YlpIRzhEVk10Cm5FUndqd3FCTGtteElKemczam
dhMU1uZm1EMTNlNWRPTUpNc2RFZHRqTCtrbG41UEVjbCtwU0FSSEtwQ0RwUGwKRnRBR1RTUGZlakg5emNNV25DVERwN28zSWtFNE12NHl0UU9LdXFlazRCcGorR2RmaFdQVDUydzBQcS9tcHJnWQp5SVgyOVRrTmRE
dERLWWZNWXIwMFhLSHNqNUJoUFBMUEpoSWtJcUNBMGRpSkU1Wk5SbUhsdGl2bG1RNVhXN1Y3CnM5ZHgvTEpPWXBRWlBBbTB6TUlJcEtqMkV0K3hOVWNkL3dPcUFISkRrdFhjQ3JrQ2RISnZzU3F5eGZkd3NlZ3YKUn
RaY2ZWWWR3TmhtCi0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
    server: https://013B428CBD396016F0448C5991171F41.gr7.us-east-1.eks.amazonaws.com
  name: arn:aws:eks:us-east-1:198177938347:cluster/test-cluster
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJRVRMZUxzRkRWczR3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2
EzVmlaWEp1WlhSbGN6QWVGdzB5TlRBMU1ETXhOak15TXpaYUZ3MHpOVEExTURFeE5qTTNNelphTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdn
Z0VLCkFvSUJBUURUaDVhYWlvTjd3NGNoeldCZ3VTcWVDejBnMExXQ3FsMzYvTHFHY1F0R3FjM1duTjI0QU1wam9DUncKbUxFeXhlVEJzYjhrTVBvWlpzcGRCbzZnVVpidGdFUHNPUkNLNWFtU0FzVmFBNklCeEpKdl
lPbFF2RDVidjZZLwphWm9FenV4SFVtdmxmeFllQUxKbCtzczQ4WVFHcWFMd240VHRJMUYwd0pNQTFPZ0FKZWl6NGZQSFdXYkFlYWN5CmJ3dVgzZkSabHliQS9VQUJmNFhWZmFteUdnN1MvNklscGI4bEtXbjUveGF4
K1c2R25VQk1ERlZpTDUzU3BwNFAKMGxXM1pZY3BiNng2RW1CVmRBTGxDbE5QWkE2OEF1S0U0NlB6TnY1eSt0WlpVTlpOWHF0K1BHcGpBZHBtajJaUgo0RnRqa1BEcmV4Nk1NdnFCNGh2QlZEQ2RvcTNiQWdNQkFBR2
pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQQkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJUQ2NxSm9hRjQwcDRydzVjS09ZNk9EUzZET09UQVYKQmd0VkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdW
ek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQ1dwRk5FaFVqRQpYVk9iNmM4eGU1dmh6MGpEK0Mrb3BZSmdOZ3p5OGxkRDdMcVFFNFp3VEt6bXVlaDd4RXpBNTAwT200bjBKWU1GCmRCRDBSR3ByaXhLU1ZEYmxQKz
Y4RzhKNlc4SFJrNk5UaXN5Zk9nTlVoN2xHa09kcmxtSlpwUk9uYjhHSW85dUgKazVydDg0dFJoeCtLejMya0I0T3VldjBDM0ZRRERSNVlPNmNvSllXQk9Qb2VZTUgwZjN4eXg1OE1Pcmx1bXZFdApIYmpIbGFsVDhn
a1hvcW4yYmk3NGpHb3M0VTQ4TVV4UWI3SGtORmduVlV1djJzWTFLcG0wQXd4WEJBZUtRS2VBCk1ZMDdwL1lBREtaQW1FOTY2dUQvNnd6UHh4YjlXT1N3dnVLd3EveDFDOVA5Q1VGeGlJeURyYmIxK2tCaWlnV2EKem
9sR3MvREVyMjJGCi0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
```

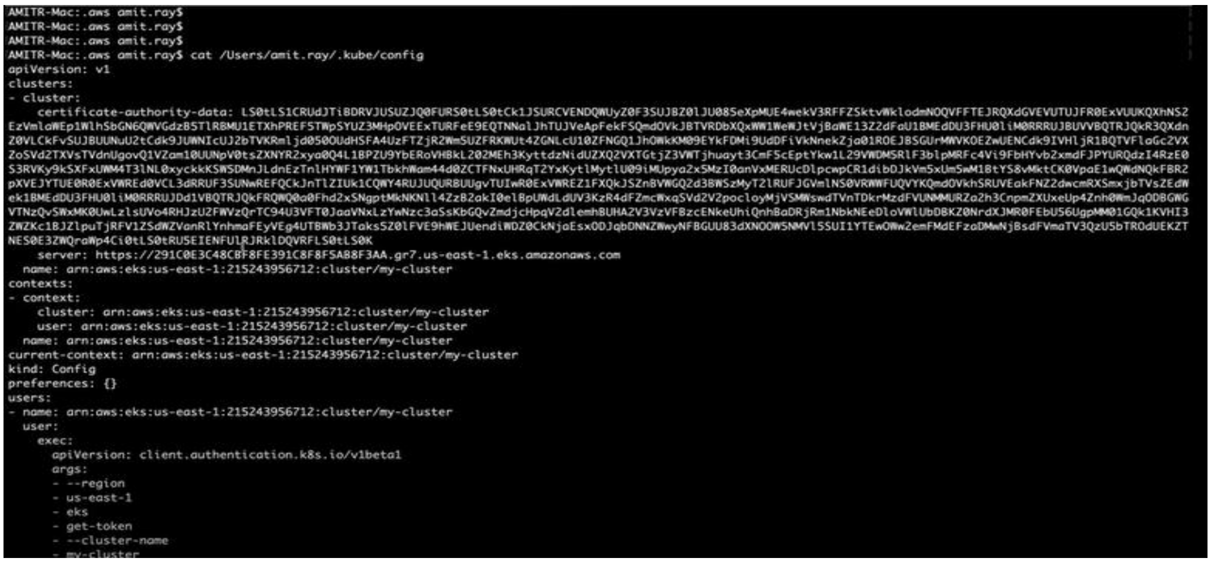
If this file show more than 1 credentials, (in above case, it's showing 3 credentials). So, delete the file and run the command again.

$ rm -f /Users/<user_name>/.kube/config

$ $ aws eks update-kubeconfig --region ap-south-1 --name eks-cluster

$ cat /Users/<user_name>/.kube/config

Now, this file will show only one credential



```
AMITR-Mac:.aws amit.ray$
AMITR-Mac:.aws amit.ray$
AMITR-Mac:.aws amit.ray$
AMITR-Mac:.aws amit.ray$ cat /Users/amit.ray/.kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJU08SeXpMUE4wekV3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2
EzVmlaWEp1WlhSbGN6QWVGdzB5TTlRBMU1ETXhPREF5TWpSYUZ3MHpOVEExTURFeE9EQTNNalJhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdn
Z0VLCkFvSUJBUUNuU2tCdk9JUWNIcUJ2bTVKRmljd050OUdHSFA4UzFTZjR2WmSUZFRKWUt4ZGNLcU10ZFNGQ1JhOWkKM09EYkFDMi9UdDFiVkNnekZja01ROEJBSGUrMWVKOEZwUENCdk9IVHljR1BQTVFlaGc2VX
ZoSVd2TXVsTVdnUgovQ1VZam10UUNpV0tsZXNYR2xya0Q4L1BPZU9YbERoVHBkL202MEh3KyttdzNidUZXQ2VXTGtjZ3VWTjhuayt3CmF5cEptYkw1L29VWDMSRlF3blpMRFc4Vi9FbHYvb2xmdFJPYURQdzI4RzE0
S3RVKy9kSXFxUWM4T3lNL0xyckkKSWSDMnJLdnEzTnlHYWF1YW1TbkhWam44d0ZCTFNxUHRqT2YxKytlMytlU09iMUpya2x5MzI0anVxMERUcDlpcwpCR1dibDJkVm5xUm5wM1BtYS8vMktCK0VpaE1wQWdNQkFBR2
pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQQkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJSZnBVWGQ2d3BWSzMyT2lRUFJGVmlNS0VRWWFUQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdW
ek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQWQ0a0Fhd2xSNgptMkNKNll4ZzB2akI0elBpUWdLdUV3KzR4dFZmcWxqSVd2V2pocloyMjVSMWswdTVnTDkrMzdFVUNMMURZa2h3CnpmZXUxeUp4Znh0WmJqODBGWG
VTNzQvSWxMK0UwLzlsUVo4RHJzU2FWVzQrTC94U3VFT0JaaVNxLzYwNzc3aSsKbGQvZmdjcHpqV2dlemhBUHA2V3VzVFBzcENkeUhiQnhBaDRjRm1NbkNEeDloVWlUbDBKZ0NrdXJMR0FEbU56UgpMM01GQk1KVHI3
ZWZKc1BJZlpuTjRFV1ZSdWZVanRlYnhmaFEyVEg4UTBWb3JTaksSZ0lFVE9hWEJUendiWDZ0CkNjaEsxODJqbDNNZWwyNFBGUU83dXNOOW5NMVl5SUI1YTEwOWw2emFMdEFzaDMwNjBsdFVmaTV3QzU5bTROdUEKZT
NES0E3ZWQraWp4Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
    server: https://291C0E3C48CBF8FE391C8F8F5A88F3AA.gr7.us-east-1.eks.amazonaws.com
  name: arn:aws:eks:us-east-1:215243956712:cluster/my-cluster
contexts:
- context:
    cluster: arn:aws:eks:us-east-1:215243956712:cluster/my-cluster
    user: arn:aws:eks:us-east-1:215243956712:cluster/my-cluster
  name: arn:aws:eks:us-east-1:215243956712:cluster/my-cluster
current-context: arn:aws:eks:us-east-1:215243956712:cluster/my-cluster
kind: Config
preferences: {}
users:
- name: arn:aws:eks:us-east-1:215243956712:cluster/my-cluster
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
      - --region
      - us-east-1
      - eks
      - get-token
      - --cluster-name
      - my-cluster
```

Step 12.4: Now Install eksctl CLI tool

[Step 2: Configure your computer to communicate with your cluster *https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html]*

a. *We need to install eksctl on our local machine, to simplify creating and managing EKS clusters.*

# Download and extract the latest release

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

# Move the extracted binary to /usr/local/bin

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

# Test that your eksclt installation was successful

```
$ eksctl version
```

Step 12.5: Now Install kubectl CLI tool

a. *We need to install kubectl, the command-line tool for interacting with your Kubernetes (EKS) cluster.*

    # Download and extract the latest release

    $ curl -O curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.11/2025-04-17/bin/linux/amd64/kubectl

    OR

    $ curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl

    $ ll

    # Change the permission to make the file executable

    $ sudo chmod +x kubectl

    # Move the extracted binary to /usr/local/bin

    $ sudo mv kubectl /usr/local/bin/

    $ sudo echo $PATH

    # Check the version to make sure that the kubectl install successfully

    $ kubectl version

    $ kubectl version --client

Step 12.6: If yet not then in the Local laptop / machine, configure the Credential of AWS user who created this EKS Cluster

a. *Follow the same steps to configure Access Key and Secrete Keys on your Laptop for the AWS user*
b. *Now run below command once more and check the nodes*

    $ aws eks update-kubeconfig --region ap-south-1 --name eks-cluster
    $ kubectl get nodes

    If this command returns Node list, then configuration is successful

    $ kubectl run pod1 --name nginx

Troubleshoot problems with Amazon EKS clusters and nodes

https://docs.aws.amazon.com/eks/latest/userguide/troubleshooting.html

Troubleshooting

1. **Node Group Fails to Create** IAM Permissions: Ensure the node role has AmazonEKSWorkerNodePolicy, AmazonEC2ContainerRegistryReadOnly, and AmazonEKS_CNI_Policy.

Subnets: Use private subnets if nodes don't need public IPs.

2. Nodes Not Joining the Cluster Check aws-auth ConfigMap (auto-created for managed node groups):

```
$ kubectl describe configmap aws-auth -n kube-system
```

Security Groups: Ensure nodes can communicate with the EKS API (port 443).

3. SSH Access Issues Ensure the key pair (my-keypair) exists in your AWS region.

Troubleshooting: Fixing "Unauthorized" Errors

Common Causes:

1. Incorrect IAM Trust Policies:

- **Cluster Role:** Must trust eks.amazonaws.com, **not** ec2.amazonaws.com.

```
# Check the trust policy
$ aws iam get-role --role-name EKSClusterRole --query "Role.AssumeRolePolicyDocument"
```

- **Node Role**: Must trust ec2.amazonaws.com.

2. Missing aws-auth ConfigMap Entries:

```
$ kubectl describe configmap aws-auth -n kube-system
```

- Ensure the node role ARN matches your IAM role.

3. Security Group Misconfigurations:

- **Worker Nodes**: Allow **outbound** traffic to the EKS API (port 443).
- **Control Plane**: Allow **inbound** traffic from worker node security groups.

```
# Example: Allow inbound traffic from worker SG
aws ec2 authorize-security-group-ingress \
  --group-id sg-controlplane \
  --protocol tcp \
  --port 443 \
  --source-group sg-worker
```

4. Terminate and Replace Nodes:

```
# Force ASG to launch new instances
aws autoscaling terminate-instance-in-auto-scaling-group \
  --instance-id i-1234567890abcdef0 \
  --should-decrement-desired-capacity
```

==

Ref: https://www.youtube.com/watch?v=VSGyxi-Vuac&list=PLTnw6NC76Hn6mrLb54nSoPjhzBuT2qdoy&index=3

Ref for Bastion Node: https://www.youtube.com/watch?v=XWaLU0alrvY