
Create Elastic Kubernetes Service (EKS) Cluster on AWS

[Edition 13]

[Last Update 210421]

For any issues/help contact : support@k21academy.com

K21 Academy

Contents

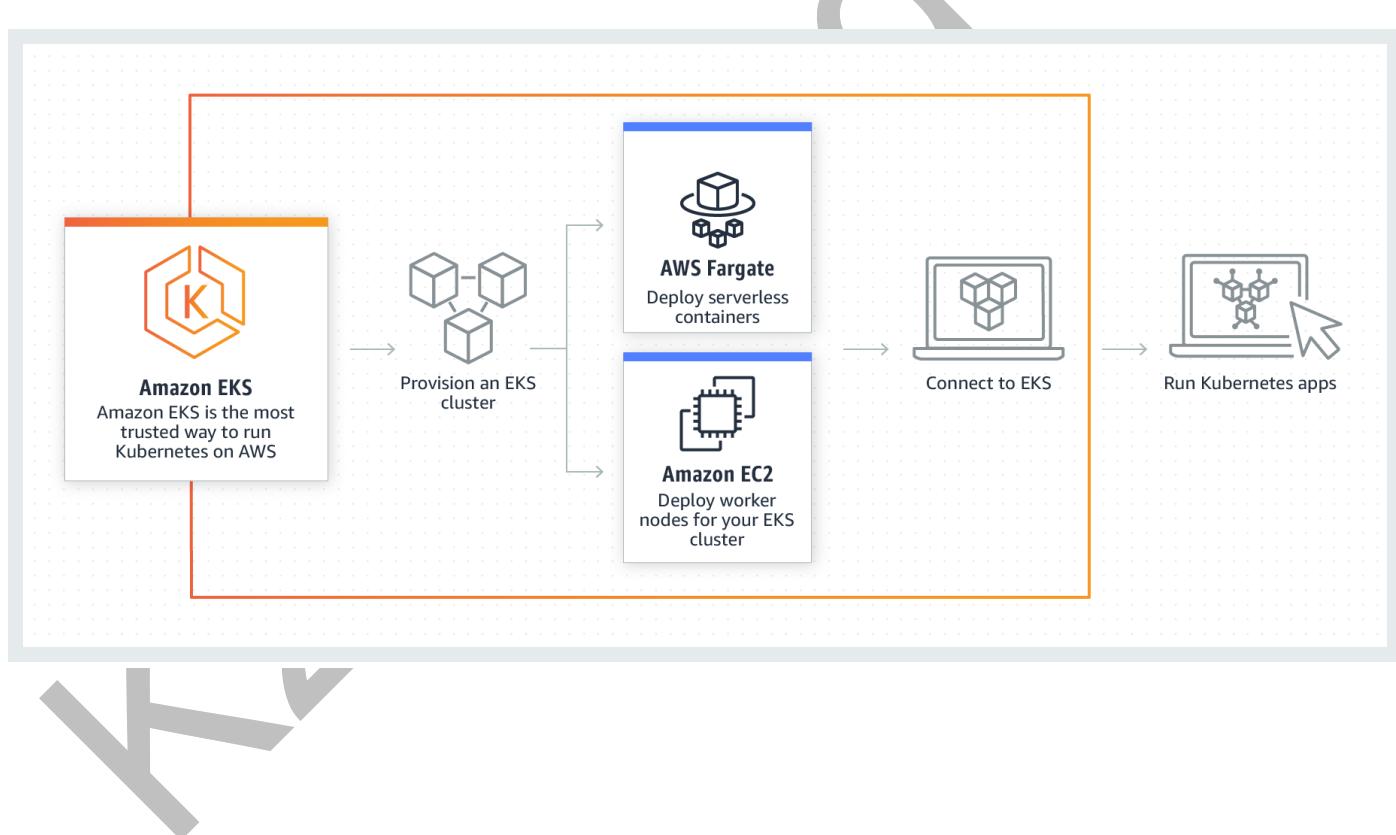
1	Introduction	3
2	Documentation.....	4
2.1	EKS Documentation.....	4
3	Pre-requisite.....	5
4	Setup EKS Cluster Master Node Using Console.....	6
4.1	Login to AWS portal	6
4.2	Create Kubernetes Cluster.....	6
4.3	Creating an EKS role	7
4.4	Back to Cluster Configuration page.....	11
4.5	Review and Create the Cluster	16
5	Configure kubectl on Client	18
5.1	Setup up the kubectl configuration	18
6	Create EKS Cluster Worker Nodes Using Console	19
6.1	Setup Worker Nodes in EKS	19
6.2	Verifying Worker node status from Kubectl.....	31
7	Delete EKS Cluster from Console	32
7.1	Delete Worker Nodes	32
7.2	Delete Master Node	33
7.3	Delete Key Pair	34
7.4	Delete IAM Roles	35
8	Setup EKS Cluster (Master & Worker Node) Using EKSCTL.....	36
8.1	Create EKS Master & Worker Nodes	36
8.2	Verify Master & Worker Node from Console.....	37
8.3	Setup up the kubectl configuration	40
8.4	Delete EKS Cluster using EKSCTL	41
9	Troubleshooting Section.....	43
9.1	CloudFormation Stack Already Exist	43
10	Summary.....	44

1 INTRODUCTION

EKS (Elastic Container Service for Kubernetes) is a managed Kubernetes service that allows you to run Kubernetes on AWS without the hassle of managing the Kubernetes control plane (Master Node).

The Kubernetes control plane plays a crucial role in a Kubernetes deployment as it is responsible for how Kubernetes communicates with your cluster — starting and stopping new containers, scheduling containers, performing health checks, and many more management tasks.

The big benefit of EKS, and other similar hosted Kubernetes services, is taking away the operational burden involved in running this control plane. You deploy cluster worker nodes using AWS Console or CLI, and EKS will provision, scale and manage the Kubernetes control plane for you to ensure high availability, security and scalability.



2 DOCUMENTATION

2.1 EKS Documentation

1. Create EKS Cluster using Console or CLI
<https://docs.aws.amazon.com/eks/latest/userguide/create-cluster.html>

K21 Academy

3 PRE-REQUSITE

- **Note:** Create account (Trial or Paid) on AWS Cloud.
*Follow Activity Guide Register_for_AWS_Free_Tier_Account_And_Login_to_AWS_Console_ed** from portal*

- **Note:** In this guide we are going using Linux machine to perfrom this lab you can provision Linux machine on Amazon.

Follow *Activity Guide Create_&_Connect_To_EC2_Linux_Instance_ed** from Portal* to create Linux machine on AWS.

- **Note:** To install & configure AWSCLI, and to install EKSCTL and kubectl.

Follow guide Activity Guide:

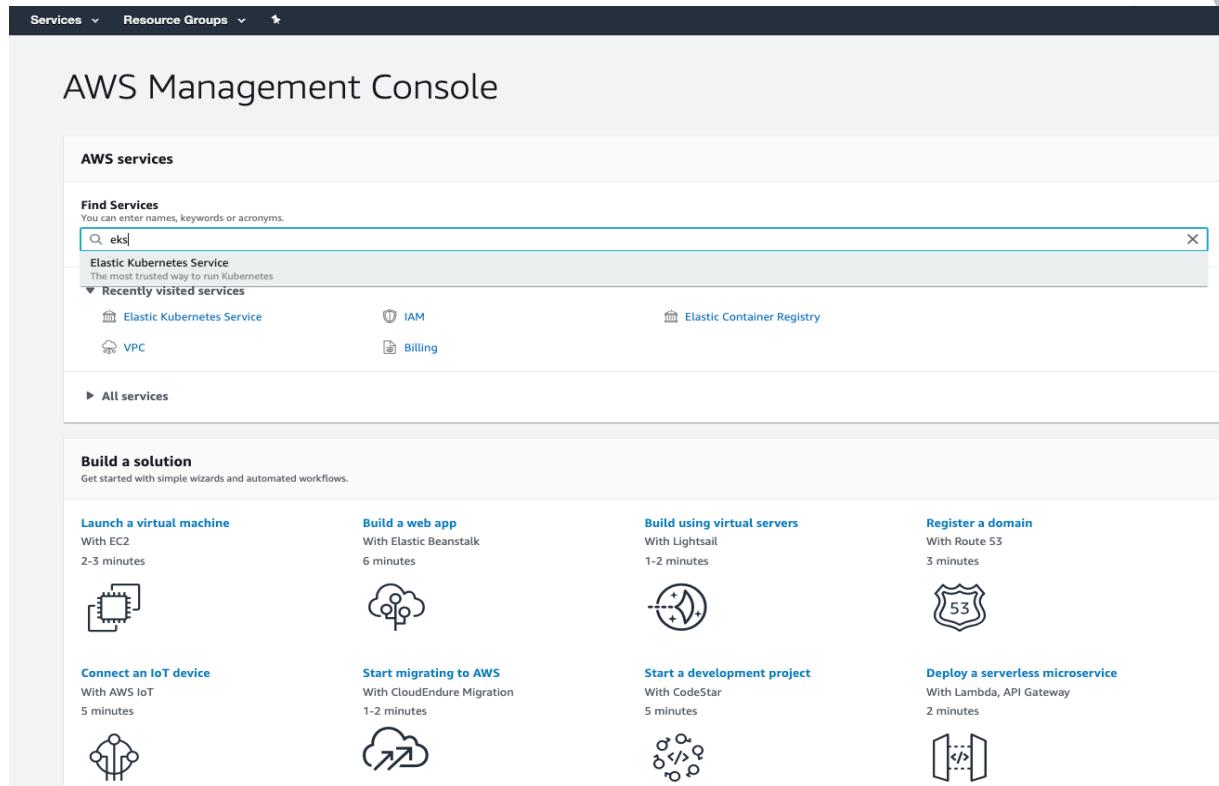
*AG_Install_Configure_AWSCLI_KUBECTL_EKSCTL_on_Linux_ed** from Portal*

- **Note:** Stop the EKS Cluster after the usage. When you stop cluster, it is shut down and you are not billed for hourly usage. If you don't stop the cluster you will be highly charged. If you decide that you no longer need an instance, you can terminate it.

4 SETUP EKS CLUSTER MASTER NODE USING CONSOLE

4.1 Login to AWS portal

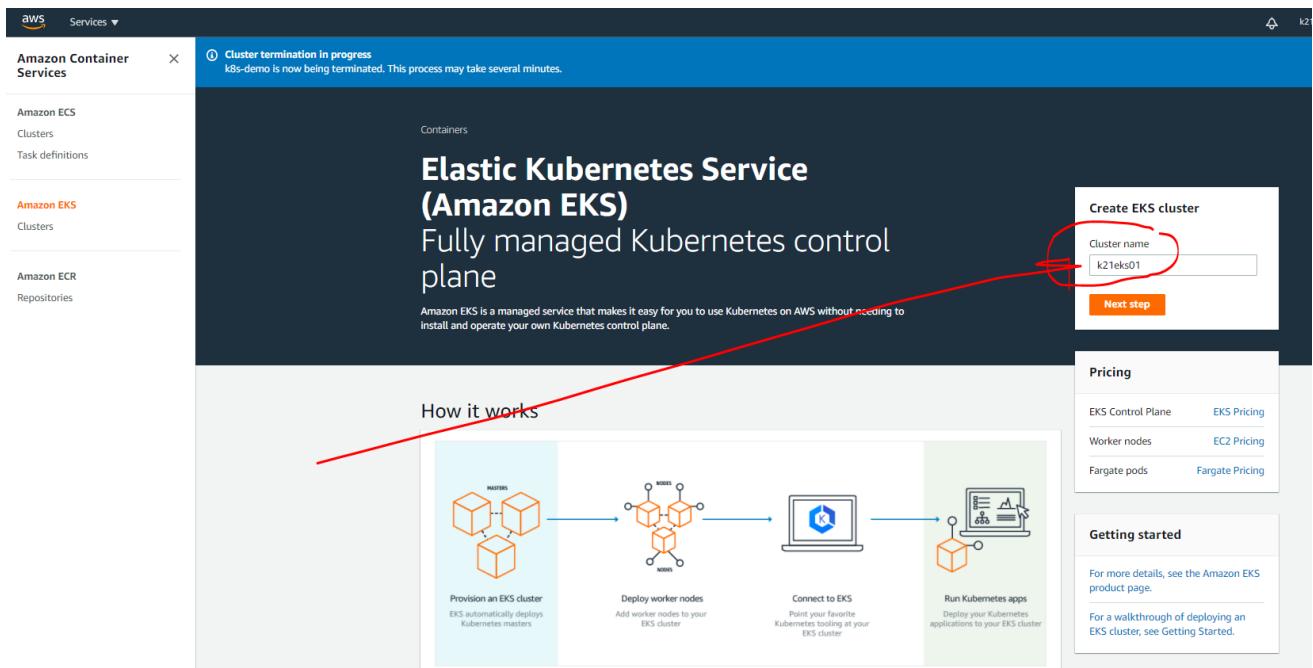
Login to AWS Portal using your account credentials. On the search bar type in EKS and select “Elastic Kubernetes Service”



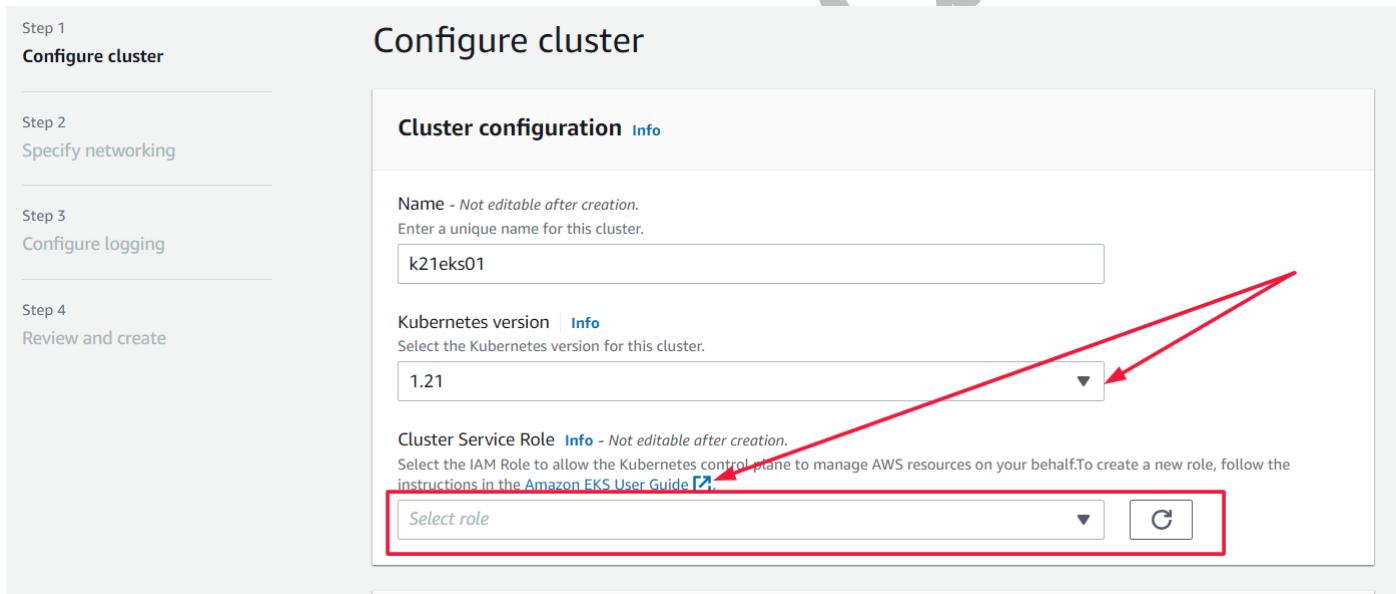
The screenshot shows the AWS Management Console interface. At the top, there's a search bar with the text "eks" typed into it. Below the search bar, the results show "Elastic Kubernetes Service" listed under "Find Services". To the right of the search bar, there are links for "IAM", "Elastic Container Registry", "Billing", and "VPC". Below the search results, there's a section titled "Build a solution" with several quick-launch options: "Launch a virtual machine", "Build a web app", "Build using virtual servers", "Register a domain", "Connect an IoT device", "Start migrating to AWS", "Start a development project", and "Deploy a serverless microservice". Each option includes a small icon and a brief description.

4.2 Create Kubernetes Cluster

1. Specify name for the Cluster



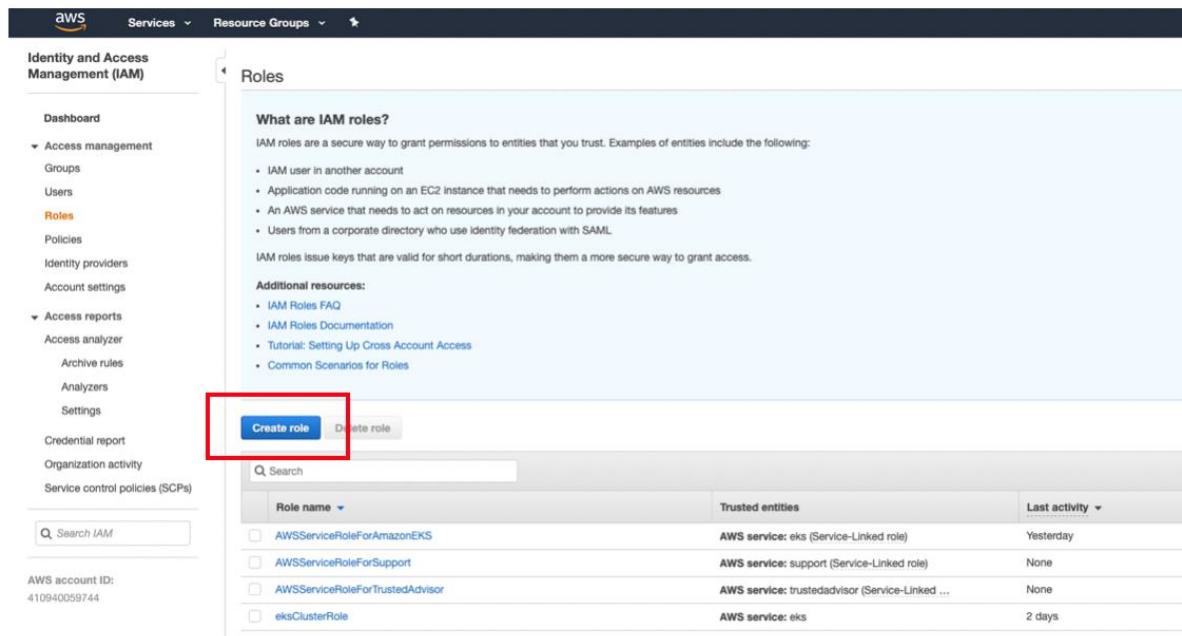
The screenshot shows the AWS EKS Create Cluster wizard. On the left, there's a sidebar with links for Amazon Container Services, Amazon ECS, Amazon ECR, and Amazon EKS. The main area displays the 'Elastic Kubernetes Service (Amazon EKS)' landing page, which includes a 'Cluster termination in progress' message. A red arrow points from the 'Cluster name' input field in the 'Create EKS cluster' step to the 'How it works' diagram. The diagram illustrates the four steps of creating an EKS cluster: Provision an EKS cluster (EKS automatically deploys Kubernetes masters), Deploy worker nodes (Add worker nodes to your EKS cluster), Connect to EKS (Point your favorite Kubernetes cluster at your EKS cluster), and Run Kubernetes apps (Deploy your Kubernetes applications to your EKS cluster). To the right, there are sections for Pricing and Getting started.



The screenshot shows the 'Configure cluster' step of the EKS Create Cluster wizard. On the left, a sidebar lists steps: Step 1 Configure cluster, Step 2 Specify networking, Step 3 Configure logging, and Step 4 Review and create. The main area shows the 'Cluster configuration' section. It includes fields for 'Name' (containing 'k21eks01'), 'Kubernetes version' (set to '1.21'), and 'Cluster Service Role' (with a note about IAM roles and a link to the User Guide). A red box highlights the 'Select role' dropdown, and a red arrow points from the 'Cluster Service Role' note to this dropdown. A large stylized 'K' graphic is visible on the left side of the page.

4.3 Creating an EKS role

1. Click on “Create role”



The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, there's a navigation sidebar with options like Dashboard, Access management, Groups, Users, Roles (which is selected), Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, and Service control policies (SCPs). Below the sidebar is a search bar labeled 'Search IAM'. At the bottom left, it says 'AWS account ID: 410940059744'. The main content area is titled 'Roles' and contains sections about what IAM roles are, additional resources, and a table of existing roles. The 'Create role' button is highlighted with a red box.

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonEKS	AWS service: eks (Service-Linked role)	Yesterday
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked ...)	None
eksClusterRole	AWS service: eks	2 days

Create role

1 2 3 4

Select type of trusted entity

 AWS service EC2, Lambda and others	 Another AWS account Belonging to you or 3rd party	 Web identity Cognito or any OpenID provider	 SAML 2.0 federation Your corporate directory
---	--	--	---

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeGuru	ElastiCache	Kinesis	RoboMaker
AWS Backup	CodeStar Notifications	Elastic Beanstalk	Lake Formation	S3
AWS Chatbot	Comprehend	Elastic Container Service	Lambda	SMS
AWS Support	Config	Elastic Transcoder	Lex	SNS
Amplify	Connect	Elastic Load Balancing	License Manager	SWF
AppStream 2.0	DMS	Forecast	Machine Learning	SageMaker
AppSync	Data Lifecycle Manager	GameLift	Macie	Security Hub
Application Auto Scaling	Data Pipeline	Global Accelerator	Managed Blockchain	Service Catalog
Application Discovery Service	DataSync	Glue	MediaConvert	Step Functions
Batch	DeepLens	Greengrass	Migration Hub	Storage Gateway
Chime	Directory Service	GuardDuty	OpsWorks	Systems Manager
CloudFormation	DynamoDB	Health Organizational View	Personalize	Textract
CloudHSM	EC2	IAM Access Analyzer	Purchase Orders	Transfer
	EC2 - Fleet	Inspector	QLDB	Trusted Advisor

* Required

Cancel

Next: Permissions

2. From the list of AWS services, select EKS and then “Next: Permissions”

Select your 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 - 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 - Nodegroup

Allows EKS to manage nodegroups on your behalf.

* Required

Cancel

Next: Permissions

- Leave the selected policies as-is, and proceed to the Review page.

Create role

Attached permissions policies

The type of role that you selected requires the following policy.

Filter policies ▾ Showing 1 result

Policy name	Used as	Description
AmazonEKSClusterPolicy	Permissions policy (1)	This policy provides Kubernetes the permission...

- Enter a name for the role (e.g. **eksClusterRole**) and hit the **Create Role** button at the bottom of the page to create the IAM role.

Create role

Review

Provide the required information below and review this role before you create it.

Role name*

eksClusterRole

Use alphanumeric and '+=-_,@_-' characters. Maximum 64 characters.

Role description

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

Maximum 1000 characters. Use alphanumeric and '+=-_,@_-' characters.

Trusted entities AWS service: eks.amazonaws.com

Policies  AmazonEKSClusterPolicy 

Permissions boundary Permissions boundary is not set

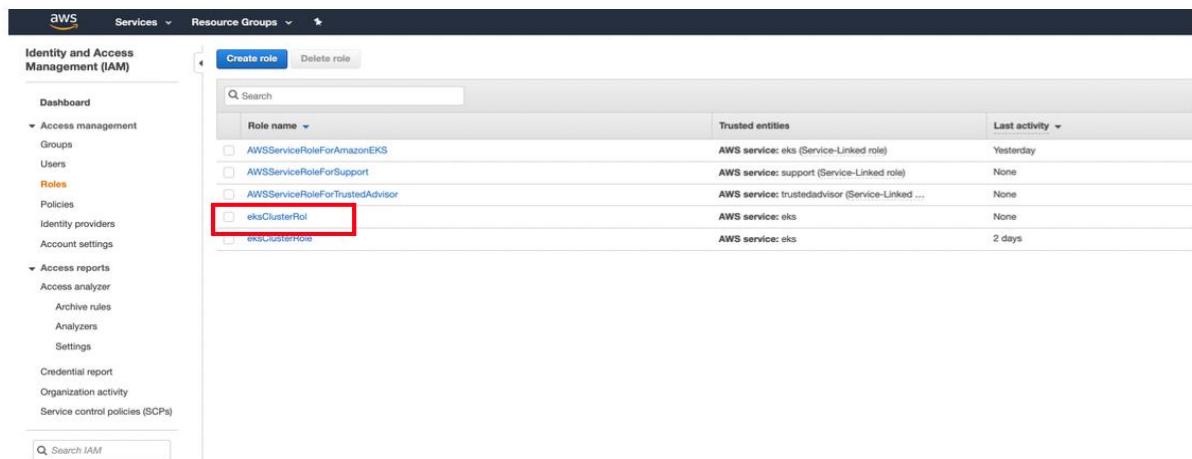
No tags were added.

Cancel

Previous

Create role

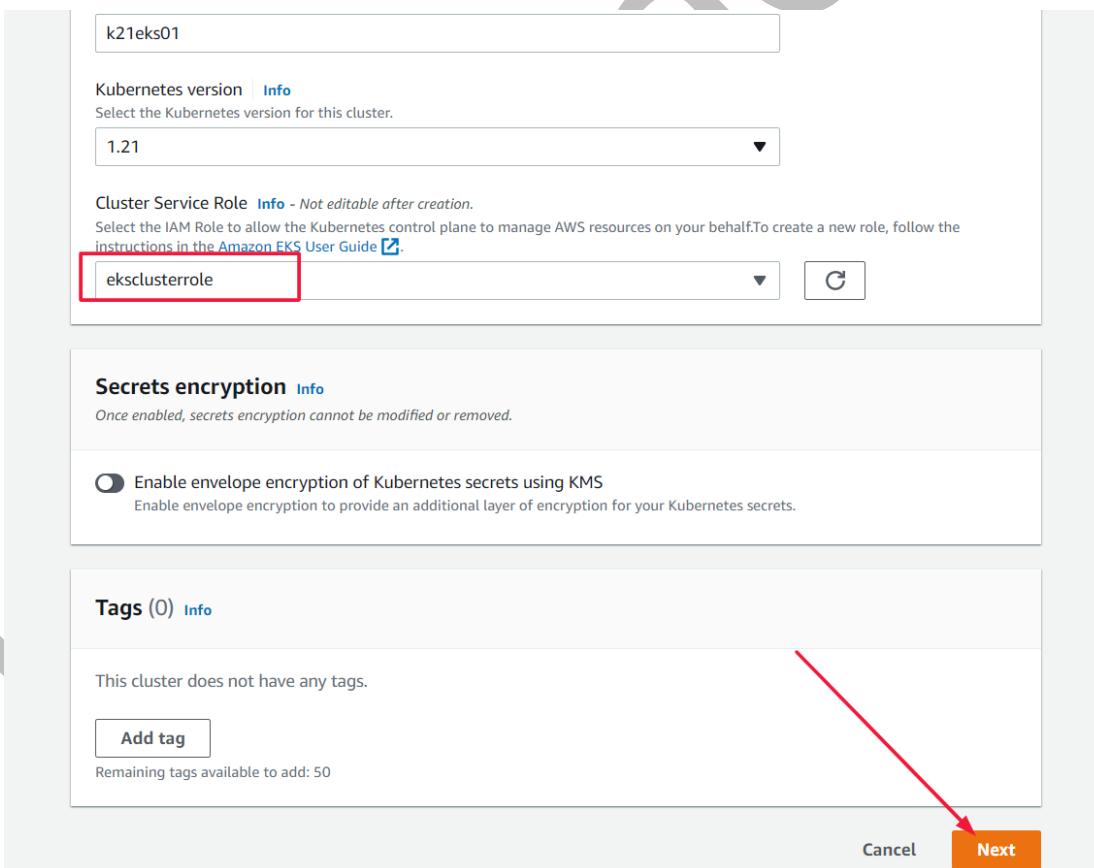
- The IAM role is created.



The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Dashboard', 'Access management', 'Roles' (which is selected and highlighted in orange), and 'Access reports'. The main area shows a table of roles. One role, 'eksClusterRole', is highlighted with a red box. The table columns include 'Role name', 'Trusted entities', and 'Last activity'.

4.4 Back to Cluster Configuration page

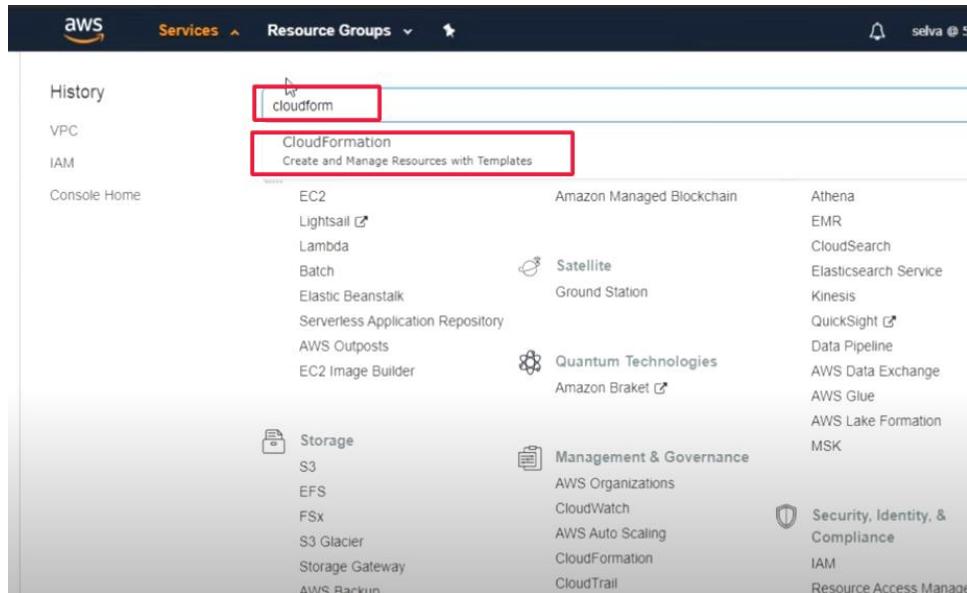
1. Select the created EKS role from the drop down and click on Next



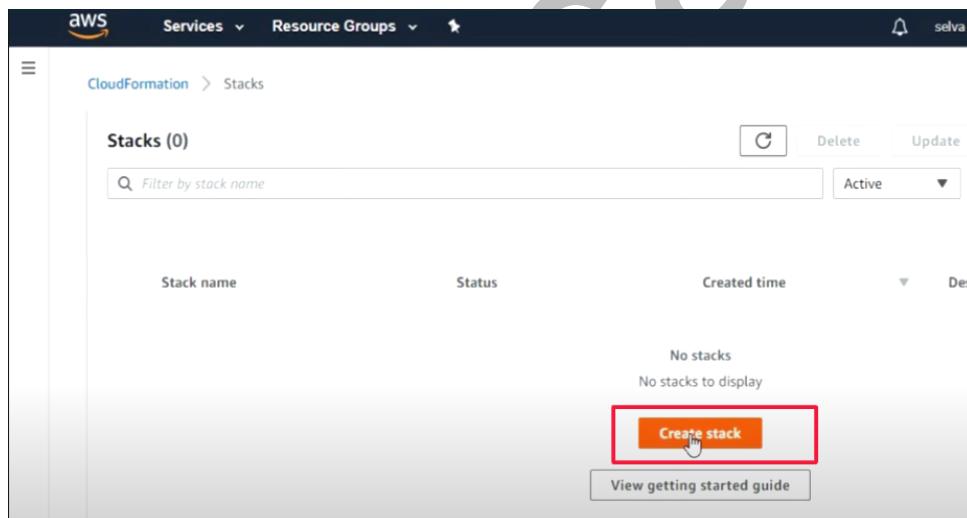
The screenshot shows the 'Create New Cluster' configuration page. It includes fields for 'Cluster Name' (k21eks01), 'Kubernetes version' (1.21 selected), 'Cluster Service Role' (eksclusterrole selected and highlighted with a red box), and 'Secrets encryption' (Enable envelope encryption of Kubernetes secrets using KMS). The 'Tags' section indicates no tags are present. A red arrow points from the 'eksclusterrole' dropdown to the 'Next' button at the bottom right.

2. Create a VPC:

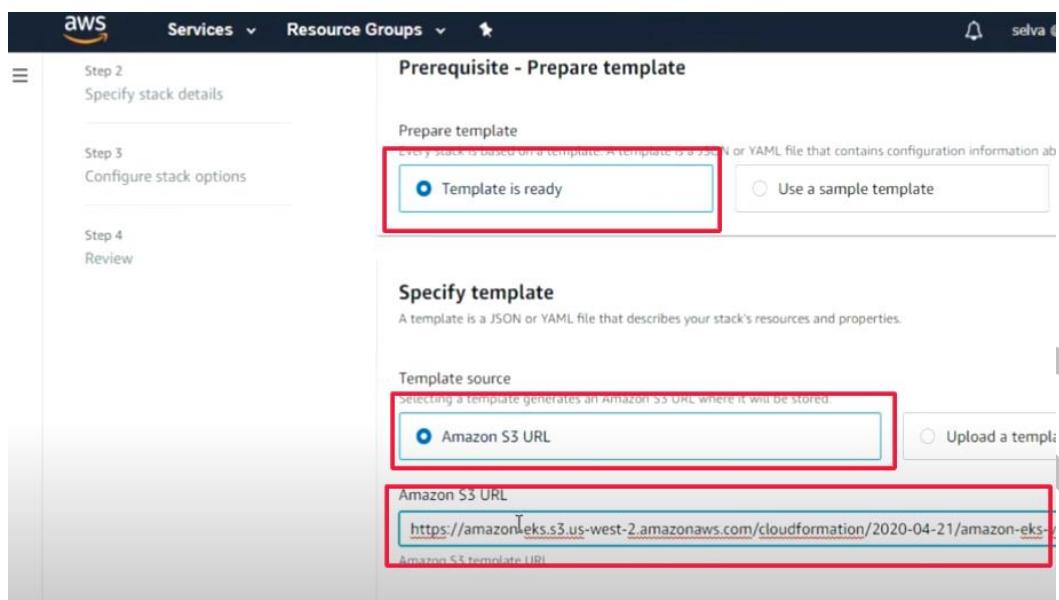
i) Search for CloudFormation



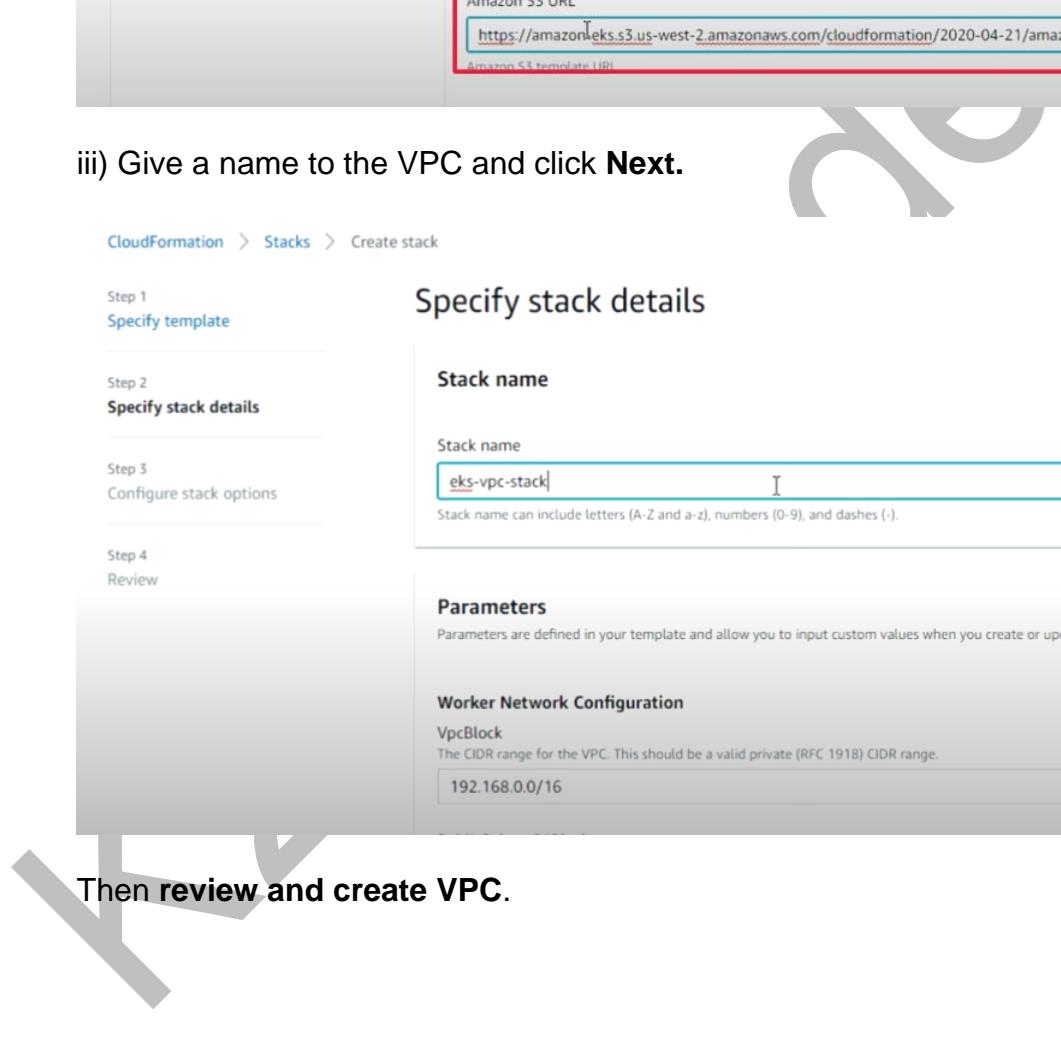
ii) Click on create VPC



- Select **Prepare template as Template is ready**
- **Template source: S3**
- **S3 URL:** <https://amazon-eks.s3.us-west-2.amazonaws.com/cloudformation/2020-07-23/amazon-eks-vpc-private-subnets.yaml>
- Then Click **Next**



iii) Give a name to the VPC and click **Next**.



CloudFormation > Stacks > Create stack

Step 1 Specify template

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review

Specify stack details

Stack name

Stack name: eks-vpc-stack

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Worker Network Configuration

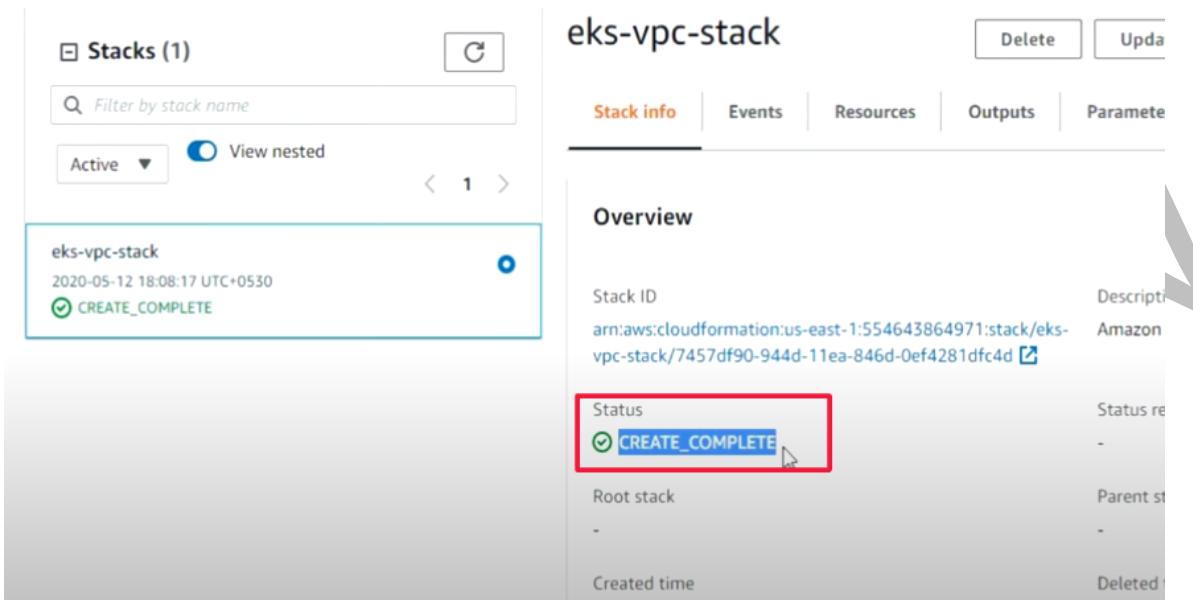
VpcBlock

The CIDR range for the VPC. This should be a valid private (RFC 1918) CIDR range.

192.168.0.0/16

Then **review and create VPC**.

CloudFormation > Stacks > eks-vpc-stack



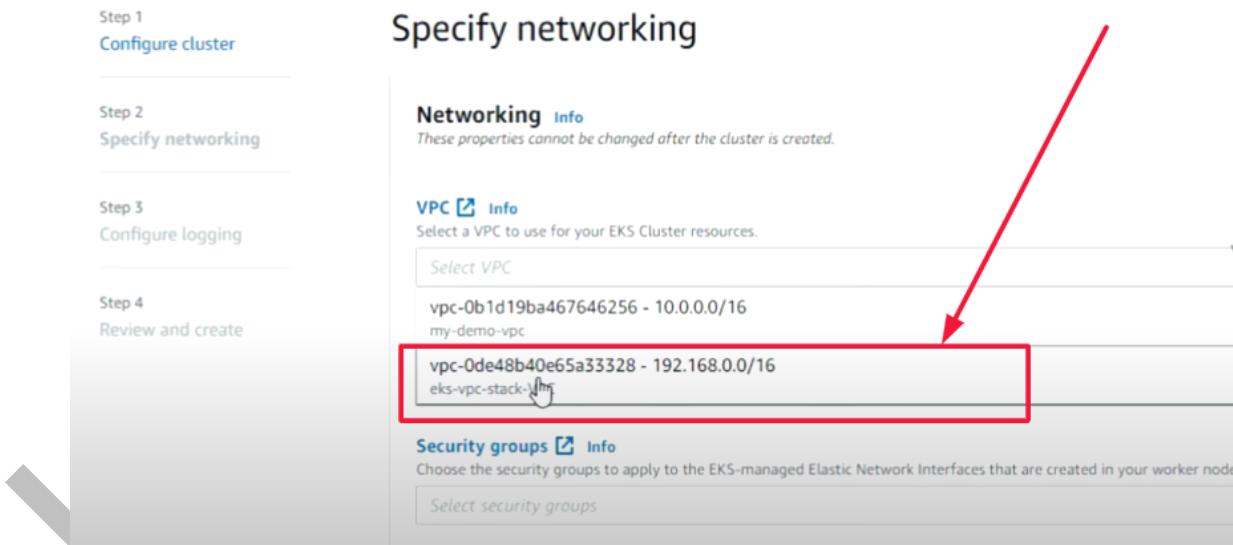
The screenshot shows the AWS CloudFormation Stacks interface. On the left, a sidebar lists 'Stacks (1)'. A search bar and a dropdown for 'Active' are at the top. Below is a table with one row:

eks-vpc-stack	2020-05-12 18:08:17 UTC+0530	
		 CREATE_COMPLETE

On the right, the 'eks-vpc-stack' details page is shown. It has tabs for 'Stack info', 'Events', 'Resources', 'Outputs', and 'Parameters'. The 'Stack info' tab is selected. It displays the Stack ID (arn:aws:cloudformation:us-east-1:554643864971:stack/eks-vpc-stack/7457df90-944d-11ea-846d-0ef4281dfc4d), Status (CREATE_COMPLETE, highlighted with a red box), Root stack (-), Created time (2020-05-12 18:08:17 UTC+0530), and Deleted time (-).

- Select the VPC that you have created.

EKS > Clusters > Create EKS cluster



The screenshot shows the 'Specify networking' step in the EKS Cluster creation wizard. On the left, a sidebar lists steps: Step 1 (Configure cluster), Step 2 (Specify networking, highlighted with a red box), Step 3 (Configure logging), and Step 4 (Review and create). The main area is titled 'Specify networking'.

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

VPC Info
 Select a VPC to use for your EKS Cluster resources.

Select VPC

- vpc-0b1d19ba467646256 - 10.0.0.0/16
my-demo-vpc
-  vpc-0de48b40e65a33328 - 192.168.0.0/16
eks-vpc-stack

Security groups Info
 Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your worker nodes.

Select security groups

- On Configure Logging select default value and click Next

EKS > Clusters > Create EKS cluster

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Review and create

Configure logging

Control Plane Logging Info

CloudWatch log group

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

API server

Logs pertaining to API requests to the cluster.

Disabled

Audit

Logs pertaining to cluster access via the Kubernetes API.

Disabled

Authenticator

Logs pertaining to authentication requests into the cluster.

Disabled

Controller manager

Logs pertaining to state of cluster controllers.

Disabled

Scheduler

Logs pertaining to scheduling decisions.

Disabled

Cancel

Previous

Next

5. Review the details and click create

Review and create

Step 1: Configure cluster

Edit

Cluster configuration

Name - *Not editable after creation.*

k21eks01

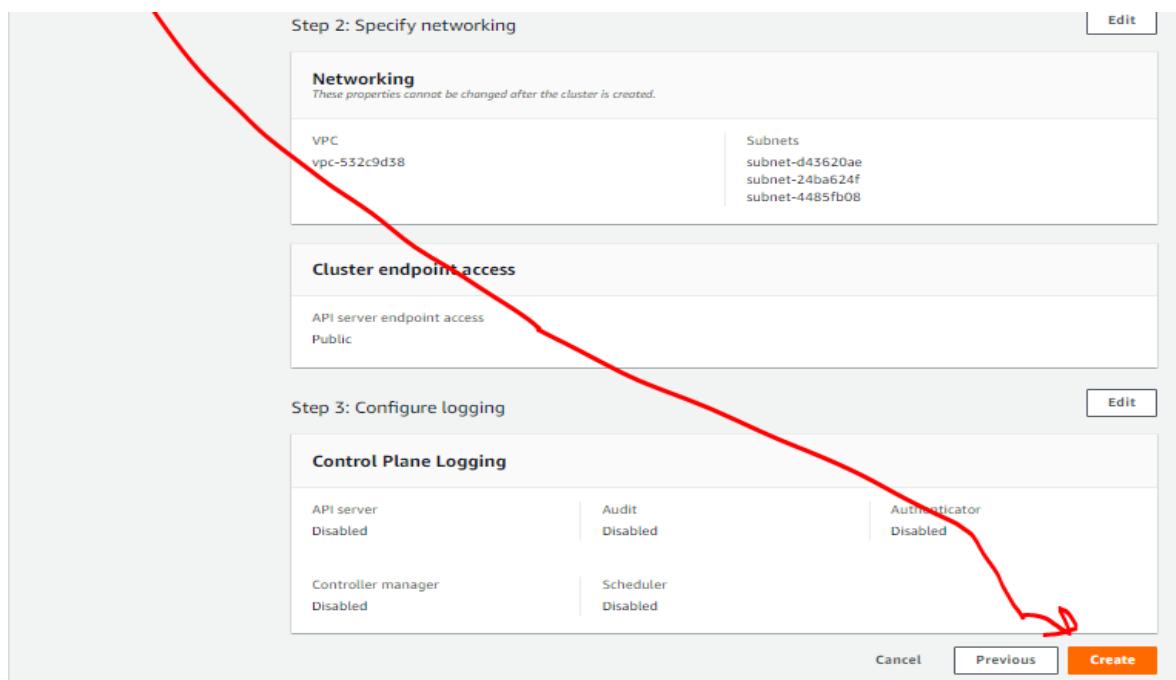
Kubernetes version

1.21

Cluster Service Role - *Not editable after creation.*

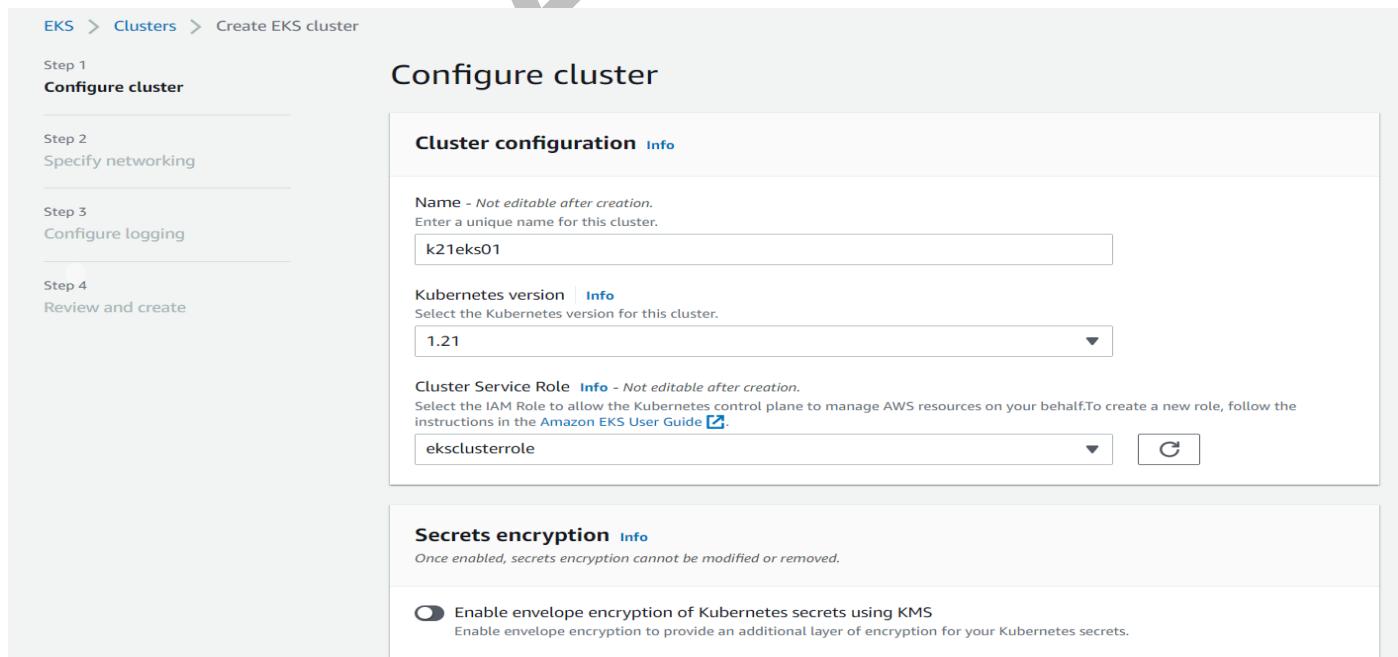
arn:aws:iam::423357588208:role/eksclusterrole





4.5 Review and Create the Cluster

1. Cluster creation in progress it will take 15-30 mins for cluster to come up



EKS > Clusters > Create EKS cluster

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Review and create

Configure cluster

Cluster configuration [Info](#)

Name - Not editable after creation.
Enter a unique name for this cluster.

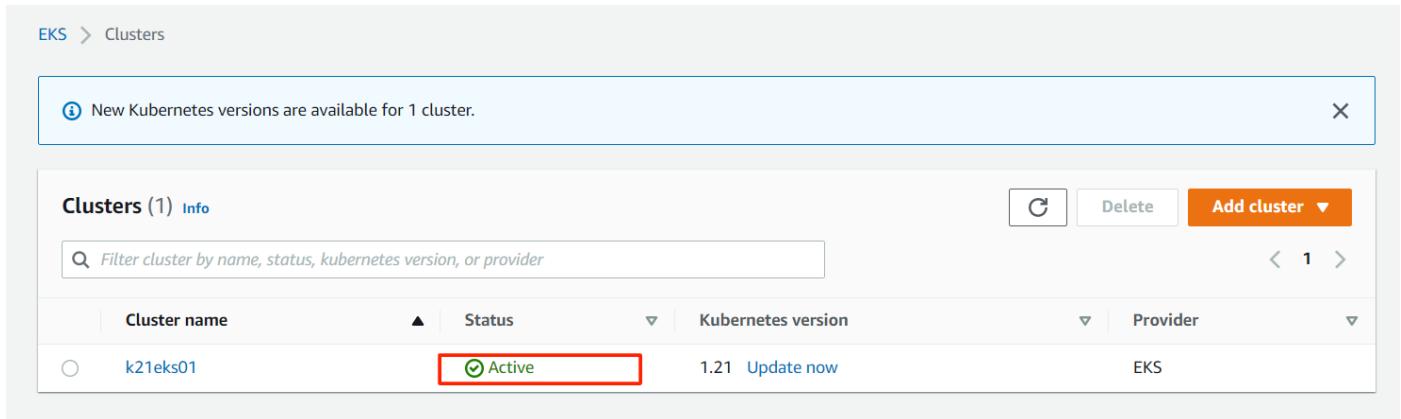
Kubernetes version [Info](#)
Select the Kubernetes version for this cluster.

Cluster Service Role [Info](#) - Not editable after creation.
Select the IAM Role to allow the Kubernetes control plane to manage AWS resources on your behalf. To create a new role, follow the instructions in the [Amazon EKS User Guide](#).

Secrets encryption [Info](#)
Once enabled, secrets encryption cannot be modified or removed.

Enable envelope encryption of Kubernetes secrets using KMS
Enable envelope encryption to provide an additional layer of encryption for your Kubernetes secrets.

2. Cluster is created with name **k21eks01** and is showing as status Active



EKS > Clusters

New Kubernetes versions are available for 1 cluster.

Clusters (1) Info		C	Delete	Add cluster ▾
		Filter cluster by name, status, kubernetes version, or provider	◀ 1 ▶	
Cluster name	Status	Kubernetes version	Provider	
k21eks01	Active	1.21 Update now	EKS	

Note: This Step will create Master node on Kubernetes Cluster.

Note: Stop the EKS Cluster after the usage. When you stop **cluster**, it is shut down and you are not billed for hourly usage. If you don't stop the instance you will be highly charged. If you decide that you no longer need an instance, you can terminate it.

5 CONFIGURE KUBECTL ON CLIENT

Note: To install & configure AWS CLI, and to install EKSCTL and kubectl please follow guide Activity Guide: *AG_Install_Configure_AWSCLI_KUBECTL_EKSCTL_on_Linux_ed** from Portal.*

5.1 Setup up the kubectl configuration

1. Check your **Cluster Name & Region Name** where EKS Master node is running from console
2. Check status of Cluster as

```
aws eks --region [EKS_Region] describe-cluster --name [EKS_Cluster_Name] --query cluster.status
aws eks --region us-east-2 describe-cluster --name k21eks01 --query cluster.status
```

```
C:\Users\44748>aws eks --region us-east-2 describe-cluster --name k21eks01 --query cluster.status
"ACTIVE"
```

3. Configure kubectl with EKS API Server credential

```
aws eks --region [EKS_Region] update-kubeconfig --name [EKS_Cluster_Name]
aws eks --region us-east-2 update-kubeconfig --name k21eks01
```

```
C:\Users\44748>aws eks --region us-east-2 update-kubeconfig --name k21eks01
Added new context arn:aws:eks:us-east-2:455508999287:cluster/k21eks01 to C:\Users\44748\.kube\config
```

4. Validate kubectl configuration to master node

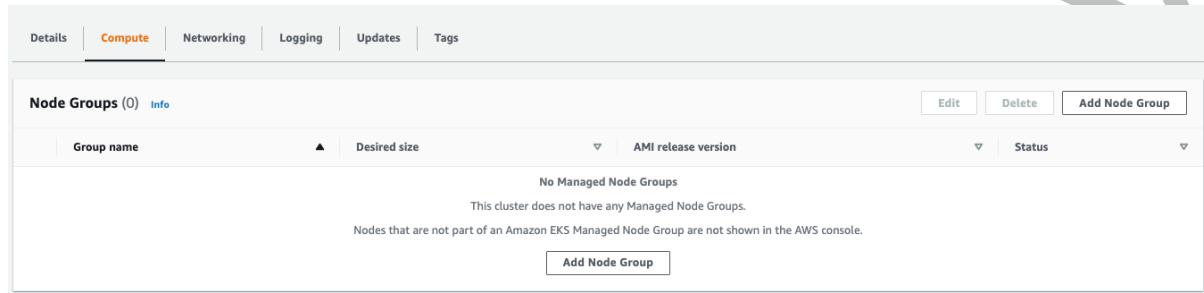
```
kubectl get svc
```

```
C:\Users\44748>kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP     40m
```

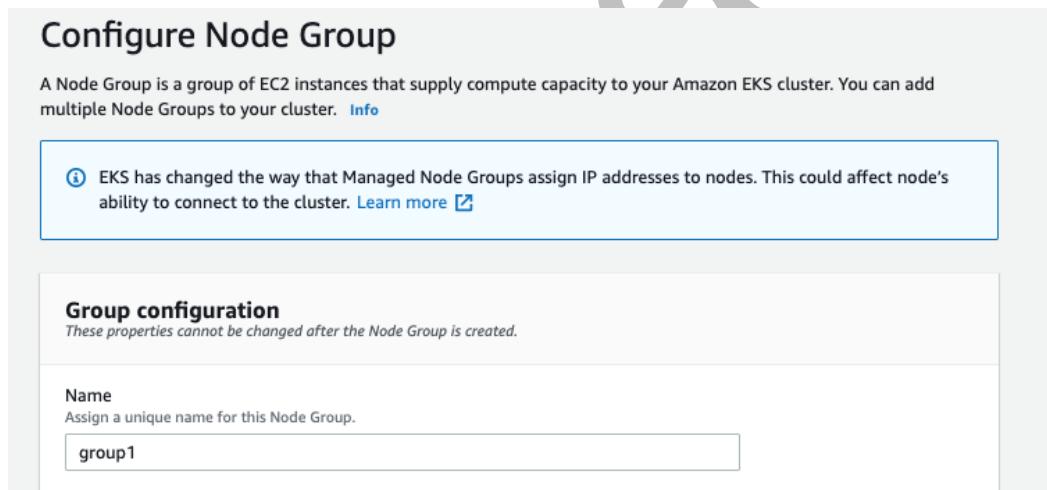
6 CREATE EKS CLUSTER WORKER NODES USING CONSOLE

6.1 Setup Worker Nodes in EKS

1. On the cluster page, select the Compute tab, and then choose Add Node Group.



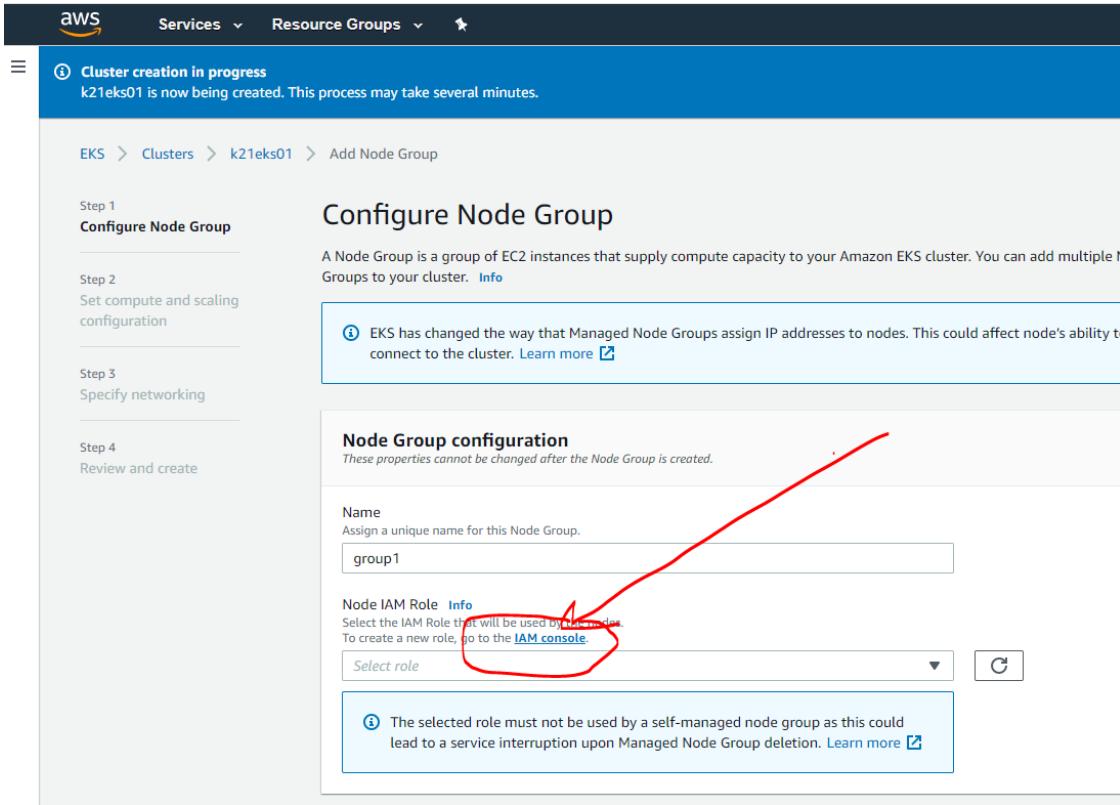
2. On the Configure node group page, fill out the parameters accordingly, and then choose Next.
 - **Name** – Enter a unique name for your managed node group.



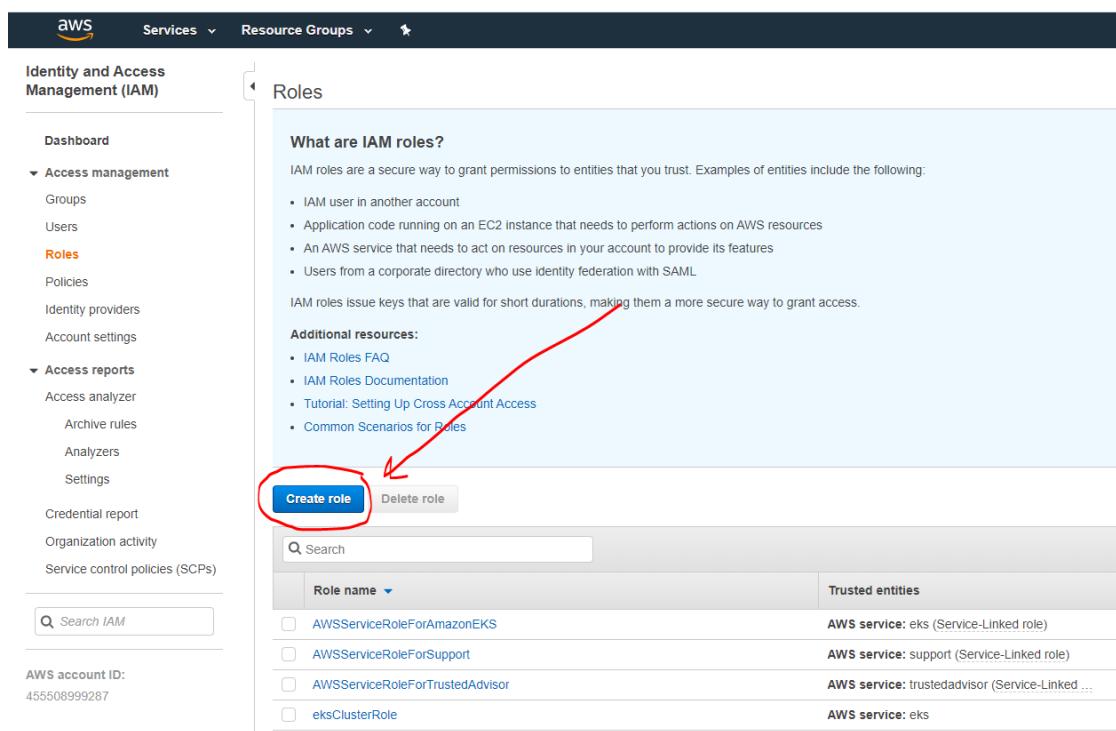
- **Node IAM role name** – Choose the node instance role to use with your node group. For more information, see Amazon EKS worker node IAM role.

To create your Amazon EKS worker node role in the IAM console

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Roles**, then **Create role**.
3. Choose **EC2** from the list of **Common use cases** under **Choose a use case**, then choose **Next: Permissions**.
4. In the **Filter policies** box, enter **AmazonEKSWorkerNodePolicy**. Check the box to the left of **AmazonEKSWorkerNodePolicy**.
5. In the **Filter policies** box, enter **AmazonEKS_CNI_Policy**. Check the box to the left of **AmazonEKS_CNI_Policy**.
6. In the **Filter policies** box, enter **AmazonEC2ContainerRegistryReadOnly**. Check the box to the left of **AmazonEC2ContainerRegistryReadOnly**.
7. Choose **Next: Tags**.
8. (Optional) Add metadata to the role by attaching tags as key-value pairs. For more information about using tags in IAM, see [Tagging IAM Entities](#) in the [IAM User Guide](#).
9. Choose **Next: Review**.
10. For **Role name**, enter a unique name for your role, such as **NodeInstanceRole**. For **Role description**, replace the current text with descriptive text such as **Amazon EKS - Node Group Role**, then choose **Create role**.



The screenshot shows the AWS EKS Node Group configuration interface. At the top, a message indicates "Cluster creation in progress" for cluster "k21eks01". The navigation path is EKS > Clusters > k21eks01 > Add Node Group. The main step is "Step 1 Configure Node Group". Below it, "Step 2 Set compute and scaling configuration" and "Step 3 Specify networking" are listed. The "Node Group configuration" section is active, showing a note that properties cannot be changed after creation. It includes fields for "Name" (set to "group1") and "Node IAM Role" (with a dropdown menu open). A red circle highlights the "Select role" dropdown. A tooltip for the role selection notes that the selected role must not be used by a self-managed node group.



Identity and Access Management (IAM)

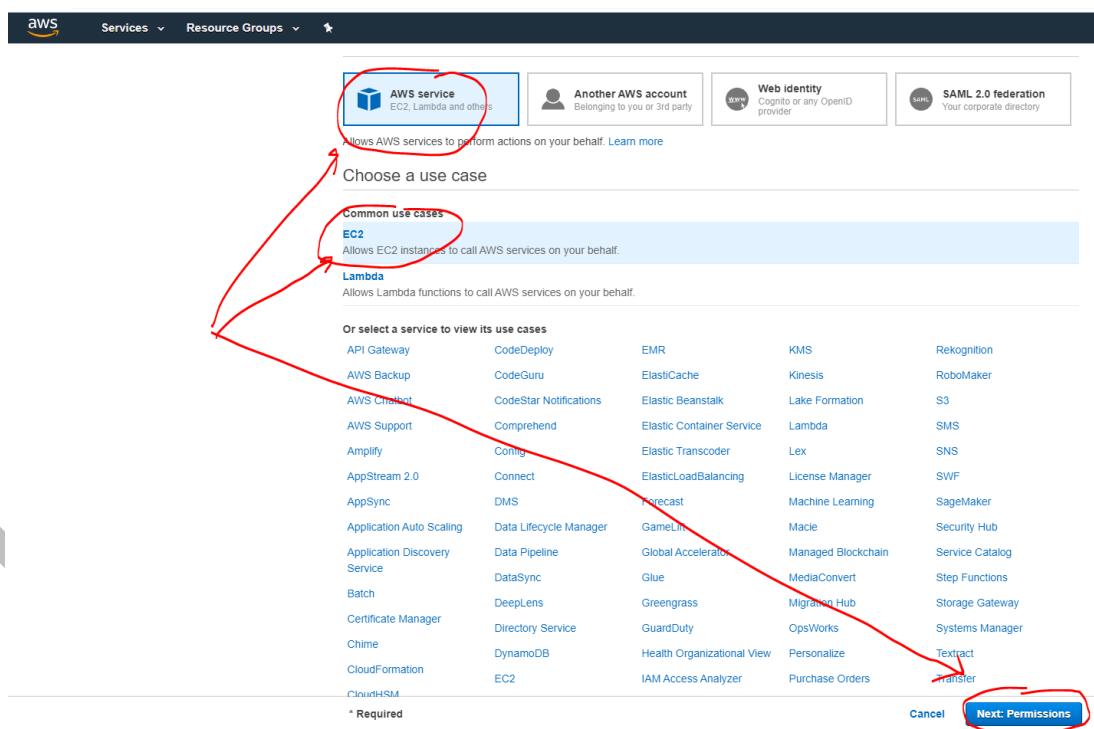
- Dashboard
- Access management
 - Groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Create role Delete role

Role name	Trusted entities
AWSServiceRoleForAmazonEKS	AWS service: eks (Service-Linked role)
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)
AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked ...)
eksClusterRole	AWS service: eks

Search IAM

AWS account ID: 455508999287



AWS service EC2, Lambda and others

Allows AWS services to perform actions on your behalf. [Learn more](#)

Another AWS account Belonging to you or 3rd party

Web identity Cognito or any OpenID provider

SAML 2.0 federation Your corporate directory

Choose a use case

Common use cases

EC2 Allows EC2 instances to call AWS services on your behalf.

Lambda Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeDeploy	EMR	KMS	Rekognition
AWS Backup	CodeGuru	ElasticCache	Kinesis	RoboMaker
AWS CloudFront	CodeStar Notifications	Elastic Beanstalk	Lake Formation	S3
AWS Support	Comprehend	Elastic Container Service	Lambda	SMS
Amplify	Config	Elastic Transcoder	Lex	SNS
AppStream 2.0	Connect	Elastic Load Balancing	License Manager	SWF
AppSync	DMS	Forecast	Machine Learning	SageMaker
Application Auto Scaling	Data Lifecycle Manager	GameLift	Macie	Security Hub
Application Discovery Service	Data Pipeline	Global Accelerator	Managed Blockchain	Service Catalog
Batch	DataSync	Glue	MediaConvert	Step Functions
Certificate Manager	DeepLens	Greengrass	Migration Hub	Storage Gateway
Chime	Directory Service	GuardDuty	OpsWorks	Systems Manager
CloudFormation	DynamoDB	Health Organizational View	Personalize	Textract
CloudHSM	EC2	IAM Access Analyzer	Purchase Orders	Transfer

* Required

Cancel **Next: Permissions**

Create role

1 2 3 4

▼ Attach permissions policies

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

[Create policy](#)

Filter policies ▾		Policy name	Used as
<input checked="" type="checkbox"/>	▶	AmazonEKS_CNI_Policy	None
<input type="checkbox"/>	▶	AmazonEKSClusterPolicy	Permissions policy (1)
<input type="checkbox"/>	▶	AmazonEKSFargatePodExecutionRolePolicy	None
<input type="checkbox"/>	▶	AmazonEKSForFargateServiceRolePolicy	None
<input type="checkbox"/>	▶	AmazonEKSServicePolicy	None
<input type="checkbox"/>	▶	AmazonEKSServiceRolePolicy	Permissions policy (1)
<input type="checkbox"/>	▶	AmazonEKSVPCResourceController	None
<input checked="" type="checkbox"/>	▶	AmazonEKSWorkerNodePolicy	None

▶ Set permissions boundary

Create role

1 2 3 4

▼ Attach permissions policies

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

[Create policy](#)

Filter policies ▾		Policy name	Used as
<input type="checkbox"/>	▶	AmazonEC2ContainerRegistryFullAccess	None
<input type="checkbox"/>	▶	AmazonEC2ContainerRegistryPowerUser	None
<input checked="" type="checkbox"/>	▶	AmazonEC2ContainerRegistryReadOnly	None
<input type="checkbox"/>	▶	AmazonEC2ContainerServiceAutoscaleRole	None
<input type="checkbox"/>	▶	AmazonEC2ContainerServiceEventsRole	None
<input type="checkbox"/>	▶	AmazonEC2ContainerServiceforEC2Role	None
<input type="checkbox"/>	▶	AmazonEC2ContainerServiceFullAccess	None
<input type="checkbox"/>	▶	AmazonEC2ContainerServiceRole	None

▶ Set permissions boundary

* Required

Cancel

Previous

Next: Tags

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* EKSNodeInstanceRole

Use alphanumeric and '+-=_,@-' characters. Maximum 64 characters.

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+-=_,@-' characters.

Trusted entities AWS service: ec2.amazonaws.com

- Policies**
-  AmazonEKS_CNI_Policy [Edit](#)
 -  AmazonEKSWorkerNodePolicy [Edit](#)
 -  AmazonEC2ContainerRegistryReadOnly [Edit](#)

Permissions boundary Permissions boundary is not set

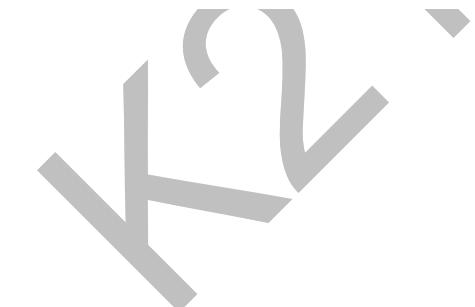
No tags were added.

* Required

[Cancel](#)

[Previous](#)

Create role



Cluster creation in progress
k21eks01 is now being created. This process may take several minutes.

EKS > Clusters > k21eks01 > Add Node Group

Step 1 Configure Node Group

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. [Info](#)

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

Name: C

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](#).

C

Launch template [Info](#)
These properties cannot be changed after the Node Group is created.

group1

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](#).

C

Launch template [Info](#)
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 [Info](#)

This Node Group does not have any labels.

Add label
Remaining labels available to add: 50

Tags (0) [Info](#)

This Node Group does not have any tags.

Add tag
Remaining tags available to add: 50

Cancel Next

EKS > Clusters > k21eks01 > 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 2 (AL2_x86_64)

Instance type **Info**
Select the EC2 instance type for nodes.
t3.medium

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

Node Group scaling configuration

T3.medi

Minimum size
Set the minimum number of nodes that the group can scale in to.
2 nodes

Maximum size
Set the maximum number of nodes that the group can scale out to.
2 nodes

Desired size
Set the desired number of nodes that the group should launch with initially.
2 nodes

Cancel **Previous** **Next**

aws Services Resource Groups

Cluster creation in progress
k21eks01 is now being created. This process may take several minutes.

EKS > Clusters > k21eks01 > Add Node Group

Step 1 Configure Node Group

Step 2 Set compute and scaling configuration

Step 3 Specify networking

Step 4 Review and create

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 [View VPC console](#).

Select subnets **C**

subnet-0c6c8bfffba9e81053 X subnet-0bbd6f43ced18f278 X
subnet-0c196ed9dadfc998a X

Allow remote access to nodes **Info**
Without remote access enabled you will not be able to directly connect to nodes after they are created.

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

Select EC2 Key Pair **C**

Required

Allow remote access from
Configure the source IP ranges that can remotely access nodes.

All Do not restrict source IPs that can remotely access nodes.
Selected security groups Specify security groups to restrict which source IPs can remotely access nodes.

Cancel **Previous** **Next**

AWS Services Resource Groups

Cluster creation in progress
k21eks01 is now being created. This process may take several minutes.

EKS > Clusters > k21eks01 > Add Node Group

Step 1 Configure Node Group

Step 2 Set compute and scaling configuration

Step 3 Specify networking

Step 4 Review and create

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-0c6c8bffa9e81053, subnet-0bbd6f42ced18f278, subnet-0c196ed9dadfc59da

Allow remote access to nodes Info
Without remote access enabled you will not be able to directly connect to nodes after they are created.

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

Select EC2 Key Pair:

⚠ Required

Allow remote access from
Configure the source IP ranges that can remotely access nodes.

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

Selected security groups: Specify security groups to restrict which source IPs can remotely access nodes.

Cancel Previous Next

New EC2 Experience Tell us what you think X

EC2 Dashboard New Events New Tags Limits Instances Instances

Key pairs

Filter key pairs

Name	Fingerprint	ID
No key pairs to display		

Actions Create key pair

K21

AWS Services Resource Groups ⚙

EC2 > Key pairs > Create key pair

Create key pair

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

File format pem For use with OpenSSH ppk For use with PuTTY

Tags (Optional)
No tags associated with the resource.

Add tag You can add 50 more tags

Cancel **Create key pair**

Resource Groups

Successfully created key pair

Key pairs (1)

Name	Fingerprint	ID
Public_Key_For_EKS	a0:b6:70:2c:5e:73:d5:7c:9c:52:43:e5:9...	key-05622f0c1e606f352

Actions Create key pair

aws Services ▾ Resource Groups ▾

☰ Cluster creation in progress
k21eks01 is now being created. This process may take several minutes.

EKS > Clusters > k21eks01 > Add Node Group

Step 1 Configure Node Group

Step 2 Set compute and scaling configuration

Step 3 Specify networking

Step 4 Review and create

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 [C](#)

subnet-0c6c8bfffba9e81053 X subnet-0bbd6f43ced18f278 X
subnet-0c196ed9dadfc998a X

Allow remote access to nodes [Info](#)
Without remote access enabled you will not be able to directly connect to nodes after they are created.

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

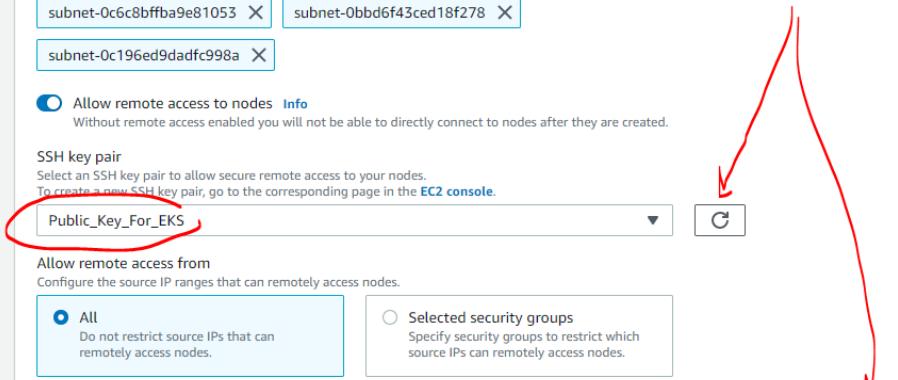
Public_Key_For_EKS [C](#)

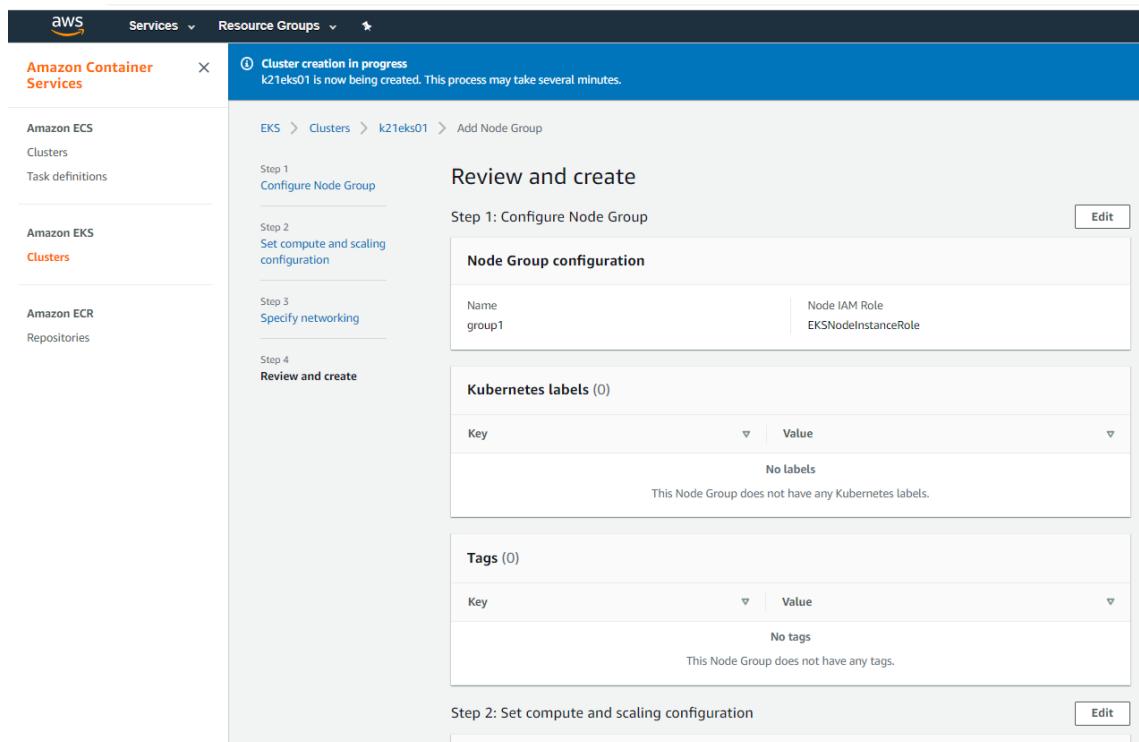
Allow remote access from
Configure the source IP ranges that can remotely access nodes.

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

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

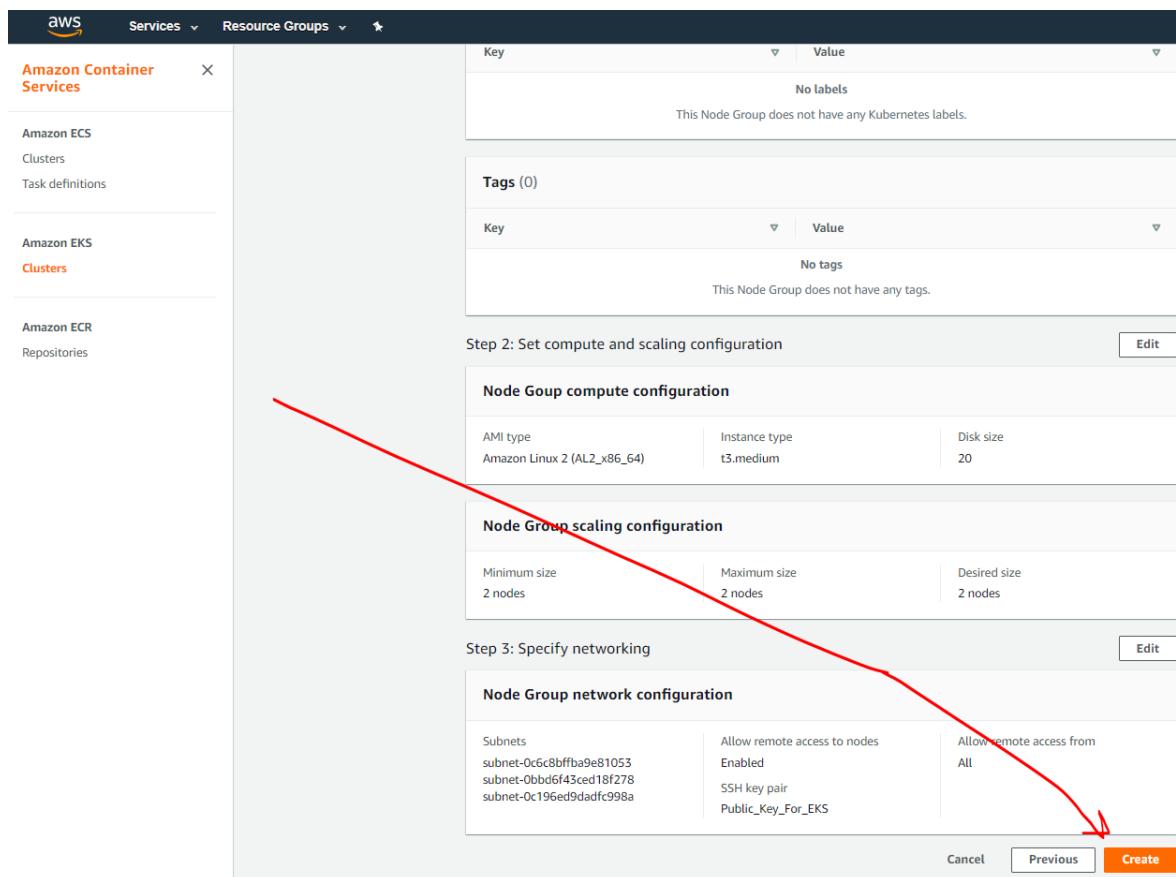
[Cancel](#) [Previous](#) [Next](#)



The screenshot shows the AWS EKS console interface for creating a new Node Group. The top navigation bar includes the AWS logo, Services dropdown, and Resource Groups button. The left sidebar for Amazon Container Services lists Amazon ECS Clusters, Task definitions, Amazon EKS Clusters, and Amazon ECR Repositories. The main content area shows a breadcrumb path: EKS > Clusters > k21eks01 > Add Node Group. A blue header bar indicates "Cluster creation in progress" for "k21eks01 is now being created. This process may take several minutes." The process is divided into four steps: Step 1: Configure Node Group (current step), Step 2: Set compute and scaling configuration, Step 3: Specify networking, and Step 4: Review and create. Step 1 is titled "Review and create" and contains a sub-section "Step 1: Configure Node Group" with a "Node Group configuration" table. The table shows a single row with "Name" set to "group1" and "Node IAM Role" set to "EKSNodeInstanceRole". There are "Edit" buttons for both columns. Below this is a section for "Kubernetes labels (0)" with a table showing "No labels" and a note that the Node Group does not have any Kubernetes labels. Another section for "Tags (0)" shows "No tags" and a note that the Node Group does not have any tags. At the bottom of the configuration section is another "Edit" button for "Step 2: Set compute and scaling configuration".

K21Ac



Step 2: Set compute and scaling configuration

AMI type	Instance type	Disk size
Amazon Linux 2 (AL2_x86_64)	t3.medium	20

Step 3: Specify networking

Subnets	Allow remote access to nodes	Allow remote access from
subnet-0c6c8bfffba9e81053 subnet-0bbd6f43ced18f278 subnet-0c196ed9dadfc998a	Enabled SSH key pair Public_Key_For_EKS	All

Cancel Previous Create

3. Worker Node Group is under creation so wait for 2-3 minutes for workers nodes to be up and working

6.2 Verifying Worker node status from Kubectl

1. Watch the status of your nodes and wait for them to reach the Ready status.

```
$ kubectl get nodes --watch
```

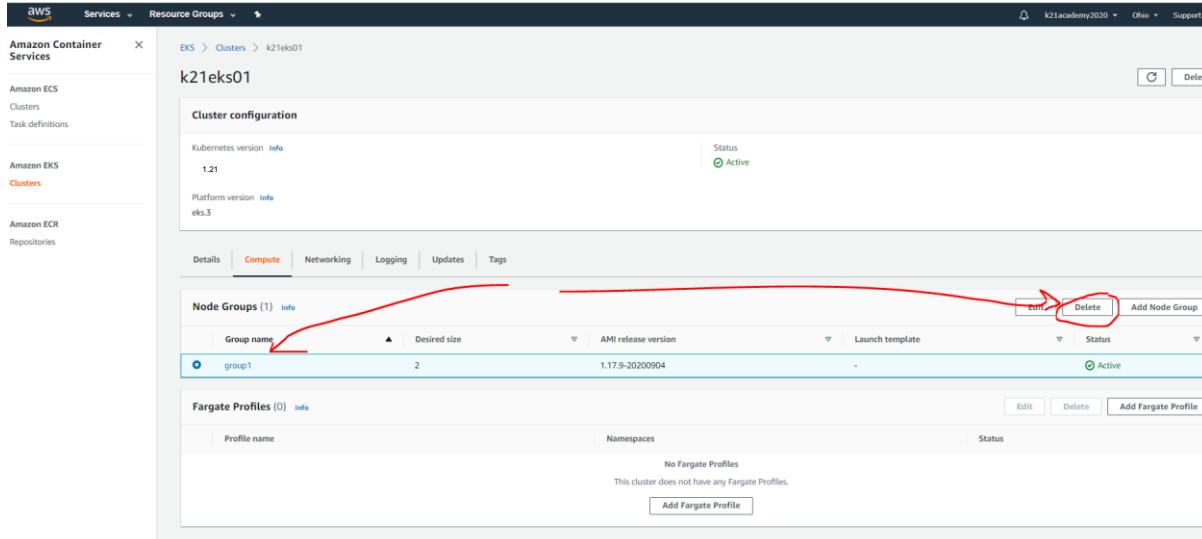
```
ubuntu@ip-172-31-20-43:~$ kubectl get nodes --watch
NAME                      STATUS   ROLES      AGE     VERSION
ip-172-31-13-0.us-east-2.compute.internal   Ready    <none>    35m    v1.21.5-eks-9017834
ip-172-31-30-233.us-east-2.compute.internal   Ready    <none>    35m    v1.21.5-eks-9017834
```

Note: Stop the EKS Cluster after the usage. When you stop **cluster**, it is shut down and you are not billed for hourly usage. If you don't stop the instance you will be highly charged. If you decide that you no longer need an instance, you can terminate it.

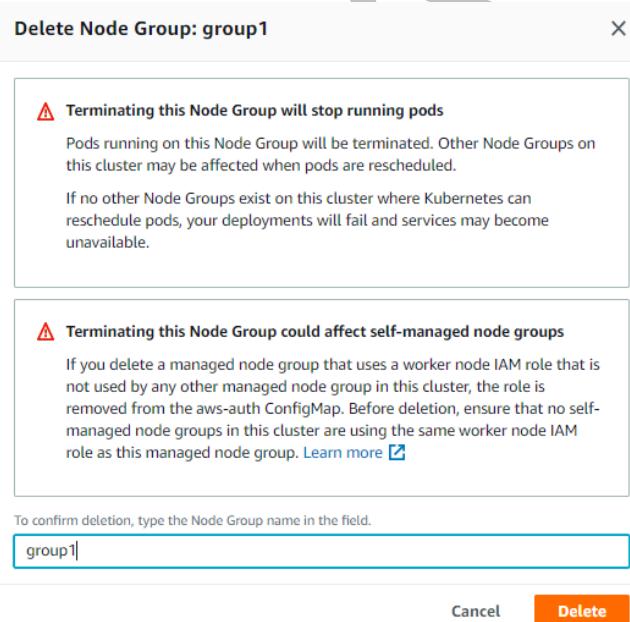
7 DELETE EKS CLUSTER FROM CONSOLE

7.1 Delete Worker Nodes

1. Delete Node Group



The screenshot shows the AWS EKS Cluster configuration page for the cluster 'k21eks01'. The 'Compute' tab is selected. In the 'Node Groups' section, there is one entry named 'group1' with a value of '2'. A red arrow points to the 'group1' entry, and another red arrow points to the 'Delete' button in the top right corner of the table.



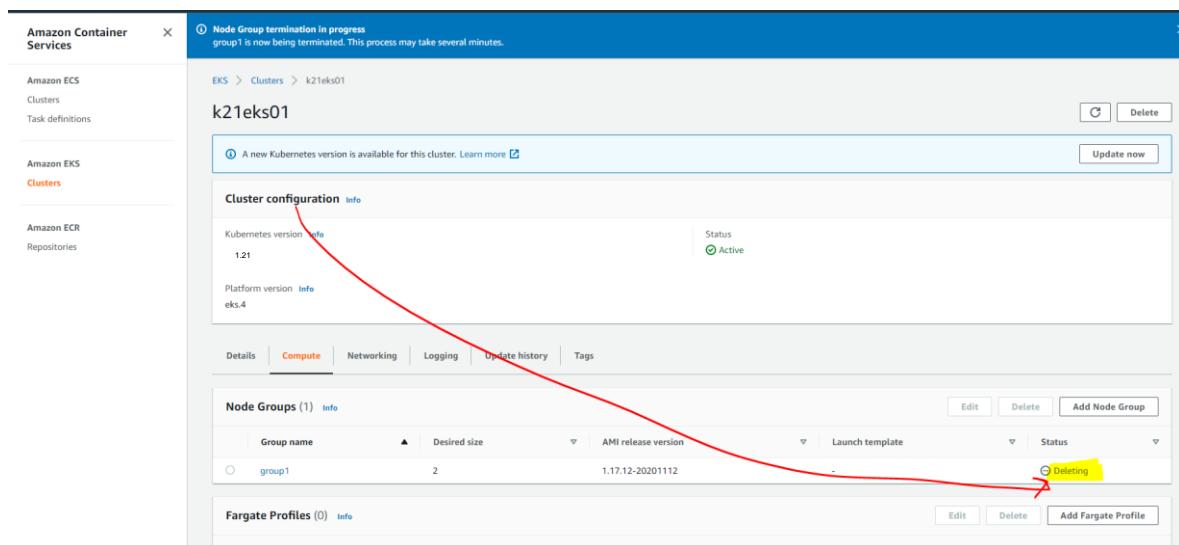
Delete Node Group: group1

⚠ Terminating this Node Group will stop running pods
Pods running on this Node Group will be terminated. Other Node Groups on this cluster may be affected when pods are rescheduled.
If no other Node Groups exist on this cluster where Kubernetes can reschedule pods, your deployments will fail and services may become unavailable.

⚠ Terminating this Node Group could affect self-managed node groups
If you delete a managed node group that uses a worker node IAM role that is not used by any other managed node group in this cluster, the role is removed from the aws-auth ConfigMap. Before deletion, ensure that no self-managed node groups in this cluster are using the same worker node IAM role as this managed node group. [Learn more](#)

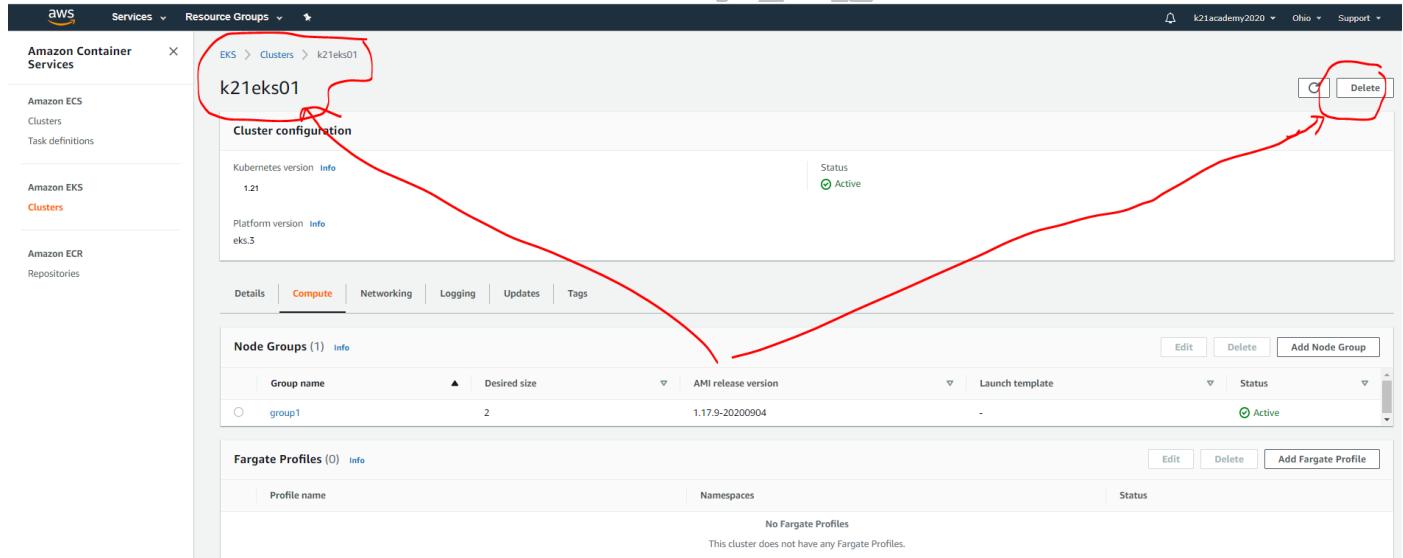
To confirm deletion, type the Node Group name in the field.

Cancel **Delete**

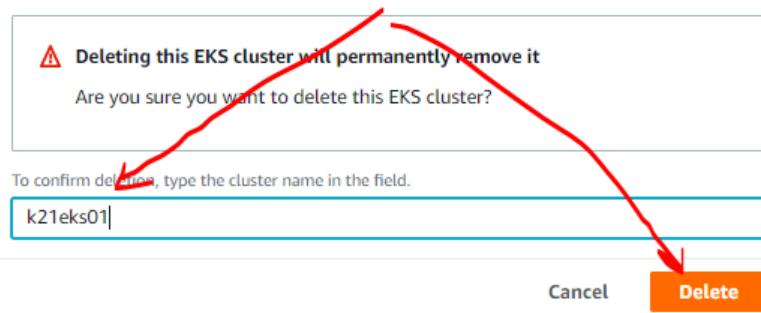


Note: This process will take approx. 10-15 minutes

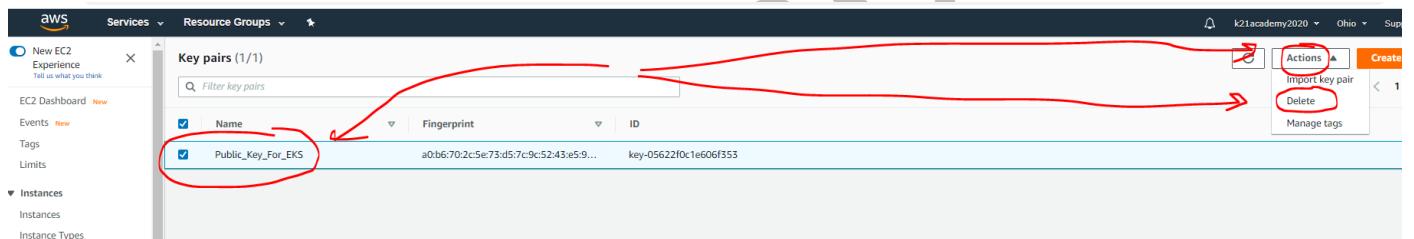
7.2 Delete Master Node



Delete Cluster: k21eks01



7.3 Delete Key Pair



Key pairs (1/1)
Filter key pairs
Name Fingerprint ID
Public_Key_For_EKS a0:b6:70:2c:5e:73:d5:7c:9c:52:43:e5:9... key-05622f0c1e606f353
Actions Import key pair Delete Manage tags

Public_Key_For_EKS could be associated with one or more instances.

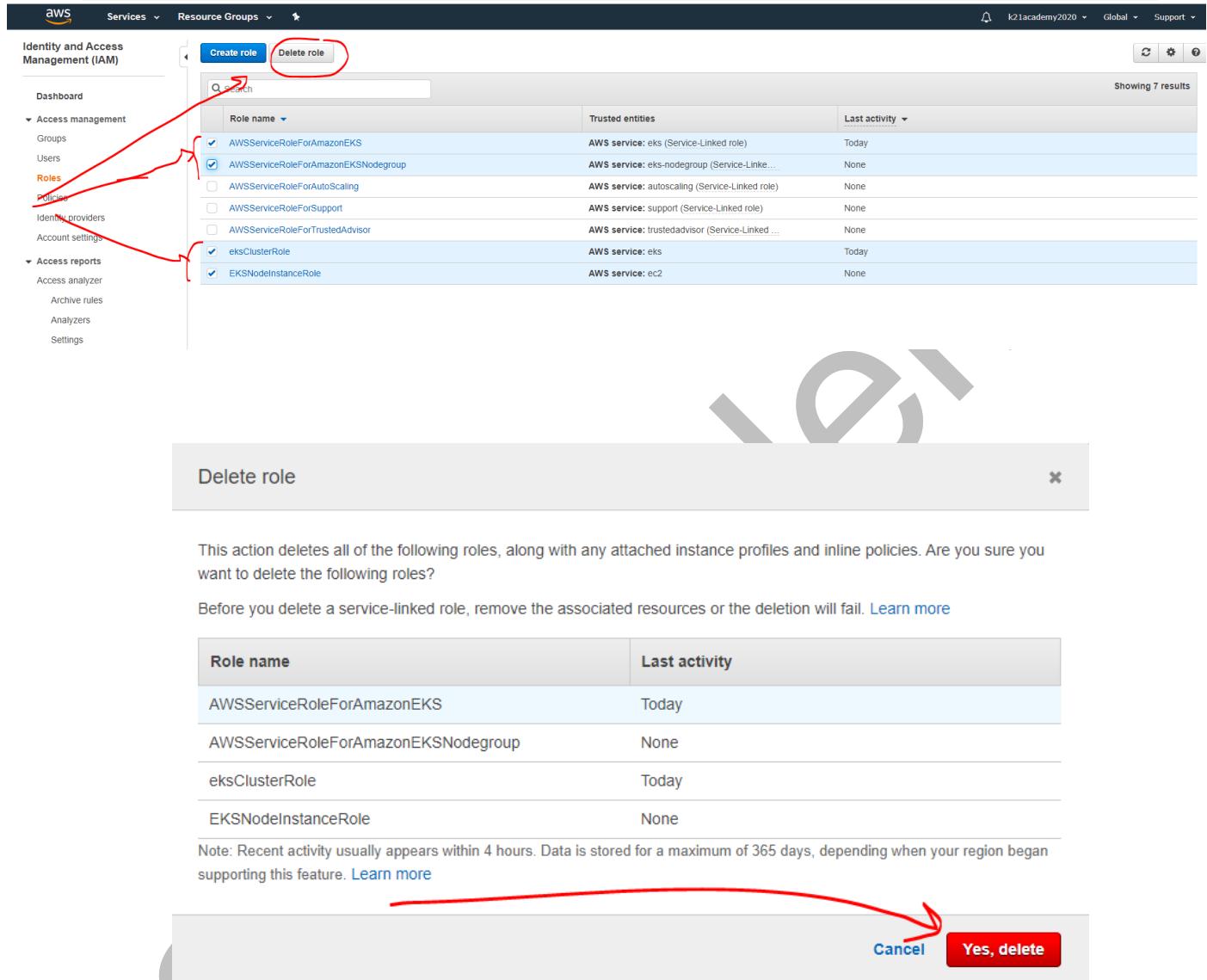
Delete Public_Key_For_EKS

To confirm deletion, type *delete* in the field

Cancel

Delete

7.4 Delete IAM Roles



The screenshot shows the AWS IAM Roles list page. The left sidebar has a red arrow pointing to the 'Roles' link under 'Access management'. The main content area shows a table of roles with several checked checkboxes. A red circle highlights the 'Delete role' button at the top right of the table. A large red arrow points from this button down to a 'Delete role' confirmation dialog. The dialog contains a list of roles to be deleted and a note about service-linked roles. At the bottom right of the dialog, there are 'Cancel' and 'Yes, delete' buttons, with a red arrow pointing to the 'Yes, delete' button.

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonEKS	AWS service: eks (Service-Linked role)	Today
AWSServiceRoleForAmazonEKSNodegroup	AWS service: eks-nodegroup (Service-Linked role)	None
AWSServiceRoleForAutoScaling	AWS service: autoscaling (Service-Linked role)	None
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked role)	None
eksClusterRole	AWS service: eks	Today
EKSNodeInstanceRole	AWS service: ec2	None

Delete role

This action deletes all of the following roles, along with any attached instance profiles and inline policies. Are you sure you want to delete the following roles?

Before you delete a service-linked role, remove the associated resources or the deletion will fail. [Learn more](#)

Role name	Last activity
AWSServiceRoleForAmazonEKS	Today
AWSServiceRoleForAmazonEKSNodegroup	None
eksClusterRole	Today
EKSNodeInstanceRole	None

Note: Recent activity usually appears within 4 hours. Data is stored for a maximum of 365 days, depending when your region began supporting this feature. [Learn more](#)

Cancel **Yes, delete**

8 SETUP EKS CLUSTER (MASTER & WORKER NODE) USING EKSCTL

8.1 Create EKS Master & Worker Nodes

1. Create your AWS EKS cluster with nodes (this will take 10-20 minutes)

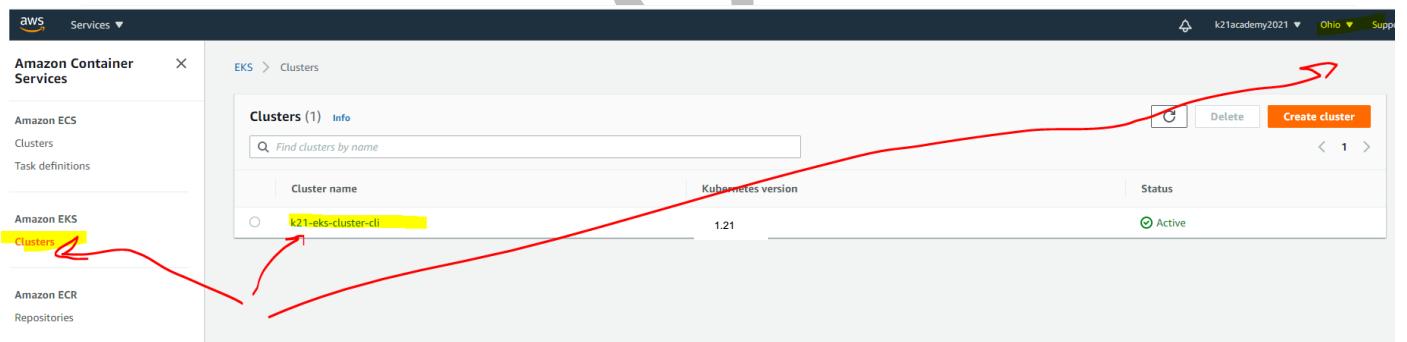
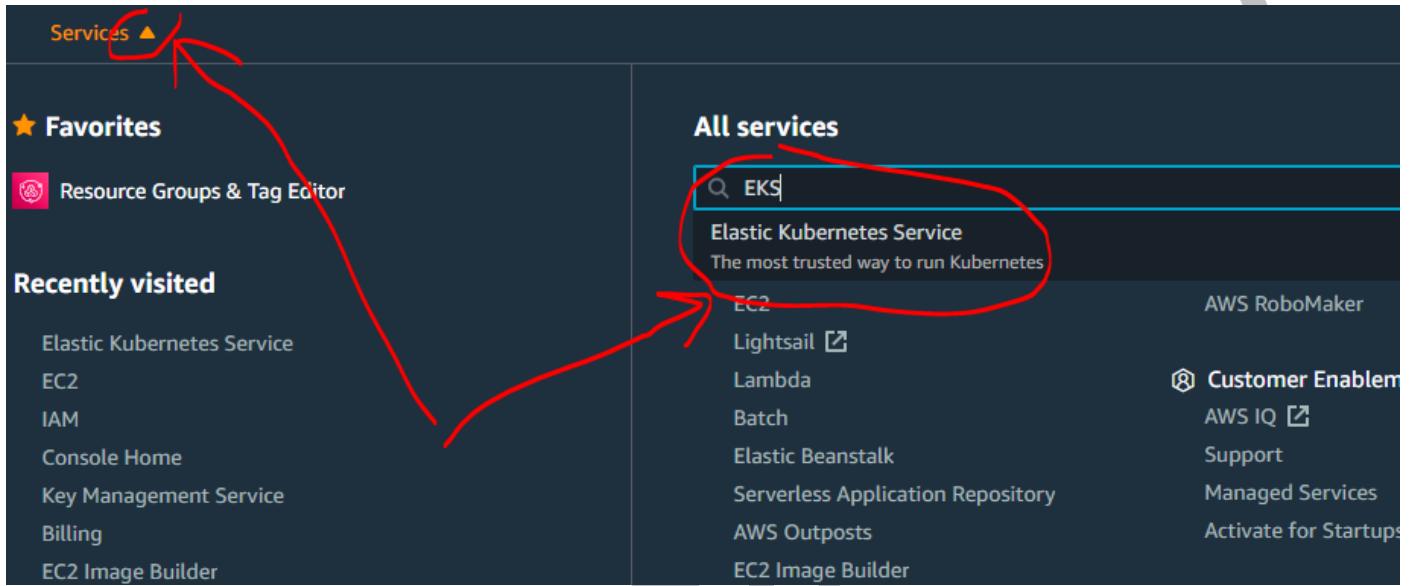
```
eksctl create cluster \
--name k21-eks-cluster-cli \
--version 1.21 \
--region us-east-2 \
--nodegroup-name linux-nodes \
--node-type t3.micro \
--managed
```

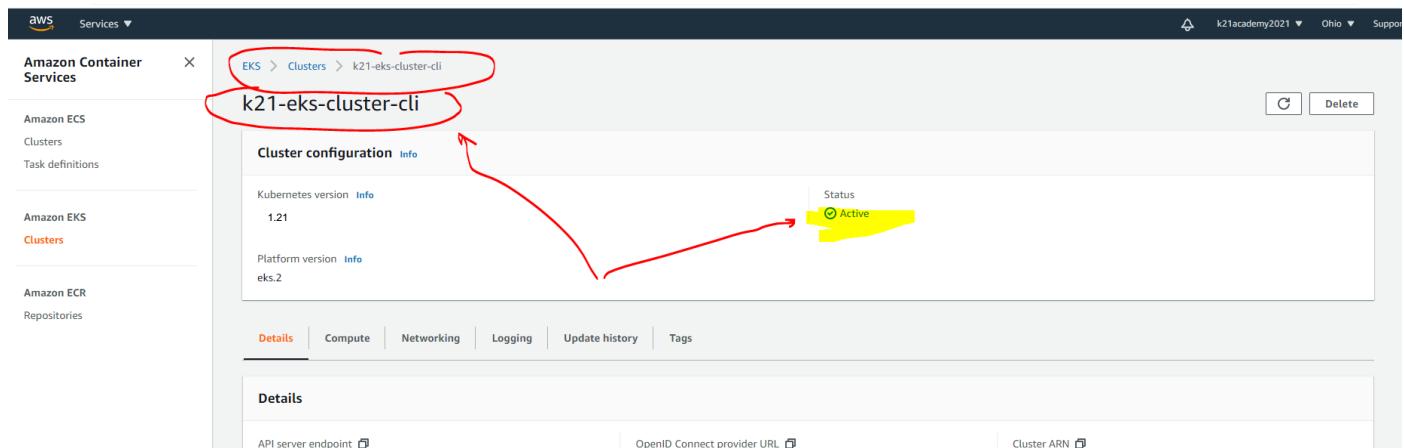
```
ubuntu@ip-172-31-33-247:~$ eksctl create cluster \
> --name k21-eks-cluster-cli \
> --version 1.21 \
> --region us-east-2 \
> --nodegroup-name linux-nodes \
> --node-type t3.micro \
> --managed

[!] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-2 --cluster=k21-eks-cluster-cli'
[!] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "k21-eks-cluster-cli" in "us-east-2"
[!] 2 sequential tasks: { create cluster control plane "k21-eks-cluster-cli", 2 sequential sub-tasks: { no tasks, create managed nodegroup "linux-nodes" } }
[!] building cluster stack "eksctl-k21-eks-cluster-cli-cluster"
[!] deploying stack "eksctl-k21-eks-cluster-cli-cluster"
[!] building managed nodegroup stack "eksctl-k21-eks-cluster-cli-nodegroup-linux-nodes"
[!] deploying stack "eksctl-k21-eks-cluster-cli-nodegroup-linux-nodes"
[!] waiting for the control plane availability...
[!] saved kubeconfig as "/home/ubuntu/.kube/config"
[!] no tasks
[!] all EKS cluster resources for "k21-eks-cluster-cli" have been created
[!] nodegroup "linux-nodes" has 2 node(s)
[!] node "ip-192-168-40-160.us-east-2.compute.internal" is ready
[!] node "ip-192-168-92-102.us-east-2.compute.internal" is ready
[!] waiting for at least 2 node(s) to become ready in "linux-nodes"
[!] nodegroup "linux-nodes" has 2 node(s)
[!] node "ip-192-168-40-160.us-east-2.compute.internal" is ready
[!] node "ip-192-168-92-102.us-east-2.compute.internal" is ready
[!] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
[!] EKS cluster "k21-eks-cluster-cli" in "us-east-2" region is ready
ubuntu@ip-172-31-33-247:~$
```

8.2 Verify Master & Worker Node from Console

1. Verify From EKS Cluster using Console





EKS > Clusters > k21-eks-cluster-cli

k21-eks-cluster-cli

Cluster configuration [Info](#)

Kubernetes version: [Info](#) 1.21

Platform version: [Info](#) eks.2

Status: Active

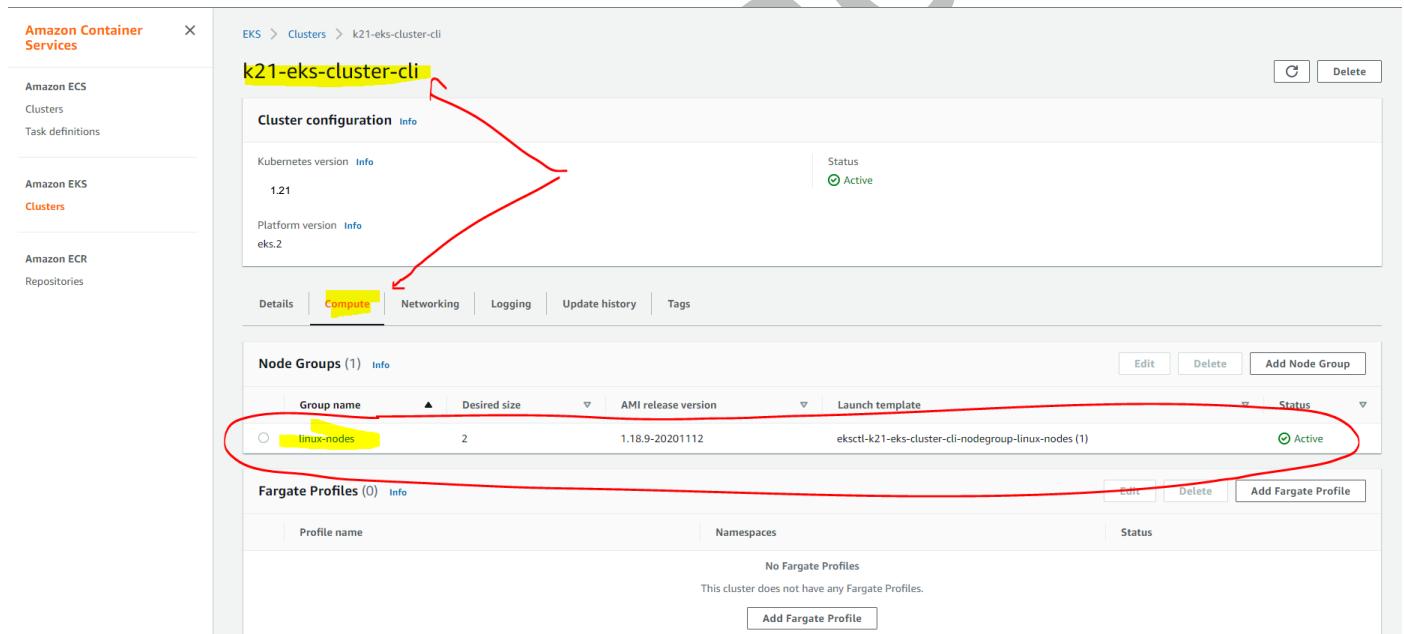
[Details](#) [Compute](#) [Networking](#) [Logging](#) [Update history](#) [Tags](#)

Details

API server endpoint [View](#) OpenID Connect provider URL [View](#) Cluster ARN [View](#)

a. Verify From EKS Nodes using Console

- Click On Compute
- Click on Node Group



EKS > Clusters > k21-eks-cluster-cli

k21-eks-cluster-cli

Cluster configuration [Info](#)

Kubernetes version: [Info](#) 1.21

Platform version: [Info](#) eks.2

Status: Active

[Details](#) [Compute](#) [Networking](#) [Logging](#) [Update history](#) [Tags](#)

Node Groups (1) [Info](#)

Group name	Desired size	AMI release version	Launch template	Status
linux-nodes	2	1.18.9-20201112	eksctl-k21-eks-cluster-cli-nodegroup-linux-nodes (1)	Active

Fargate Profiles (0) [Info](#)

No Fargate Profiles

This cluster does not have any Fargate Profiles.

[Add Fargate Profile](#)

Amazon Container Services X

EKS > Clusters > k21-eks-cluster-cli > Node Group: linux-nodes

linux-nodes

Node Group configuration [Info](#)

Kubernetes version 1.21	AMI type Info AL2_x86_64	Launch template eksctl-k21-eks-cluster-cli-nodegroup-linux-nodes	Status  Active
AMI release version Info 1.18.9-20201112	Instance type t3.micro	Launch template version 1	Disk size Specified in launch template

Details | [Health issues](#)  | [Kubernetes labels](#) | [Update history](#) | [Tags](#)

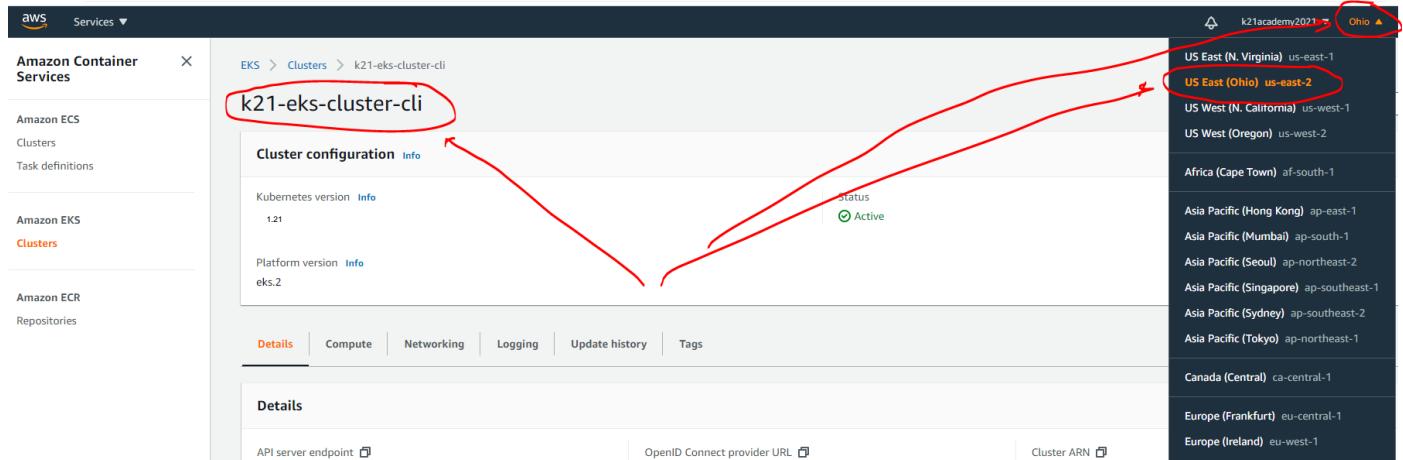
Details

Node Group ARN Info arn:aws:eks:us-east-2:265208284733:nodegroup/k21-eks-cluster-cli/linux-nodes/2ebaeb28-c9e2-d27d-d137-cb5e01ff51dc	Autoscaling group name eks-2ebaeb28-c9e2-d27d-d137-cb5e01ff51dc	Minimum size 2 nodes	Subnets subnet-028e0a9db64c12aa6  subnet-06f526640d8fee30d  subnet-01b35eff835dc2b78 
Node IAM Role ARN Info arn:aws:iam::265208284733:role/eksctl-k21-eks-cluster-cli-nodegr-NodeInstanceRole-9Y1YA92YP86P	Maximum size 2 nodes	Desired size 2 nodes	Allow remote access to nodes Disabled
Creation time Nov 17th 2020 at 1:07 PM			

K21 Academy

8.3 Setup up the kubectl configuration

1. Check your **Cluster Name & Region Name** where EKS Master node is running from console



The screenshot shows the AWS EKS service console. On the left, there's a sidebar with 'Amazon Container Services' (Amazon ECS Clusters, Amazon EKS Clusters), 'Amazon ECR Repositories'. In the center, under 'Clusters', it shows 'EKS > Clusters > k21-eks-cluster-cli'. The main panel displays 'Cluster configuration' with 'Kubernetes version: 1.21' and 'Platform version: eks.2'. Below this are tabs for 'Details', 'Compute', 'Networking', 'Logging', 'Update history', and 'Tags'. At the bottom are fields for 'API server endpoint', 'OpenID Connect provider URL', and 'Cluster ARN'. On the right, a dropdown menu lists various regions: US East (N. Virginia) us-east-1, US East (Ohio) us-east-2 (highlighted with a red circle), US West (N. California) us-west-1, US West (Oregon) us-west-2, Africa (Cape Town) af-south-1, Asia Pacific (Hong Kong) ap-east-1, Asia Pacific (Mumbai) ap-south-1, Asia Pacific (Seoul) ap-northeast-2, Asia Pacific (Singapore) ap-southeast-1, Asia Pacific (Sydney) ap-southeast-2, Asia Pacific (Tokyo) ap-northeast-1, Canada (Central) ca-central-1, Europe (Frankfurt) eu-central-1, and Europe (Ireland) eu-west-1. A red circle also highlights the 'Ohio' entry in the dropdown.

2. Check status of Cluster as

```
aws eks --region [EKS_Region] describe-cluster --name [EKS_Cluster_Name] --query cluster.status
```

```
aws eks --region us-east-2 describe-cluster --name k21-eks-cluster-cli --query cluster.status
```

```
ubuntu@ip-172-31-33-247:~$ aws eks --region us-east-2 describe-cluster --name k21-eks-cluster-cli --query cluster.status
"ACTIVE"
ubuntu@ip-172-31-33-247:~$
```

3. Configure kubectl with EKS API Server credential

```
aws eks --region [EKS_Region] update-kubeconfig --name [EKS_Cluster_Name]
```

```
aws eks --region us-east-2 update-kubeconfig --name k21-eks-cluster-cli
```

```
ubuntu@ip-172-31-33-247:~$ aws eks --region us-east-2 update-kubeconfig --name k21-eks-cluster-cli
Added new context arn:aws:eks:us-east-2:265208284733:cluster/k21-eks-cluster-cli
to /home/ubuntu/.kube/config
ubuntu@ip-172-31-33-247:~$
```

4. Validate kubectl configuration to master node

```
kubectl get svc
```

```
ubuntu@ip-172-31-33-247:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP     38m
```

5. Validate worker nodes

```
kubectl get nodes
```

```
ubuntu@ip-172-31-20-43:~$ kubectl get nodes --watch
NAME                      STATUS  ROLES   AGE  VERSION
ip-172-31-13-0.us-east-2.compute.internal  Ready  <none>  35m  v1.21.5-eks-9017834
ip-172-31-30-233.us-east-2.compute.internal  Ready  <none>  35m  v1.21.5-eks-9017834
```

6. Get all the details

```
kubectl get all
```

```
ubuntu@ip-172-31-33-247:~$ kubectl get all
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP     42m
ubuntu@ip-172-31-33-247:~$
```

8.4 Delete EKS Cluster using EKSCTL

1. Delete your AWS EKS cluster with nodes (this will take 10-20 minutes)

```
eksctl delete cluster \
--name k21-eks-cluster-cli \
--region us-east-2
```

```
ubuntu@ip-172-31-33-247:~/Kubernetes$ eksctl delete cluster \
> --name k21-eks-cluster-cli \
> --region us-east-2
[!] eksctl version 0.31.0
[!] using region us-east-2
[!] deleting EKS cluster "k21-eks-cluster-cli"
[!] deleted 0 Fargate profile(s)
[▼] kubeconfig has been updated
[!] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
[!] 2 sequential tasks: { delete nodegroup "linux-nodes", delete cluster controller plane "k21-eks-cluster-cli" [async] }
[!] will delete stack "eksctl-k21-eks-cluster-cli-nodegroup-linux-nodes"
[!] waiting for stack "eksctl-k21-eks-cluster-cli-nodegroup-linux-nodes" to get deleted
[!] will delete stack "eksctl-k21-eks-cluster-cli-cluster"
[▼] all cluster resources were deleted
ubuntu@ip-172-31-33-247:~/Kubernetes$
```

K21 Academy

9 TROUBLESHOOTING SECTION

9.1 CloudFormation Stack Already Exist

Issue: When we are creating EKSCTL cluster using EKSCTL and getting Cloudformation already exist error.

```
eksctl create cluster --name asif-eks-cli --version 1.18 --region us-east-1 --zones "us-east-1a us-east-1b us-east-1c" --nodegroup-name linux-nodes --node-type t3.medium --managed
```

Fix: We got this type of error because of the command we have entered, please check your command that you have entered, there should not be any extra spaces in command.

```
eksctl create cluster --name asif-eks-cli --version 1.18 --region us-east-1 --zones "us-east-1a,us-east-1b,us-east-1c" --nodegroup-name linux-nodes --node-type t3.medium --managed
```

10 SUMMARY

In this guide we covered:

Setting up AWS Kubernetes Cluster (EKS)

- Login to AWS portal and sign in using your account credentials.
- Create EKS Cluster Master Node
- Creating an EKS role
- Configure kubectl
- Create EKS Worker nodes
- Verifying Worker node status from kubectl
- Delete Master & Worker node of EKS

K21 Academy