**Load balancer**

An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances that are registered with the load balancer.

The Ingress Controller is an application that runs in a cluster and configures an HTTP load balancer according to Ingress resources. The load balancer can be a software load balancer running in the cluster or a hardware or cloud load balancer running externally.

A listener is a process that checks for connection requests. You define a listener when you create your load balancer, and you can add listeners to your load balancer at any time.

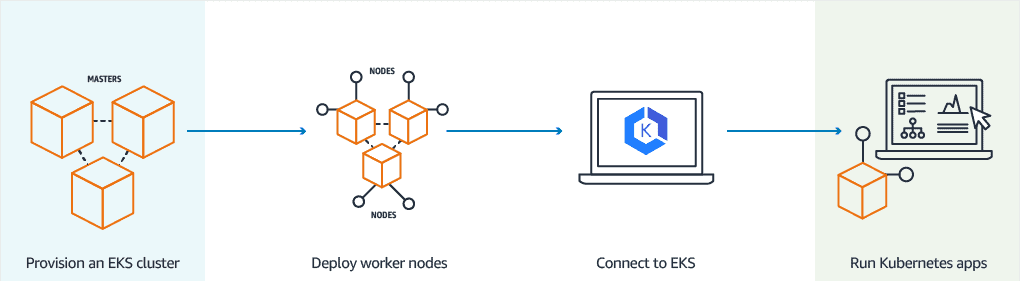
DNS name - k8s-notifica-notifica-9f6118cd12-747068323.us-east-2.elb.amazonaws.com

**Amazon EKS** is a managed service that is used to run Kubernetes on AWS. Using EKS users don’t have to maintain a Kubernetes control plan on their own. It is used to automate the deployment, scaling, and maintenance of the containerized application. It works with most operating systems.

EKS is integrated with various AWS services:

* ECR (Elastic Container Registry) for container images.
* Elastic Load Balancer for distributing traffic.
* IAM for providing authentication and authorization.
* VPC (Virtual Private Cloud) for isolating resources.

**Amazon EKS Workflow**



1. Provision EKS cluster using AWS Console, AWS CLI, or one of the AWS SDKs.
2. Deploy worker nodes to the EKS cluster. There is already a predefined template that will automatically configure nodes.
3. Now we configure Kubernetes tools such as kubctl to communicate with the Kubernetes cluster.
4. We are now all set to deploy an application on the Kubernetes cluster.

**Create EKS Kubernetes Cluster Using GUI**

The important steps involved are:

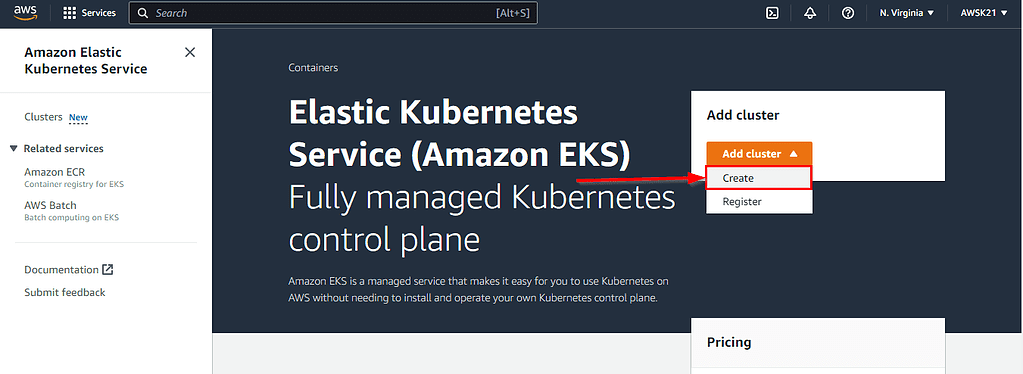
1. **Creating a Master Node**
2. **Installing and Configuring AWS CLI & kubectl**
3. **Creating a Worker Node**

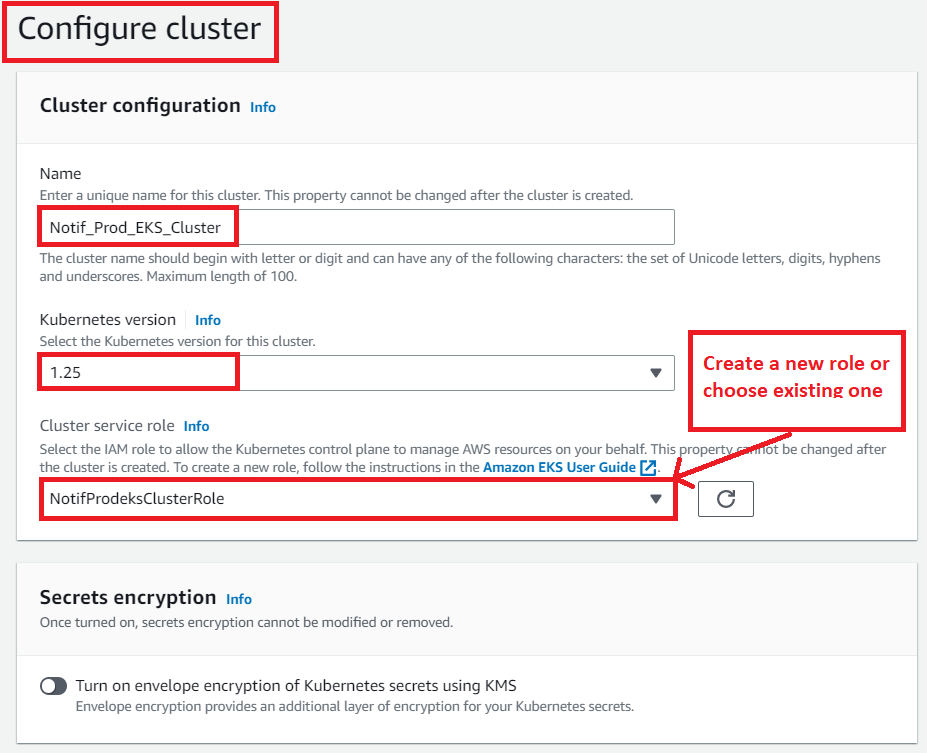
The detailed Steps are.

**Step 1:**The very first thing is to create an **AWS** account.

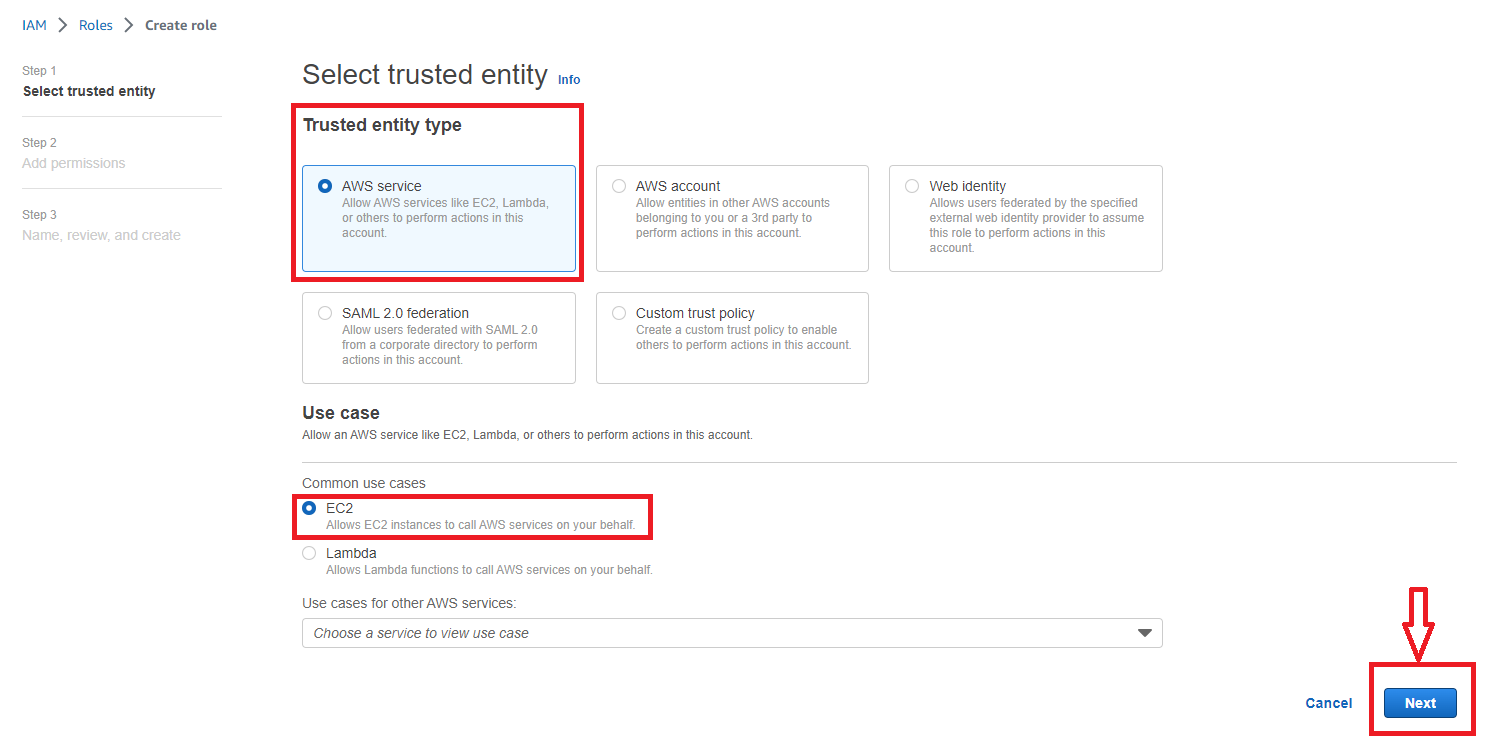
**Step 2:** Next step is to create a**Master Node**, follow the below steps to create one.

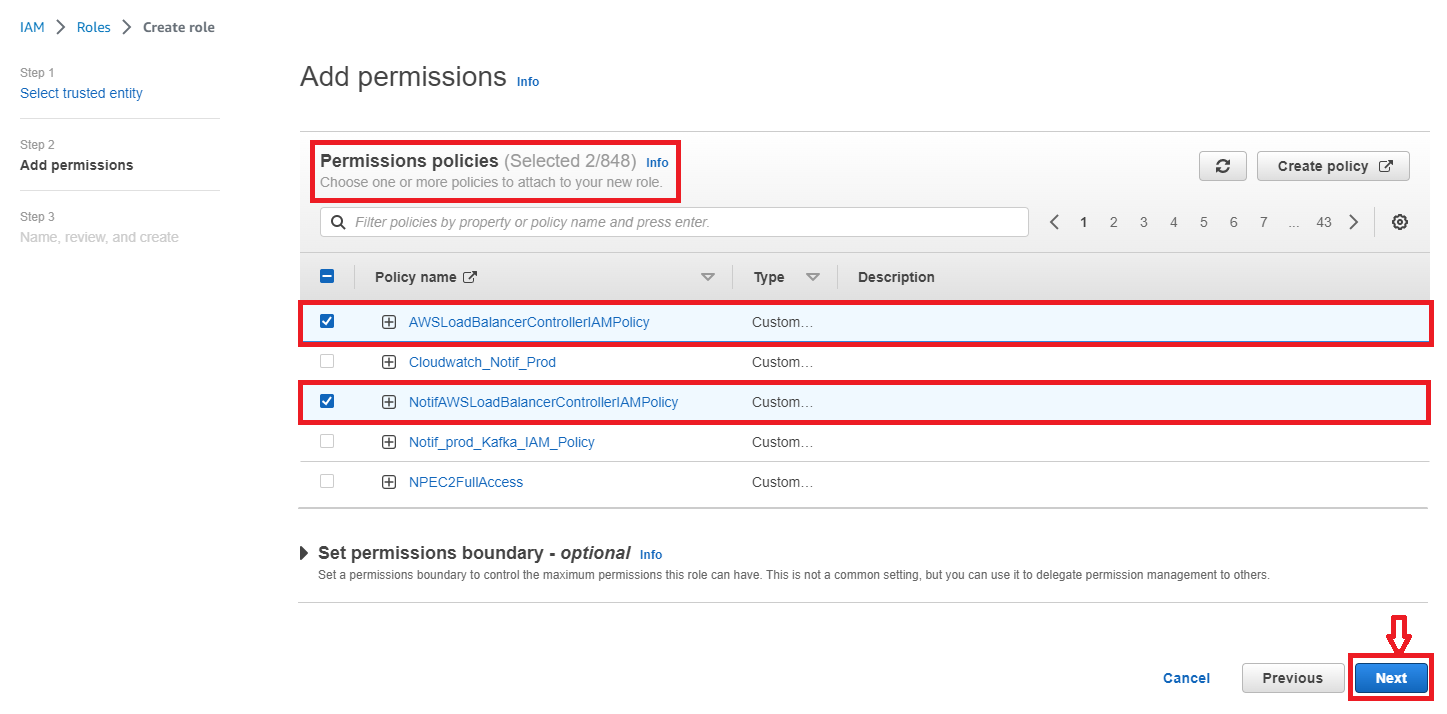
1. Log in to the AWS portal, find the Kubernetes Service by searching for **EKS** and click on **Create Kubernetes Cluster** and then specify the name for the Cluster.





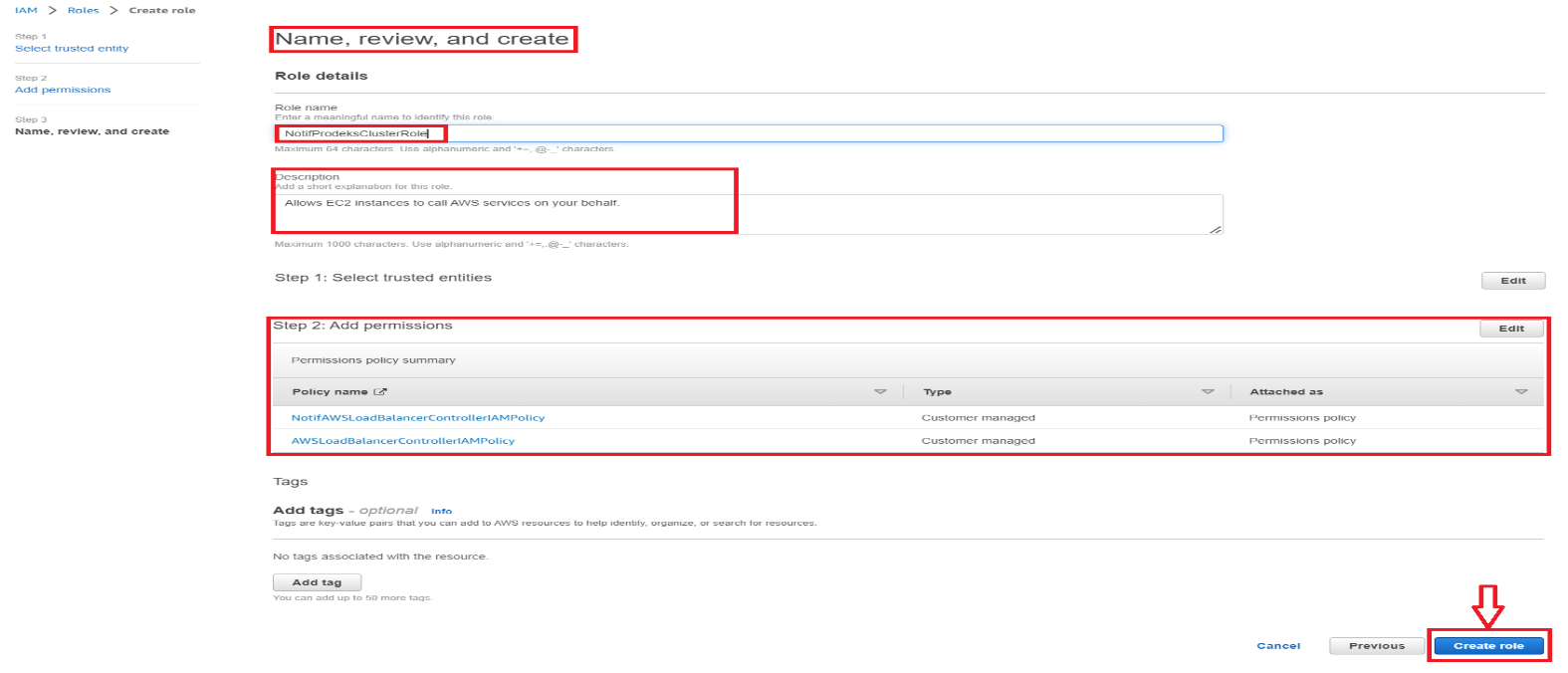
**b)**Next is to create the role, click on “Create role” -> AWS Service -> EKS (from AWS Services) -> Select EKS Cluster -> Next Permissions.



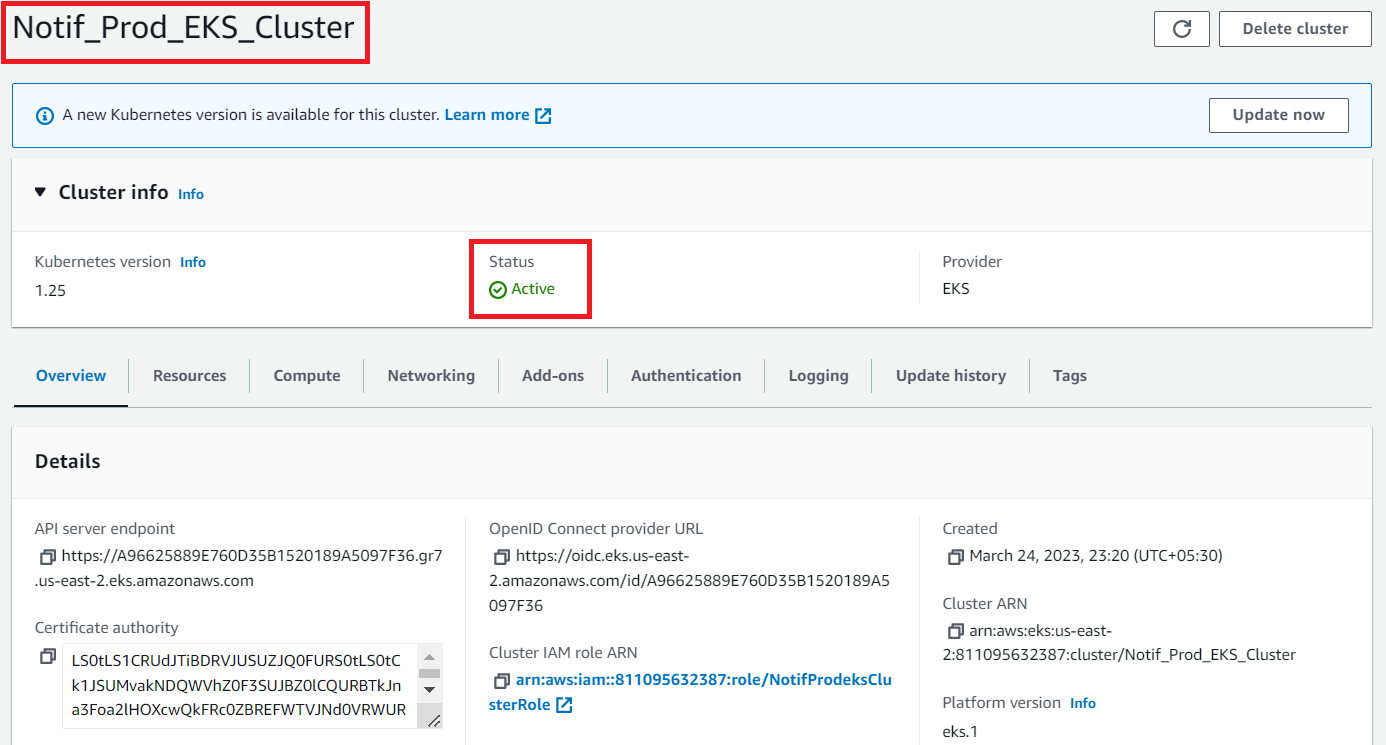


**c)**Leave the selected policies as-it-is and click on **Review** Page.

**d)**Enter a name for the role (e.g., **NotifProdeksClusterRole**) and hit the **Create role** button at the bottom of the page to create the IAM role. The IAM role is created.



**e)**Now Master node will be created in approximately **15-20** minutes.



**Step 3:**Next step is to Install & configure**AWS** CLI on the EC2 instance. Follow the below links and steps for the same.

* To complete this step, you need to create an Amazon Linux Instance first,
* To get the Latest EKS UserGuide visit Getting Started AWS Console
* Know more about the CLI UserGuide from CLI Configure Quickstart,

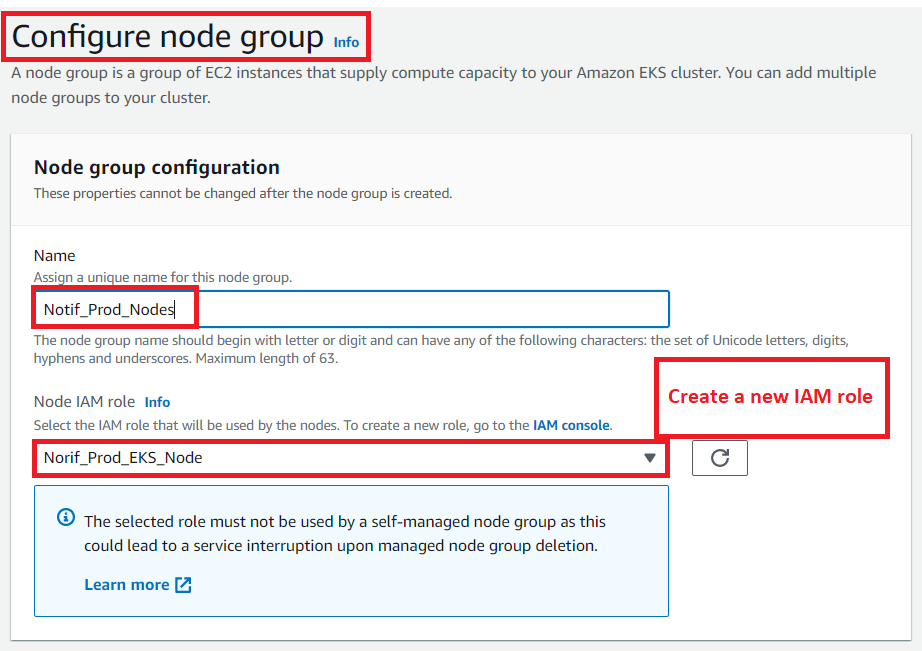
**Step 4:**Next is to install & configure the kubectl, by checking your Cluster Name & Region Name where the EKS Master node is running from the console. Check the status of Cluster and Configure kubectl with EKS API Server and validate kubectl configuration to master node.

**Step 5:**The final step is to create the **Worker Node.**

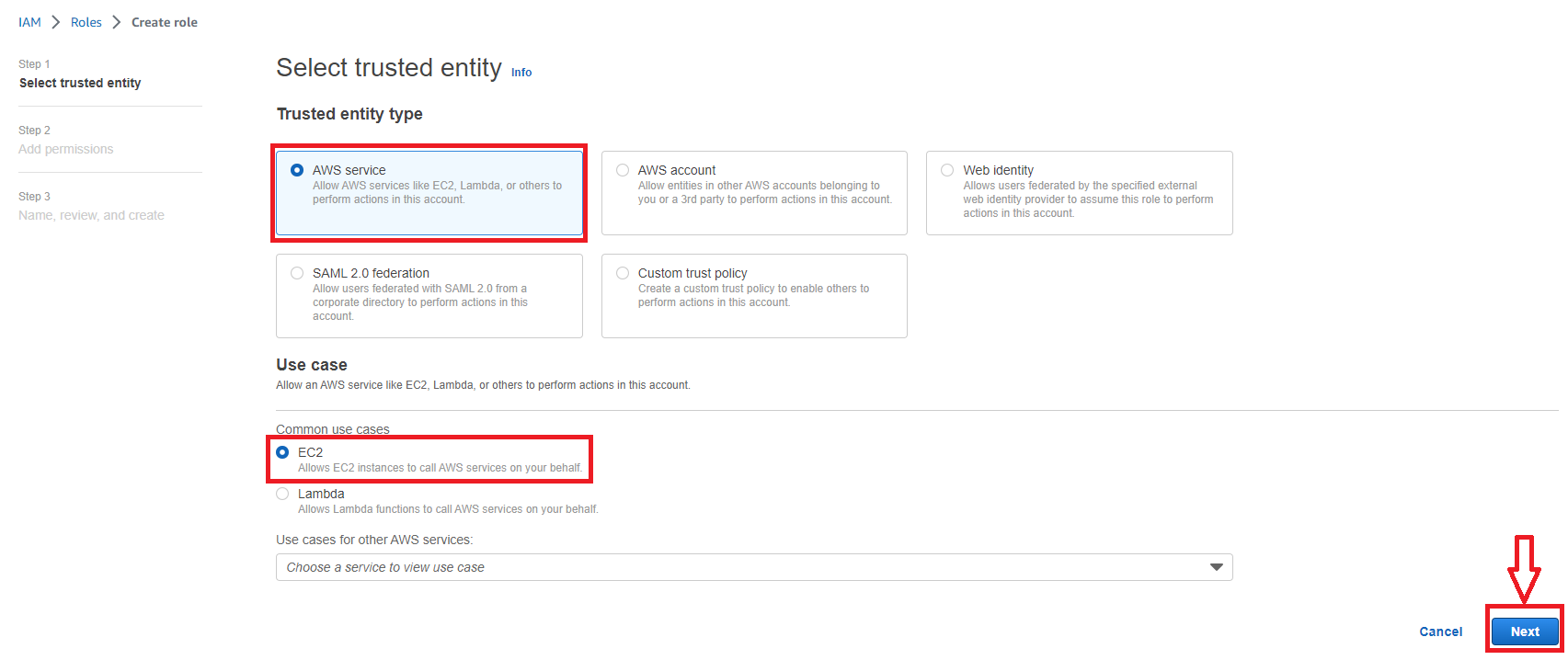
**a)**On the cluster page, select the Compute tab, and then choose Add Node Group.

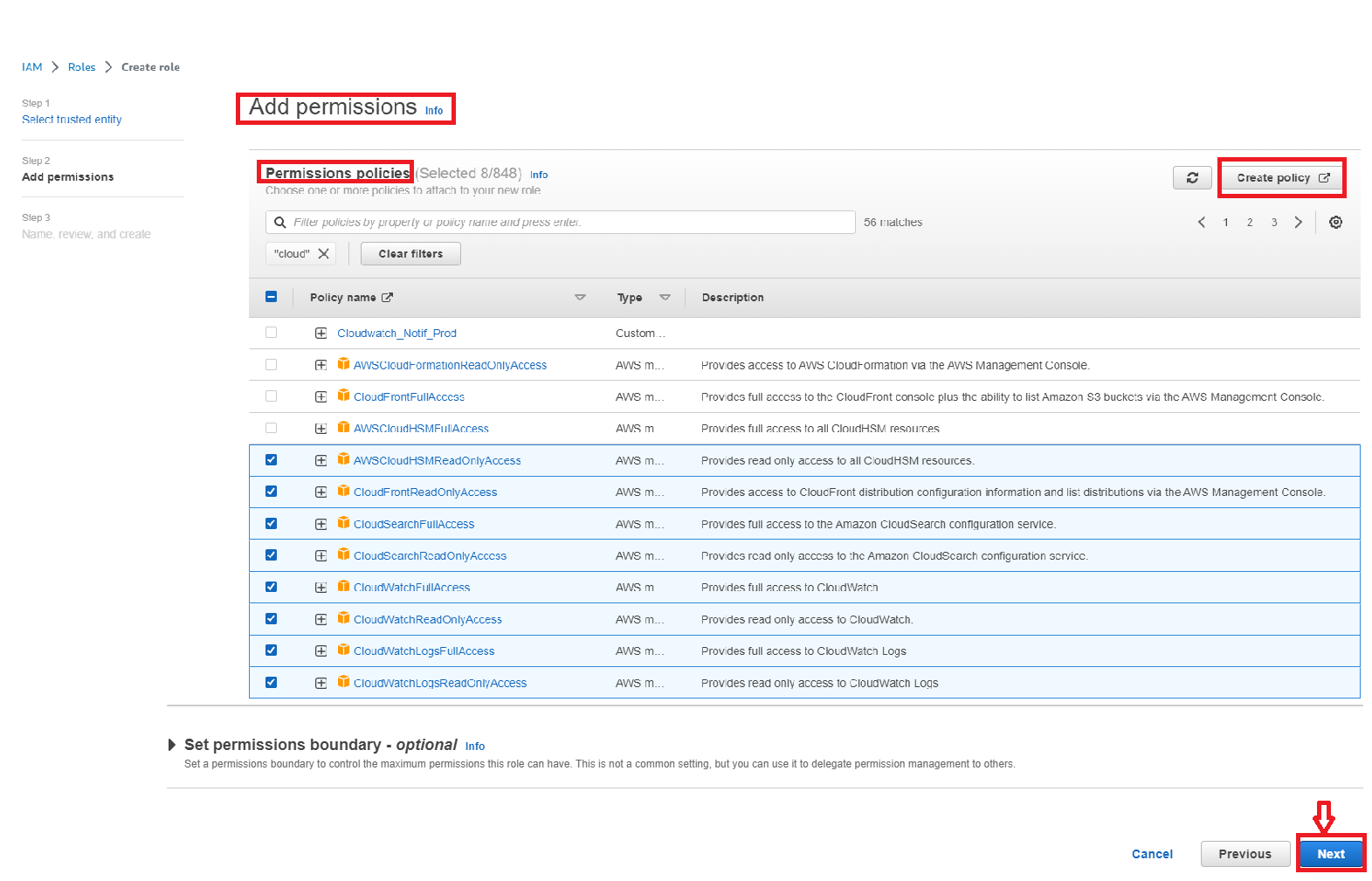
**b)**On the Configure node group page, fill out the parameters accordingly, and then choose Next.

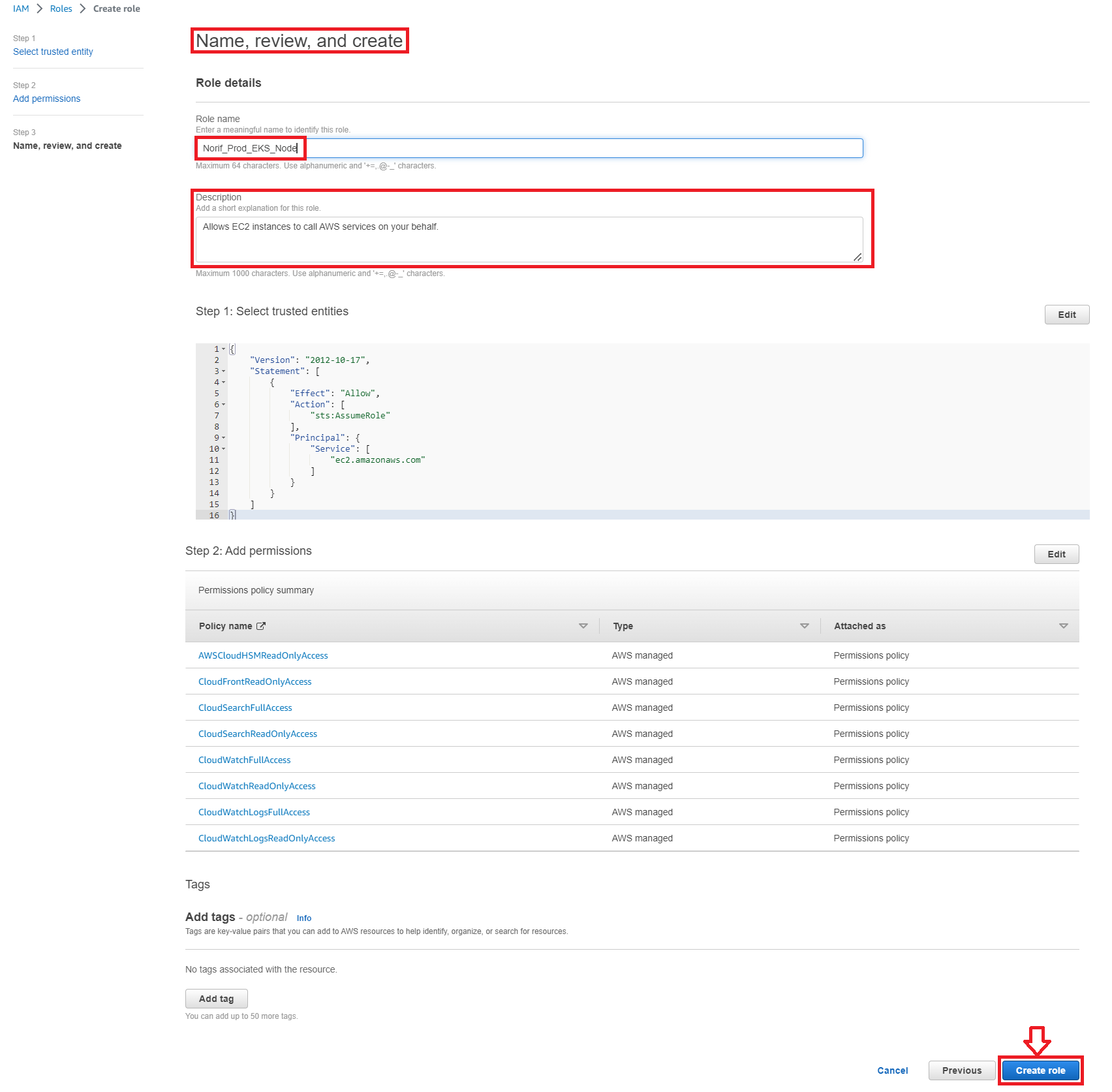
* **Name**- Enter a unique name for your managed node group.
* **Node IAM role name** - Choose the node instance role to use with your node group. For more information, see the Amazon EKS worker node IAM role,



To Create an IAM role follow the below-given steps, and to open the IAM Console,







After following all the above steps, leave the other settings to default and proceed further.

**Step 6:**The final step is to verify the Worker node status from Kubectl. For the same perform the given command:

**#kubectl get nodes --watch**

