**Create EKS cluster**: -

1. Login as IAM user
2. Create an IAM **policy** for EKS, select service as EKS

Graphical user interface, text, application, email

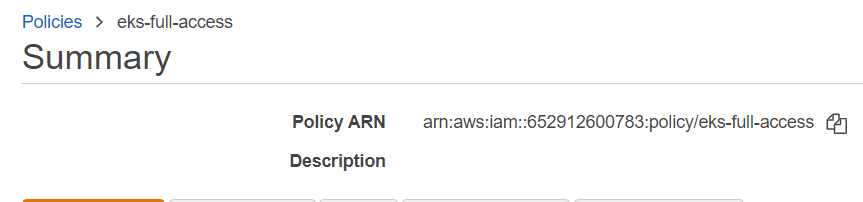
Description automatically generated

Select **resources as all**

A picture containing text

Description automatically generated

And create.



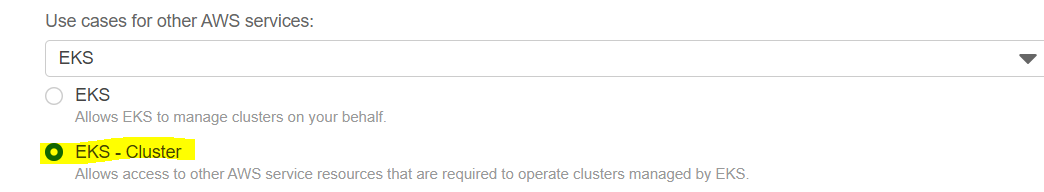
1. Create EKS cluster Role:

Choose **Roles**, then **Create role**.

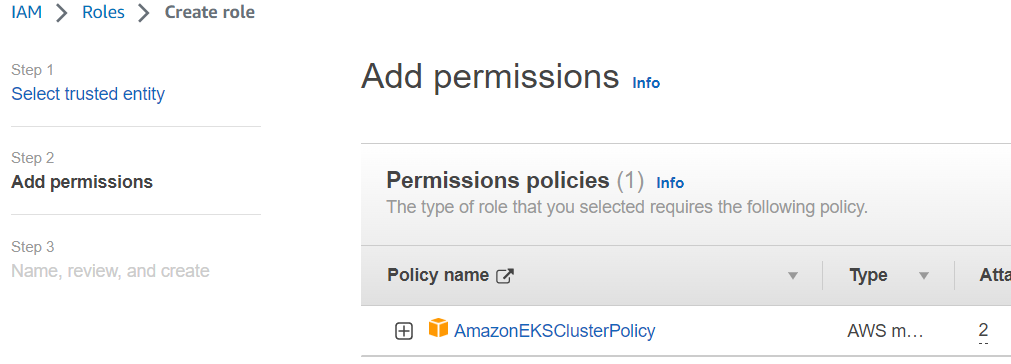
Under **Trusted entity type**, select **AWS service**.

From the **Use cases for other AWS services** dropdown list, choose **EKS**.

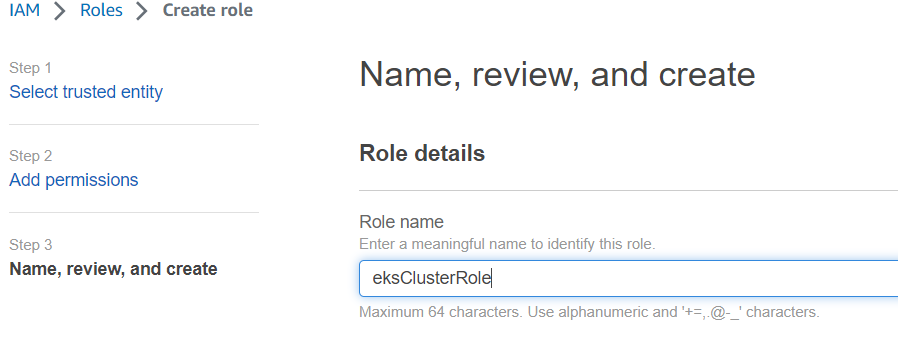
Choose **EKS - Cluster** for your use case, and then choose **Next**.



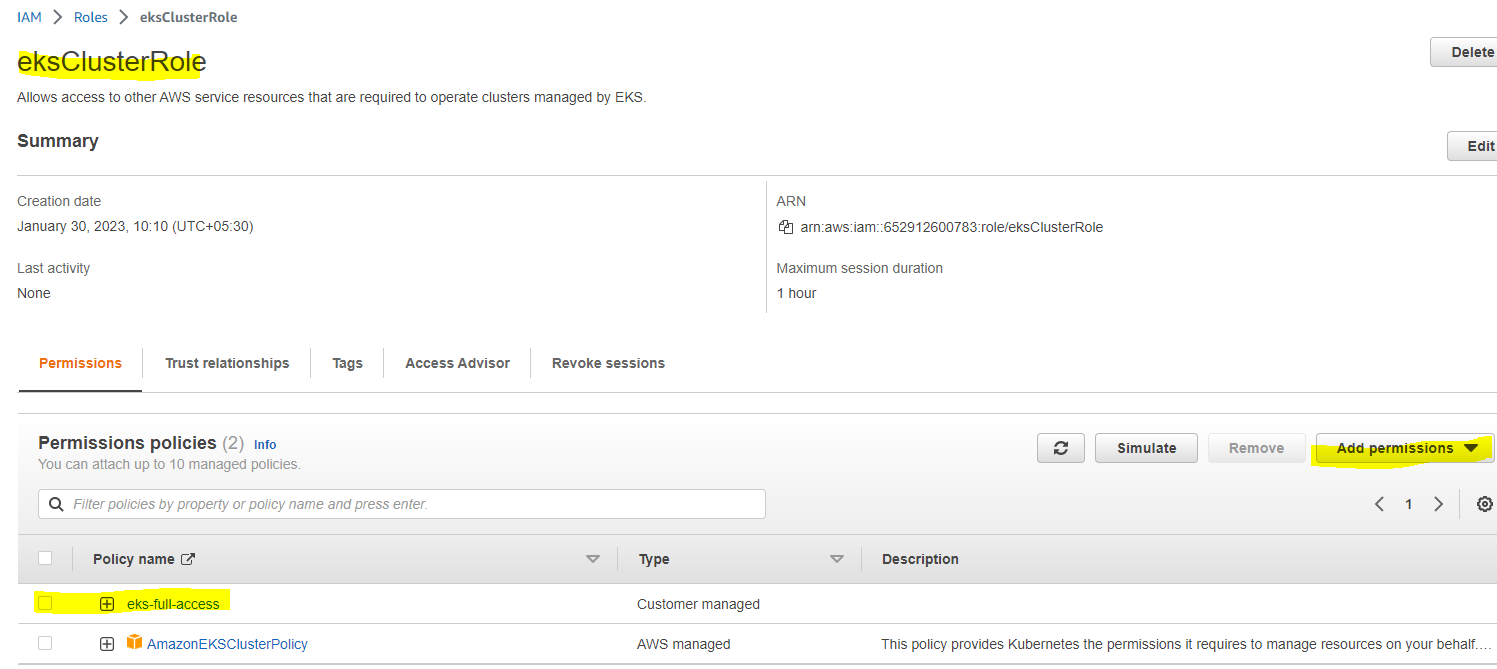
On the **Add permissions** tab, choose **Next** (by default EKS cluster policy will be selected)



For **Role name**, enter a unique name for your role, such as **eksClusterRole**.



Once your role is ready, search role in IAM role-> Add permission and search policy from step 2

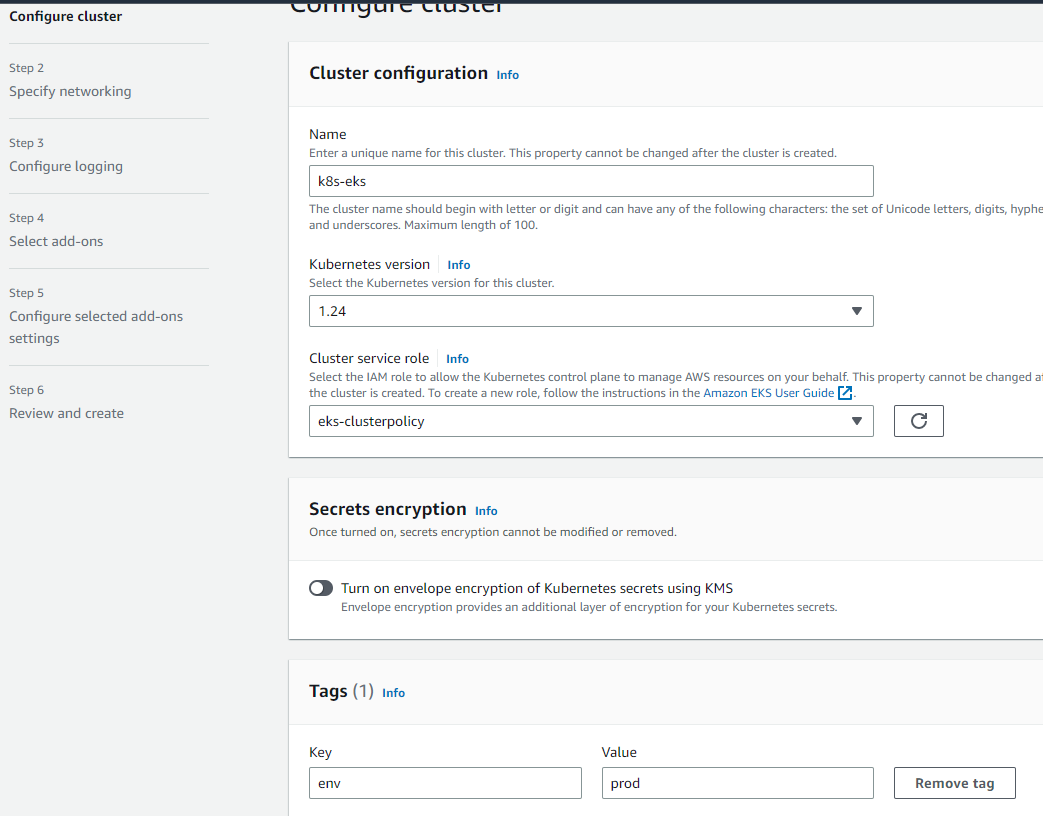


1. Create an EKS cluster and attach Role from previous steps.

Graphical user interface, text, application

Description automatically generated

Assign tag your EKS cluster i.e “env=prod”



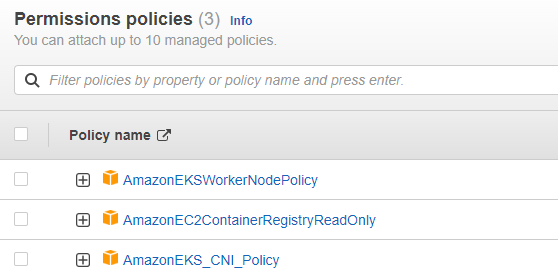
Go ahead with the default setting and create EKS control panel (you can change setting as per your requirements)

1. Create an IAM role AmazonEKSNodeRole

(so that Node can connect to EKS control panel & ECR container registry)

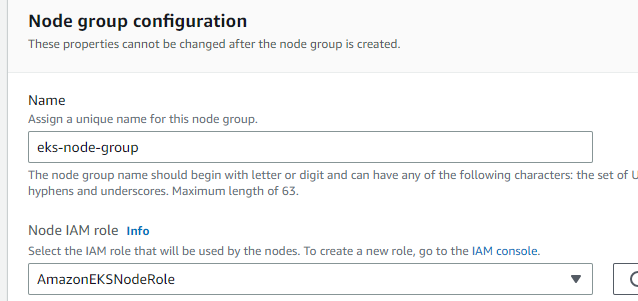
Go to the IAM role -> Create New Role -> select EC2 -> attach below policy to this Role.

([AmazonEKSWorkerNodePolicy](https://us-east-1.console.aws.amazon.com/iam/home" \l "/policies/arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy" \t "_blank) , [AmazonEC2ContainerRegistryReadOnly](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly), [AmazonEKS\_CNI\_Policy](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy))

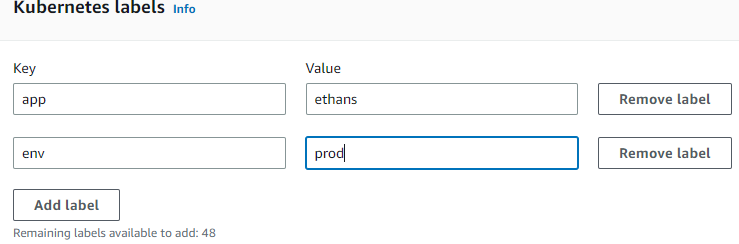


1. Once EKS cluster is ready, Create EKS node group (worker Node Group, where k8s runs pod/container)

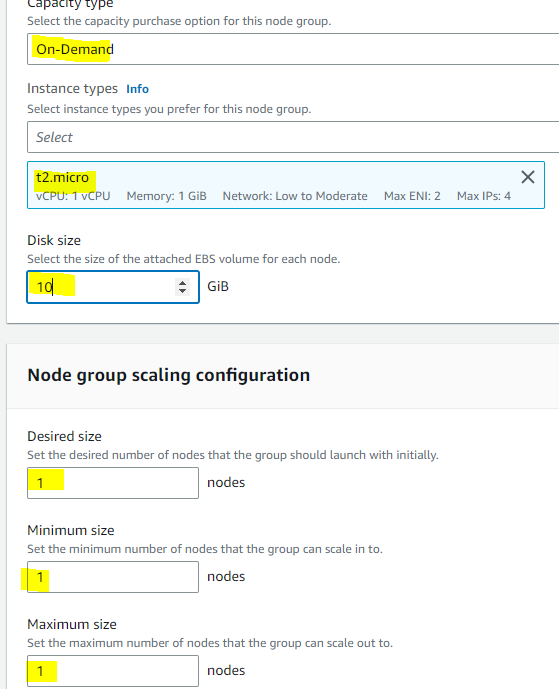
**EKS cluster-> Compute-> Add node group -> attach previously created Role “**AmazonEKSNodeRole**”**



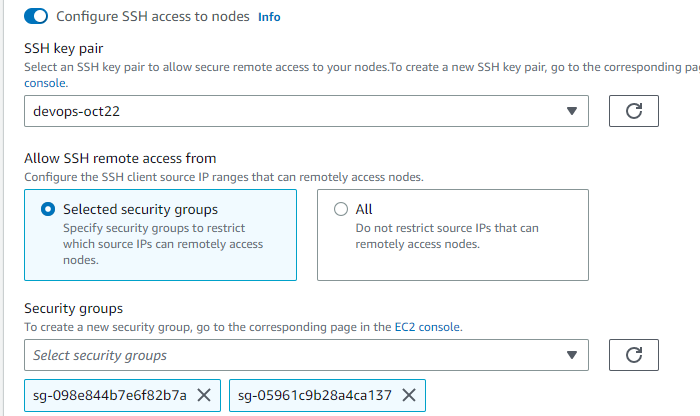
**Assign Kubernetes labels** (as per your application or environment or some meaningful)



Select instance type as t3.small/t2micro or any other based on your requirements.



You can also enable ssh connection. (so that you can connect to worker node and troubleshoot if required)



1. You can connect to EKS cluster, once EKS node group is ready

Open AWS cloudshell on AWS portal/ or powershell or cmd on your windows OS



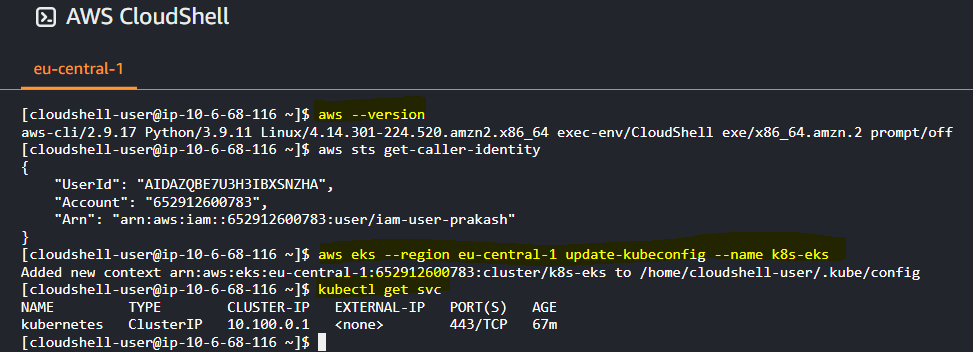
Cloud shell in available on AWS portal beside your region in the right top corner.

configure your IAM credentials using Access key & Secret key

**aws configure**

and provide Access key , secret key and default EKs region

------From Cloud shell



follow below link to connect EKS from CloudShell or PowerShell

<https://aws.amazon.com/premiumsupport/knowledge-center/eks-cluster-connection/>

aws --version

aws sts get-caller-identity

aws eks --region <**region**> update-kubeconfig --name <**cluster\_name**>

ex: aws eks --region eu-central-1 update-kubeconfig --name k8s-eks

kubectl get pods --kubeconfig ./.kube/config

kubectl get svc

---------From powershell

Text

Description automatically generated