

Create The K8s EKS and Deploy the nginx Application

Create a IAM and add policy to the user to access the EKS cluster

The screenshot shows the AWS IAM console 'Create user' page. The breadcrumb navigation is 'IAM > Users > Create user'. On the left, a sidebar shows the steps: Step 1 (Specify user details, selected), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled 'Specify user details'. Under 'User details', the 'User name' field contains 'EKS-User'. Below this, there is a checkbox for 'Provide user access to the AWS Management Console - optional' which is unchecked. A note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)'. Another note says: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A blue information box contains text about generating access keys for programmatic access. At the bottom right, there are 'Cancel' and 'Next' buttons.

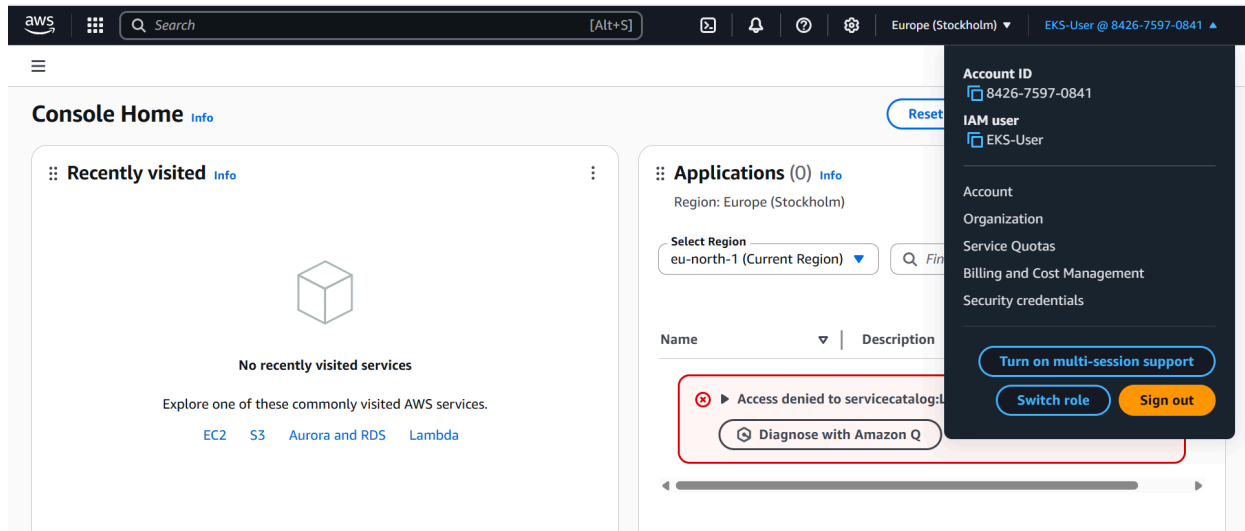
Providing Policy to the user and create the user

The screenshot shows the AWS IAM console 'Create user' page at Step 3: Review and create. The breadcrumb navigation is 'IAM > Users > Create user'. The sidebar shows Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create, selected), and Step 4 (Retrieve password). The main content area shows a summary of the user and permissions. The 'User name' is 'EKS-User'. The 'Console password type' is 'Custom password'. The 'Require password reset' is 'No'. Below this is a 'Permissions summary' table with 3 columns: Name, Type, and Used as. The table lists 8 AWS managed policies.

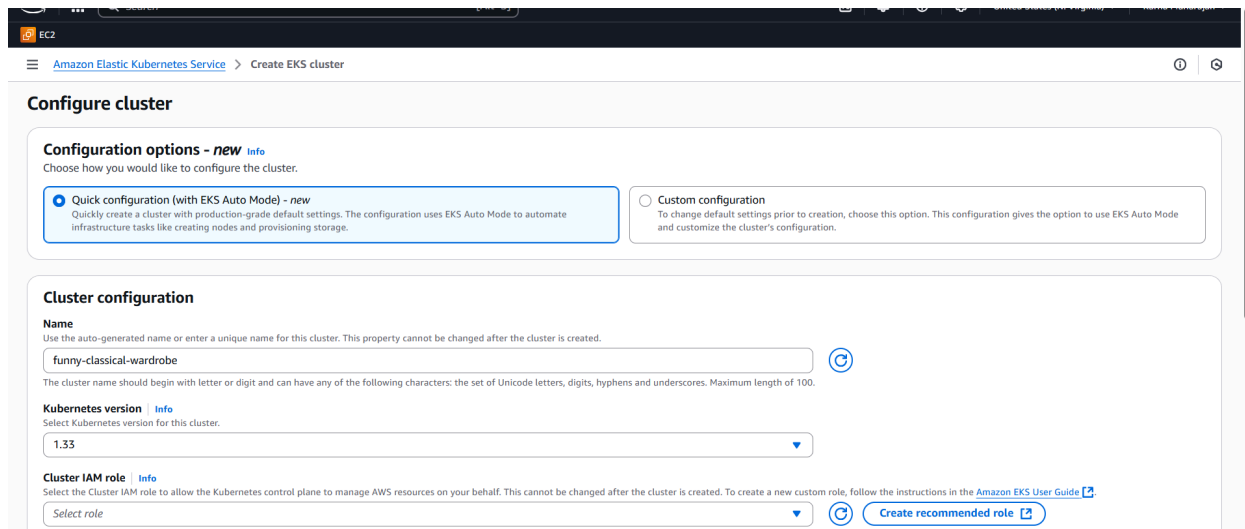
Name	Type	Used as
AmazonEC2ContainerRegistryFullAccess	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonECS_FullAccess	AWS managed	Permissions policy
AmazonEKSClusterPolicy	AWS managed	Permissions policy
AmazonEKSServicePolicy	AWS managed	Permissions policy
AmazonVPCFullAccess	AWS managed	Permissions policy
AWSCloudFormationFullAccess	AWS managed	Permissions policy
IAMFullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Once the user is created, the login with the user credential.

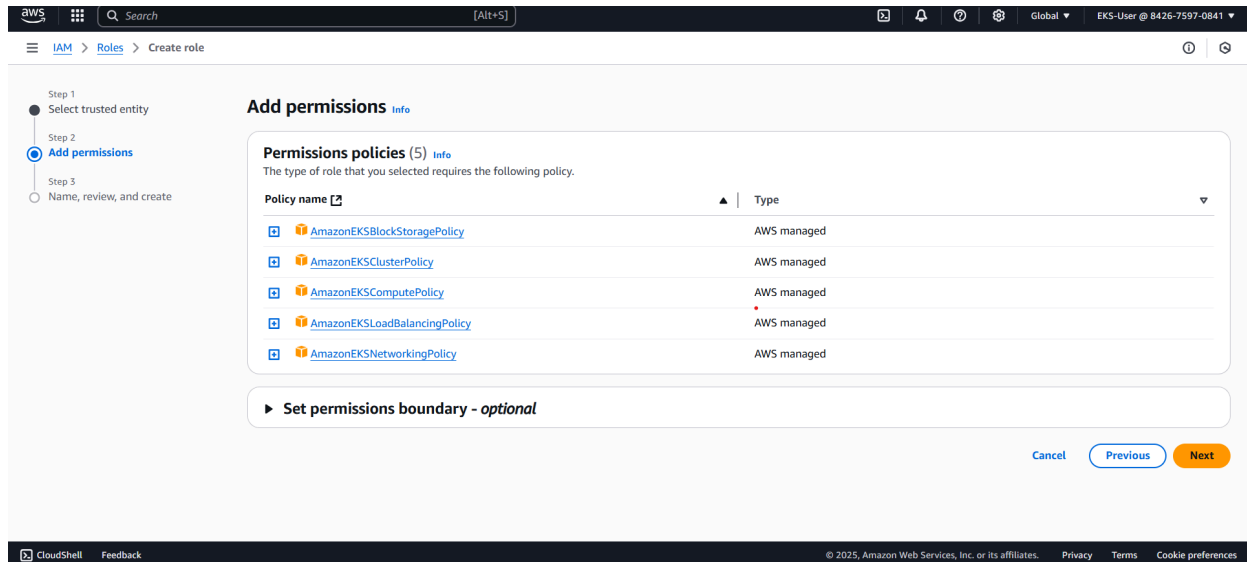
Logging with an IAM user.



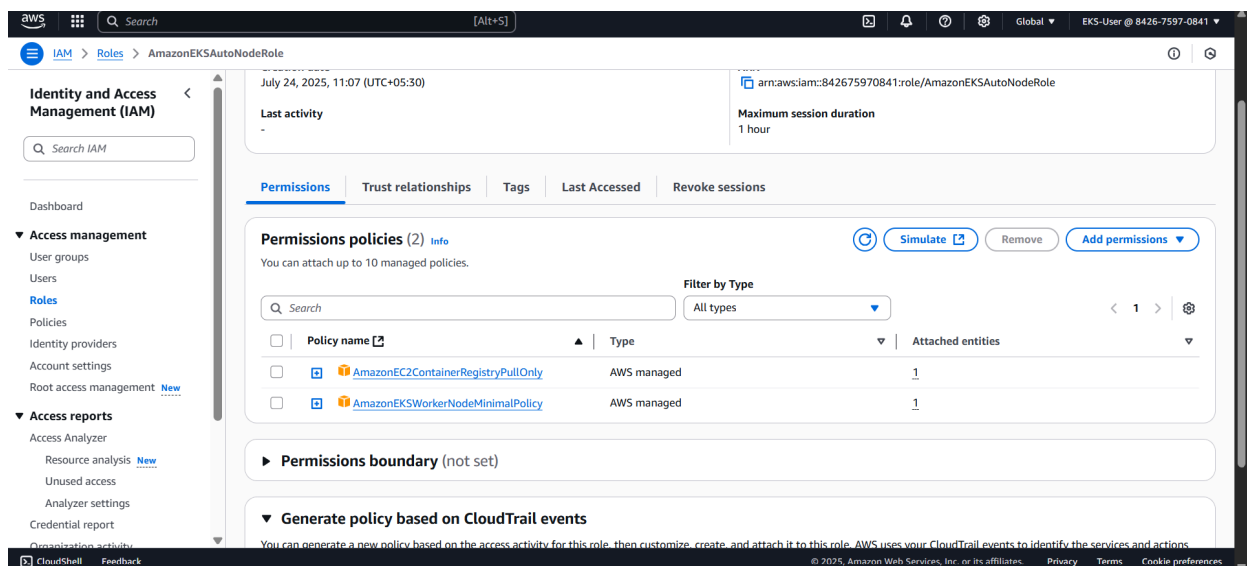
Create EKS Cluster and provide the required configuration.



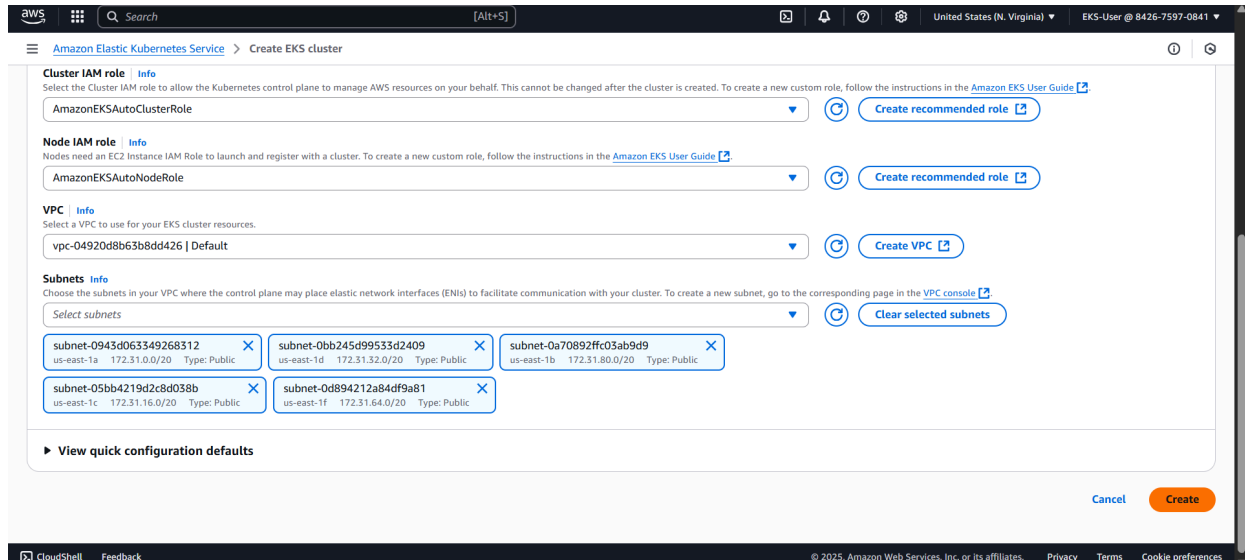
Adding Cluster IAM role.



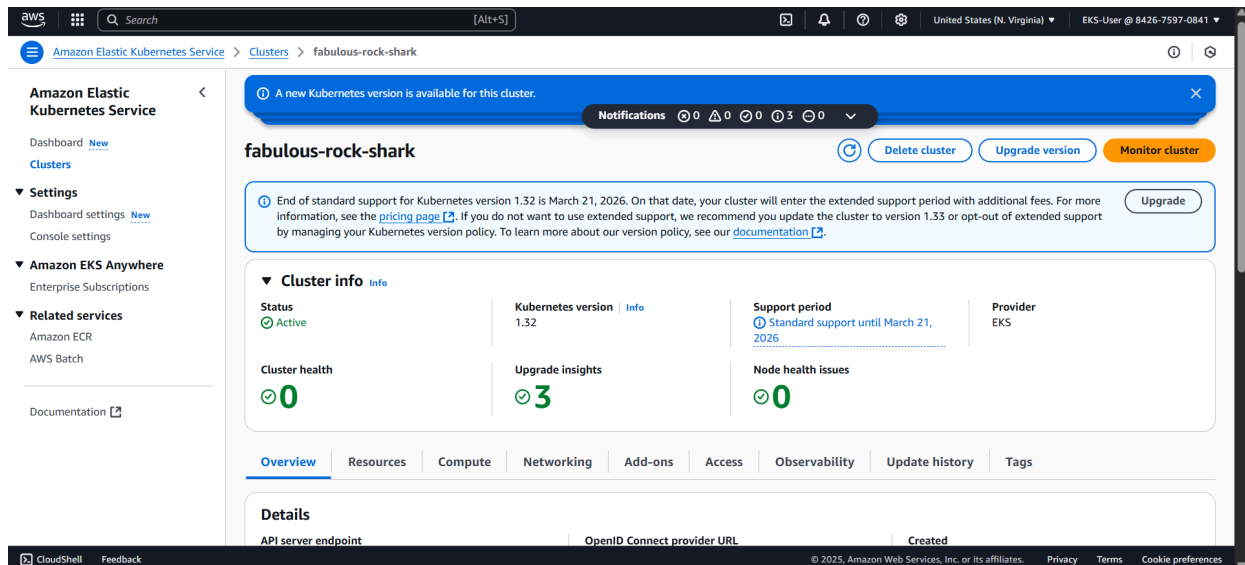
Creating the cluster IAM role.



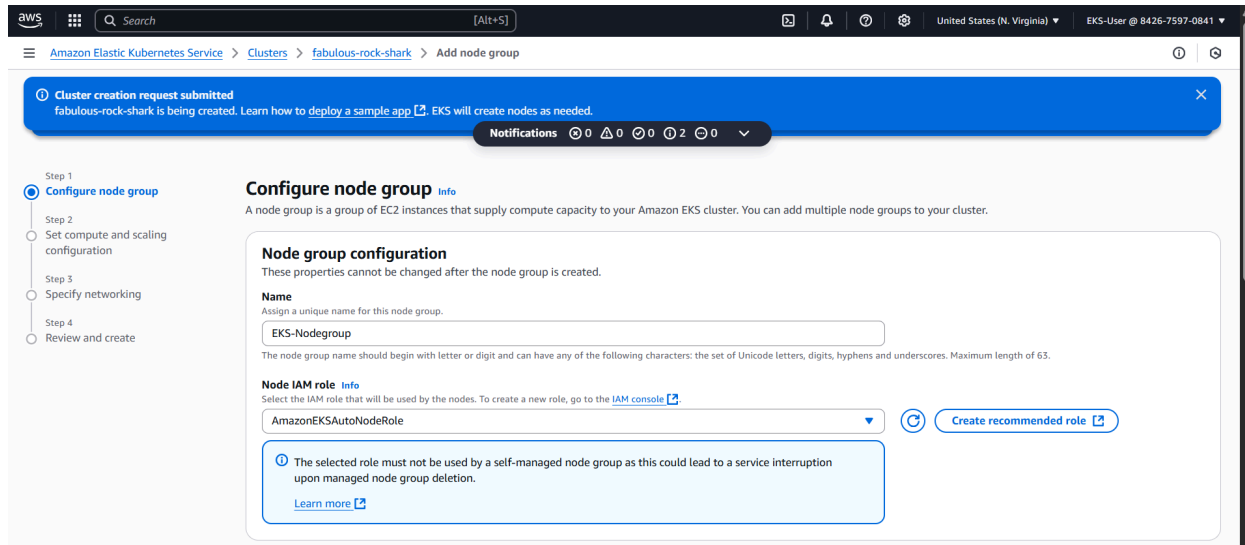
Add Node IAM role and add default vpc and subnet



Finally the Cluster is Created



Once the Cluster is Created, create nodegroup in the cluster compute setting.



Nodes are also created.

Finally switch to console create the deployment and services,

Login with aws access ID and Secret key

```
PS C:\K8-D> cd ..
PS C:\> aws configure
AWS Access Key ID [None]: AKIA4IM3G4MMURCXGUET
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
PS C:\> aws sts get-caller-identity
{
  "UserId": "AIDA4IM3G4MMWR4MAR6Q3",
  "Account": "842675970841",
  "Arn": "arn:aws:iam::842675970841:user/EKS-User"
}
PS C:\> |
```

Below picture shows how the deployment and service are created.

```
PS C:\> aws sts get-caller-identity
{
  "UserId": "AIDA4IM3G4MMWR4MAR6Q3",
  "Account": "842675970841",
  "Arn": "arn:aws:iam::842675970841:user/EKS-User"
}

PS C:\> aws eks update-kubeconfig --region us-east-1 --name fabulous-rock-shark
Added new context arn:aws:eks:us-east-1:842675970841:cluster/fabulous-rock-shark to C:\Users\Happy\.kube
\config
PS C:\> kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
i-0da7cac0eaaaa4bdc Ready     <none>    72m   v1.32.5-eks-98436be
PS C:\> kubectl create deployment nginx-deployment --image=nginx --replica=2
error: unknown flag: --replica
See 'kubectl create deployment --help' for usage.
PS C:\> kubectl create deployment nginx-deployment --image=nginx --replicas=2
deployment.apps/nginx-deployment created
PS C:\> kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
nginx-deployment-6cfb98644c-kzk78   0/1      Pending   0           19s
nginx-deployment-6cfb98644c-l7l1cg  0/1      Pending   0           19s
PS C:\> kubectl expose deployment nginx-deployment --port=80 --type=LoadBalancer
service/nginx-deployment exposed
PS C:\> kubectl get services
NAME                TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP      10.100.0.1   <none>        443/TCP          84m
nginx-deployment    LoadBalancer  10.100.241.88 <pending>     80:31791/TCP     25s
```

The nginx is created and exposed with Loadbalancer...

```
=====#####=====#####=====#####=====
=====
```