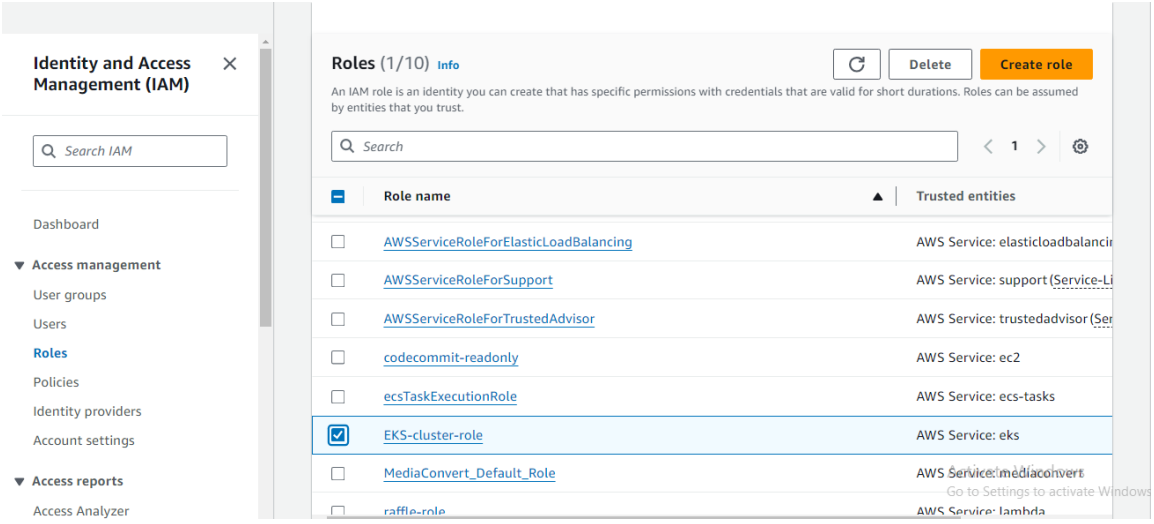
Creating a Scalable Application Using AWS EKS via AWS Console

Project Summary:

- learning how to create an EKS Cluster through the frontend (AWS Console) with all its configurations, make it scalable using Cluster-AutoScaler,and deploy an application on it.
- Creating the components needed by the EKS Cluster, which are: IAM Role and VPC.
- Creating the EKS Cluster and connect it to CloudShell.
- Creating the IAM Role required by the Node Group and create the Node Group itself.
- Setting up the Auto Scaling for the Node Group.
- Finally, adding a demo site to the EKS Cluster and test how it scale and adapts to incoming load.

Steps:

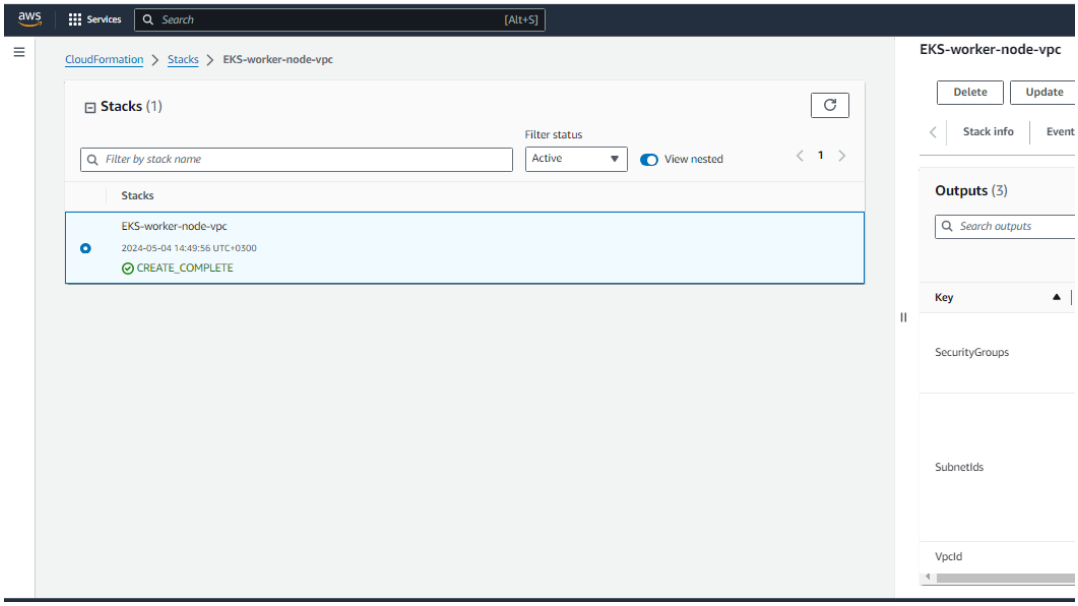
1- create a IAM role for EKS by IAM service

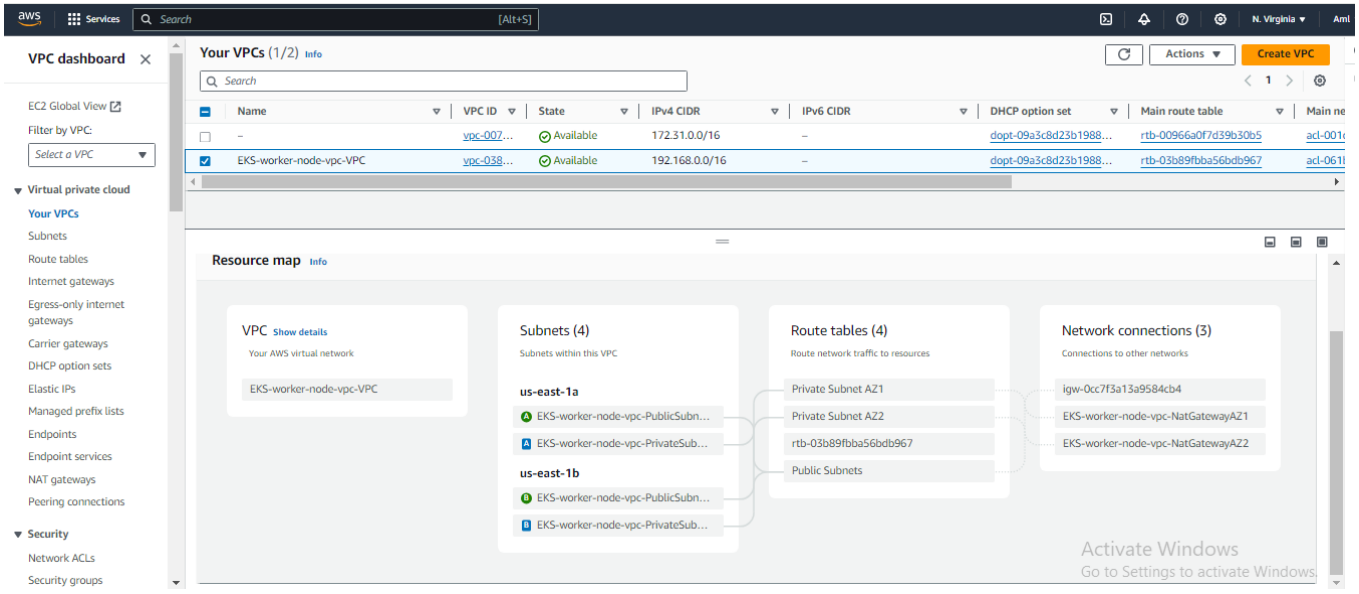


2- **<u>Create a VPC for worker nodes with special specification</u>**:
I'll use CloudFormation service for creating this VPC with these resources:

- pair of public and private subnets
- two Availability Zones
- internet gateway, with a default route on the public subnets.
- a pair of NAT gateways (one in each AZ)

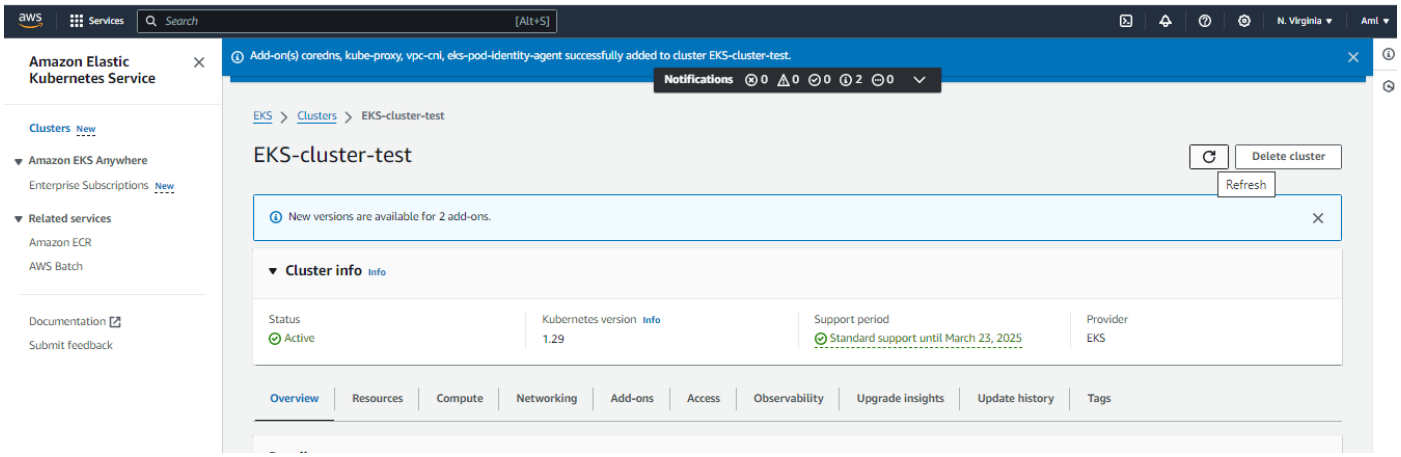
After choosing from the console cloudformation → create stack → Choose an existing template →Upload a template file →upload file (previous prepared) →submit





3- <u>Creating EKS cluster and connect to it by CloudShell:</u>

Choose EKS from console → Add cluster → create →choose eks-cluster-role (pre-created) → choose VPC (pre-created) → choose security group (pre-created) → create



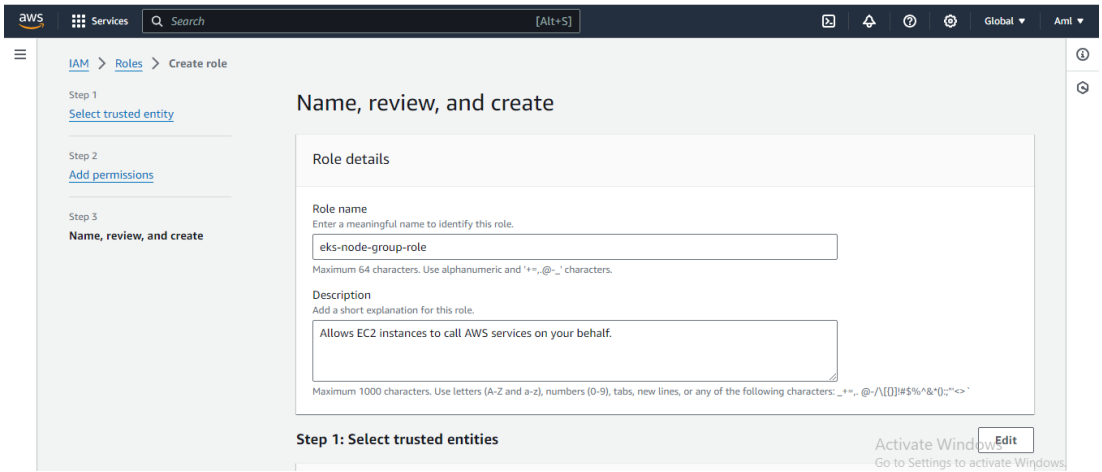
- After applying previous steps the master nodes should be created
- I'll use CloudShell to connect to the cluster
- Type these commands

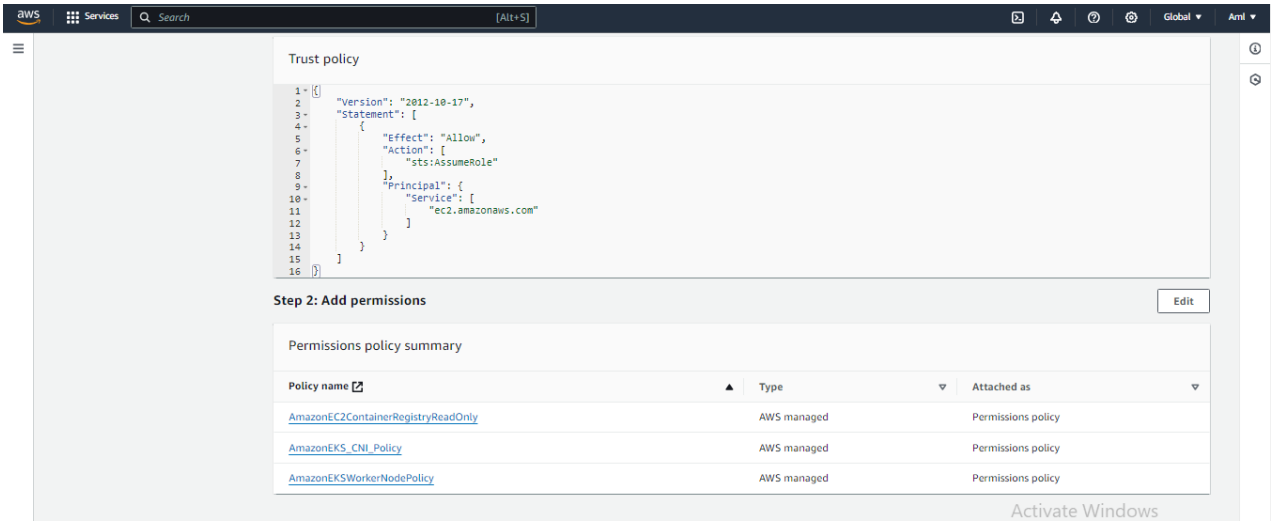
  $aws eks update-kubeconfig --name eks-cluster-test

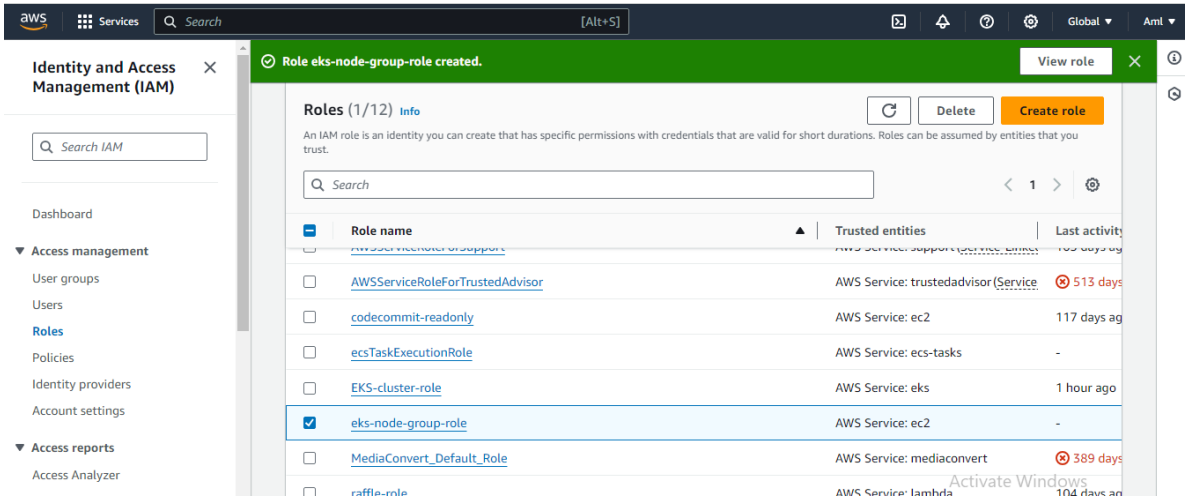
  $kubectl get nodes → *to ensure that I'm connected with the cluster*

4- <u>Creating Node group and associated IAM role:</u>

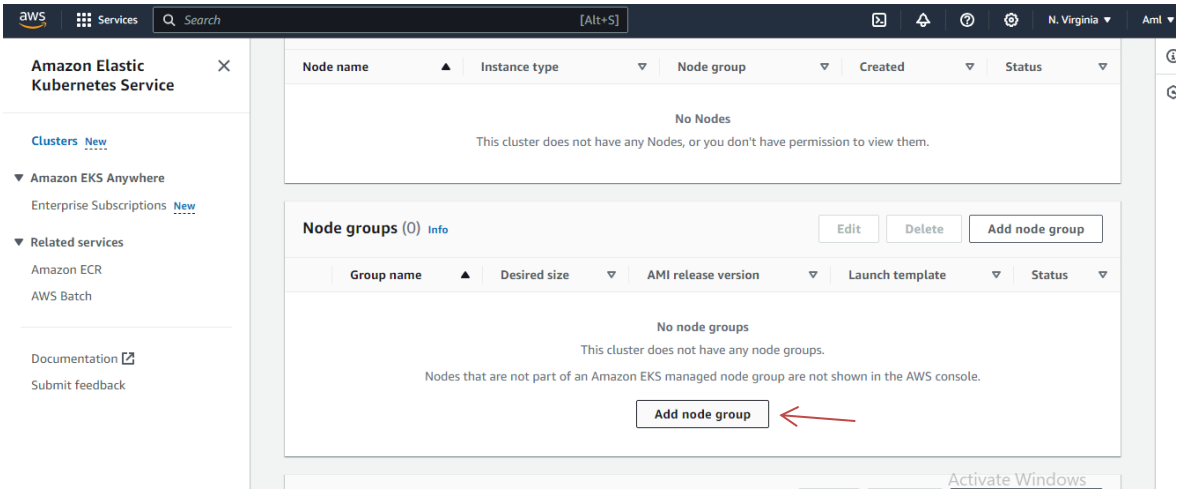
- Creating IAM role

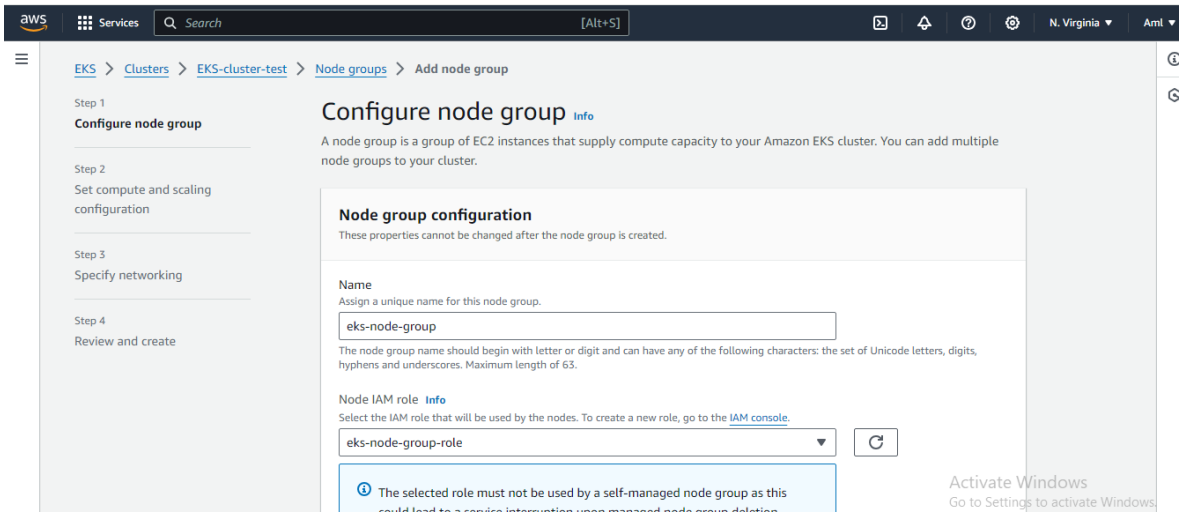


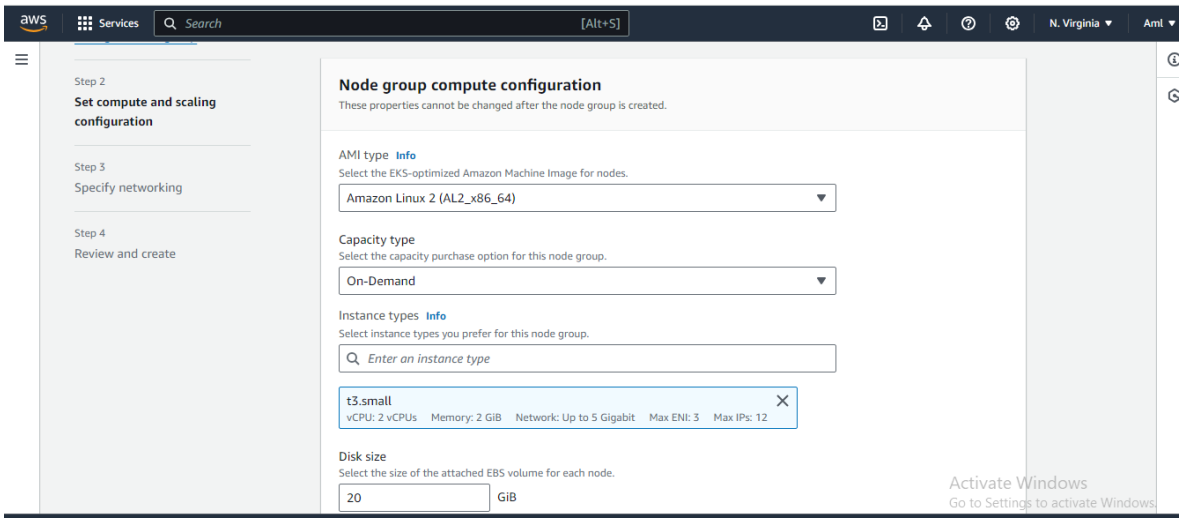


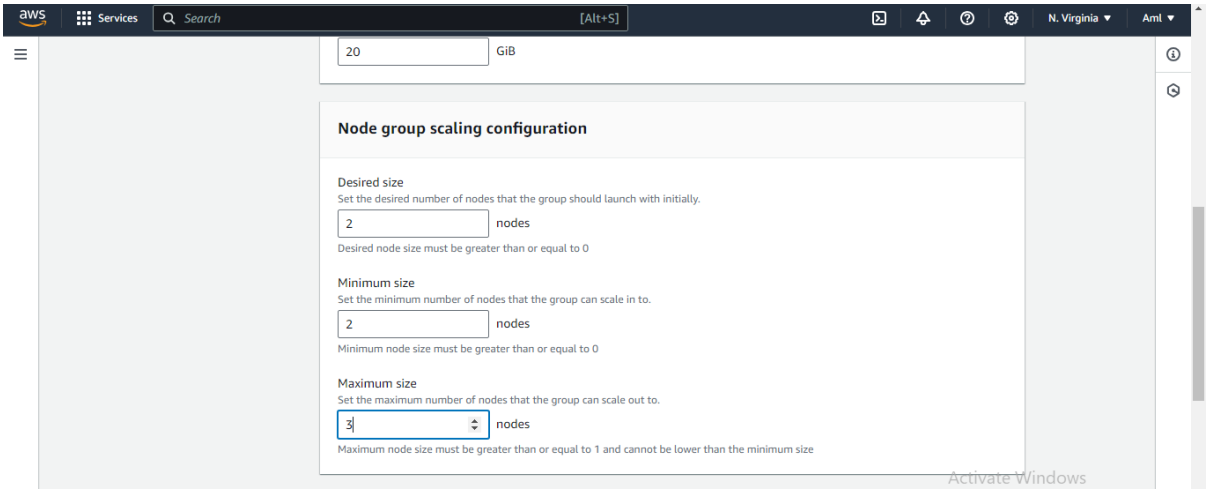


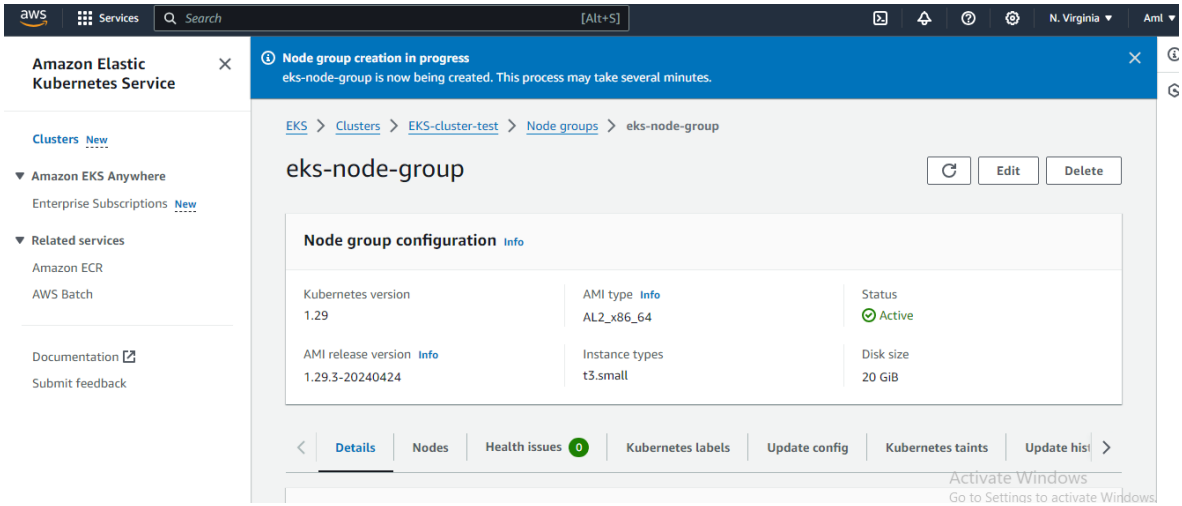
- Creating Node Group (Group consist of EC2 instances):

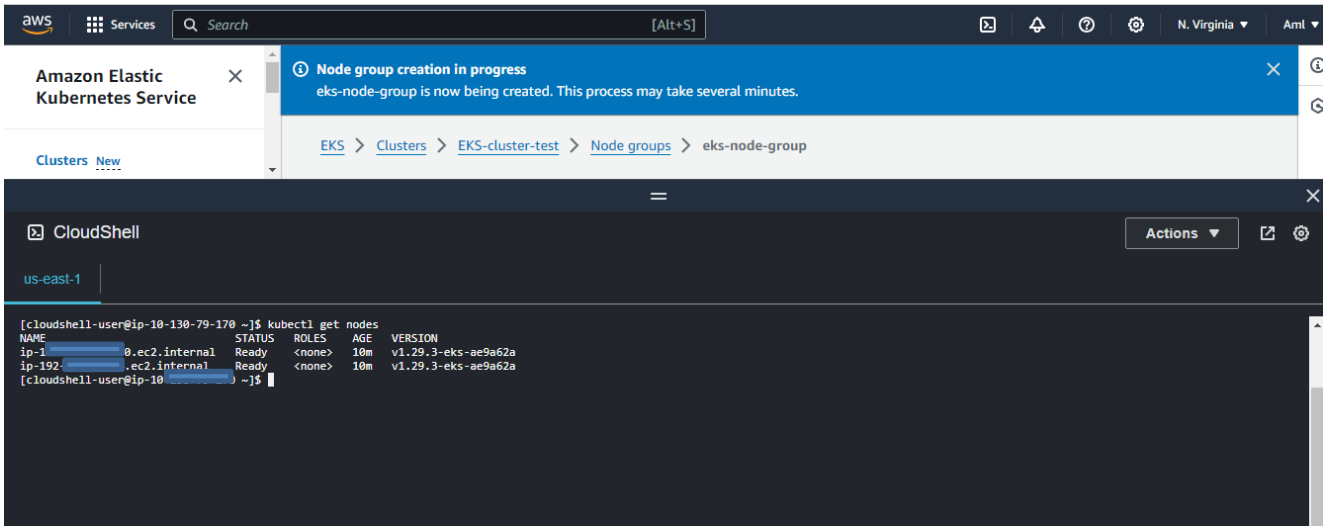



aws
Services
Search
[Alt+S]
N. Virginia
Aml
EKS > Clusters > EKS-cluster-test > Node groups > Add node group
Step 1
Configure node group
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create
Configure node group Info
A node group is a group of EC2 instances that 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-node-group-role
The selected role must not be used by a self-managed node group as this
Activate Windows
Go to Settings to activate Windows.


aws
Services
Search
[Alt+S]
N. Virginia
Aml
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create
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)
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.small
vCPU: 2 vCPUs Memory: 2 GiB Network: Up to 5 Gigabit Max ENI: 3 Max IPs: 12
Disk size
Select the size of the attached EBS volume for each node.
20
GiB
Activate Windows
Go to Settings to activate Windows.

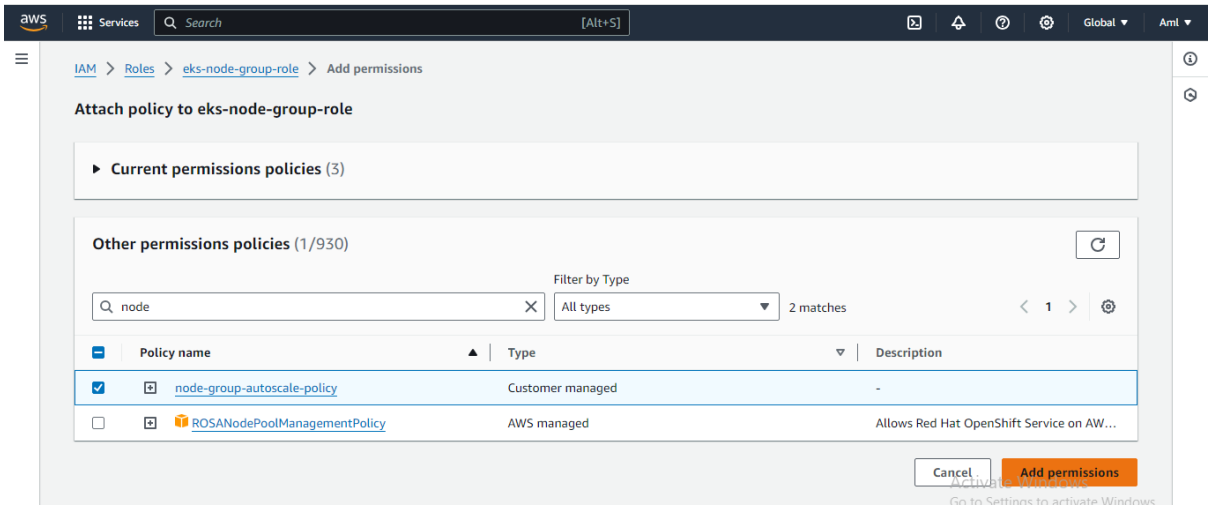

aws
Services
Search
[Alt+S]
N. Virginia
Aml
20
GiB
Node group scaling configuration
Desired size
Set the desired number of nodes that the group should launch with initially.
2
nodes
Desired node size must be greater than or equal to 0
Minimum size
Set the minimum number of nodes that the group can scale in to.
2
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.
3
nodes
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size
Activate Windows



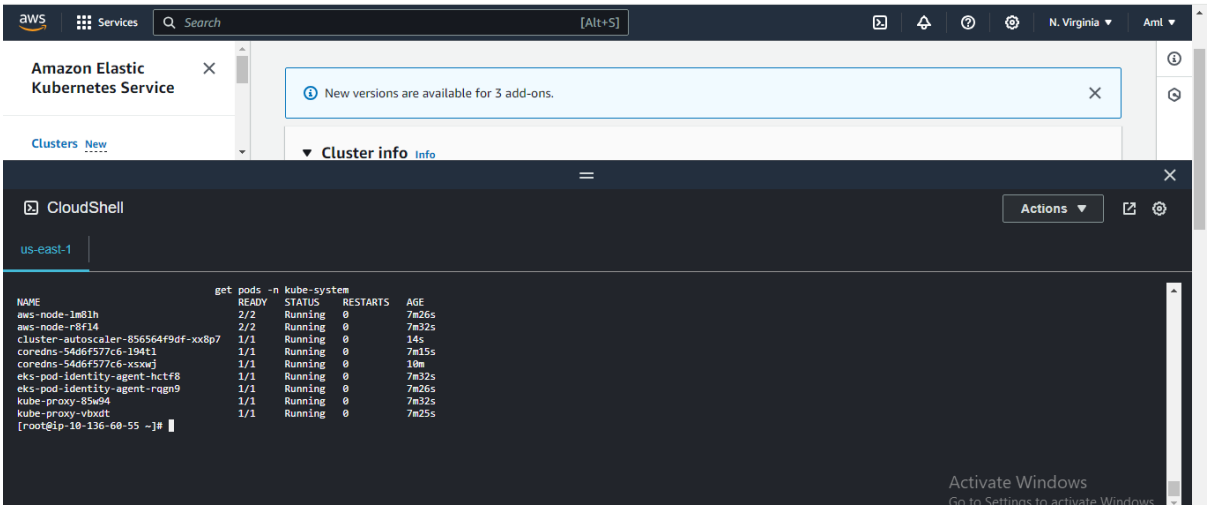


5- Adding Cluster-Autoscaler:

Policy → create policy → use json file → attatch policy to role



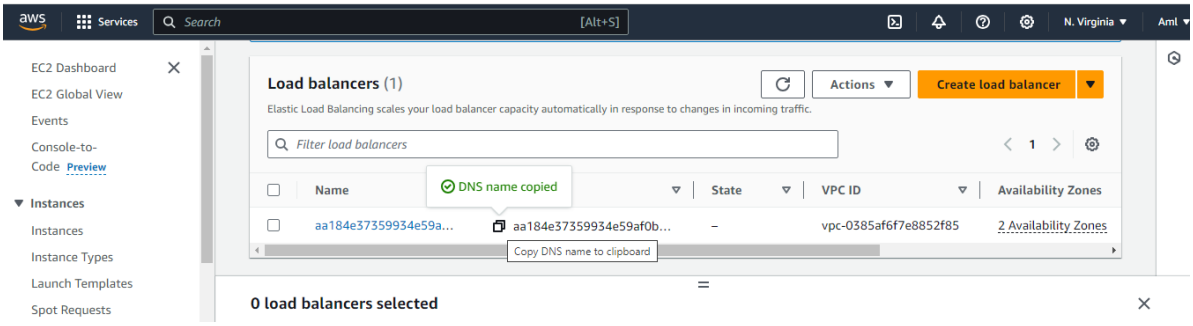
- Using cluster-autoscaling.yaml



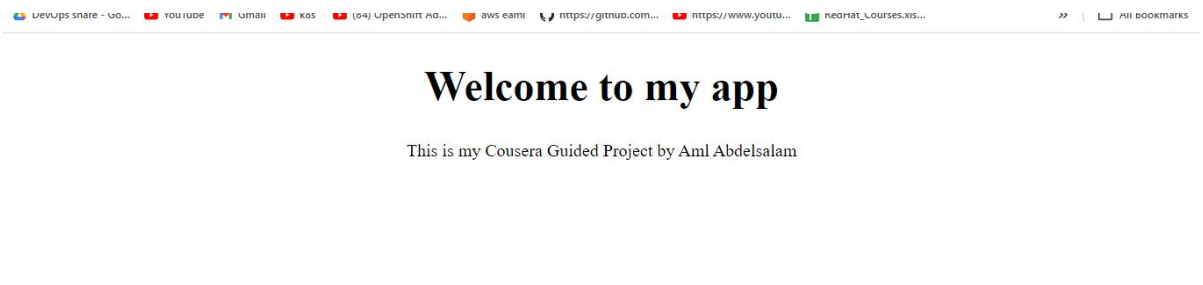
6- <u>Applying the application to the EKS cluster:</u>

Pasting the nginx.yaml (Apllication) file to the shell

- Check load balancers
-



- Copy the DNS name and paste it to the browser



- To test the autoscaler I edited the deployment new-app , editing the replicas to be 8 which will test the load
  then use $ kubectl get pods → that will reply with 8 pods
  then use $kubectl get nodes → that will reply with another creating nodes (you can also check EC2 instances)