

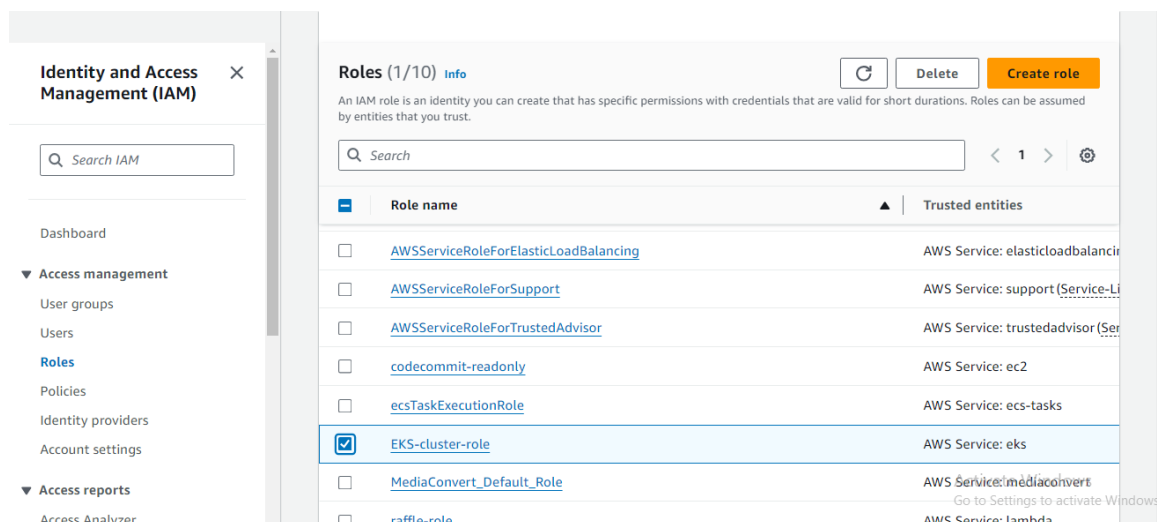
Creating a Scalable Application Using AWS EKS via AWS Console

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

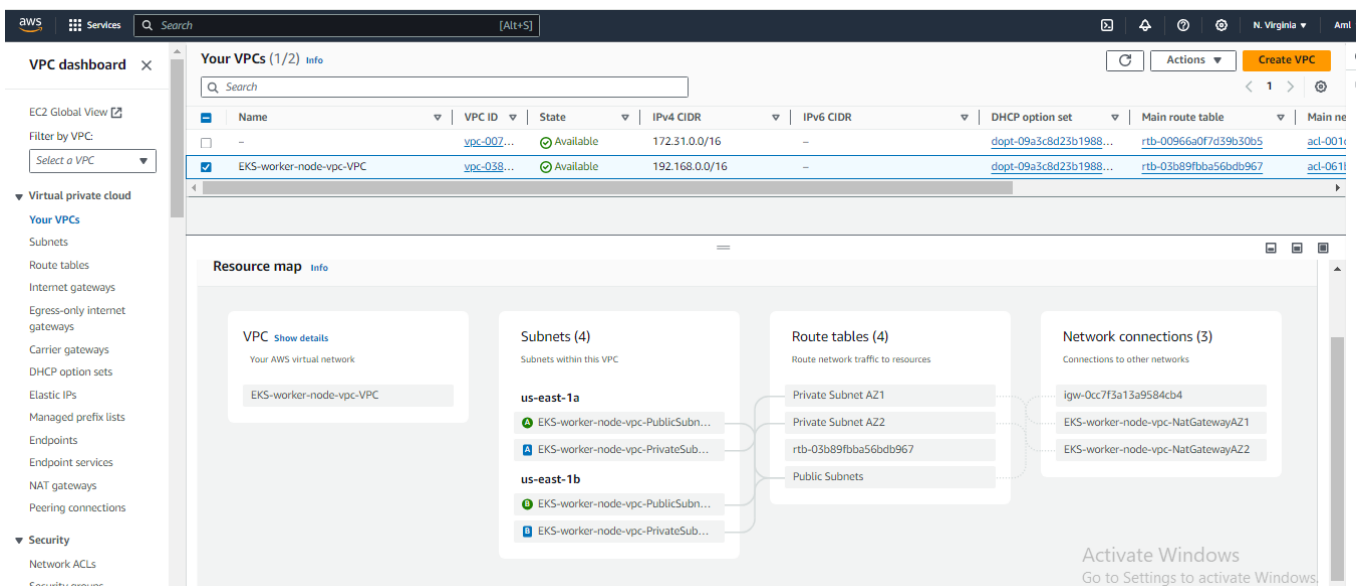
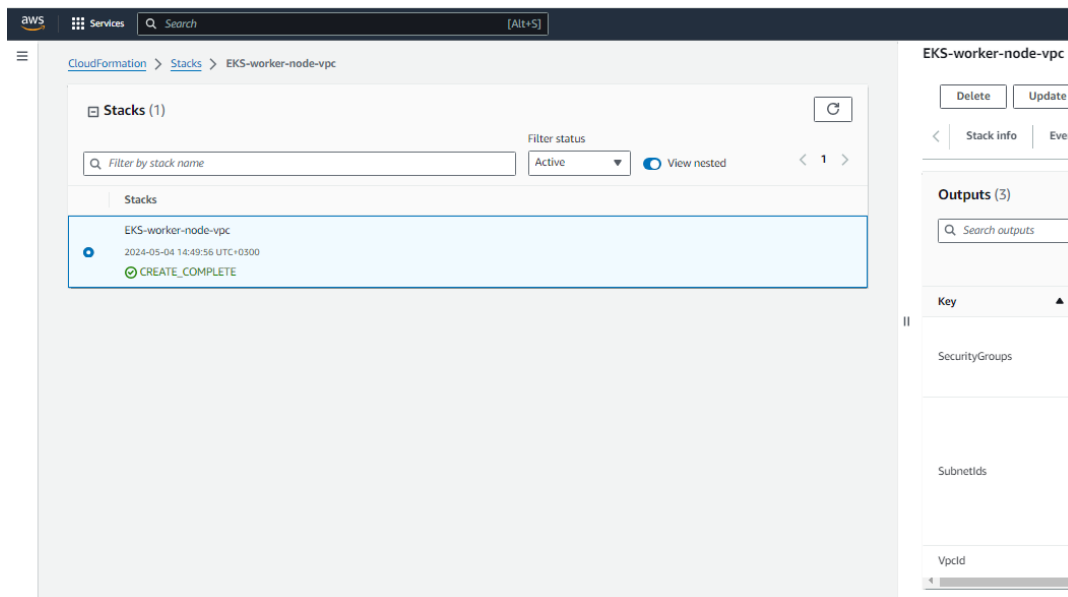


2- Create a VPC for worker nodes with special specification:

I'll use CloudFormation service for creating this VPC with these resources:

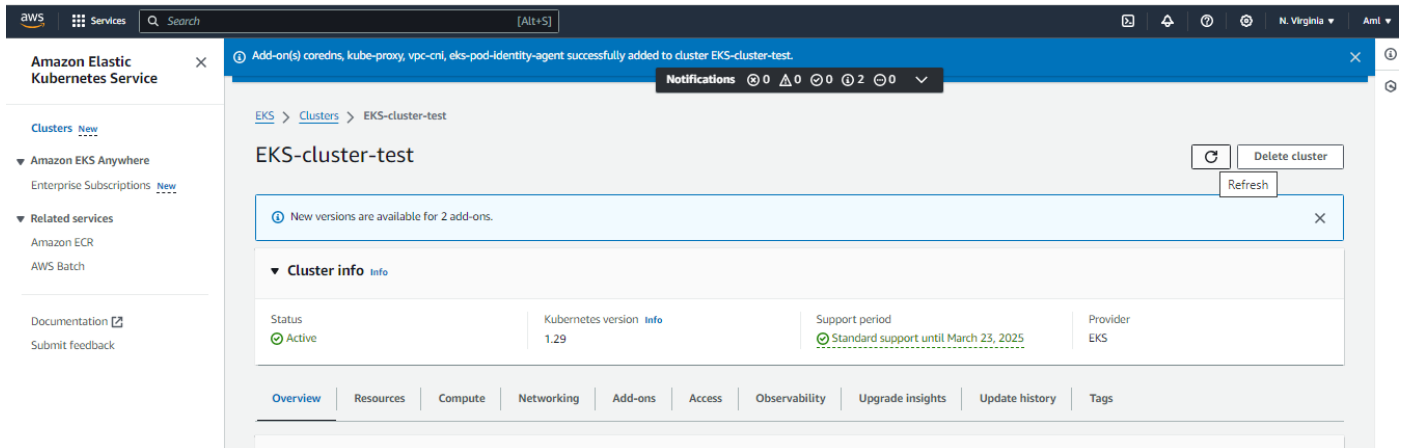
- pair of public and private subnets
- two Availability Zones
- internet gateway, with a default route on the public subnets.
- a pair of NAT gateways (one in each AZ)

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



3- Creating EKS cluster and connect to it by CloudShell:

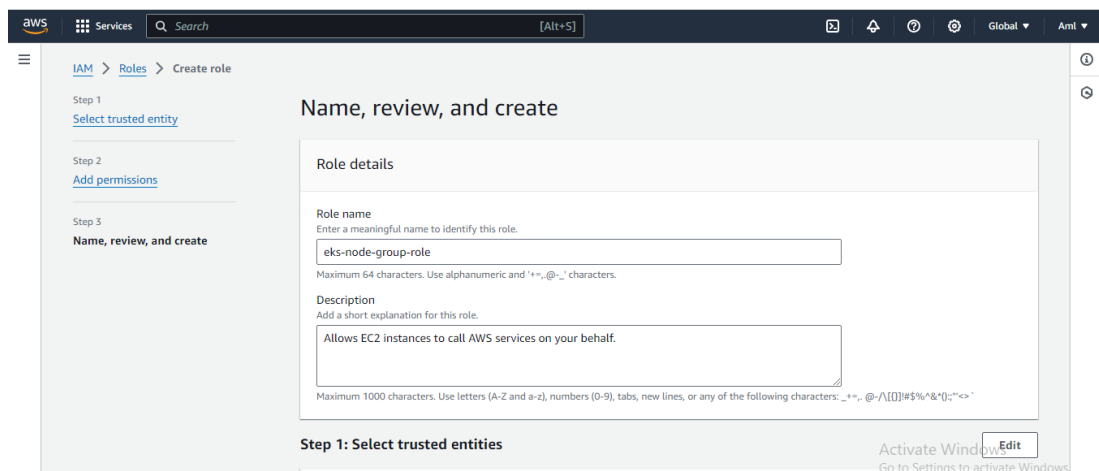
Choose EKS from console → Add cluster → create → choose eks-cluster-role (pre-created) → choose VPC (pre-created) → choose security group (pre-created) → create



- 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- Creating Node group and associated IAM role:

- Creating IAM role



Trust policy

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-       "Principal": {
10-        "Service": [
11-          "ec2.amazonaws.com"
12-        ]
13-      }
14-    ]
15-  }
16- }
```

Step 2: Add permissions Edit

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2ContainerRegistryReadOnly	AWS managed	Permissions policy
AmazonEKS_CNI_Policy	AWS managed	Permissions policy
AmazonEKSWorkerNodePolicy	AWS managed	Permissions policy

Identity and Access Management (IAM)

Search IAM

Dashboard

- Access management
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access Analyzer

Role eks-node-group-role created. View role

Roles (1/12) Info Refresh Delete Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

Role name	Trusted entities	Last activity
AmazonEC2ContainerRegistryReadOnly	AWS Service: ec2	117 days ago
<input type="checkbox"/> AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	513 days ago
<input type="checkbox"/> codecommit-readonly	AWS Service: ec2	117 days ago
<input type="checkbox"/> ecsTaskExecutionRole	AWS Service: ecs-tasks	-
<input type="checkbox"/> EKS-cluster-role	AWS Service: eks	1 hour ago
<input checked="" type="checkbox"/> eks-node-group-role	AWS Service: ec2	-
<input type="checkbox"/> MediaConvert_Default_Role	AWS Service: mediaconvert	389 days ago
<input type="checkbox"/> raffia-role	AWS Service: lambda	104 days ago

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

Amazon Elastic Kubernetes Service

Clusters New

- Amazon EKS Anywhere
 - Enterprise Subscriptions New
- Related services
 - Amazon ECR
 - AWS Batch

Documentation Feedback

Node name Instance type Node group Created Status

No Nodes
This cluster does not have any Nodes, or you don't have permission to view them.

Node groups (0) Info Edit Delete Add node group

Group name	Desired size	AMI release version	Launch template	Status
No node groups This cluster does not have any node groups. Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.				

Add node group ←

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.

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](#).

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon manual node group deletion.

aws

Services

Search

[Alt+S]

N. Virginia

Aml

EKS > Clusters > EKS-cluster-test > Node groups > Add node group

Step 2
Set compute and scaling configuration

Step 3
Specify networking

Step 4
Review and create

Node group compute configuration Info

These properties cannot be changed after the node group is created.

AMI type Info
Select the EKS-optimized Amazon Machine Image for nodes.

Capacity type
Select the capacity purchase option for this node group.

Instance types Info
Select instance types you prefer for this node group.

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.

GiB

aws

Services

Search

[Alt+S]

N. Virginia

Aml

EKS > Clusters > EKS-cluster-test > Node groups > Add node group

Step 2
Set compute and scaling configuration

Step 3
Specify networking

Step 4
Review and create

Node group scaling configuration

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

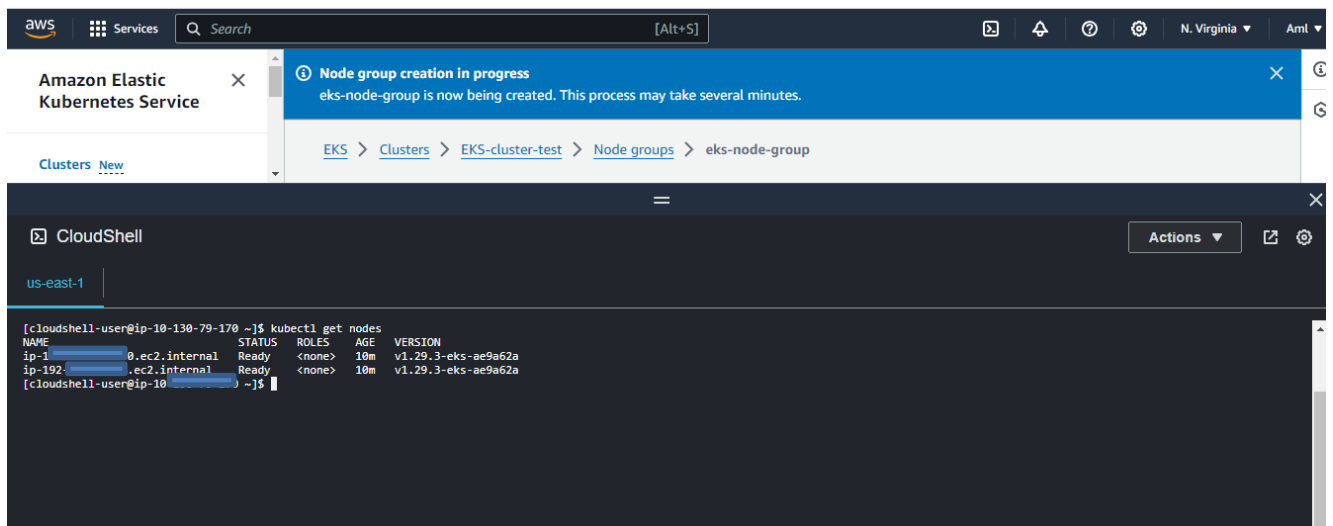
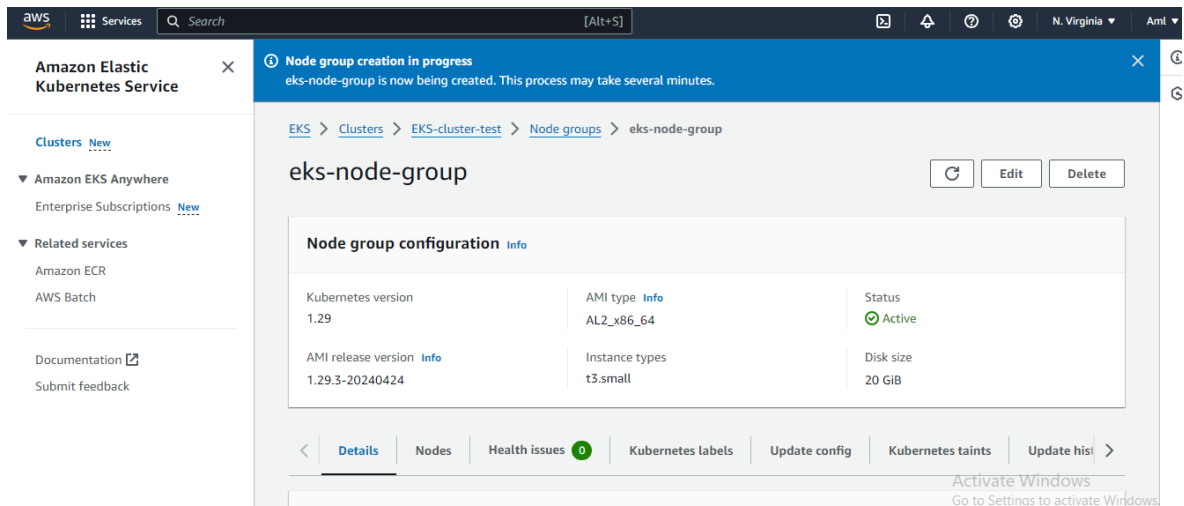
nodes
Desired node size must be greater than or equal to 0

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

nodes
Minimum node size must be greater than or equal to 0

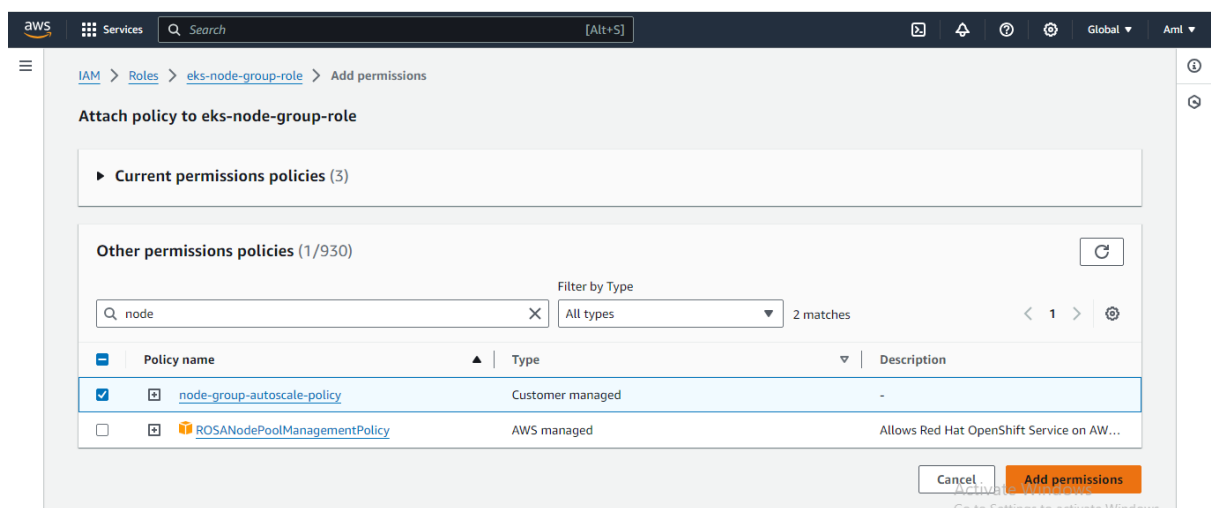
Maximum size
Set the maximum number of nodes that the group can scale out to.

nodes
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size

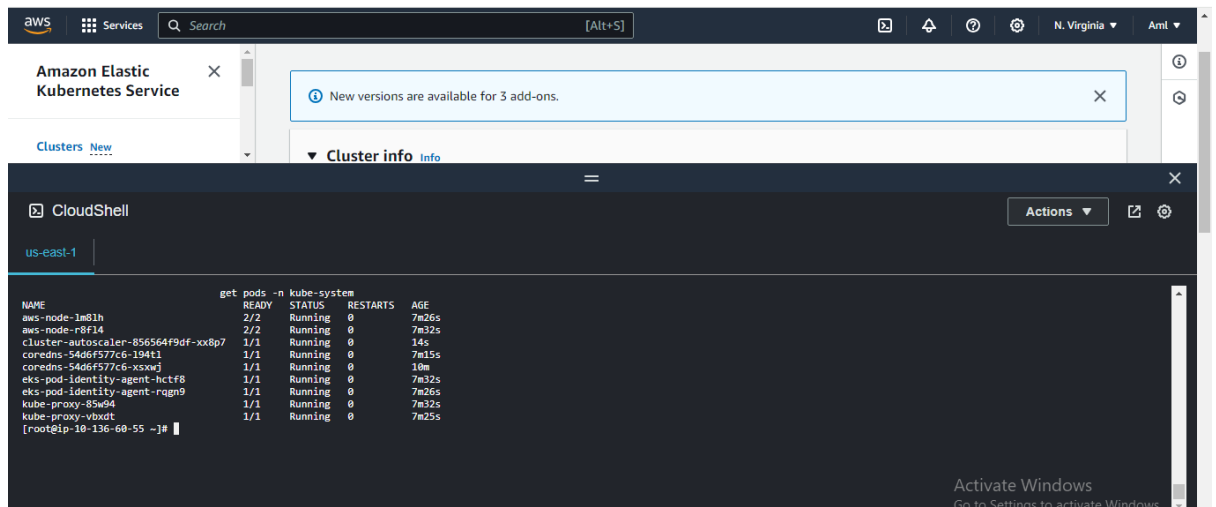


5- Adding Cluster-Autoscaler:

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

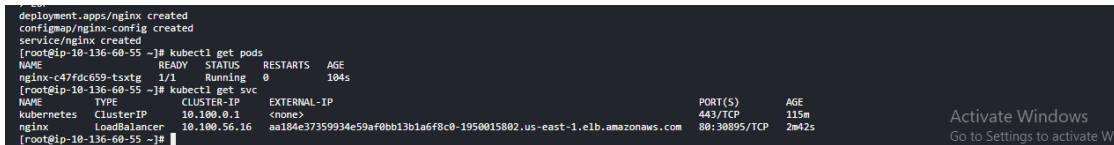


- Using cluster-autoscaling.yaml



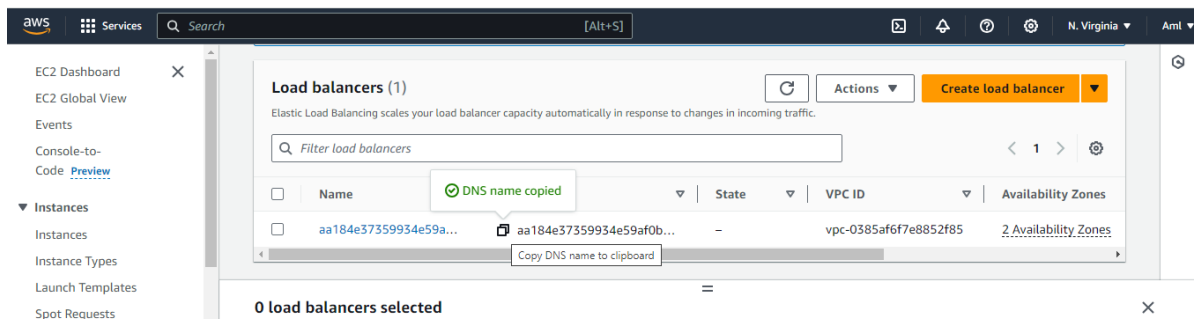
6- Applying the application to the EKS cluster:

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



- Check load balancers

-



- Copy the DNS name and paste it to the browser

Welcome to my app

This is my Cousera Guided Project by Aml Abdelsalam

- 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)