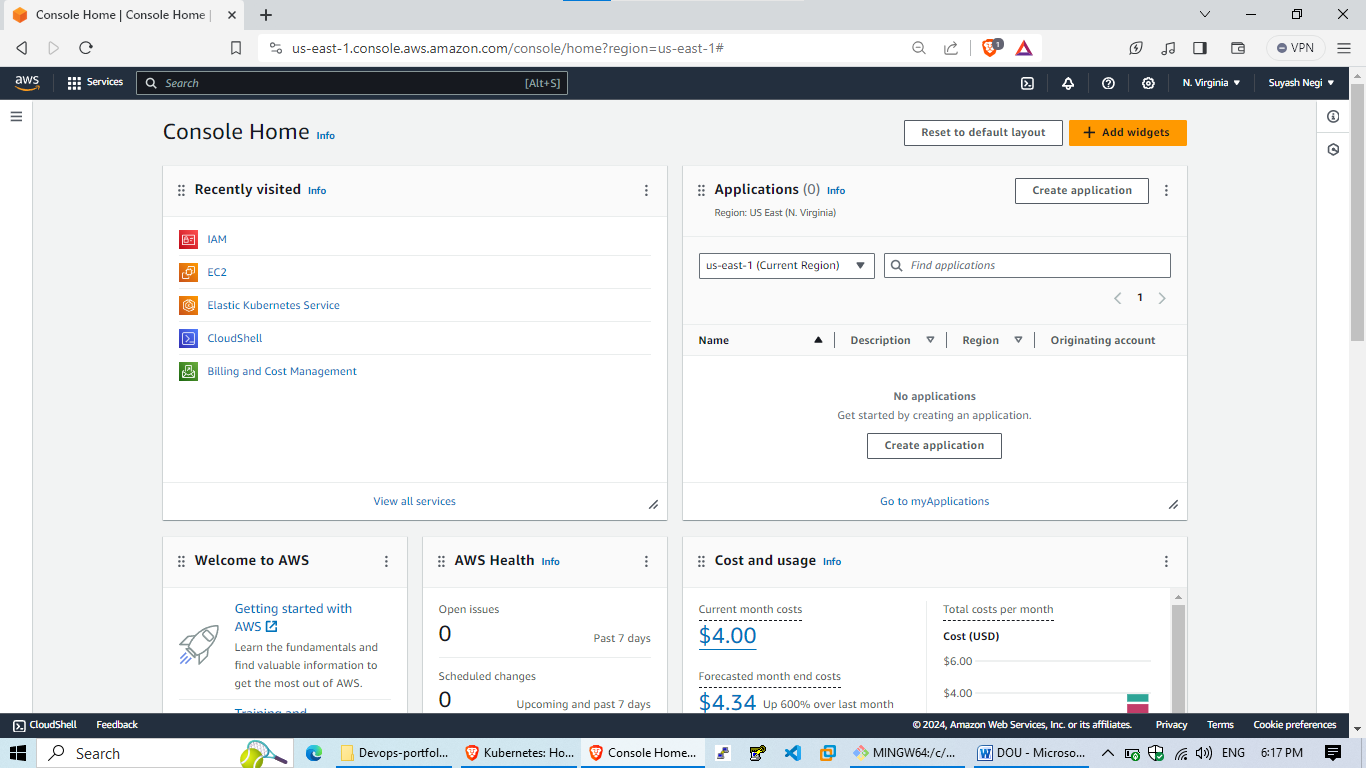
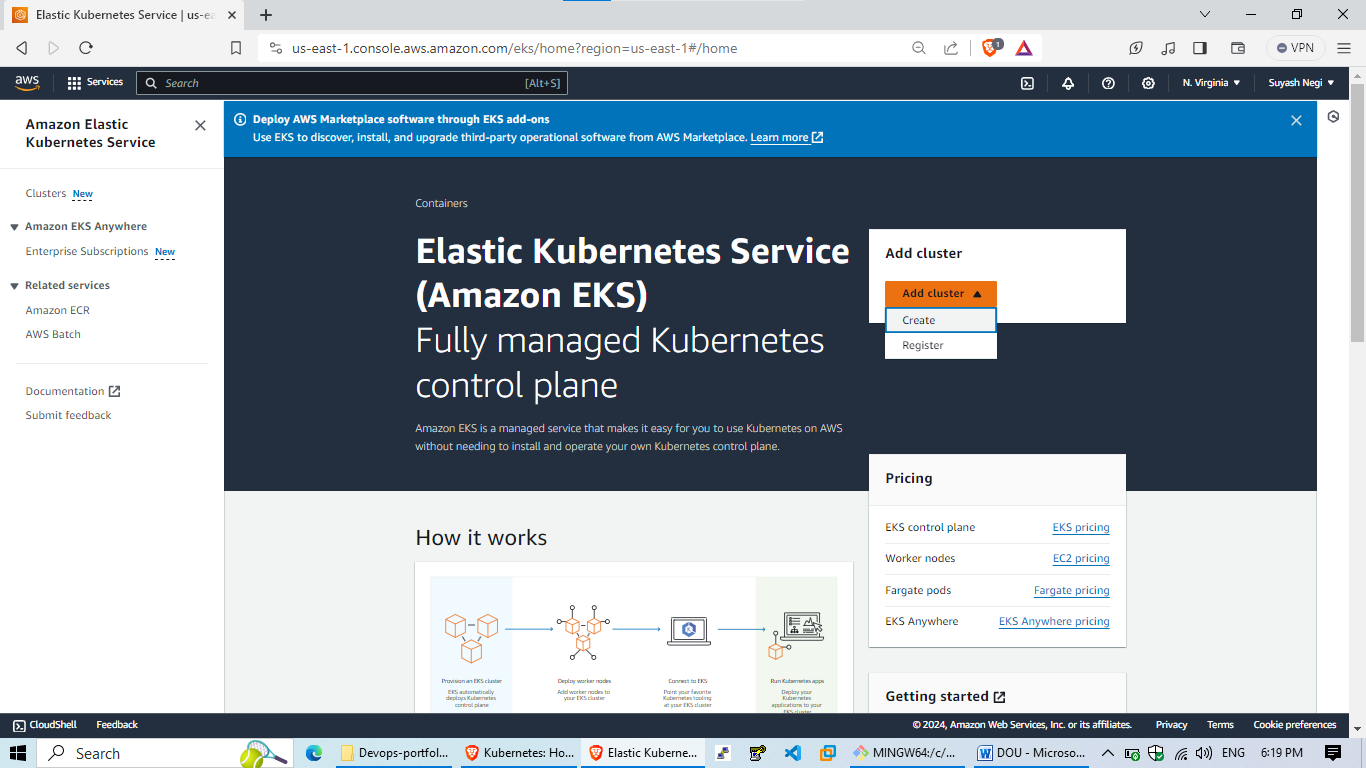
Project One: – 2048 Game

Note: we are setting up an EKS cluster and node group for compute after success of it; we will be able to play 2048.

Step 1: Log into AWS environment.



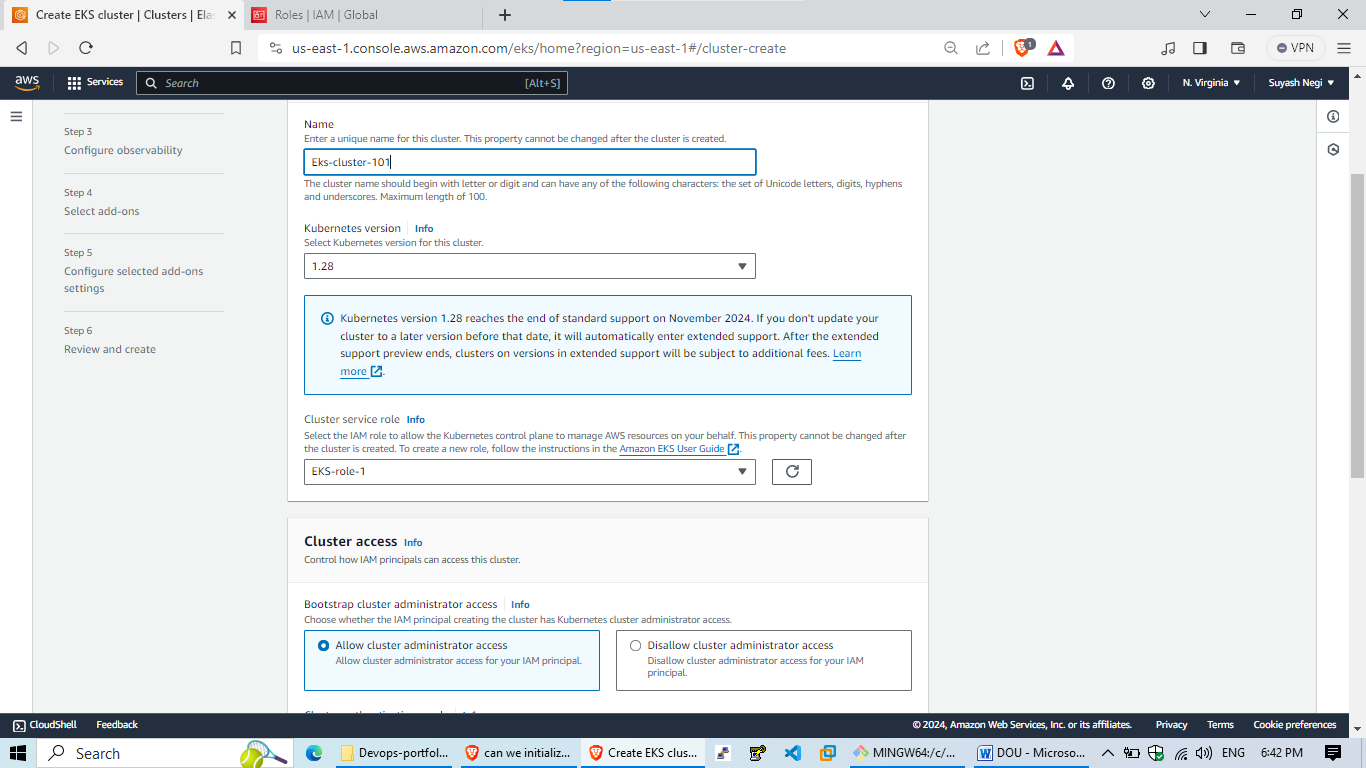
Step 2: Go to EKS platform through AWS console and create a cluster.



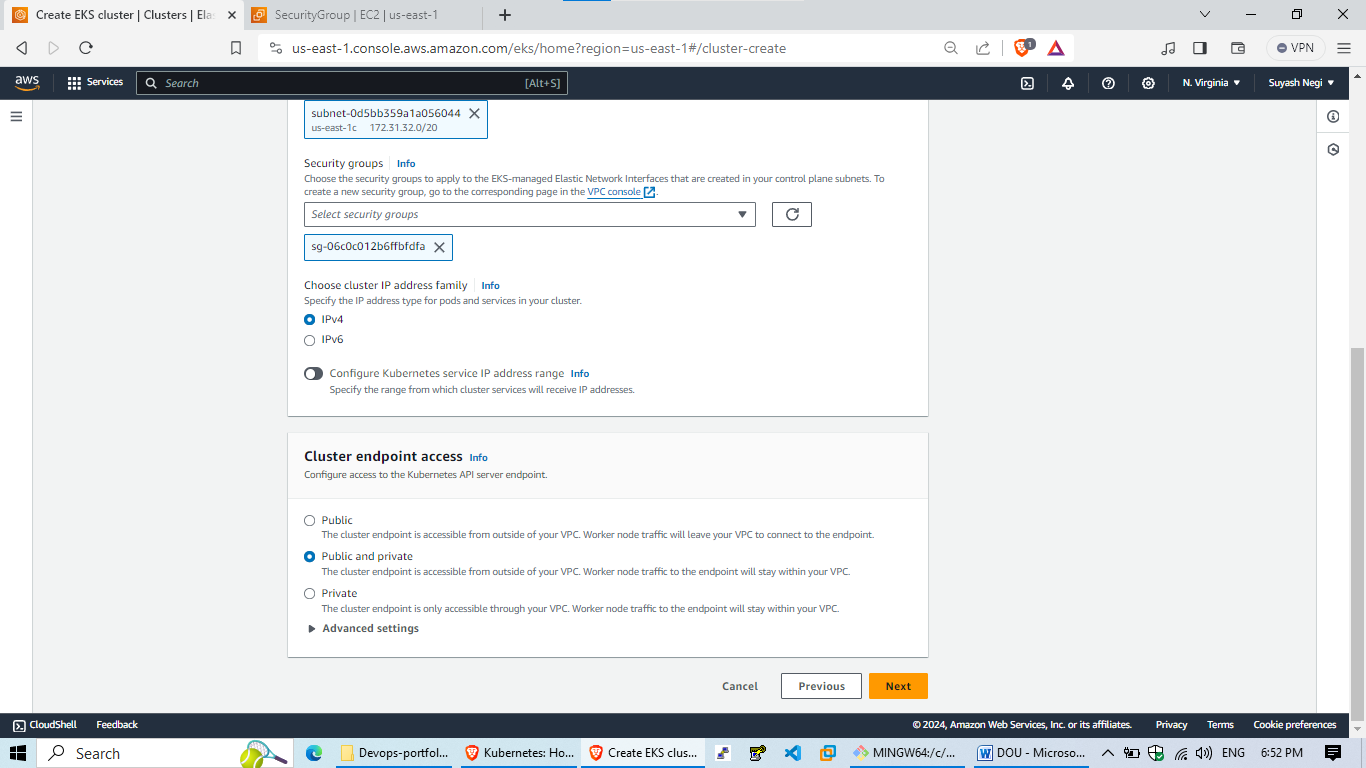
Bonus point Create an IAM role first:

IAM-> Roles-> Create a role-> Trust entity type (AWS service) -> Use Case (Service: EKS type it and select EKS cluster use case) -> Next-> Add permissions (will already have an existing EKS cluster policy no need to change it, do Next) -> Name (EKS-role-1) -> Create role.

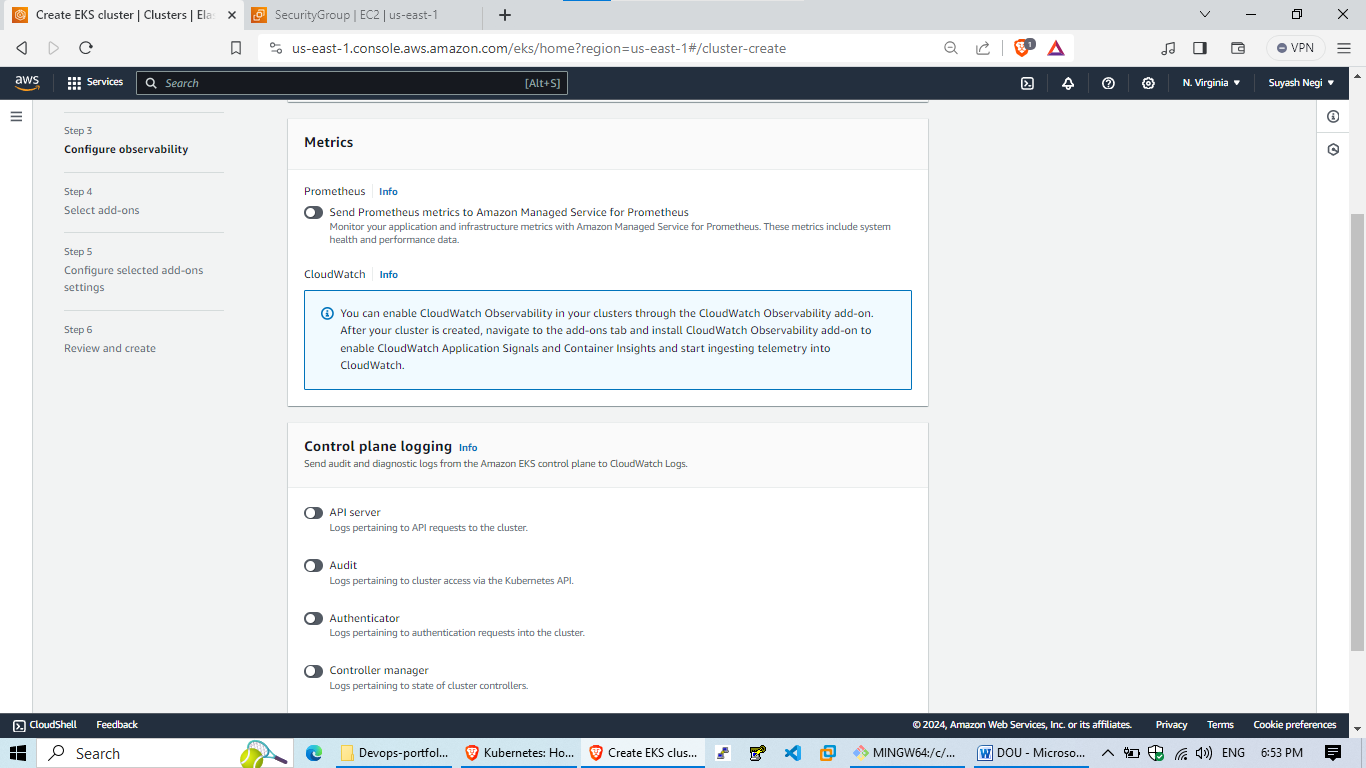
Step 3: Add name of the cluster, always select latest version for K8’s and create a service role to it through IAM.

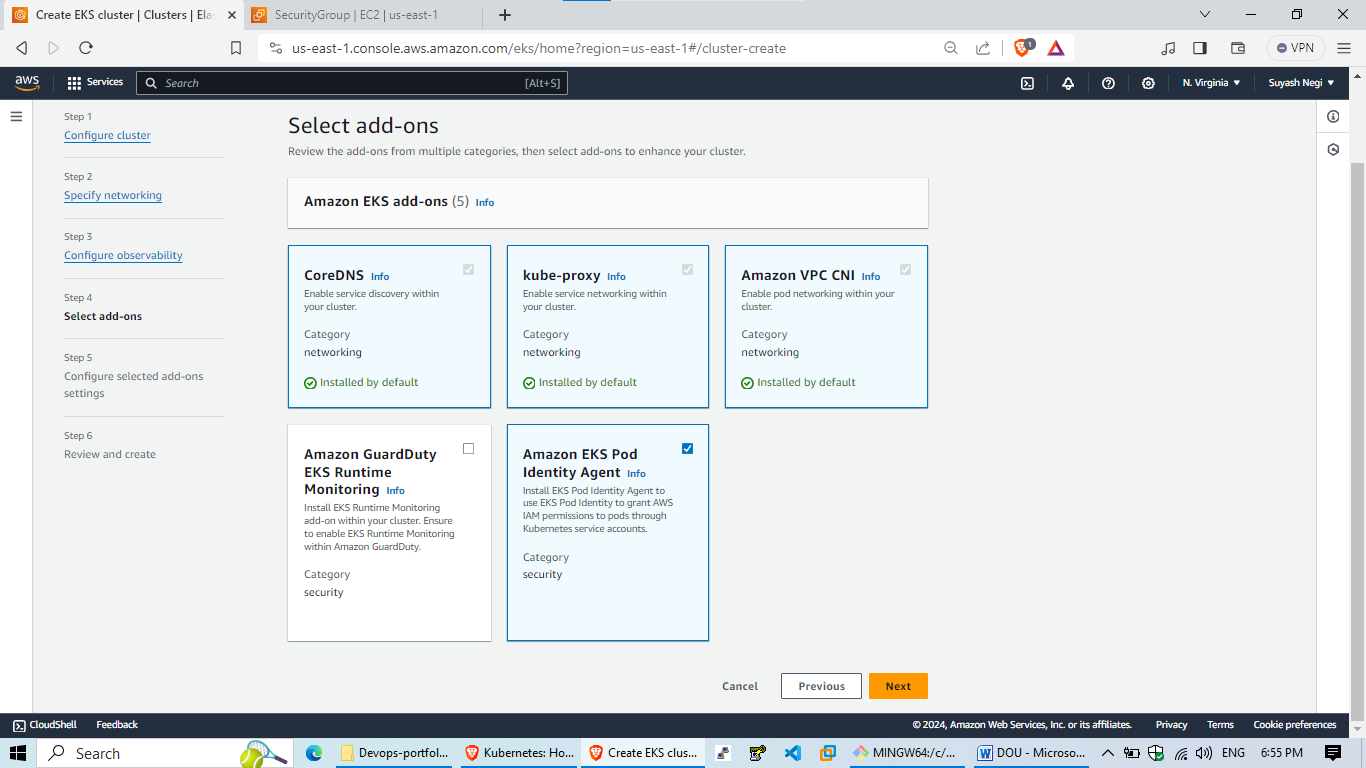


Step 4: Go to next page then select APV you can select DFLT VPC for now and do remove AZ-1E. Create a SG with Name will eks-sg1 then select the dflt vpc and then add inbound rules ssh & http anywhere and add custom tcp pot 8080 anywhere create sg. Come back to eks screen add Security group and then leave everything as it is.

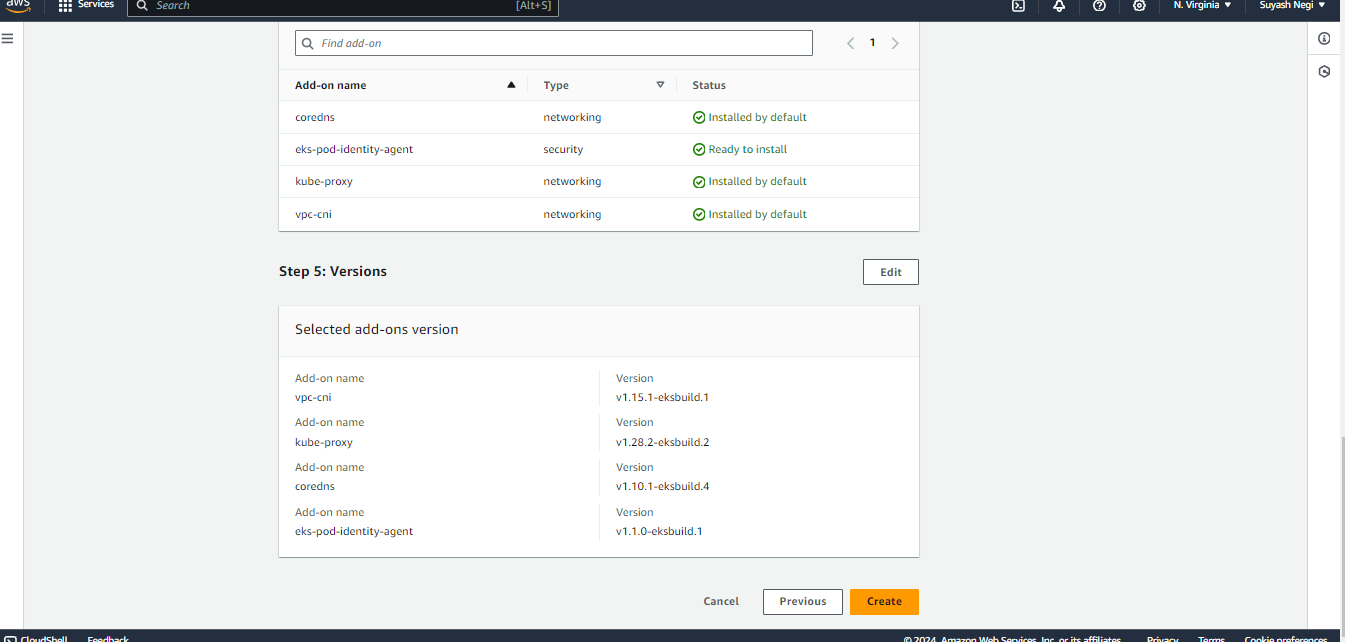


Step 5: Leave everything default in these two upcoming pages and click next.

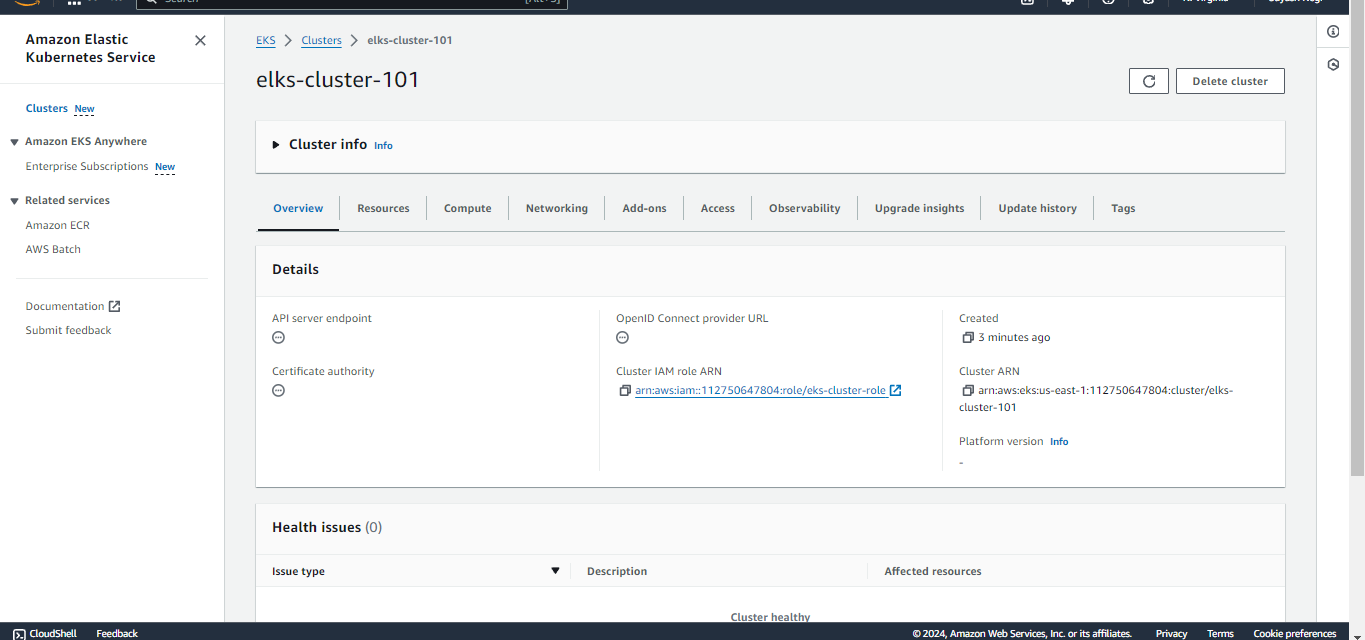


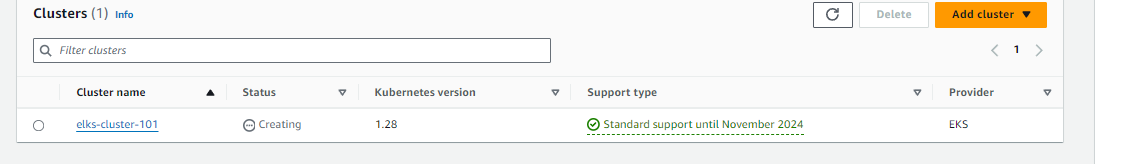


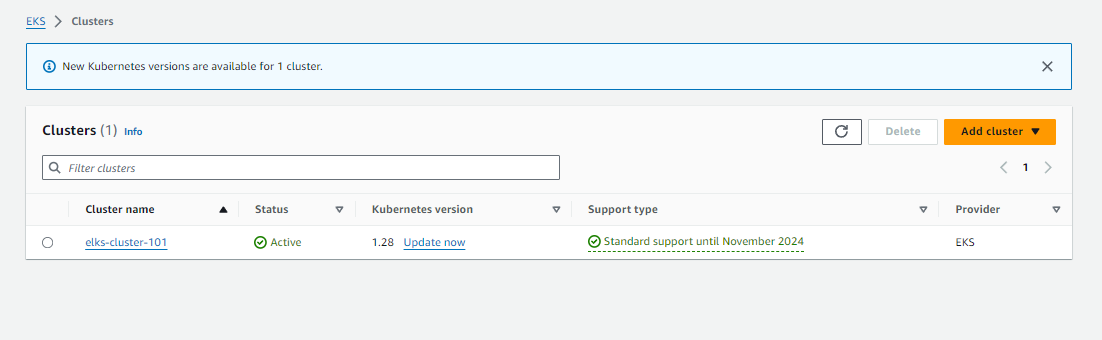
Step 6: Final cluster page:----------------Click on create.



Step 7: Then wait for 15 to 20mins(Time taken by aws to create a cluster)





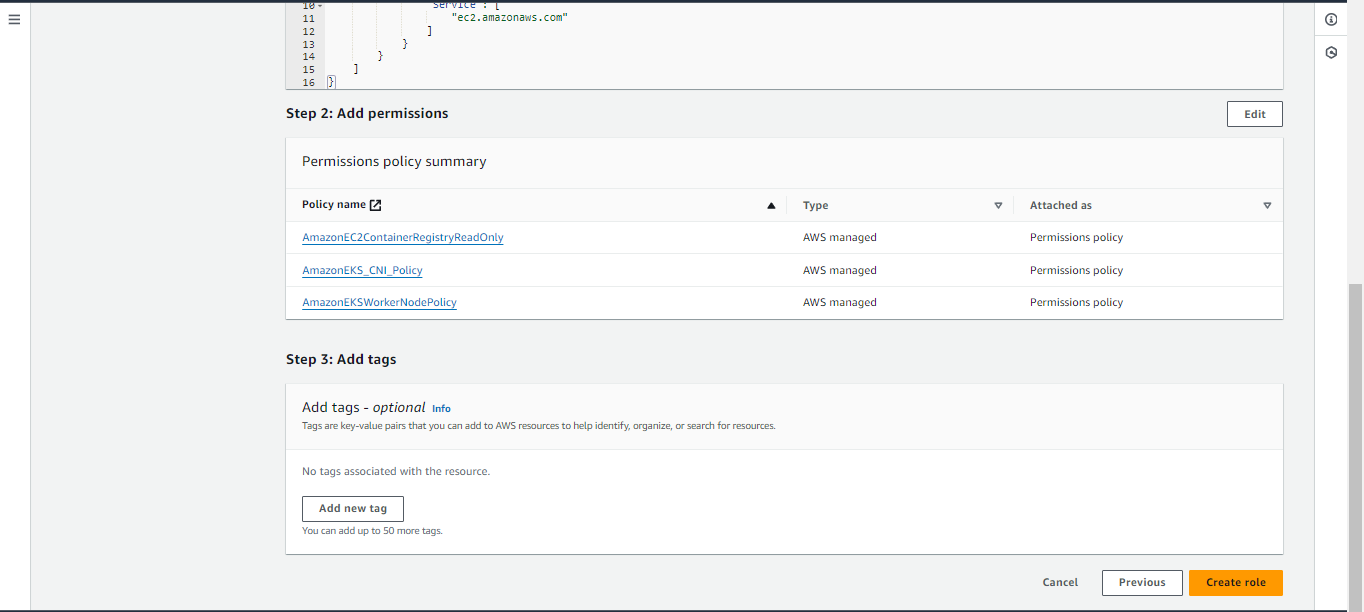
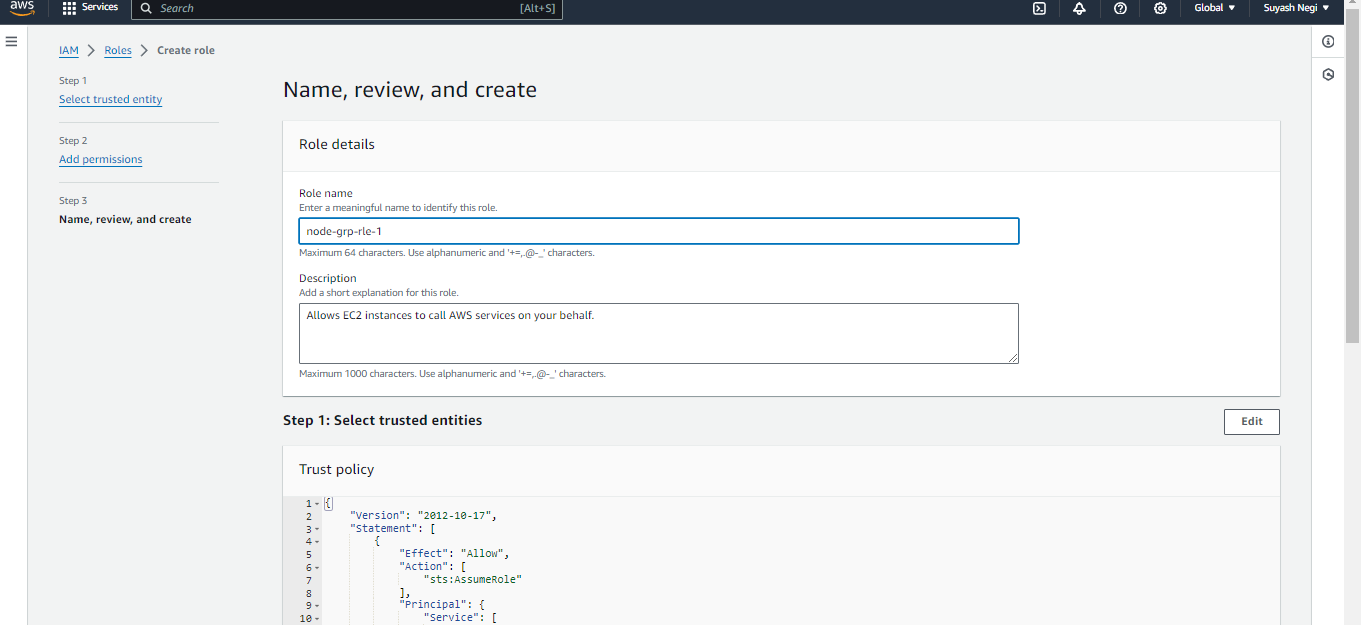
Note: Ready status

Step 8: Now before adding compute for our control plane we will create a new iam role for our node ec2 (It allows EC2 instances to call AWS services on your behalf.)-- Go to iam-roles-create a role—AWS service—type: ec2—next— add these 3 policies:

- EKSWorkerNodePolicy

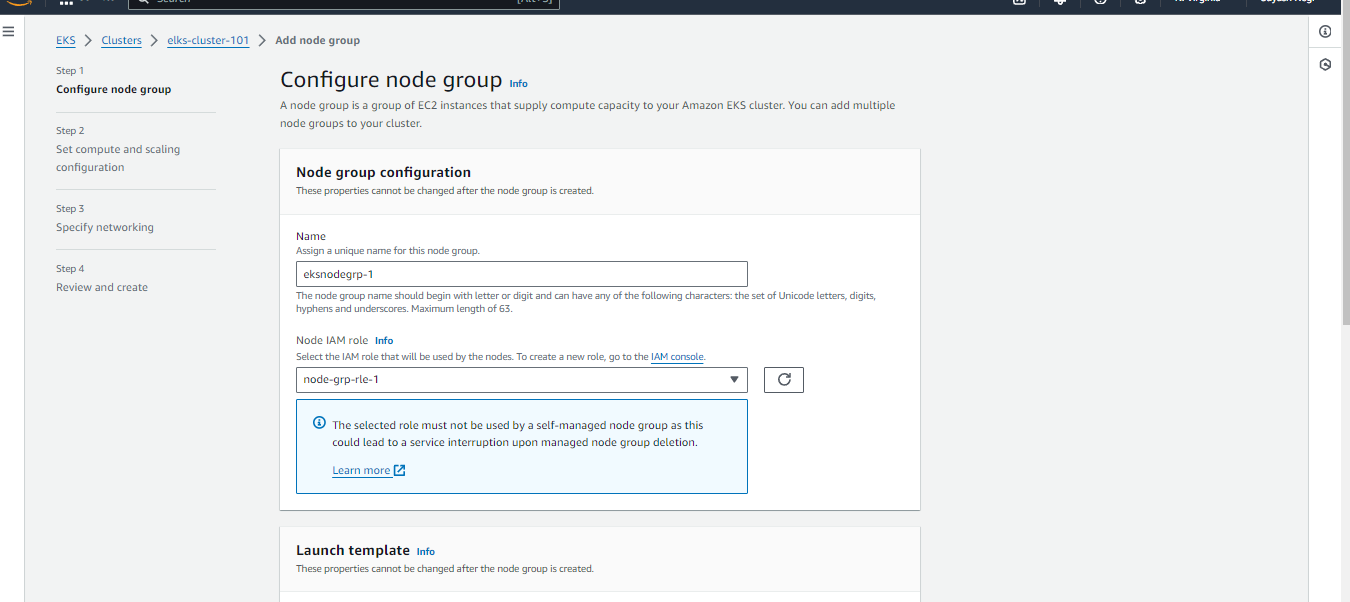
- EC2ContainerRegistryReadOnly

- EKS\_CNI\_Policy. Click next then review the role.

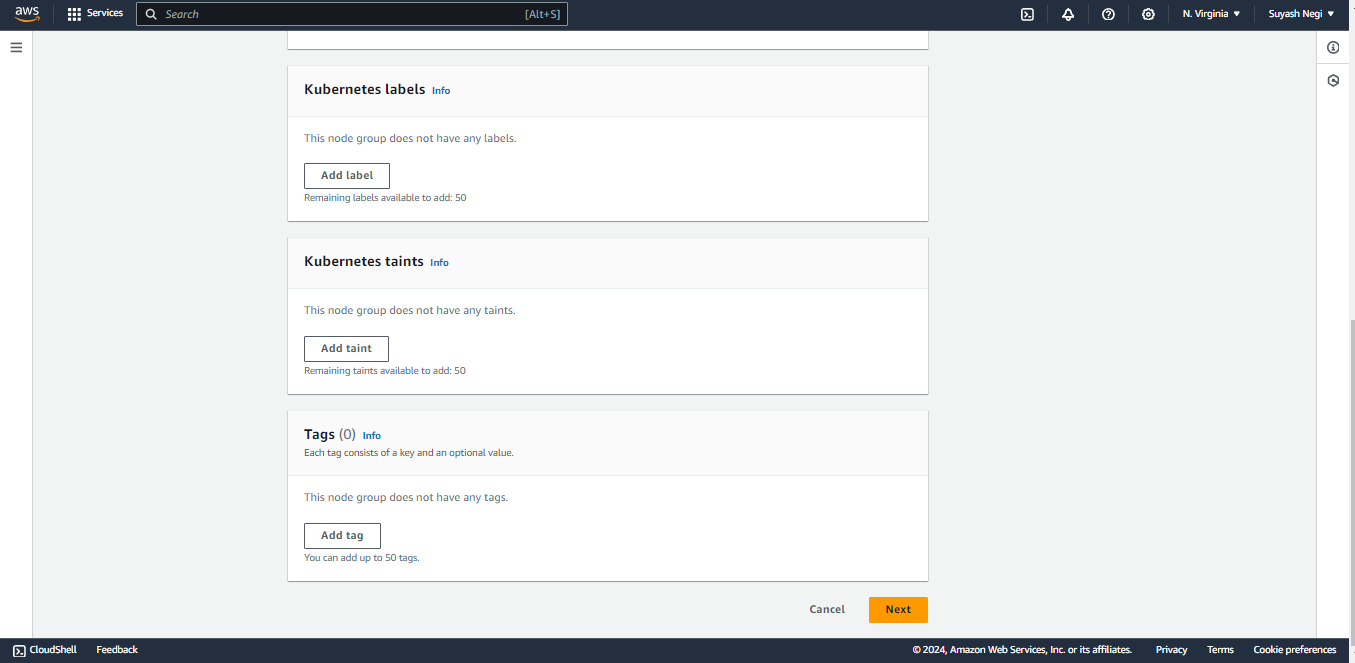


Finally create it.

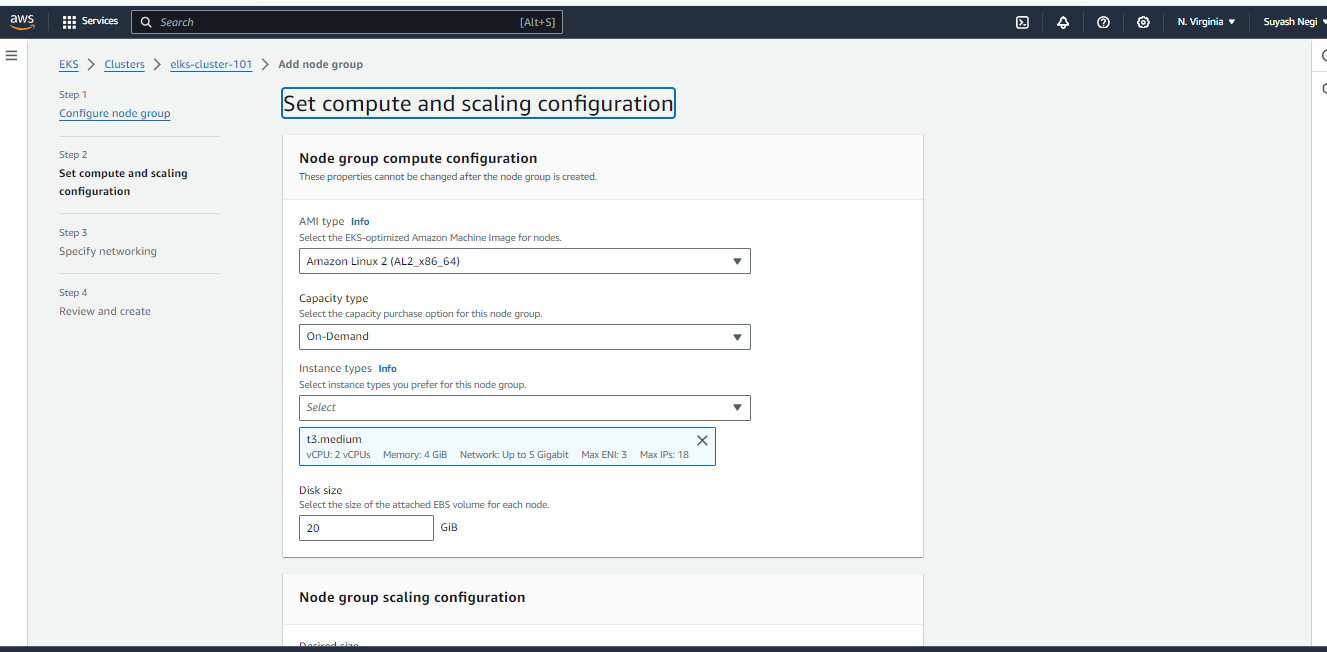
Step 9: Go to eks – compute add node group (Name the following node group and also attach the iam role which we have created)—



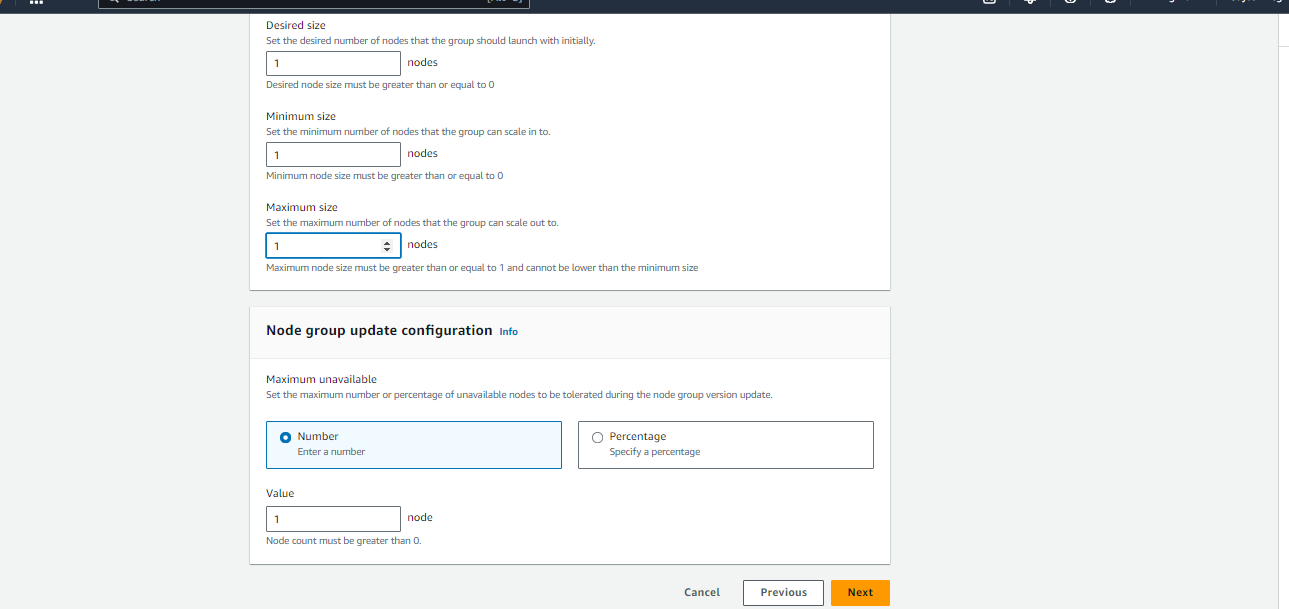
Then click on next:



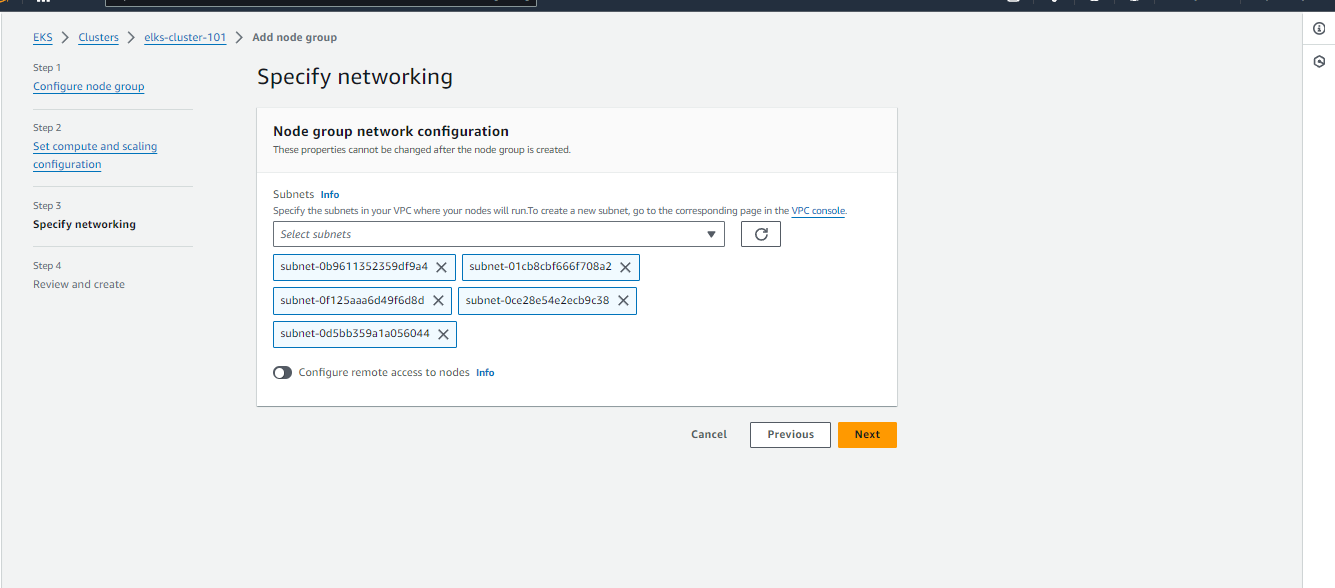
Step 10: Leave all deflt, if you still want to change you can chnge instance type to t2.medium not less than that:



Node group scaling configuration: Select these then click next;

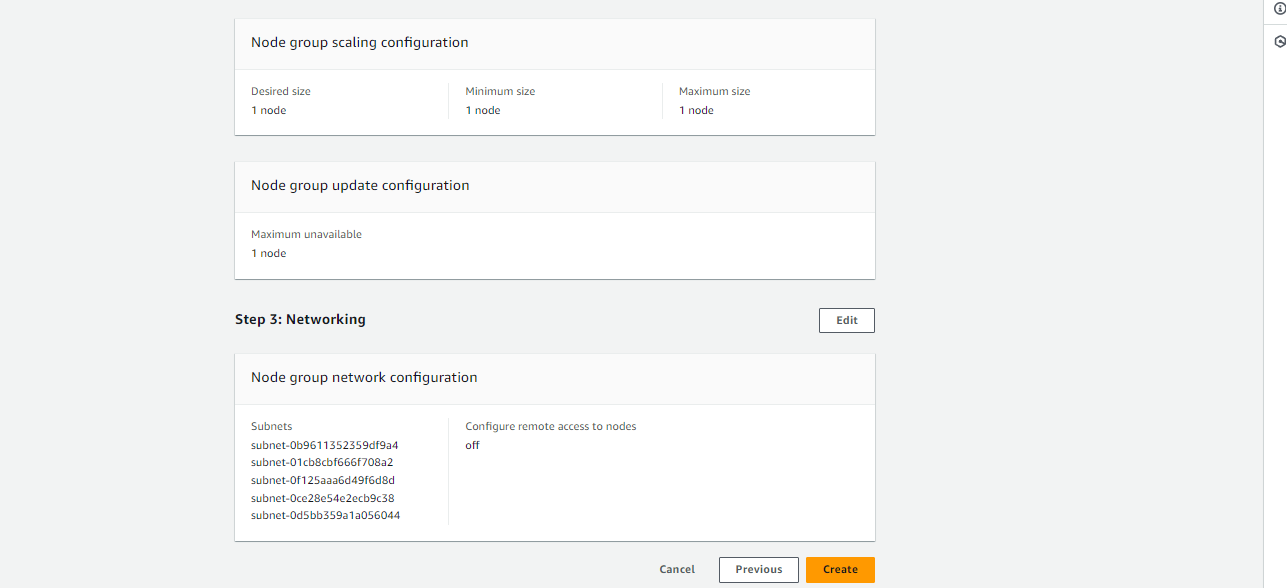


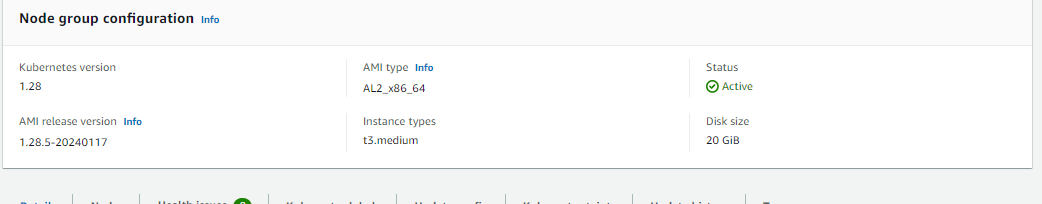
Specify networking

**Node group network configuration: select subnets and then click next.**

**Finally on next page swipe all the way down and click on create☺:**

**(Wait for 5 minutes to create it).**

****

****

Step 11: Authenticate to this cluster

===================================

Reference:

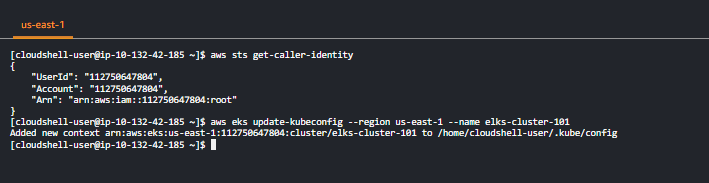
https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html

Open cloudshell# Type on your AWS CLI window # observe your account and user id details:

**aws sts get-caller-identity**

# Create a kubeconfig file where it stores the credentials for EKS, kubeconfig configuration allows you to connect to your cluster using the kubectl command line.

**aws eks update-kubeconfig --region us-east-1 --name elks-cluster-101**

****

**Now create pod manifest file which will hold the configuration for our legendary game:**

****

Use this

### code starts ###

apiVersion: v1

kind: Pod

metadata:

name: 2048-pod

labels:

app: 2048-ws

spec:

containers:

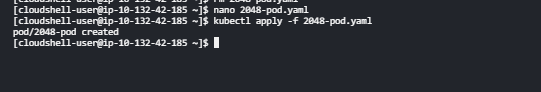
- name: 2048-container

image: blackicebird/2048

ports:

- containerPort: 80

**:wq!**

****

**Now deploying a service—load balancer which will be our end point to access the game: Manifest file:**

apiVersion: v1

kind: Service

metadata:

name: mygame-svc

spec:

selector:

app: 2048-ws

ports:

- protocol: TCP

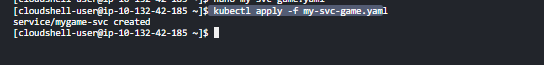
port: 80

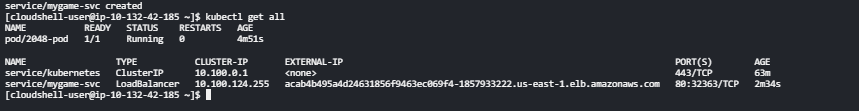
targetPort: 80

type: LoadBalancer

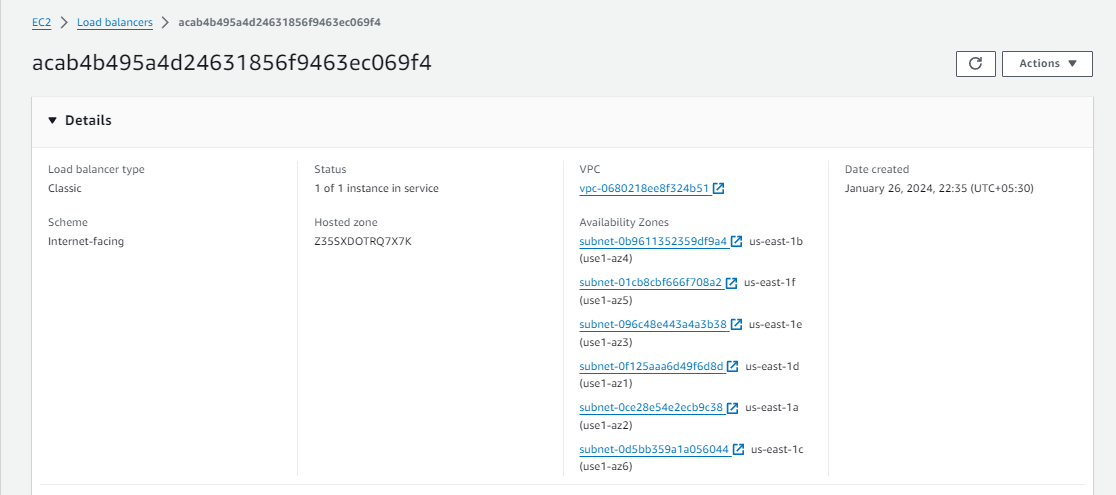
:wq!

**Then --- kubectl apply -f my-svc-game.yaml**

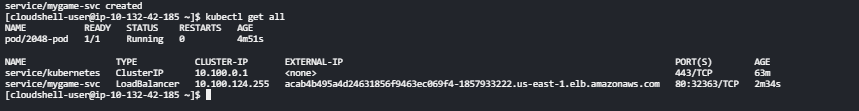
****

****

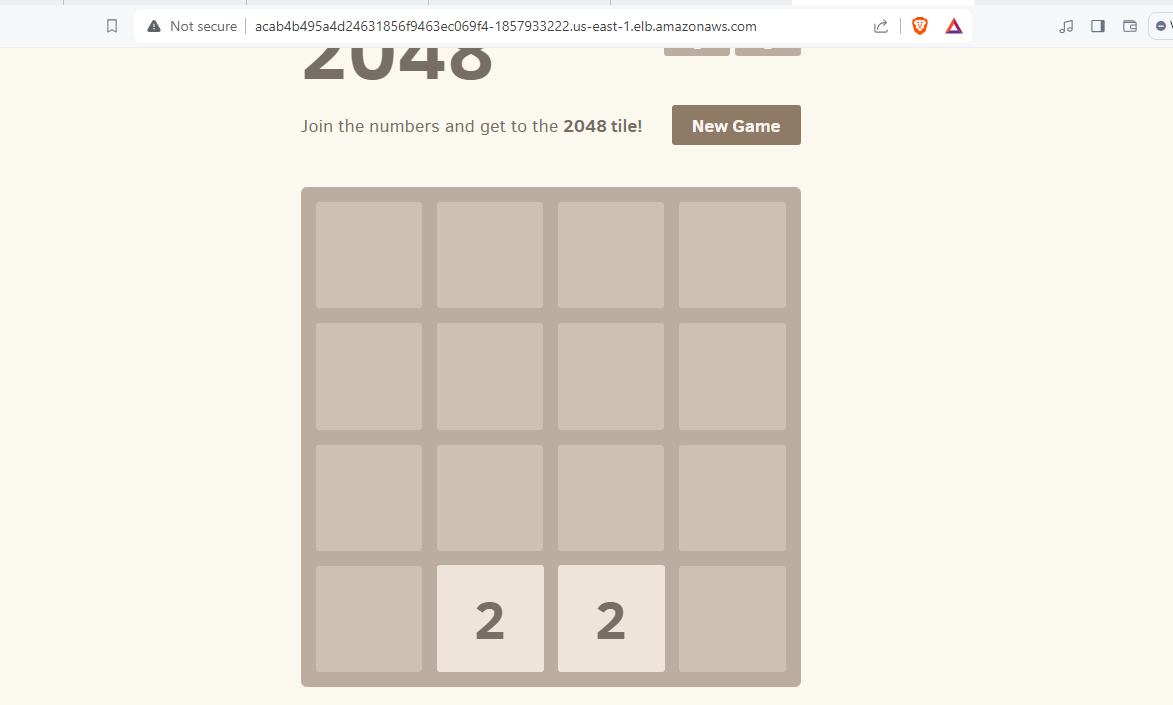
**Go to loadbalancers in aws console and then go to lbs☺(We are looking for status active 1of 1 instances in service):**

****

**Final steps copy the external ip of the lb from here and search it on new tab of your browser:**



**Results:**

****

# The end