EKS-1

1. Create eks cluster using eksctl
   During creation, Specify
   - Cluster name
   - Kubernetes version
   - Control plane role
   - Subnets for Control Plane
   - Control Plane security Group
   - Add tag: owner, purpose on Control Plane
   - Node Group Name
   - Node Instance Role
   - Subnets for Node Group
   - Node Instance SSH key pair
   - Node Instance Security Group
   - Node Instance Instance Type
   - Node Instance Disk
   - Add tag: owner, purpose on Node Group
   - Node Group Size: min, max

First we have to create a yml file for the cluster

```
nodeGroups:
        - name: t34ak-cluster
          iam:
                instanceProfileARN: "arn:aws:iam::187632318301:instance-profile/EKSNodeInstanceRole"
          availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c"]
          ssh:
                allow: true
                publicKeyName: t34ak
          securityGroups:
                withShared: true
                withLocal: false
                attachIDs: [sg-0997df8eb9d968770]
          instanceType: t3.medium
          desiredCapacity: 2
          minSize: 1
          maxSize: 5
          volumeSize: 8
                                                                    43,0-1        Bot
```

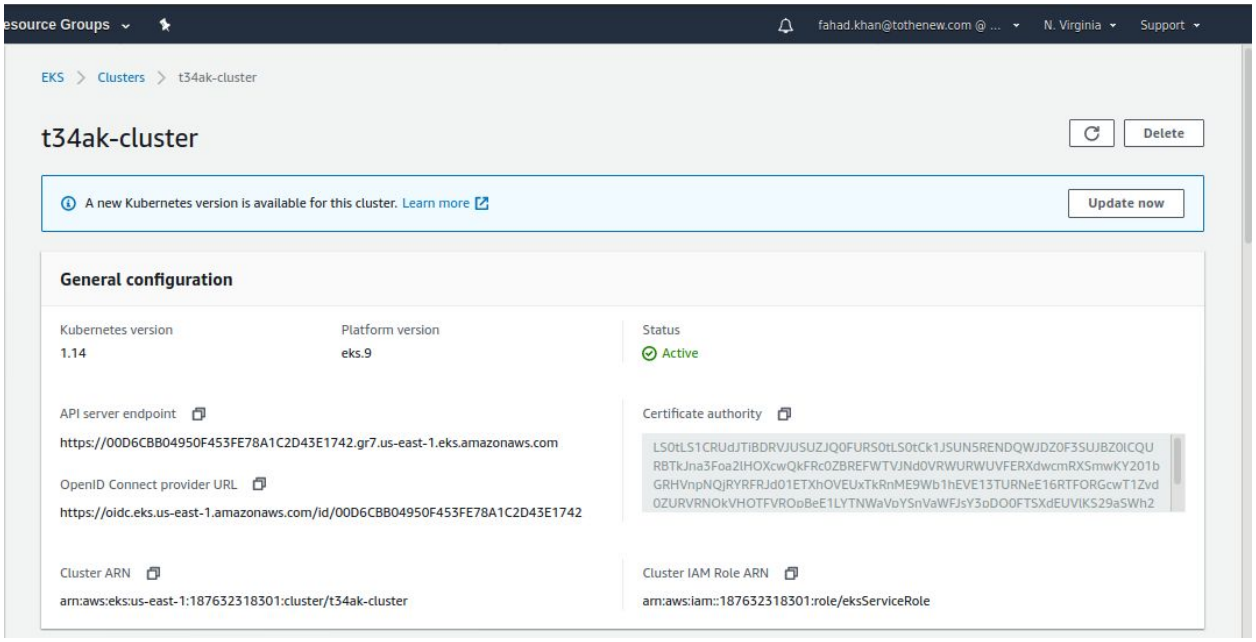
Now run eksctl create cluster command to create the cluster and node group in it

```
fahad@fahad ~/cluster  ‹master*›
➤ eksctl create cluster -f cluster.yml
[ℹ]  eksctl version 0.13.0
[ℹ]  using region us-east-1
[✔]  using existing VPC (vpc-0b061c711cd6ec803) and subnets (private:[] public:[subnet-0a5a6b106347d1b70 subnet-033003c9
2989d26d9 subnet-070c80956ba0dde0a])
[!]  custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configu
ration of VPC/subnets
[ℹ]  nodegroup "t34ak-cluster" will use "ami-087a82f6b78a07557" [AmazonLinux2/1.14]
[ℹ]  using EC2 key pair "t34ak"
[ℹ]  using Kubernetes version 1.14
[ℹ]  creating EKS cluster "t34ak-cluster" in "us-east-1" region with un-managed nodes
[ℹ]  1 nodegroup (t34ak-cluster) was included (based on the include/exclude rules)
[ℹ]  will create a CloudFormation stack for cluster itself and 1 nodegroup stack(s)
[ℹ]  will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
[ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 -
-cluster=t34ak-cluster'
[ℹ]  CloudWatch logging will not be enabled for cluster "t34ak-cluster" in "us-east-1"
[ℹ]  you can enable it with 'eksctl utils update-cluster-logging --region=us-east-1 --cluster=t34ak-cluster'
[ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "t34ak-clus
ter" in "us-east-1"
[ℹ]  2 sequential tasks: { create cluster control plane "t34ak-cluster", create nodegroup "t34ak-cluster" }
[ℹ]  building cluster stack "eksctl-t34ak-cluster-cluster"
[ℹ]  deploying stack "eksctl-t34ak-cluster-cluster"
[ℹ]  building nodegroup stack "eksctl-t34ak-cluster-nodegroup-t34ak-cluster"
[ℹ]  deploying stack "eksctl-t34ak-cluster-nodegroup-t34ak-cluster"
[✔]  all EKS cluster resources for "t34ak-cluster" have been created
[!]  unable to write kubeconfig , please retry with 'eksctl utils write-kubeconfig -n t34ak-cluster': unable to read exi
sting kubeconfig file "/home/fahad/.kube/config": Error loading config file "/home/fahad/.kube/config": open /home/fahad
/.kube/config: permission denied
[ℹ]  adding identity "arn:aws:iam::187632318301:role/EKSNodeInstanceRole" to auth ConfigMap
[ℹ]  nodegroup "t34ak-cluster" has 0 node(s)
[ℹ]  waiting for at least 1 node(s) to become ready in "t34ak-cluster"
[ℹ]  nodegroup "t34ak-cluster" has 2 node(s)
[ℹ]  node "ip-192-168-148-213.ec2.internal" is ready
```

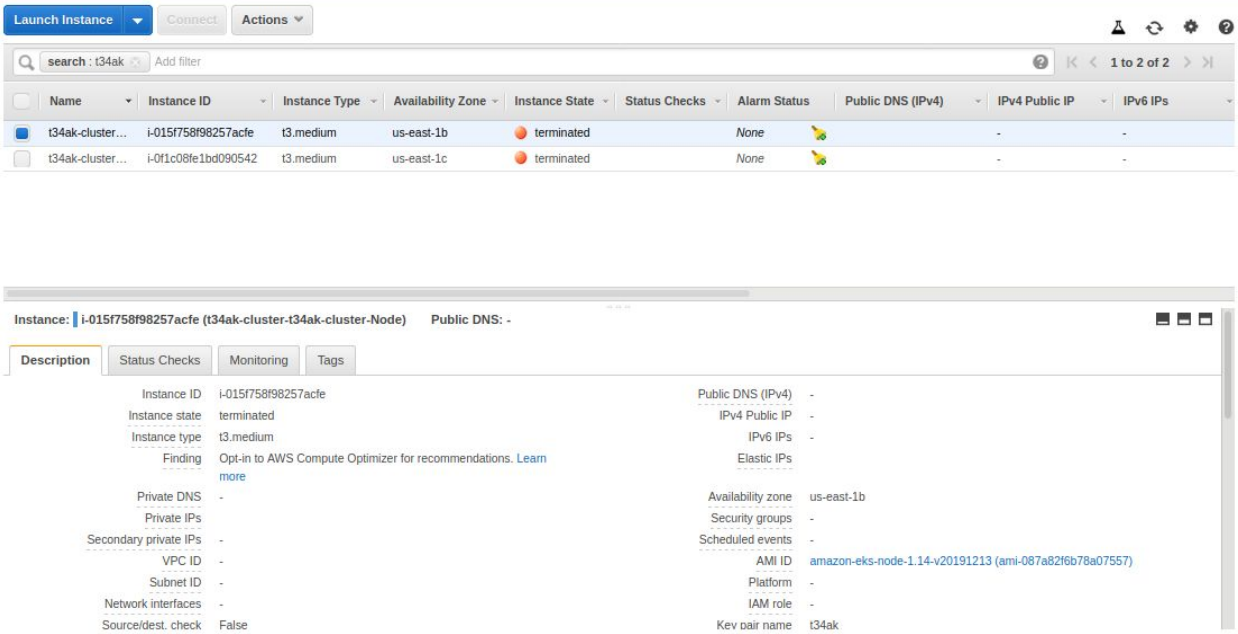
We can see list all nodes by kubectl get nodes command

```
root@fahad:cluster # kubectl get nodes
NAME                                STATUS   ROLES    AGE   VERSION
ip-192-168-148-213.ec2.internal   Ready    <none>   15m   v1.14.8-eks-b8860f
ip-192-168-235-201.ec2.internal   Ready    <none>   15m   v1.14.8-eks-b8860f
root@fahad:cluster #
```

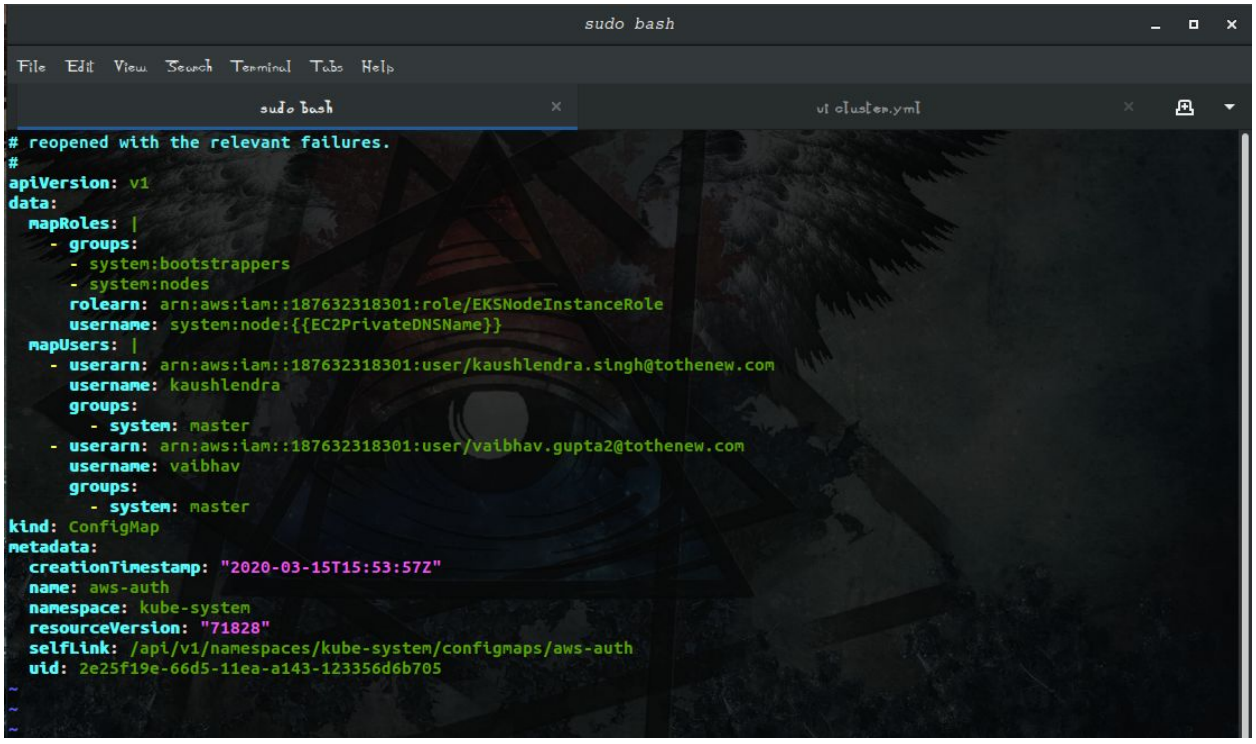
Now we can see that cluster is created



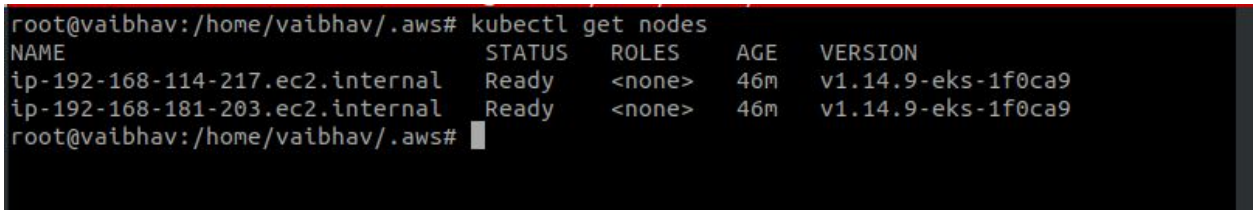
And the instances are also created and running



2. Authentication Management
    a. Add new 2 IAM user into the cluster
    b. Add arn of the users whom we want to give access to the cluster

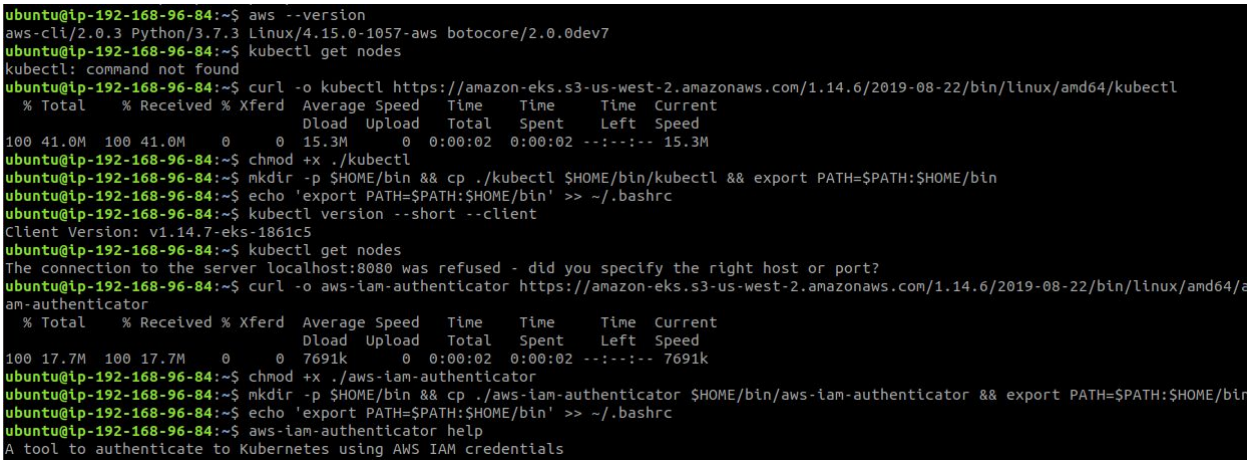


Now we can see the nodes using kubectl get nodes on that user's

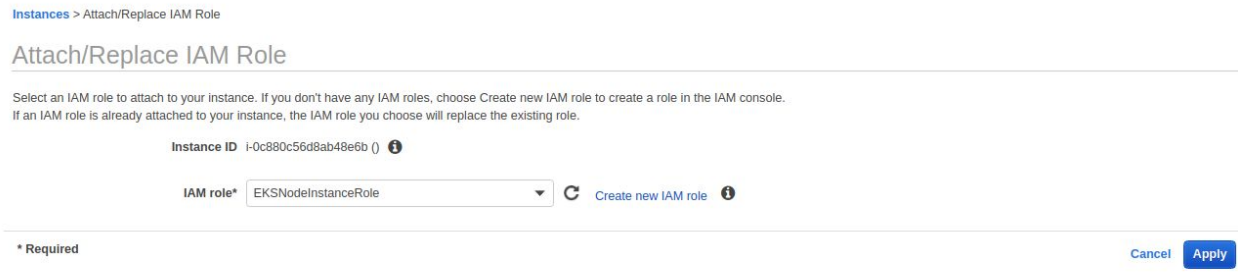


c.Enable a EC2 server to access Cluster master API without using access/secret key

Login to a instance and download aws, kubectl and aws-iam-authenticator



Attach a role whose ARN is present in the aws auth file for cluster



Then we will be able to see all the nodes

```
Use "aws-iam-authenticator [command] --help" for more information about a command.
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-192-168-96-84:~$ aws eks --region us-east-1 update-kubeconfig --name group5
Added new context arn:aws:eks:us-east-1:187632318301:cluster/group5 to /home/ubuntu/.kube/config
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
error: You must be logged in to the server (Unauthorized)
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
NAME                                        STATUS   ROLES    AGE   VERSION
ip-192-168-114-217.ec2.internal   Ready    <none>   35m   v1.14.9-eks-1f0ca9
ip-192-168-181-203.ec2.internal   Ready    <none>   35m   v1.14.9-eks-1f0ca9
ubuntu@ip-192-168-96-84:~$
```

3. Eksctl command to terminate the stack

```
rishabh@rishabh:Downloads $ eksctl delete cluster --region=us-east-1 --name=group5
[ℹ]  eksctl version 0.14.0
[ℹ]  using region us-east-1
[ℹ]  deleting EKS cluster "group5"
[ℹ]  deleted 0 Fargate profile(s)
[✔]  kubeconfig has been updated
[ℹ]  cleaning up LoadBalancer services
[ℹ]  2 sequential tasks: { delete nodegroup "g5-node1", delete cluster control plane "group5" [async] }
[ℹ]  will delete stack "eksctl-group5-nodegroup-g5-node1"
[ℹ]  waiting for stack "eksctl-group5-nodegroup-g5-node1" to get deleted
[ℹ]  will delete stack "eksctl-group5-cluster"
[✔]  all cluster resources were deleted
```

-