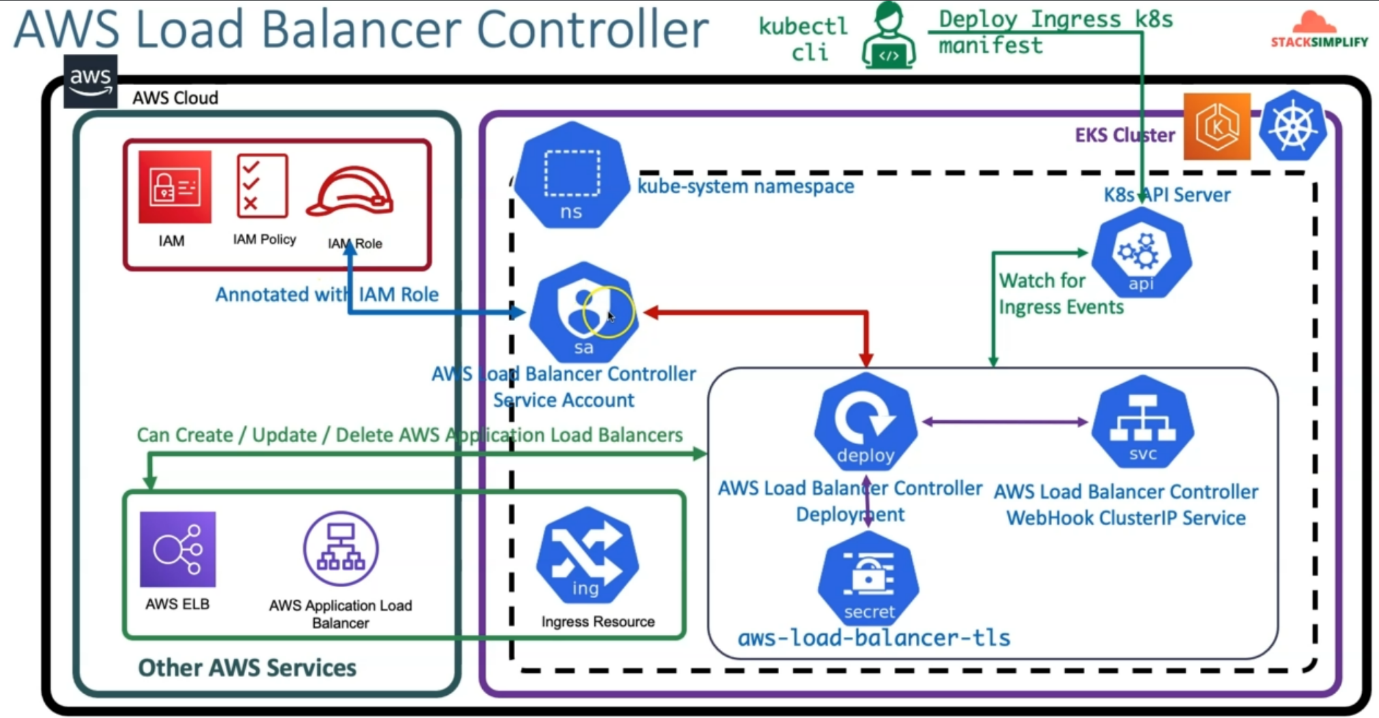
**02. Verify Pre-requisites**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/08-NEW-ELB-Application-LoadBalancers/08-01-Load-Balancer-Controller-Install>

**Introduction**



1. Create IAM Policy and make a note of Policy ARN
2. Create IAM Role and k8s Service Account and bound them together
3. Install AWS Load Balancer Controller using HELM3 CLI
4. Understand IngressClass Concept and create a default Ingress Class

**Pre-requisites**

**Pre-requisite-1: eksctl & kubectl Command Line Utility should be installed**

--- **note** - Should be the latest eksctl version (should be installed latest eksctl version)

**# Verify eksctl version**

--- **eksctl version**

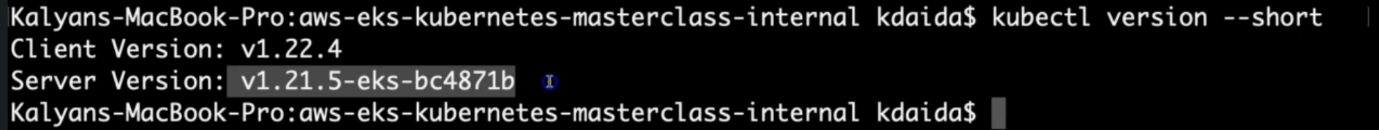
**0.82.0** – as of now it is the latest version, it may be different for you.

**# For installing or upgrading latest eksctl version**

--- **https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html**

**# Verify EKS Cluster version**

--- **kubectl version –short**



--- **note** – it will give us the both client and server version. As our client version is v1.22.4 and it will support the server version of **v1.21, v1.22** and **v1.23**.

--- **kubectl version**

--- **Important Note**: You must use a kubectl version that is within one minor version difference of your Amazon EKS cluster control plane. For example, a 1.20 kubectl client works with Kubernetes 1.19, 1.20 and 1.21 clusters.

**# For installing kubectl cli**

--- <https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

**Pre-requisite-2: Create EKS Cluster and Worker Nodes (if not created)**

**# Create Cluster (Section-01-02)**

--- **eksctl create cluster --name=eksdemo1 \**

**--region=us-east-1 \**

**--zones=us-east-1a,us-east-1b \**

**--version="1.21" \**

**--without-nodegroup**

**# Get List of clusters (Section-01-02)**

--- **eksctl get cluster**

**# Template (Section**-01-02)

--- **eksctl utils associate-iam-oidc-provider \**

**--region region-code \**

**--cluster <cluter-name> \**

**--approve**

**# Replace with region & cluster name (Section-01-02)**

**eksctl utils associate-iam-oidc-provider \**

**--region us-east-1 \**

**--cluster eksdemo1 \**

**--approve**

**# Create EKS NodeGroup in VPC Private Subnets (Section-07-01)**

--- **eksctl create nodegroup --cluster=eksdemo1 \**

**--region=us-east-1 \**

**--name=eksdemo1-ng-private1 \**

**--node-type=t3.medium \**

**--nodes-min=2 \**

**--nodes-max=4 \**

**--node-volume-size=20 \**

**--ssh-access \**

**--ssh-public-key=kube-demo \**

**--managed \**

**--asg-access \**

**--external-dns-access \**

**--full-ecr-access \**

**--appmesh-access \**

**--alb-ingress-access \**

**--node-private-networking**

--- **note** – after the pre-requisites-2, we will have eks cluster and nodes groups in private subnet.

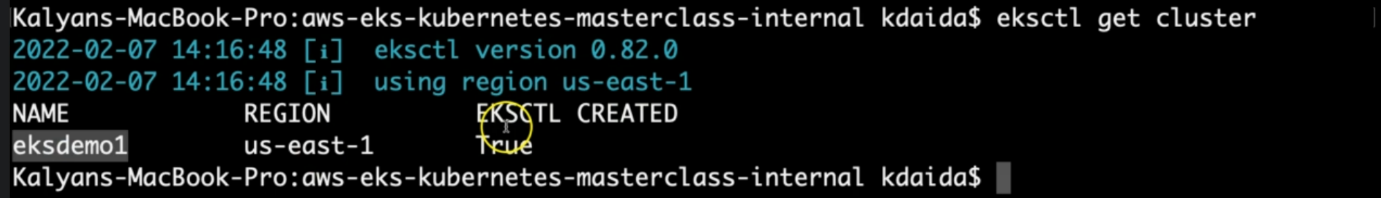
**Pre-requisite-3: Verify Cluster, Node Groups and configure kubectl cli if not configured**

--- EKS Cluster

--- EKS Node Groups in Private Subnet

**# Verify EKS Cluster**

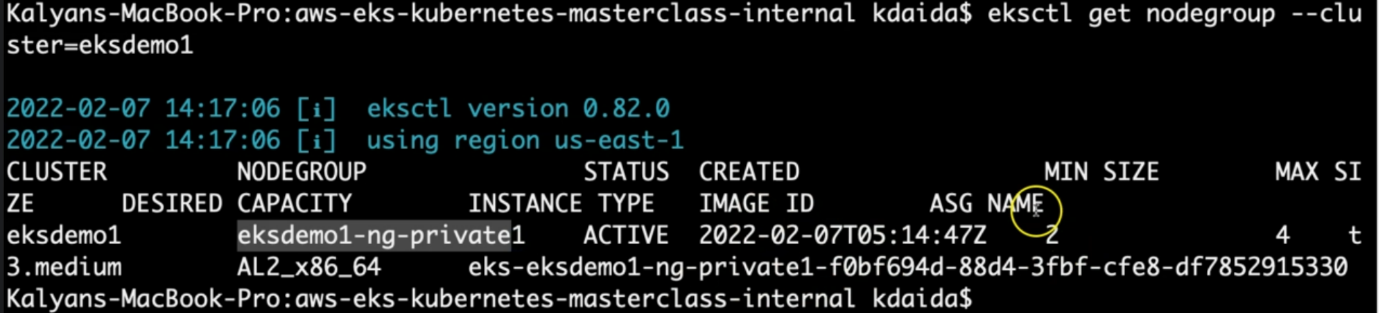
--- **eksctl get cluster**



--- our EKS cluster eksdemo1 is up and running and it is created in EKSCTL. It is in the us-east-1 region.

**# Verify EKS Node Groups**

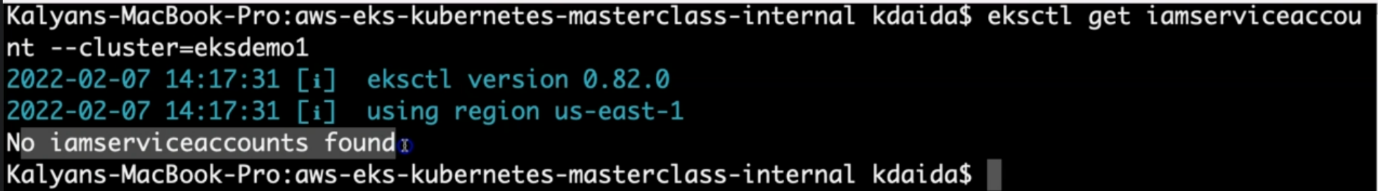
--- **eksctl get nodegroup --cluster=eksdemo1**



--- the nodegroup name is eksdemo1-ng-private1 and the instances in the nodes group is 2.

**# Verify if any IAM Service Accounts present in EKS Cluster**

--- **eksctl get iamserviceaccount --cluster=eksdemo1**



--- there are no IAM accounts present inside of this eksdemo1 cluster.

**Observation:**

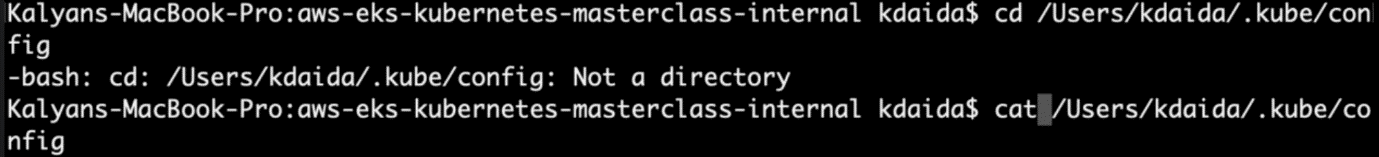
1. No k8s Service accounts as of now.

**# Configure kubeconfig for kubectl**

--- **eksctl get cluster** - TO GET CLUSTER NAME

--- **aws eks --region <region-code> update-kubeconfig --name <cluster\_name>**

--- **aws eks --region us-east-1 update-kubeconfig --name eksdemo1**



--- if you’re not configured kubectl kube config, whenever the eks cluster is created, it will automatically create kube config.

--- **note** – this will have the information of eks cluster connecting. This file will contain all the authentication information.

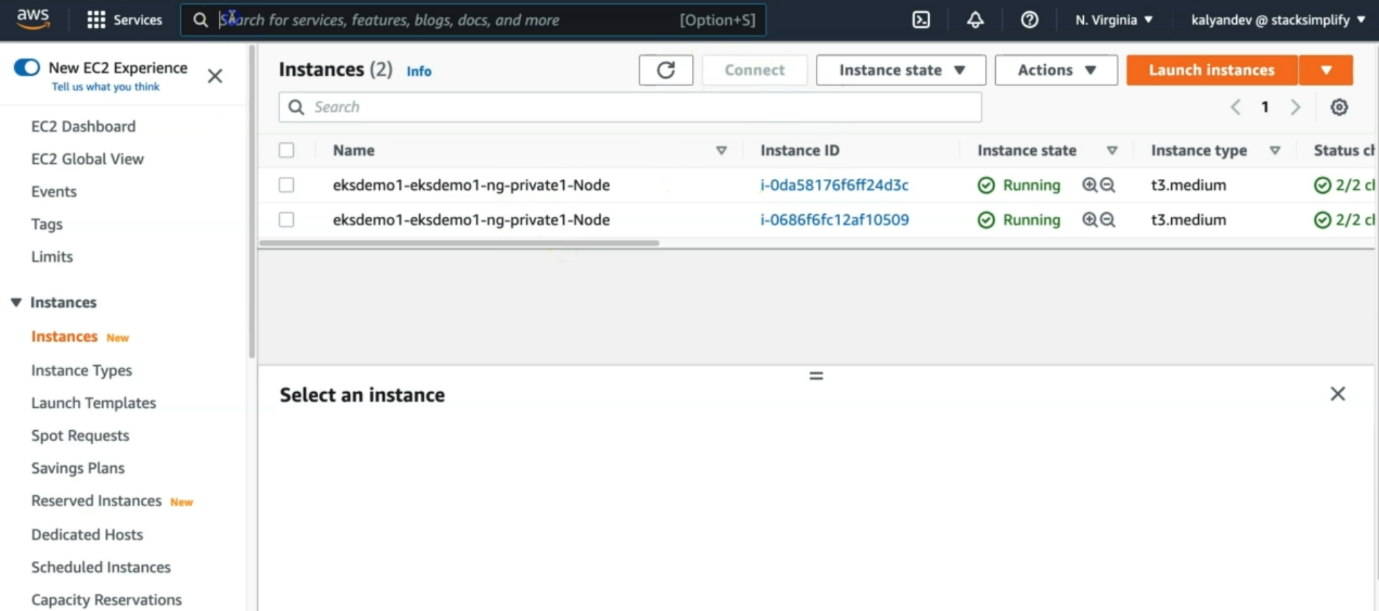
--- **note** – from your local instance you’re not able to connect your remote cluster information like nodes...etc then you can apply command on your local instance. It will create kube config file in your local and add the content to the kube config file.

**# Verify EKS Nodes in EKS Cluster using kubectl**

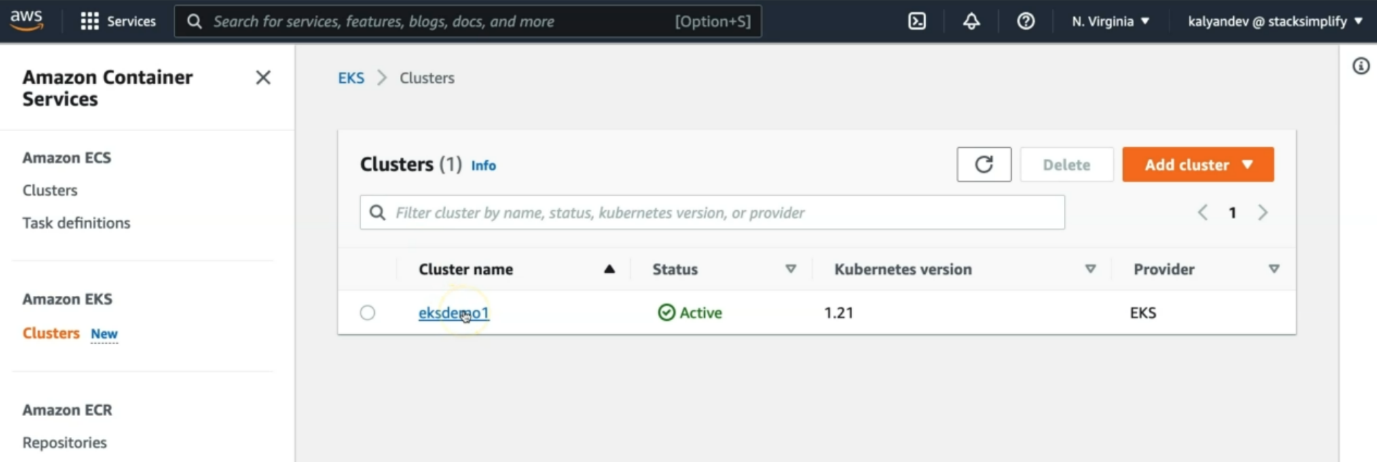
--- **kubectl get nodes**

**# Verify using AWS Management Console**

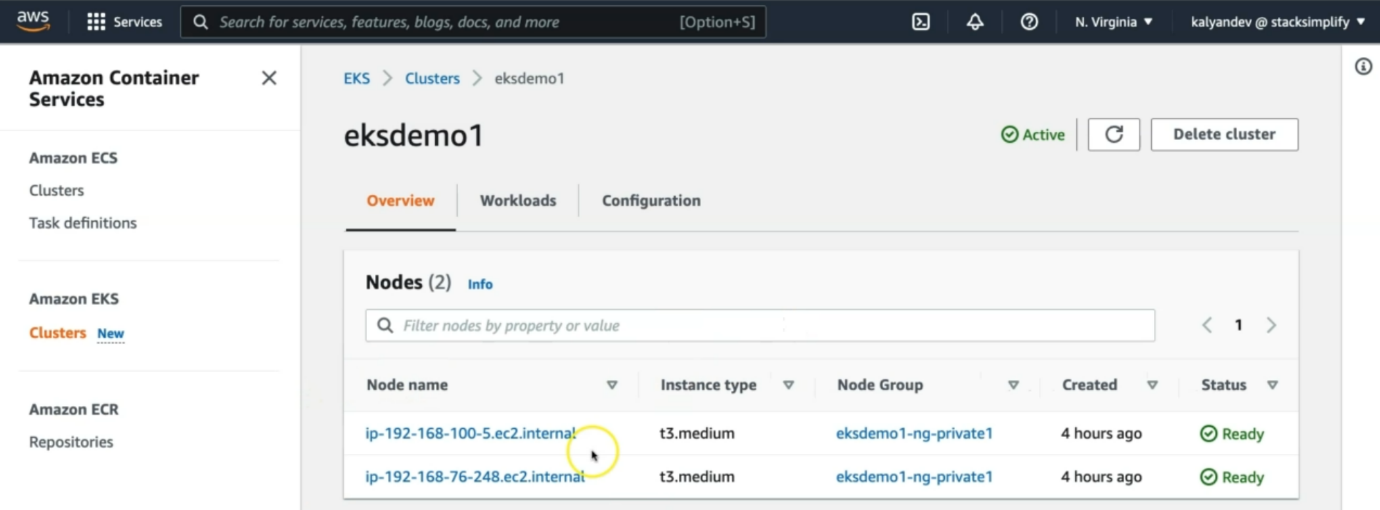
1. EKS EC2 Nodes (Verify Subnet in Networking Tab)
2. EKS Cluster



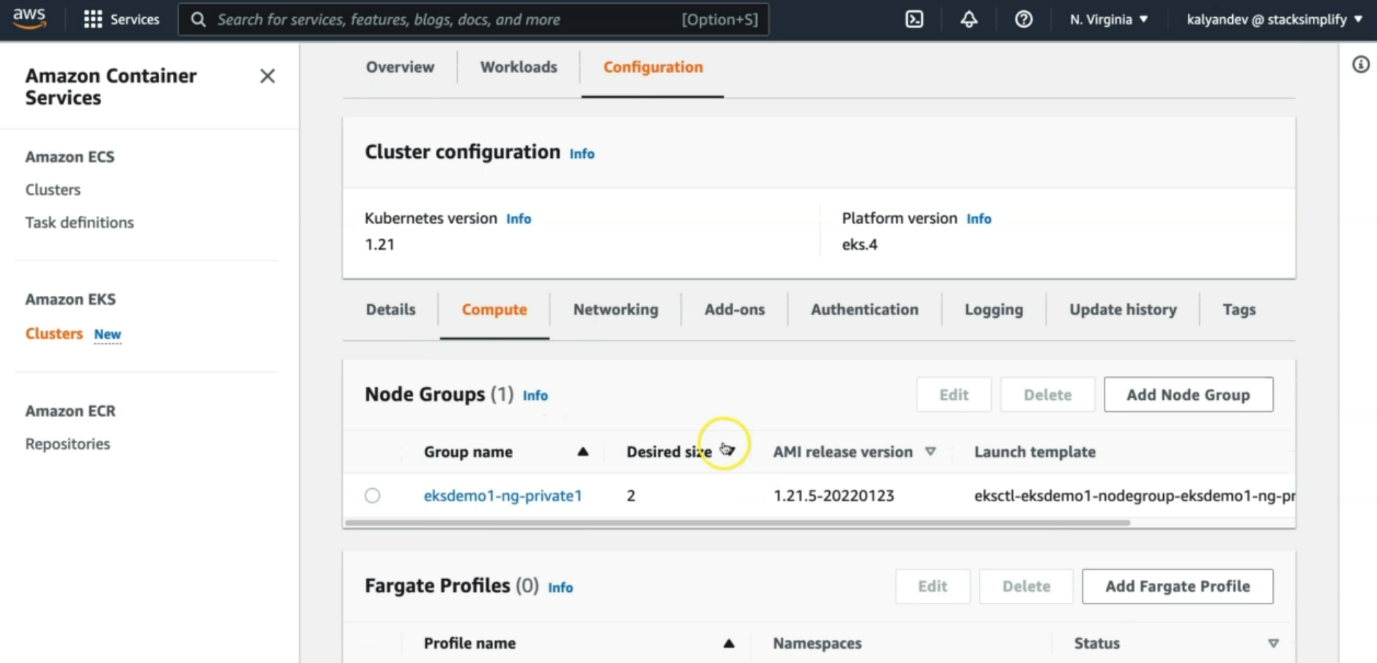
--- **note** - we have the 2 private instances running in the node group. Go to the EKS kubernetes service and verify the same.



--- click on the cluster name. here we can also see the kubernetes version.



--- **note** – we can see that the same private instances are running inside of kubernetes cluster.



--- click on the configuration to see the cluster configuration information.