Kubernetes Task-2

Task Description:

Create the K8s EKS,further you have to do the deployment of the Nginx application and access the application outside the cluster.

`eksctl create cluster` output:


```
Kustomize version: v5.4.2
vineeth@vineeth-pc: $ eksctl create cluster --name dev-cluster --region ap-south-1 --nodegroup-name dev-nodes --node-type t3.medium --nodes 2 --nodes-min 1 --nodes-max 3 --managed
2025-08-25 11:19:52 [i]  eksctl version 0.212.0
2025-08-25 11:19:52 [i]  using region ap-south-1
2025-08-25 11:19:52 [i]  setting availability zones to [ap-south-1c ap-south-1a ap-south-1b]
2025-08-25 11:19:52 [i]  subnets for ap-south-1c - public:192.168.0.0/19 private:192.168.96.0/19
2025-08-25 11:19:52 [i]  subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19
2025-08-25 11:19:52 [i]  subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19
2025-08-25 11:19:52 [i]  nodegroup "dev-nodes" will use "" [AmazonLinux2023/1.32]
2025-08-25 11:19:52 [i]  using Kubernetes version 1.32
2025-08-25 11:19:52 [i]  creating EKS cluster "dev-cluster" in "ap-south-1" region with managed nodes
2025-08-25 11:19:52 [i]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-08-25 11:19:52 [i]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=dev-cluster'
2025-08-25 11:19:52 [i]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "dev-cluster" in "ap-south-1"
2025-08-25 11:19:52 [i]  CloudWatch logging will not be enabled for cluster "dev-cluster" in "ap-south-1"
2025-08-25 11:19:52 [i]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=dev-cluster'
2025-08-25 11:19:52 [i]  default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-08-25 11:19:52 [i]
2 sequential tasks: { create cluster control plane "dev-cluster",
    2 sequential sub-tasks: {
        2 sequential sub-tasks: {
            1 task: { create addons },
            wait for control plane to become ready,
        },
        create managed nodegroup "dev-nodes",
    }
}
2025-08-25 11:19:52 [i]  building cluster stack "eksctl-dev-cluster-cluster"
2025-08-25 11:19:52 [i]  deploying stack "eksctl-dev-cluster-cluster"
2025-08-25 11:20:22 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:20:53 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:21:53 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:22:53 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:23:53 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:24:54 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:25:54 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:26:55 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:27:55 [i]  waiting for CloudFormation stack "eksctl-dev-cluster-cluster"
2025-08-25 11:27:56 [!]  recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide I
AM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under `addon.PodIdentityAssociations`, and run `eks
ctl update addon`
2025-08-25 11:27:56 [i]  creating addon: vpc-cni
2025-08-25 11:27:56 [i]  successfully created addon: vpc-cni
2025-08-25 11:27:57 [i]  creating addon: kube-proxy
2025-08-25 11:27:57 [i]  successfully created addon: kube-proxy
2025-08-25 11:27:58 [i]  creating addon: coredns
```

kubectl get nodes:


```
vineeth@vineeth-pc:~/eks-nginx$ kubectl get nodes
NAME                                          STATUS   ROLES    AGE   VERSION
ip-192-168-5-178.ap-south-1.compute.internal    Ready    <none>   12m   v1.32.7-eks-3abbec1
ip-192-168-67-123.ap-south-1.compute.internal   Ready    <none>   12m   v1.32.7-eks-3abbec1
vineeth@vineeth-pc:~/eks-nginx$
```

kubectl get deployments:


```
vineeth@vineeth-pc:~/eks-nginx$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           8m11s
vineeth@vineeth-pc:~/eks-nginx$
```

kubectl get svc:


```
vineeth@vineeth-pc:~/eks-nginx$ kubectl get svc
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP                                                                    PORT(S)        AGE
kubernetes      ClusterIP      10.100.0.1      <none>                                                                         443/TCP        19m
nginx-service   LoadBalancer   10.100.70.191   a811f65d855f84aaaa3d909fd0054a86-914978965.ap-south-1.elb.amazonaws.com       80:31301/TCP   7m14s
vineeth@vineeth-pc:~/eks-nginx$
```