EKS assignment

Task: Complete the list of tasks below.

Task 1:

- Create your cluster and then create a Ngnix deployment yaml file with a service.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sample-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: public.ecr.aws/nginx/nginx:1.19.6
          ports:
            - name: http
              containerPort: 80

---
apiVersion: v1
kind: Service
metadata:
  name: nginx-service-nodeport
spec:
  type: NodePort
```

```
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

- Create another yaml file for the ingress controller

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: simple-ingress
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/target-type: instance
spec:
  rules:
    - http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service-nodeport
                port:
                  number: 80
```

Task 2:

- You will need to add OpenID connect to your cluster. Enter the command below to view cluster ID:

```
 aws eks describe-cluster --name {your cluster name} \
--query "cluster.identity.oidc.issuer" --output text
```

- Then enter this command for the to view if OpenID connect is connected to your cluster:

```
aws iam list-open-id-connect-providers
```

- Enter this command to add OpenID to the cluster:

```
eksctl utils associate-iam-oidc-provider --cluster {your cluster name} --approve
```

Task 3:

- Download the Role Base Access Control:

```
curl -o rbac-role.yaml \

https://raw.githubusercontent.com/RobinNagpal/kubernetes-tutorials/master/06_tools/007_alb_ingress/01_eks/rbac-role.yaml
```

- You will see a yaml file called rbac-role.yaml (explore the file)
- Apply the file with `kubectl apply -f rbac-role.yaml`
- Next download the iam policy with the following command below:

```
curl -o iam_policy.json
https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.3.0/docs/install/iam_policy.json
```

- You will see a file called iam_policy.json (explore the file)
- Next you will create the AWS policy with the following command:

```
aws iam create-policy \
    --policy-name AWSLoadBalancerControllerIAMPolicy \
    --policy-document file://iam_policy.json
```

- Next create the service account:

```
eksctl create iamserviceaccount \
  --cluster=mycluster01 \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --attach-policy-arn=arn:aws:iam::266686430719:policy/AWSLoadBalancerControllerIAMPolicy \
  --override-existing-serviceaccounts \
  --approve
```

- Next create certificate manager for the ingress controller:

```
kubectl apply \
    --validate=false \
    -f https://github.com/jetstack/cert-manager/releases/download/v1.5.4/cert-manager.yaml
```

Task 4:

- Time to make the load balancer controller by downloading and running the following commands:

```
curl -Lo v2_3_0_full.yaml
https://github.com/kubernetes-sigs/aws-load-balancer-controller/releases/download/v2.3.0/v2_3_0_full.yaml
```

- Edit the file that was downloaded v2_3_0_full.yaml (replace {`cluster-name=`*`your-cluster-name`*`}` with your cluster name)
- Now enter the following command below:

```
kubectl apply -f v2_3_0_full.yaml
```

- Use this command to view the controller:

```
kubectl get deployment -n kube-system aws-load-balancer-controller
```

Task 5 :

- Create your application in EKS by creating the deployment and service yaml file. Also the ingress yaml file.

- Test out your application by checking the ALB url