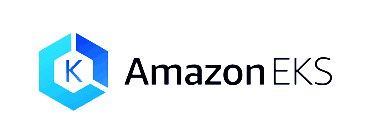
**Kubernetes!!**



# Objective:

Create an application environment with EKS.

# Task: Complete the list of tasks below.

Task 1:

Below is the deployment.yaml file you should create in your directory. Replace image with your image and replace port numbers with your port numbers. You also might want to change the name (optional).

|  |
| --- |
| apiVersion: apps/v1 kind: Deployment metadata:  labels:  app: url-app  name: url-app spec:  replicas: 2  selector:  matchLabels:  app: url-app  template:  metadata:  labels:  app: url-app  spec:  containers:  - image: You enter  name: You enter  ports:  - name: http  containerPort: 80 --- apiVersion: v1 kind: Service metadata:  labels:  app: url-app  name: url-app spec:  ports:  - name: you enter  port: you enter  protocol: TCP  targetPort: you enter  selector:  app: url-app  type: NodePort |

* Below is the ingress.yaml file

|  |
| --- |
| apiVersion: networking.k8s.io/v1 kind: Ingress metadata:  name: url-app  annotations:  kubernetes.io/ingress.class: alb  alb.ingress.kubernetes.io/scheme: internet-facing  alb.ingress.kubernetes.io/target-type: instance spec:  rules:   - http:  paths:  - backend:  service:  name: url-app  port:  number: you enter  path: /  pathType: Prefix |

Task 2:

* Create your cluster using the command below (This will take awhile):

eksctl create cluster --name cluster001

* You will need to add OpenID connect to your cluster.
* Enter this command to add OpenID to the cluster:

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

* Enter this command for the to view if OpenID connect is connected to your cluster:

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

* The output of the above command will show you an ARN. Copy the AWS ID number in between iam and oidc-provider. I provided an example below:

"arn:aws:iam::266686430719:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/55EB49DA2FA3AE8CAB259699883A3EE8"

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={your cluster name} \  --namespace=kube-system \  --name=aws-load-balancer-controller \  --attach-policy-arn=arn:aws:iam::{AWS ID number}: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 -o 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