# AWS EKS

## Install kubectl

```
$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.1/2023-04-19/bin/linux/amd64/kubectl
```

```
$ chmod +x ./kubectl
```

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
```

```
kubectl version --short --client
```

## Create EKS Role to manage AWS services

- From IAM Service creat Role
- select AWS Service
- EKS , then EKS-Cluster
- attach : AmazonEKSClusterPolicy
- Next
- give a name to the role
- create role

## Creat Role for node group

- From IAM Service creat Role
- select AWS Service
- Ec2
- attach : AmazonEKSWorkerNodePolicy, AmazonEKS_CNI_Policy, AmazonEC2ContainerRegistryReadOnly
- next
- give a name

- create role

## Create Cluster

- EKS Service
- enter name
- keep the default version
- Role : choose the eks cluster role which created earlier
- networkin: keep default
- Cluster endpoint access: select public
- next
- Configure logging: keep default
- Select add-ons: keep default
- next
- Configure selected add-ons settings: keep default
- next
- review

## Add Node Group to the Cluster (EC2 instances or fargate, here we choose EC2 insatnces)

- Select cluster
- Compute Tab
- enter the name
- choose the role which created earlier for the node group
- everything else is default
- next
- Node group compute configuration:
- change as needed
- next
- networking: default
- next
- review and create

## Cluster Authentication

### Creating or updating a kubeconfig file for an Amazon EKS cluster

- make sure that kubectl installed earlier
- automatically create or update a kubeconfig file for your Amazon EKS cluster with the AWS CLI update-kubeconfig command.
- First run the following on CloudShell, to see the user who created the cluster

```
aws sts get-caller-identity
```

- user

- amazon cli should be 1.27.160 or later

```
aws --version
```

- create the config file, replace the region and the cluster name:

```
aws eks update-kubeconfig --region region-code --name my-cluster
```

- now we should be able to run kubectl command:

```
kubectl get nodes
```

- Deploy application to the cluster:

see the file "EKS Deploy Application.pdf"