Demo – 1 :  EKS  Node Group

**Prerequisites:**

* AWS Account with Admin Privileges
* AWS CLI Access to use Kubectl utility
* Sign into AWS as a user, create access key and secret keys for that user
* Navigate to vs code on terminal to and configure aws and pass creds using the below command

- aws configure

**Step by step procedure:**

Step – 1 : Create **IAM role** for EKS Cluster

Step – 2 : Create **dedicated VPC** for the EKS Cluster

Step – 3 : Create **EKS Cluster**

Step – 4 : Install & Setup IAM Authenticator, Kubectl utility and Updated AWS CLI

Step – 5 : Create **IAM Role** for EKS Worker Nodes

Step – 6 : Create **Worker Nodes**

Step – 7 : Deploying Demo Application

Step – 1 : Create IAM role for EKS Cluster

* Login AWS console
* Navigate to IAM Console

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a screenshot of a screenshot

Description automatically generated

A screenshot of a computer

Description automatically generated

Step – 2 : Create dedicated VPC for the EKS Cluster

* Navigate to CloudFormation Console

A screenshot of a computer

Description automatically generated

* Use the below url in Amazon S3 Url section
* https://amazon-eks.s3.us-west-2.amazonaws.com/cloudformation/2020-04-21/amazon-eks-vpc-private-subnets.yaml

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Step – 3 : Create EKS Cluster

* Navigate to EKS Console

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

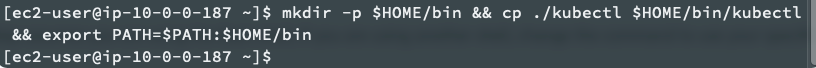
Step – 4 : Install Kubectl utility

* We have to install Kubectl as a part of prerequisite on our local terminal
* Navigate to below link
  + <https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>
* Follow all the commands and install kubectl

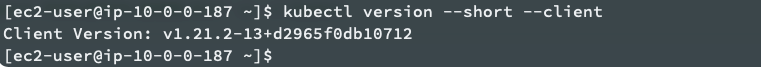
A screen shot of a computer

Description automatically generated









* Kubectl installed successfully
* Upgrade the AWS CLI
* curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"
* unzip awscliv2.zip
* sudo ./aws/install --update

* We have to run below command to add EKS cluster to kubeconfig
  + aws eks --region <region where cluster is deployed> update-kubeconfig --name <name of cluster>

A screenshot of a computer

Description automatically generated

* Run below command to setup kubeconfig in environment variable
  + export KUBECONFIG=~/.kube/config
* Run below command to get services available
  + kubectl get svc
* you should see something like below

A screen shot of a computer

Description automatically generated

* Run below command to get namespace available
  + **kubectl get ns**

A screenshot of a computer

Description automatically generated

* Everything looks good now let’s setup worker nodes

Step – 5 : **Create IAM Role for EKS Worker Nodes**

* Navigate to IAM Console

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Attach below three policies in the next step

A group of blue text

Description automatically generated

A screenshot of a computer

Description automatically generated

Step – 6 : Create Worker Nodes

* Navigate to EKS Console
* Select cluster you just created
* select compute and finally add node group

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

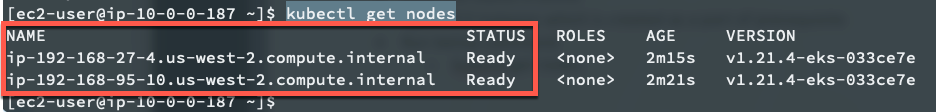
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Once the status is active
* Run below command on vs code terminal
  + kubectl get nodes



https://phoenixnap.com/kb/wp-content/uploads/2021/11/kubectl-commands-cheat-sheet-by-pnap.pdf

https://phoenixnap.com/kb/kubectl-commands-cheat-sheet

Step – 7 : Deploying Demo Application

* Clone the code
  + git clone <https://github.com/avinashmamidi/aws.git>
  + cd aws/kubernetes/knote-example/
* Deploy app by running below commands
  + kubectl apply -f mongo.yaml
  + kubectl apply -f knote.yaml
* Check the services running
  + kubectl get svc
  + kubectl get pods

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Cleanup The Environment:

1. Delete the NodeGroup on the EKS cluster
2. Delete EKS Cluster
3. Delete IAM Roles
4. Delete VPC