EKS Cluster SetUp(Managed K8S-Cluster)

1. Create IAM Role for EKS Cluster

2. AWS EKS VPC CloudFormation

3. Create EKS Cluster

4. Create IAM Role for EKS Worker Nodes
A. AmazonEKSWorkerNodePolicy
B. AmazonEKS_CNI Policy
C.AmazonEC2ContainerRegisteyReadOnlyPloicy

5. Create Worker Nodes

6. Create an Instance, Install Kubectl, AWS-Cli, configure AWS,IAM Authenticator(/kube/config) aws eks update-kubeconfig --name democluster --region ap-south-1

Note: If i deploy a Cluster-Autoscaler , it has access to K8s API's as well as AWS-API's.
Note: I am not able to Achieve Cluster-AutoScaler if i use Self-Managed/Bare Metal Cluster.But I achieve HPA for that Self-Managed/Bare Metal Cluster

7). ***Cluster AutoScaler Deployment in EKS***

Note: Cluster-AutoScaler also running as a Pod in K8S-Cluster

A) Create AWS Policy with below Actions OR Create an IAM Policy For Worker Node
---

-> After the creation of EKS, The Cluster Autoscaler requires the following IAM permissions to make calls to AWS APIs on your behalf.
-> I need some AutoScaling Permissions to adjust the desiren number.

-> IAM Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeTags",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "ec2:DescribeLaunchTemplateVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

B) Attach Policy to IAM Role(EKS-WorkerNode-Role) which is used in EKS Node Group(EKS-WorkerNode Group)

-> If i have attached this Role to the server then ClusterAutoScaler will be able to communicate with AWS API's

C)***Deploy the Cluster Autoscaler***

-> Deploy the Cluster Autoscaler to your cluster using this .yaml file

https://raw.githubusercontent.com/kubernetes/autoscaler/master/cluster-autoscaler/cloudprovider/aws/examples/cluster-autoscaler-autodiscover.yaml

Note: I above “cluster-autoscaler-autodiscover.yaml “ UPdate Your cluster Name and Your AWS Region Name

UPdate Your cluster Name Here..
UPdate Your AWS Region Name Here in Which You created the EKS-Cluster

***Note: We will get some StorageClass configured by default after deploying a Cluster-Autoscaler.

*** Note: If you want create a Stateful Application in your K8S-Cluster
-> an AWS EBS volume to be provisioned by the ebs.csi.aws.com provisioner.
-> This could mean that the CSI (Container Storage Interface) driver for AWS EBS.

8)AWS(EKS) EBS Volume to be provisioned by the ebs.csi.aws.com provisioner****

A) Check if the EBS CSI Driver is Installed:

-> $ kubectl get pods -n kube-system

B) Install the EBS CSI Driver (if not installed):

**** Note: Before Executing following command first Install the Git, if not installed.**

-> $ kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/ecr/?ref=release-1.11"

C) Check the StorageClass Configuration:

*****Note: We will get some StorageClass configured by default after deploying a Cluster-Autoscaler.**

D) Verify IAM Role Permissions:

-> Ensure that the IAM role associated with your EKS worker nodes has the necessary permissions to manage EBS volumes. Attach the following IAM policy to the node role:

i) Create a policy with the necessary permissions for the EBS CSI driver. Below is an example IAM policy JSON for the EBS CSI driver:

EKS_EBS_CSI_Policy.JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVolume",
        "ec2:DeleteVolume",
        "ec2:AttachVolume",
        "ec2:DetachVolume",
        "ec2:ModifyVolume",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVolumeAttribute",
        "ec2:CreateTags",
        "ec2:DeleteTags",
```

```
                "ec2:DescribeInstances",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets"
              ],
              "Resource": "*"
            }
        ]
    }
```

ii)Attach the IAM Policy to the Worker Node Role:

-> Go to Roles and find the role used by your EKS worker nodes and attach EKS_EBS_CSI_Policy.

iii) Verify the EBS CSI Driver is Running:

-> List Pods in the kube-system Namespace:
Ex: $ kubectl get pods -n kube-system -l app=ebs-csi-controller
-> Check Logs of the EBS CSI Controller:
Ex: kubectl logs -n kube-system <ebs-csi-controller-pod-name>

9) * Finally,**
--> Deploy Demo Application to check Nodes Automatically Adjusting by ClusterAutoScaler
--> Verify the Volumes(PVC) are Provisioning Dynamically or not by Creating Stateful Applications Like MongoDB and Springapp ***

==> For Mongodb ReplicaSet
Ex: kubectl apply -f mongodbstatefulset.yaml
==> To check Springapp internally connected and working with MongoDB statefulset or not
Ex: kubectl apply -f springappstatefulset.yaml