**AWS EKS Cluster Configuration, Setup & App Deployment**

This documentation provides detailed instructions for setting up and configuring an Amazon Elastic Kubernetes Service (EKS) cluster in the **us-east-1** region. The steps include installing AWS CLI, setting up the EKS cluster, updating kubeconfig, installing kubectl, installing eksctl, associating IAM OIDC provider, creating a service account with required permissions, deploying the AWS EBS CSI Driver, and applying a custom manifest file.

**Step 1: Install AWS CLI**

Follow these steps to install the AWS Command Line Interface (CLI):

1. Download the AWS CLI installer:

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

1. Install the unzip utility (if not already installed):

sudo apt install unzip

1. Unzip the installer:

unzip awscliv2.zip

1. Run the installer:

sudo ./aws/install

1. Configure AWS CLI:

aws configure

**Step 2: Clone the Terraform Project and Deploy EKS**

1. Clone the Terraform project folder and navigate to it:

git clone <terraform\_project\_repo>

cd <terraform\_project\_folder>

1. Initialize Terraform:

terraform init

1. Deploy the resources:

terraform apply --auto-approve

**Step 3: Update kubeconfig for the EKS Cluster**

Run the following command to update the kubeconfig file with your EKS cluster details:

**aws eks --region us-east-1 update-kubeconfig --name devopsshack-cluster**

This command retrieves the kubeconfig configuration for the devopsshack-cluster and stores it in the default kubeconfig file (e.g., ~/.kube/config).

**Step 4: Install kubectl**

Follow these steps to install the latest version of kubectl:

1. Download the kubectl binary:

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

1. Make the binary executable:

chmod +x kubectl

1. Move the binary to a system PATH directory:

sudo mv kubectl /usr/local/bin/

1. Verify the installation:

kubectl version --client

**Step 5: Install eksctl**

Follow these steps to install eksctl, a CLI tool for managing EKS clusters:

1. Download the eksctl tarball:

curl -LO "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_Linux\_amd64.tar.gz"

1. Extract the tarball:

tar -xzf eksctl\_Linux\_amd64.tar.gz

1. Move the binary to a system PATH directory:

sudo mv eksctl /usr/local/bin

1. Verify the installation:

eksctl version

**Step 6: Associate IAM OIDC Provider**

Associate an IAM OIDC provider with your EKS cluster to enable IAM roles for Kubernetes service accounts:

eksctl utils associate-iam-oidc-provider --region us-east-1 --cluster devopsshack-cluster --approve

This command associates the OIDC provider required for enabling IAM roles for service accounts in the EKS cluster.

**Step 7: Create an IAM Service Account**

Create a Kubernetes service account with IAM permissions for the Amazon EBS CSI Driver:

eksctl create iamserviceaccount \

--region us-east-1 \

--name ebs-csi-controller-sa \

--namespace kube-system \

--cluster devopsshack-cluster \

--attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \

--approve \

--override-existing-serviceaccounts

* --name ebs-csi-controller-sa: Name of the service account.
* --namespace kube-system: Namespace where the service account will be created.
* --attach-policy-arn: IAM policy ARN for EBS CSI Driver permissions.
* --approve: Automatically approve the creation.

**Step 8: Deploy the AWS EBS CSI Driver**

Deploy the AWS EBS CSI Driver in the cluster using the following command:

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

This command deploys the latest stable release of the EBS CSI Driver from the official repository.

**Step 9: Deploy Application Using Manifest File**

After setting up the cluster, deploy your application using a Kubernetes manifest file:

kubectl apply -f manifest.yaml

Replace manifest.yaml with the path to your Kubernetes manifest file.

**Verification**

1. Verify that the kubeconfig is updated and the cluster is accessible:

kubectl get nodes

1. Verify that the service account is created:

kubectl get serviceaccount ebs-csi-controller-sa -n kube-system

1. Verify the deployment of the EBS CSI Driver:

kubectl get pods -n kube-system -l app.kubernetes.io/name=aws-ebs-csi-driver

1. Verify the application deployment:

kubectl get pods