EKS Cluster Setup

1. Prerequisites for EKS Setup

1. **AWS Account**

    - **Status:** An active AWS account is required.
    - **IAM User:** Ensure you have admin permissions to create EKS clusters and EC2 instances.
    - Commands can be executed on an EC2 server as mentioned earlier.

2. **Required Permissions for IAM Role**
   Ensure the IAM role has the following permissions:

    - AmazonEKSClusterPolicy
    - AmazonEKSWorkerNodePolicy
    - AmazonEC2FullAccess

3. **AWS CLI (Command Line Interface)**

    - Install command:

      curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

      unzip awscliv2.zip

      sudo ./aws/install

    - Verify installation:

      aws –version

4. **kubectl (Kubernetes CLI)**

    - Install command:

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

      chmod +x kubectl

      sudo mv kubectl /usr/local/bin/

    - Verify installation:

      kubectl version –client

5. **eksctl (EKS CLI Tool)**

- Install command:

  curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz" | tar xz -C /tmp

  sudo mv /tmp/eksctl /usr/local/bin

- Verify installation:

  eksctl version

6. **AWS CLI Configuration**

   Configure AWS CLI to access your account:

   aws configure

   Enter Access Key ID, Secret Access Key, Region, and Output Format (json).

2. Create an EKS Cluster

1. **Basic Cluster Creation Command**

   eksctl create cluster --name EKS --region eu-north-1 --nodes 2 --node-type t3.medium --version 1.31

Explanation:

- --name: Cluster name (e.g., my-eks-cluster).
- --region: AWS region (e.g., us-east-1).
- --nodes: Number of worker nodes (e.g., 2).
- --node-type: Instance type for nodes (e.g., t3.medium).
- --version: Kubernetes version (e.g., 1.25).

2. **Monitor the Creation Process**
   Cluster creation will take 10–15 minutes. Logs will show details for:

   - VPC setup.
   - IAM roles creation.
   - Worker nodes provisioning.

3. **Verify Cluster is Ready**
   Check the cluster:

   eksctl get cluster

   Output: Cluster name, status (ACTIVE), version, region.

4. **Update kubectl Configuration**
   Connect kubectl to the EKS cluster:

aws eks --region us-east-1 update-kubeconfig --name my-eks-cluster

Verify connection:

kubectl get nodes

Output: List of worker nodes.

3. Deploy Application in EKS

1. **Create a Kubernetes Deployment Manifest**

Create a file named nginx-deployment.yaml and add your deployment configuration.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.21
        ports:
        - containerPort: 80
```

2. **Apply the Deployment to the Cluster**

Run the following command to create the deployment:

kubectl apply -f nginx-deployment.yaml

Check the deployment status:

kubectl get deployments

3. **Expose the Deployment Using a Service**

Create a file named nginx-service.yaml and add your service configuration.

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: LoadBalancer
```

Apply the service file:

```
kubectl apply -f nginx-service.yaml
```

Check the service status:

```
kubectl get svc
```

4. **Access the Application**
   Note down the EXTERNAL-IP of the nginx-service using:

   ```
   kubectl get svc nginx-service
   ```

   Paste the external IP into your browser. You should see the **Nginx Welcome Page!**

4. Destroy EKS Setup

1. **Delete Kubernetes Resources**
   Delete all applications, services, and HPA deployed in the cluster:

   ```
   kubectl delete all --all
   ```

   Verify:

   ```
   kubectl get all
   ```

   Output: Everything should be empty.

2. **Delete the EKS Cluster**
   Use a single command to delete the cluster:

   ```
   eksctl delete cluster --name my-eks-cluster
   ```

What Happens:

- Cluster gets deleted.
- Worker nodes and attached resources (VPC, IAM roles) are cleaned up.

Verify:

```
eksctl get cluster
```

Ensure your cluster name is not listed.

3. **Check and Delete Leftover AWS Resources (If Any)**
   Sometimes eksctl may fail to clean up resources. Check manually:

   - **EC2 Instances:** Verify no EKS worker nodes are running in the EC2 dashboard.
   - **VPC:** Check the VPC dashboard to confirm deletion of EKS-created VPC.
   - **IAM Roles:** Ensure roles created for the EKS cluster (eksctl-*) are deleted.
   - **Load Balancers:** Check the EC2 > Load Balancers section for deletion of EKS-created load balancers.