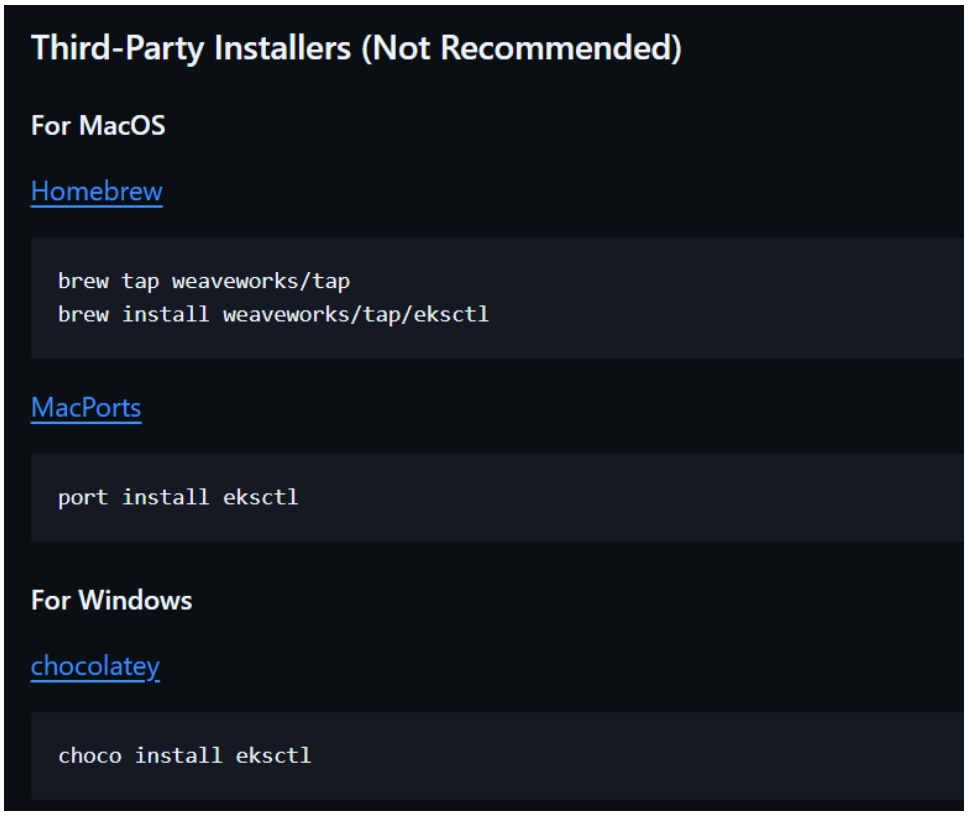
Demo Project: Create EKS cluster with eksctl

This guide demonstrates how to create an Amazon EKS cluster using the **eksctl** tool, which simplifies and automates much of the manual setup process.

Step 1: Install eksctl

1. **Download and Install eksctl**

   Follow the installation instructions from the eksctl GitHub repository (https://github.com/eksctl-io/eksctl)



2. **Verify Installation**

   Run the following command to confirm that **eksctl** is installed: `eksctl version`

Step 2: Connect eksctl with AWS account

- **Configure AWS CLI (if not already done)**

  Make sure your AWS CLI is set up by running: `aws configure list`

  This will display your AWS Access Key, Secret Access Key, region, and output format.

Step 3: Create EKS cluster

1. **Create the Cluster Using eksctl**

   Run the following command to create a new EKS cluster. This command creates the control plane, a default node group, and configures autoscaling

parameters:

```
eksctl create cluster \
--name demo-cluster \
--version 1.31 \
--region us-east-1 \
--nodegroup-name demo-nodes \
--node-type t2.micro \
--nodes 2 \
--nodes-min 1 \
--nodes-max 3
```

Note: The creation process may take around 30 minutes.

2. Alternative: Create a Cluster Using a Config File
   For more advanced configurations, you can create a `cluster.yaml` file (see eksctl documentation) eksctl.io and then run: `eksctl create cluster -f cluster.yaml`

   This method allows you to customize many aspects of your cluster in a single file.

Customize your cluster by using a config file. Just run

```
eksctl create cluster -f cluster.yaml
```

to apply a `cluster.yaml` file:

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: basic-cluster
  region: eu-north-1

nodeGroups:
  - name: ng-1
    instanceType: m5.large
    desiredCapacity: 10
  - name: ng-2
    instanceType: m5.xlarge
    desiredCapacity: 2
```

Step 4: Review the Created EKS Cluster

1. **Verify kubeconfig**

   After the cluster is created, your `~/.kube/config` file is updated automatically.

2. **List Nodes**

   Check the cluster nodes: `kubectl get nodes`

3. **Review in AWS Console**

   - **IAM Roles:** Check for new roles created (e.g., `eks-cluster-role` and `eks-node-group-role`).
   - **VPC & Subnets:** Confirm that a new VPC was created along with public and private subnets.

- **Cluster Overview:** Verify the public API endpoint of your EKS cluster.
- **EC2 Instances:** Under EC2, confirm the presence of the two worker nodes.

Note: At this point, there will be no application pods running yet (use `kubectl get pod` to confirm).

Step 5: Clean Up

When finished, you can clean up your resources:

1. **Delete the Deployment** (if needed): `kubectl delete deployment <deployment-name>`
2. **Optionally, Delete the EKS Cluster**: `eksctl delete cluster --name <your-cluster-name> --region <your-region>`
