How to Create EKS Cluster on AWS using Eksctl

To create EKS Cluster on AWS using Eksctl Command, so that you can have your Kubernetes environment on AWS CLI.

Prerequisites:

- AWS CLI Access with Admin Privileges.
- One Instance (AWS Cli must have installed)
- SSH Key (Optional).

Step1:

Install eksctl and kubectl utility:

Eksctl:

```
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -
s)_amd64.tar.gz" | tar xz -C /tmp
```

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

```
eksctl version
```

kubectl:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.17.7/2020-07-
08/bin/linux/amd64/kubectl
```

```
chmod +x ./kubectl
```

```
sudo mv ./kubectl /usr/local/bin
```

```
kubectl version --short –client
```

Step 2: Create EKS Cluster on AWS using the Eksctl utility:

```
eksctl create cluster
--name eks-cluster-demo
--version 1.15
--region us-west-1
--nodegroup-name eks-worker-nodes
--node-type t3.medium
--nodes 2
--nodes-min 2
--nodes-max 4
--ssh-access
--ssh-public-key selvakey.pub
--managed
```

Step 3: Deploying Demo Application.

```
git clone https://github.com/learnitguide/kubernetes-knote.git
```

How to Create EKS Cluster on AWS using Console:

Prerequisites:

1) AWS Account with Admin Privileges
2) AWS Cli Access to use the Kubectl utility
3) Instance (To manage cluster by using Kubectl)
4) Create an IAM role for EKS Cluster
   a) Select service role "EKS"➔EKS cluster(AmazonEKSclusterpolicy)
   b) Select EC2 ➔AmazonEKS_CNI_policy && AmazonEC2ContainerRegistryReadOnly && AmazonEKSWorkerNodepolicy
5) Create a Dedicated VPC for the EKS Cluster
Create Public and Private Subnet using CloudFormation: https://s3.ap-south-1.amazonaws.com/awsb30suresh.xyz2/amazon-eks-vpc-private-subnets.yaml

6) Install & Setup IAM Authenticator and Kubectl Utility.

Kubectl:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.17.7/2020-07-08/bin/linux/amd64/kubectl
```

```
chmod +x ./kubectl &&sudo mv ./kubectl /usr/local/bin
```

```
kubectl version --short –client
```

IAM Authenticator:

```
curl -Lo aws-iam-authenticator https://github.com/kubernetes-sigs/aws-iam-authenticator/releases/download/v0.5.9/aws-iam-authenticator_0.5.9_linux_amd64
```

```
chmod +x ./aws-iam-authenticator
```

```
mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH
```

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
```

```
aws-iam-authenticator help
```

Create EKS Cluster:

1) Give a name of cluster and select role for cluster.
2) Select stack created VPC and subnets (Public and private).
3) The Cluster need to have public and private access.
4) If need the remaining options we can able to provide the access-related options (which is optional)
5) Click on create. The cluster will be ready within 10 minutes.
6) Using the below command, you can able get the configuration file. (You have run the all command in one server which is deployed initially)

   ```
   Aws eks --region <cluster-region> update-kubeconfig --name <clustername>
   ```

Create worker node group:

1) Give the name of node groups and select a role.
2) Select stack vpc and subnets and default security group.
3) Select the key pair name for accessing the worker nodes.
4) Select the size of the worker node and keep desired and current size of worker nodes.
5) Once you have created the worker node group and check the worker in cli.

   ```
   kubectl get nodes
   ```

You have the active cluster to your application deployment in eks cluster.