Provisioning AWS EKS cluster using eksctl

Step-1) install latest aws cli

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo yum install unzip -y
unzip awscliv2.zip
sudo ./aws/install

step-2)

1. Create an IAM user/role with Route53, CFT, VPC, EC2, IAM and S3 full access

And we need `eksadministrator policy attached to thet role`

`Create custom policy with name eksadministrator(you can change the name it's you wish)`

`Attach the following in JSON`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "eksadministrator",
            "Effect": "Allow",
            "Action": "eks:*",
            "Resource": "*"
        }
    ]
}
```

`Now create policy and attach it to the role along with the above required policies`

2. Attach IAM role to the server

or

Configure your AWS CLI credentials

Step-3)
Install **eksctl**

To install or upgrade `eksctl` on Linux using `curl`

1)Download and extract the latest release of `eksctl` with the following command.

**curl --silent --location " https://github.com/weaveworks/eksctl/releases/download/0.19.0-rc.0/eksctl_$(uname -s)_amd64.tar.gz
" | tar xz -C /tmp**

2) Move the extracted binary to `/usr/local/bin`.

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

3) Test that your installation was successful with the following command.

eksctl version

step-4)
Install and configure kubectl for Amazon EKS(centos)

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
Or add the content directly in the mentioned location and
sudo chmod 0755 kubernetes.repo(gives execute permission to centos user as
well)
sudo yum install -y kubectl
```

Step-5)
Create your Amazon EKS cluster and worker nodes

```
eksctl create cluster \
--name prod \
--region region-code \
--nodegroup-name standard-workers \
--node-type t3.medium \
--nodes 3 \
--nodes-min 1 \
--nodes-max 4 \
--ssh-access \
--ssh-public-key my-public-key.pub \
--managed
```

If you want to ssh into worker nodes we can use the following

eksctl create cluster --name demo-ekscluster --region us-east-1 --nodegroup-name demo-workergroup --node-type t2.medium --nodes 2 --nodes-min 2 --nodes-max 2 --ssh-access --ssh-public-key aws-keypair.pem --managed

(or)

If you don't want to ssh into worker nodes we can use the following

eksctl create cluster --name demo-ekscluster --region us-east-1 --nodegroup-name demo-workergroup --node-type t2.micro --nodes 2 --nodes-min 1 --nodes-max 2 –managed

now, the cluster is created but it will take 10-15 min to create the cluster

To delete the cluster

```
eksctl delete cluster --name prod
```

Namespaces

Install kubens and kubectx for convienience

```
1)git clone https://github.com/ahmetb/kubectx.git ~/.kubectx
```

2)cd ~

3) cd .kubectx/

4)chmod +x kubens

5)chmod +x kubectx

6)cd ~

```
7) COMPDIR=$(pkg-config --variable=completionsdir bash-completion)
```

8)cd ~

```
9) cd .kubectx/completion/
```

```
10)chmod +x kubens.bash

11)chmod +x kubens.bash

12)sudo ln -sf ~/.kubectx/completion/kubens.bash $COMPDIR/kubens

13)sudo ln -sf ~/.kubectx/completion/kubectx.bash $COMPDIR/kubectx

14)cat << FOE >> ~/.bashrc

#kubectx and kubens

export PATH=~/.kubectx:\$PATH

FOE

15)source ~/.bashrc
```

Now try kubens and kubectx it should work.

```
kubectl get pods --namespace=<insert-namespace-name-here>
```

Setting the namespace preference

```
kubectl config set-context --current --namespace=<insert-namespace-name-here>
# Validate it
kubectl config view --minify | grep namespace:
```

To list the namespaces

Kubectl get ns

To create ns

Kubectl create ns testenv

To list pods in a particular ns

kubectl --namespace kube-system get pods

to list all the resoureces in a name space(note never ever mess with the kube-system namespace)

kubectl get all -n kube-system

Errors:

If you face any error says

kubectl error You must be logged in to the server (Unauthorized)

check the iam user accesskey, secret access key

else create new keys and do the aws configure in eks control plane node.

If you got any error regarding the permissions please check the role permissions attached to eks-control plane or please check with the user permissions (ie.., the level of permissions does the user got to access other resources in aws)