**Kubernetes Installation on AWS EC2 Instance**

Kops installation On Linux

**wget https://github.com/kubernetes/kops/releases/download/1.10.0/kops-linux-amd64**

**chmod +x kops-linux-amd64**

**mv kops-linux-amd64 /usr/local/bin/kops**

Kubectl Installation on Linux

curl -Lo kubectl https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

chmod +x ./kubectl

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

**Setup your environment**

Create IAM user (ex. kops) with given below permissions—

AmazonEC2FullAccess

AmazonRoute53FullAccess

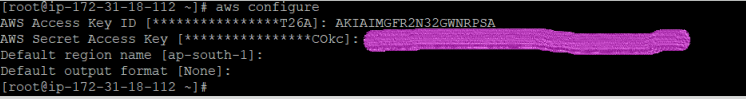
AmazonS3FullAccess

IAMFullAccess

AmazonVPCFullAccess

You should record the SecretAccessKey and AccessKeyID after creating the user.

Now , we need to configure aws client to use your IAM user-



export AWS\_ACCESS\_KEY\_ID=aws\_access\_key\_id

export AWS\_SECRET\_ACCESS\_KEY=aws\_secret\_access\_key

**Cluster State storage**

In order to store the state of your cluster, and the representation of your cluster, we need to create a dedicated S3 bucket for kops to use. This bucket will become the source of truth for our cluster configuration.

Create s3 bucket .

Note: We **STRONGLY** recommend versioning your S3 bucket in case you ever need to revert or recover a previous state store.

# Creating your first cluster

export NAME= myfirstcluster.k8s.local

export KOPS\_STATE\_STORE=s3://the\_name\_of\_your\_bucket

For a gossip-based cluster, make sure the name ends with k8s.local

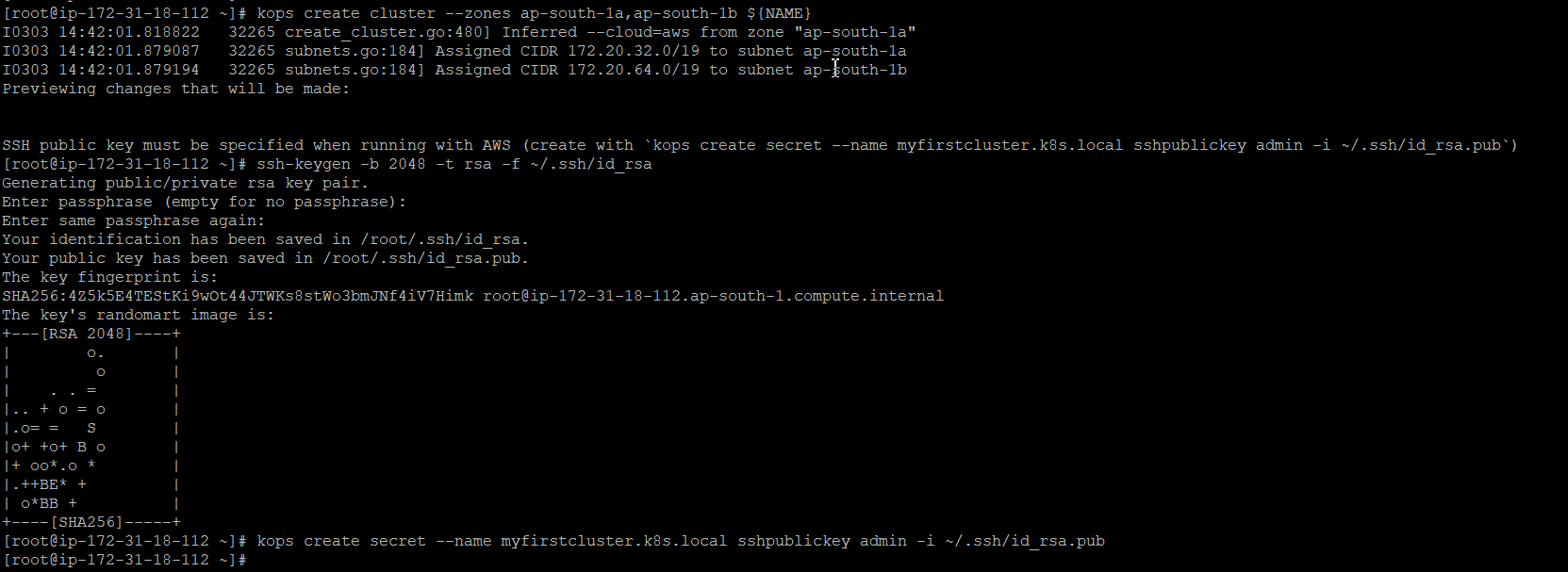
## Create cluster configuration

## We will need to note which availability zones are available to us. In this example we will be deploying our cluster to the ap-south-1 region

aws ec2 describe-availability-zones --region ap-south-1

Below is a create cluster command.

kops create cluster --zones ap-south1a,ap-south-1b ${NAME}



All instances created by kops will be built within ASG (Auto Scaling Groups), which means each instance will be automatically monitored and rebuilt by AWS if it suffers any failure.

## Customize Cluster Configuration

Now we have a cluster configuration, we can look at every aspect that defines our cluster by editing the description.

kops edit cluster ${NAME}

This opens your editor (as defined by $EDITOR) and allows you to edit the configuration. The configuration is loaded from the S3 bucket we created earlier, and automatically updated when we save and exit the editor.

We'll leave everything set to the defaults for now, but the rest of the kops documentation covers additional settings and configuration you can enable.

## Build the Cluster

Now we take the final step of actually building the cluster. This'll take a while. Once it finishes you'll have to wait longer while the booted instances finish downloading Kubernetes components and reach a "ready" state.

kops update cluster ${NAME} --yes

This will take few minutes to functioning the cluster. You can check the nodes and cluster by giving below commands—

Kubectl validate cluster

Kubectl get nodes