**Kubernetes installation in local VM**

# Install Python

apt-get update && apt-get install python-pip

# Install Kubectl

wget -O 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

# Install Kops

wget -O kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x ./kops

sudo mv ./kops /usr/local/bin/

# Install AWS CLI

pip install awscli

# Create IAM roles and assign to Kubernetes CLI EC2 Instance

# Create S3 Bucket name= madhavaa567

export KOPS\_STATE\_STORE=s3://madhava567

vi /root/.bashrc

export KOPS\_STATE\_STORE=s3://madhava567

# Route53 domain creation

Go to Route53 + Creeate Hosted Zone + Domain name = madhava567 + Type = private hosted zone for Amazon VPC + VPC ID = us east north Virginia

# Create SSH Keys

ssh-keygen

#Create Cluster

kops create cluster --cloud=aws --zones=us-east-1a --name=madhu.com --dns-zone=madhava567 --dns private

Suggestions:

\* list clusters with: kops get cluster

\* edit this cluster with: kops edit cluster madhu.com

\* edit your node instance group: kops edit ig --name=madhu.com nodes

\* edit your master instance group: kops edit ig --name=madhu.com master-us-east-1a

Finally configure your cluster with: kops update cluster madhu.com --yes

kops get cluster

kops delete cluster madhu.com --yes

### Manual update

* kops edit cluster $NAME
* set the KubernetesVersion to the target version (e.g. v1.3.5)
* kops update cluster $NAME to preview, then kops update cluster $NAME --yes
* kops rolling-update cluster $NAME to preview, then kops rolling-update cluster $NAME --yes

### Automated update

* kops upgrade cluster $NAME to preview, then kops upgrade cluster $NAME --yes

In future the upgrade step will likely perform the update immediately (and possibly even without a node restart), but currently you must:

* kops update cluster $NAME to preview, then kops update cluster $NAME --yes
* kops rolling-update cluster $NAME to preview, then kops rolling-update cluster $NAME --yes

# Export Config File #

sudo cp /etc/kubernetes/admin.conf $HOME/

sudo chown $(id -u):$(id -g) $HOME/admin.conf

export KUBECONFIG=$HOME/admin.conf

# Add Flannel Networking #

sudo curl -sSL "https://github.com/coreos/flannel/blob/master/Documentation/kube-flannel.yml?raw=true" | kubectl --namespace=kube-system create -f -

# Add for Dashboard BUT IT IS NOT WORKING #

sudo kubectl apply -f https://rawgit.com/kubernetes/dashboard/master/src/deploy/kubernetes-dashboard.yaml --namespace=kube-system

# Export Config File

sudo cp /tmp/admin.conf $HOME/

sudo chown $(id -u):$(id -g) $HOME/admin.conf

export KUBECONFIG=$HOME/admin.conf

**Kubernetes Cluster creation in AWS Methode-1**

# Requirements

1. EC2 instances Ubuntu OS
2. AWS CLI
3. KOPS # it is a tool (Kubernetes operation). Help of kops create cluster very easy in AWS
4. S3 Bucket
5. Route 53
6. IAM Roles :

Amazon Route 53 Domain Full Access

Administrator Access

# AWS CLI installation

cd /opt/aws # AWS

apt-get update && apt-get install python-pip

pip install awscli

aws --version

vim aws\_cred.txt

[default]

aws\_assess\_key\_id =

aws\_secret\_access\_key =

region =

:wq!

Export AWS\_CONFIG\_FILE = “/opt/aws\_cred.txt”

Aws ec2 describe-reson – output text

Aws ec2 run-instance – image-id=ami-a6a7e7f4 –instance-type=t1.micro –key-name=getcloudready –security-group=webservers –min-count=1 –max-count=1

Aws ec2 terminate-instance –instance-ids=i-8643989 # Delete or Terminate EC2 instances in AWS

**Kubernetes Cluster creation in AWS Methode-2**

# Install Python PIP

Curl -O <https://bootstrap.pypa.io/get-pip.py>

Python get -pip.py --user

Export PATH= ~/.bash\_profile

# Install AWS CLI

pip install awscli

# Create Access key & Secured Access key

Go to AWS + account + my security credentials + continue to security credentials + access keys (Access key & Secured Access key) + Create new access key + and copy or down load key and save

# Configure AWS key in EC2

aws configure

AWS ACCESS KEY:

AWS SECRET KEY:

DEFAULT RESION NAME [NONE]:

DEFAULT OUT PUT FORMAT [NONE]: [JSN OR JEST ENTRY]

Aws ec2 describe-instances #Show all configurations

**Kubectl installation**

sudo apt-get update && apt-get upgrade -y

sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

sudo cat <<EOF > /etc/apt/sources.list.d/kubernetes.list

deb http://apt.kubernetes.io/ kubernetes-xenial main

EOF

sudo apt-get update -y

sudo apt-get install -y kubectl

**KOPS installation**

wget https://github.com/kubernetes/kops/releases/download/1.6.1/kops-linux-amd64

chmod +x kops-linux-amd64

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

export KOPS\_STATE\_STORE=s3://madhava567

mahesh@idmetrics.com

# Cluster creation command

Kops create cluster \

-- cloude=aws \

--name = cluster.kubernetes-aws.io \

--zone=us-east-1d \

--master-size =t2.micro \

--node-size = t2.micro

--state=s3://bucket-name \

--dns-zone= DNS Name \

--dns=private

--num-master=1

--num-node=2

# Master m3.medium = less than 5 nodes

m3.large = 6-10 Nodes

m3.XLarge = 11-100 Nodes

