I**nstalling Kubernetes on Ubuntu:**

<https://medium.com/htc-research-engineering-blog/install-a-kubernetes-cluster-with-kubeadm-on-ubuntu-step-by-stepff-c118514bc5e0>

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

1. apt-get update

2. apt install docker.io

3. **Install https support:**

sudo apt-get update && sudo apt-get install -y apt-transport-https curl

4. **Get kubernetes repo key:**

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

5. **Add kubernetes repo to manifest**:

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

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

EOF

6. sudo apt-get update

7. **Install kubeadm** :

sudo apt-get install -y kubelet kubeadm kubectl

# 8. Create the cluster ,Initial the kubeadm:

# kubeadm init --apiserver-advertise-address=172.168.3.235 --pod-network-cidr=192.168.0.0/16

**Warning : 2CPU**

kubeadm init --apiserver-advertise-address=**172.168.3.235** --pod-network-cidr=192.168.0.0/16 –-ignore-preflight-errors=NumCPU

**Result:**

Your Kubernetes control-plane has initialized successfully!

**To start using your cluster, you need to run the following as a regular user:**

**mkdir -p $HOME/.kube**

**sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**

**sudo chown $(id -u):$(id -g) $HOME/.kube/config**

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

**kubeadm join 172.31.39.132:6443 --token frglis.4wccpa6o1sr7m0ua \**

**--discovery-token-ca-cert-hash sha256:ae2452810478511f756cf5324479645f9437c4ce3c6ee07a18c73738d84854ea**

root@ip-172-31-39-132:~#

root@ip-172-31-39-132:~# **kubectl get nodes**

NAME STATUS ROLES AGE VERSION

ip-172-31-39-132 **NotReady** master 12m v1.18.0

ip-172-31-93-8 **NotReady** <none> 5m4s v1.18.0

root@ip-172-31-39-132:~# **kubectl get pods --all-namespaces**

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system calico-kube-controllers-75d56dfc47-l5vsr 1/1 Running 0 2m40s

kube-system calico-node-6ckcr 1/1 Running 0 2m39s

kube-system calico-node-wp4dk 1/1 Running 0 2m40s

kube-system coredns-66bff467f8-5vjpm 1/1 Running 0 16m

kube-system coredns-66bff467f8-dc7z2 1/1 Running 0 16m

kube-system etcd-ip-172-31-39-132 1/1 Running 0 17m

kube-system kube-apiserver-ip-172-31-39-132 1/1 Running 0 17m

kube-system kube-controller-manager-ip-172-31-39-132 1/1 Running 0 17m

kube-system kube-proxy-bjrvv 1/1 Running 0 9m41s

kube-system kube-proxy-r44s2 1/1 Running 0 16m

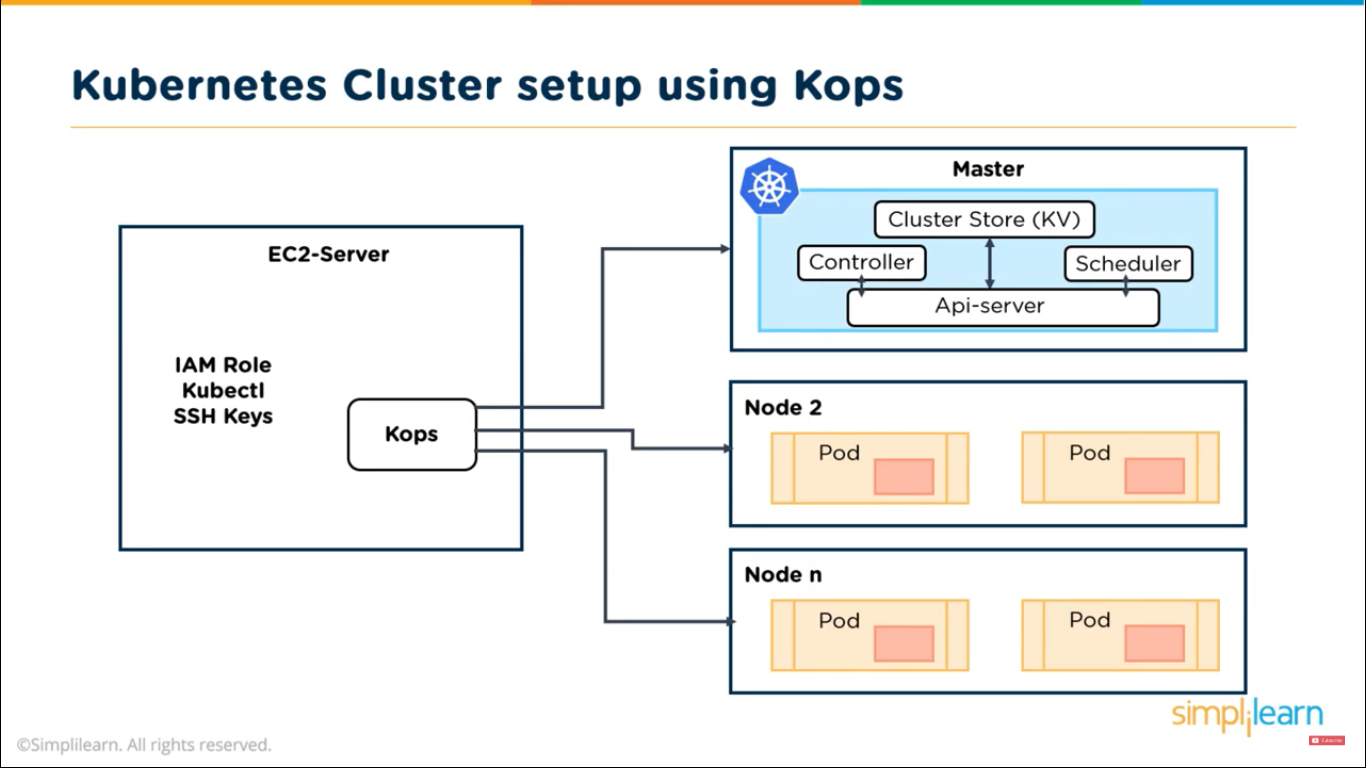
kube-system kube-scheduler-ip-172-31-39-132 1/1 Running 0 17m

root@ip-172-31-39-132:~#

**KOPS ::** Kubernetes operations

It’s a tool used to setup a cluster on production environment.

<https://github.com/simplilearn-github/kubernetes/blob/master/simplilearn-kops-setup.MD>



# Installing Kubernetes on AWS using Kops

### Launch Linux EC2 instance in AWS, we will use this EC2 instance to launch our Kubernetes cluster by installing Kops on it.

### Create and attach IAM role to EC2 Instance.

Kops would need permissions following permissions in order to operate:

S3

EC2

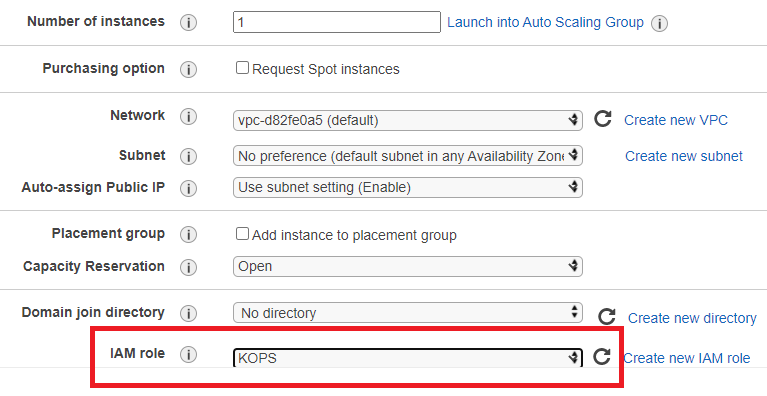
VPC

Route53

Autoscaling

etc..

Create a Role for the above specified permissions and then attach it to the running EC2 instance

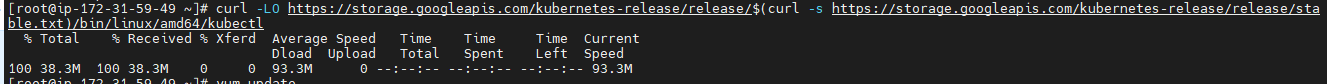


**3. Install Kops on EC2**

curl -LO 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-linux-amd64

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



### 4. Install kubectl

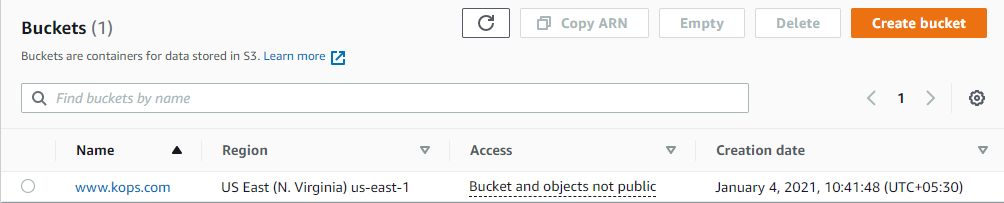
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s <https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl>

chmod +x ./kubectl

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

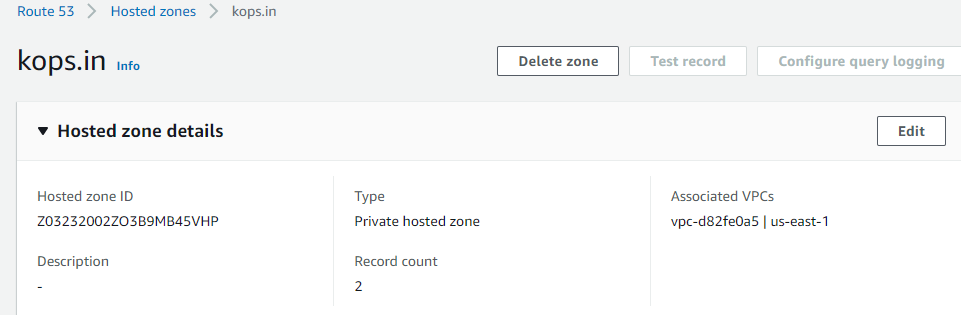
### 5.Create S3 bucket in AWS

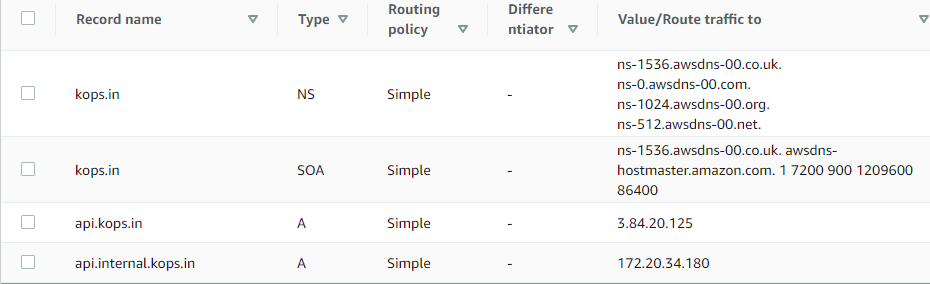
S3 bucket is used by kubernetes to persist cluster state and so, create any S3 bucket of your choice



### 6. Create private hosted zone in AWS Route53

1. Head over to aws Route53 and create a private hosted zone
2. Choose a name of your choice - kops.in
3. Choose type as private hosted zone for VPC
4. Select default vpc in the region you are setting up your cluster

****



**NS-**

### 7.Configure environment variables.

Open .bashrc file

vi ~/.bashrc

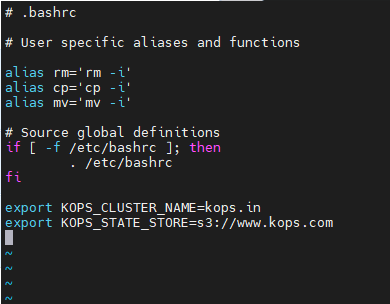
Add following content into .bashrc, you can choose any arbitary name for cluster and make sure buck name matches the one you created in previous step.

export KOPS\_CLUSTER\_NAME=kops.in

export KOPS\_STATE\_STORE=s3://[www.kops.com](https://s3.console.aws.amazon.com/s3/buckets/www.kops.com?region=us-east-1)

Then running command to reflect variables added to .bashrc

source ~/.bashrc



### 8.Create ssh key pair

This keypair is used for ssh into kubernetes cluster

ssh-keygen

### 9.Create a Kubernetes cluster definition.

kops create cluster \

--state=${KOPS\_STATE\_STORE} \

--node-count=2 \

--master-size=t2.micro \

--node-size=t2.micro \

--zones=us-east-1a,us-east-1b \

--name=${KOPS\_CLUSTER\_NAME} \

--dns private \

--master-count 1

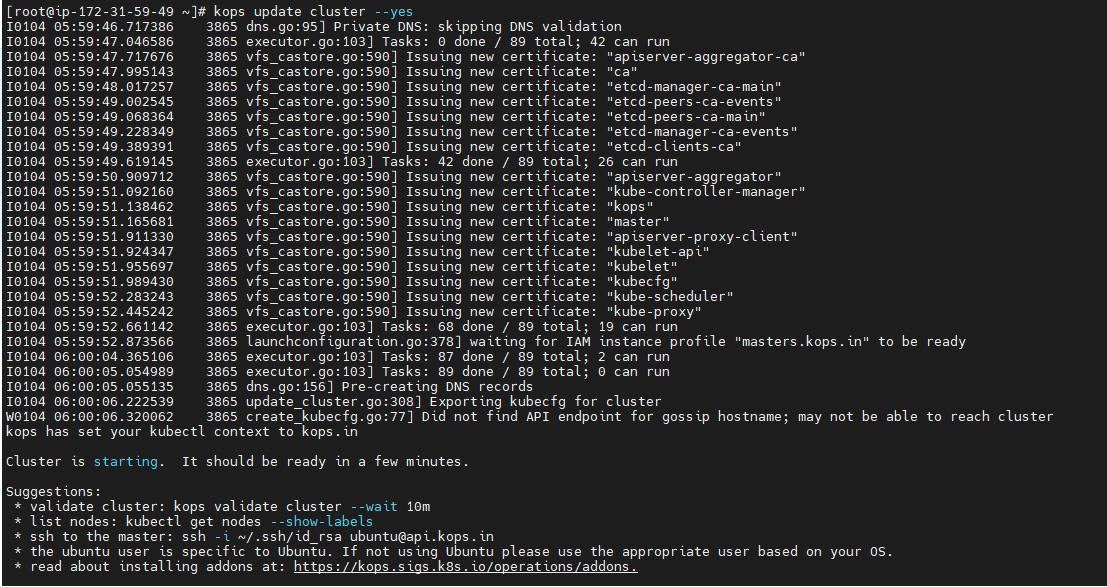
### 10.Create kubernetes cluster

kops update cluster –yes

Above command may take some time to create the required infrastructure resources on AWS. Execute the validate command to check its status and wait until the cluster becomes ready

kops validate cluster

For the above above command, you might see validation failed error initially when you create cluster and it is expected behaviour, you have to wait for some more time and check again.



Cluster is starting. It should be ready in a few minutes.

Suggestions:

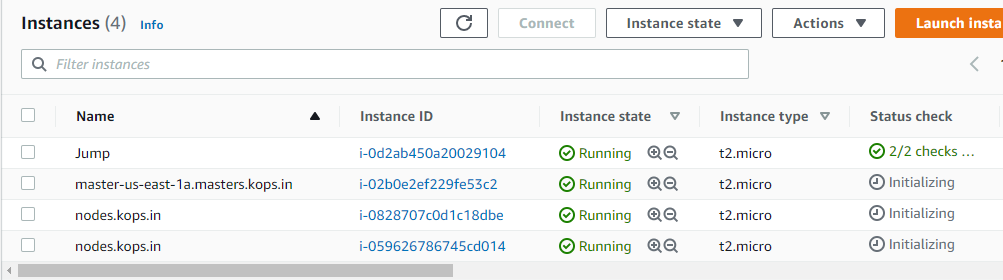
\* validate cluster: **kops validate cluster** --wait 10m

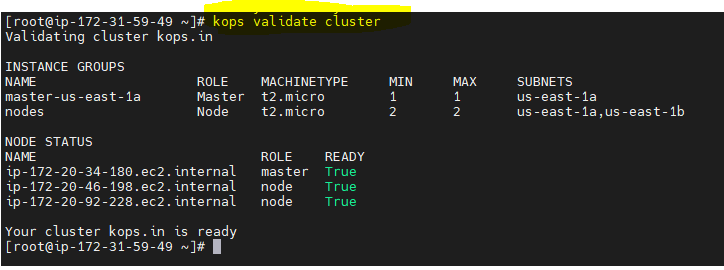
\* list nodes: **kubectl get nodes --show-labels**

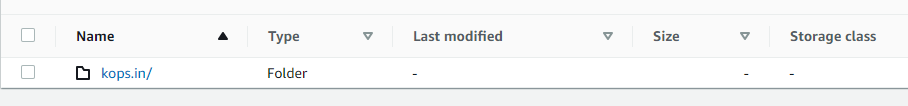
\* ssh to the master**: ssh -i ~/.ssh/id\_rsa ubuntu@api.kops.in**

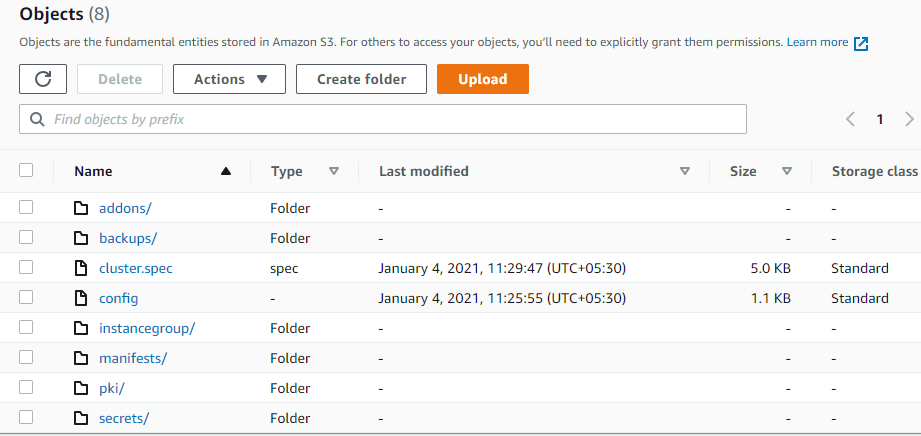
\* the ubuntu user is specific to Ubuntu. If not using Ubuntu please use the appropriate user based on your OS.

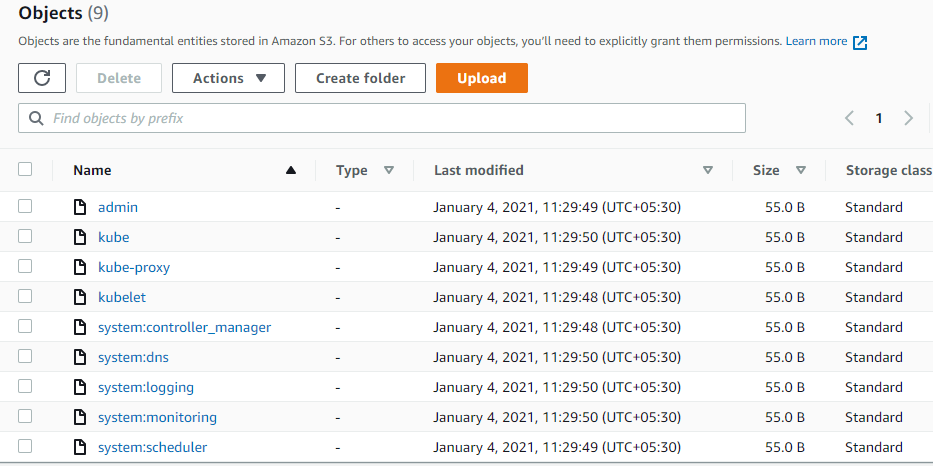
\* read about installing addons at: https://kops.sigs.k8s.io/operations/addons.





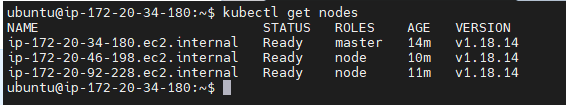


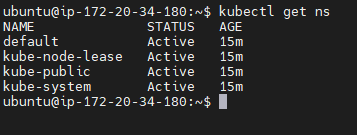


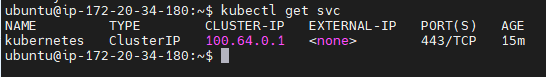


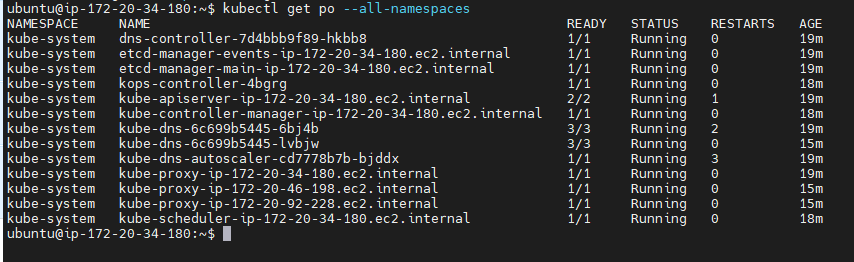
### 11) To connect to the master

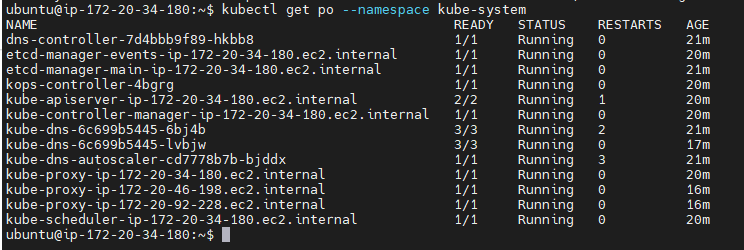
ssh -i ~/.ssh/id\_rsa [ubuntu@api.kops.in](mailto:ubuntu@api.kops.in)

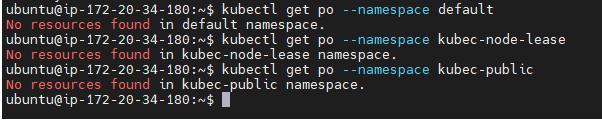












# 12) destroy the kubernetes cluster

# kops delete cluster --yes