

Kubernetes on AWS using Kops

1. Launch Linux EC2 instance in AWS (Kubernetes Client)

2. Create and attach IAM role to EC2 Instance.

Kops need permissions to access

- S3, EC2, VPC, Route53, Autoscaling etc..

3. Install AWS CLI

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

4. 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`

5. Install kubectl

- `curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"`
- `sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl`

(Note: If you do not have root access on the target system, you can still install kubectl to the `~/.local/bin` directory)

- `chmod +x kubectl`
- `mkdir -p ~/.local/bin/kubectl`
- `mv ./kubectl ~/.local/bin/kubectl`

(# and then add `~/.local/bin/kubectl` to `$PATH`)

6. Create S3 bucket in AWS

S3 bucket is used by Kubernetes to persist cluster state, let's create S3 bucket using AWS CLI

Note: Make sure you choose bucket name that is unique across all AWS accounts

Example: `aws s3 mb s3://javahome.in.k8s --region ap-south-1`

7. Create private hosted zone in AWS Route53

1. Head over to **AWS Route53** and **create hosted zone**
2. **Choose name for example (javahome.in)**
3. Choose type as private hosted zone for VPC
4. Select default VPC in the region you are setting up your cluster
5. Hit create

8. Configure environment variables.

Open .bashrc file

- `vi ~/.bashrc`

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

```
export KOPS_CLUSTER_NAME=javahome.inexport KOPS_STATE_STORE=s3://javahome.in.k8s
```

Then running command to reflect variables added to **.bashrc**

- `source ~/.bashrc`

9. Create ssh key pair

This keypair is used for SSH into kubernetes cluster

```
ssh-keygen
```

10. Create a Kubernetes cluster definition.

```
kops create cluster \--state=${KOPS_STATE_STORE} \--node-count=2 \--master-size=t3.medium \--node-size=t3.medium \--zones=ap-south-1a,ap-south-1b \--name=${KOPS_CLUSTER_NAME} \--dns private \--master-count 1
```

11. Create kubernetes cluster

```
kops update cluster --yes --admin
```

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 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.

12. To connect to the master

ssh admin@api.javahome.in

To delete the kubernetes cluster ----> kops delete cluster --yes

Update Nodes and Master in the cluster

We can change number of nodes and number of masters using following commands

- kops edit ig nodes change minSize and maxSize to 0
- kops get ig- to get master node name
- kops edit ig - change min and max size to 0
- kops update cluster --yes

Optional (Create terraform scripts through kops)

<https://github.com/kubernetes/kops/blob/master/docs/terraform.md>