**Setup Kubernetes (K8s) Cluster on AWS**

1. Create Ubuntu EC2 instance
2. install AWSCLI
3. curl https://s3.amazonaws.com/aws-cli/awscli-bundle.zip -o awscli-bundle.zip
4. apt install unzip python
5. unzip awscli-bundle.zip
6. #sudo apt-get install unzip - if you dont have unzip in your system

./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws

1. Install kubectl
2. curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
3. chmod +x ./kubectl

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

1. Create an IAM user/role with Route53, EC2, IAM and S3 full access
2. Attach IAM role to ubuntu server

**Note: If you create IAM user with programmatic access then provide Access keys.**

aws configure

1. Install kops on ubuntu instance:
2. 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
3. chmod +x kops-linux-amd64

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

1. Create a Route53 private hosted zone (you can create Public hosted zone if you have a domain)
2. create an S3 bucket

aws s3 mb s3://dev.k8s.valaxy.in

1. Expose environment variable:

export KOPS\_STATE\_STORE=s3://dev.k8s.valaxy.in

1. Create sshkeys before creating cluster

ssh-keygen

1. Create kubernetes cluster definitions on S3 bucket

kops create cluster --cloud=aws --zones=ap-southeast-1b --name=dev.k8s.valaxy.in --dns-zone=valaxy.in --dns private

1. Create kubernetes cluser

kops update cluster dev.k8s.valaxy.in --yes

1. Validate your cluster

kops validate cluster

1. To list nodes

kubectl get nodes

**Deploying Nginx container on Kubernetes**

1. Deploying Nginx Container
2. kubectl run sample-nginx --image=nginx --replicas=2 --port=80
3. kubectl get pods

kubectl get deployments

1. Expose the deployment as service. This will create an ELB in front of those 2 containers and allow us to publicly access them:
2. kubectl expose deployment sample-nginx --port=80 --type=LoadBalancer

kubectl get services -o wide

1. To delete cluster

kops delete cluster dev.k8s.valaxy.in --yes