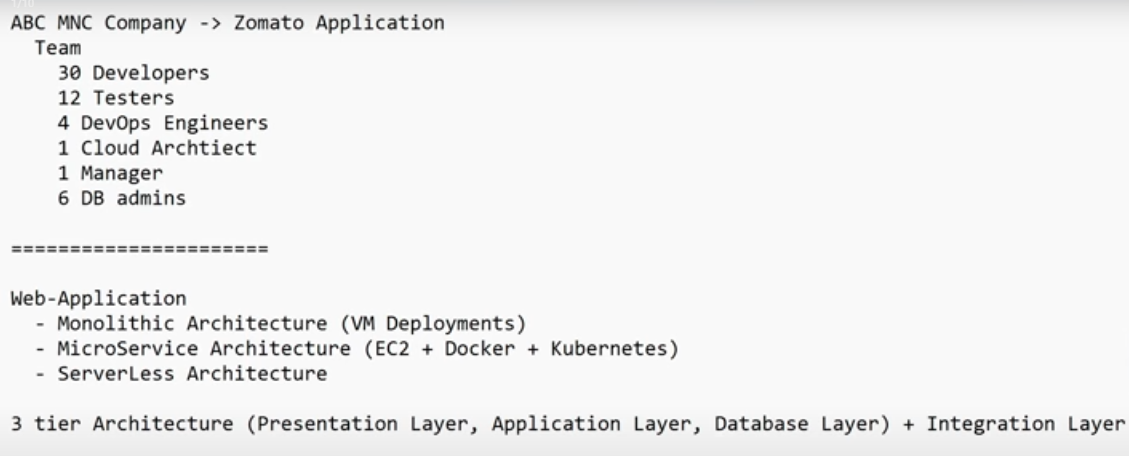
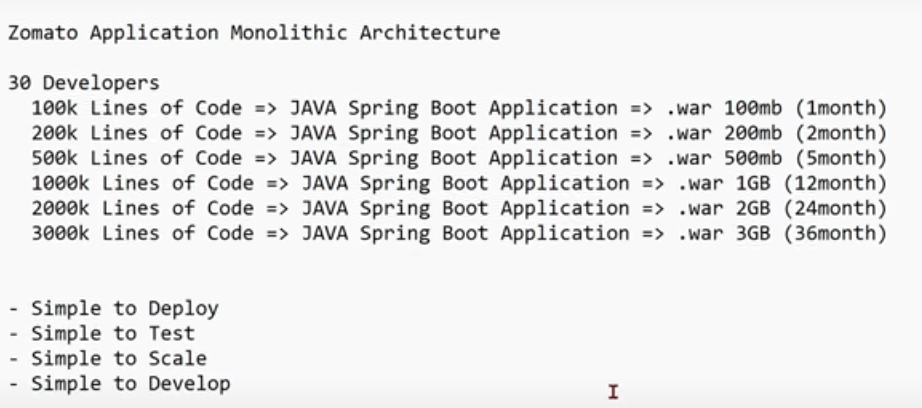
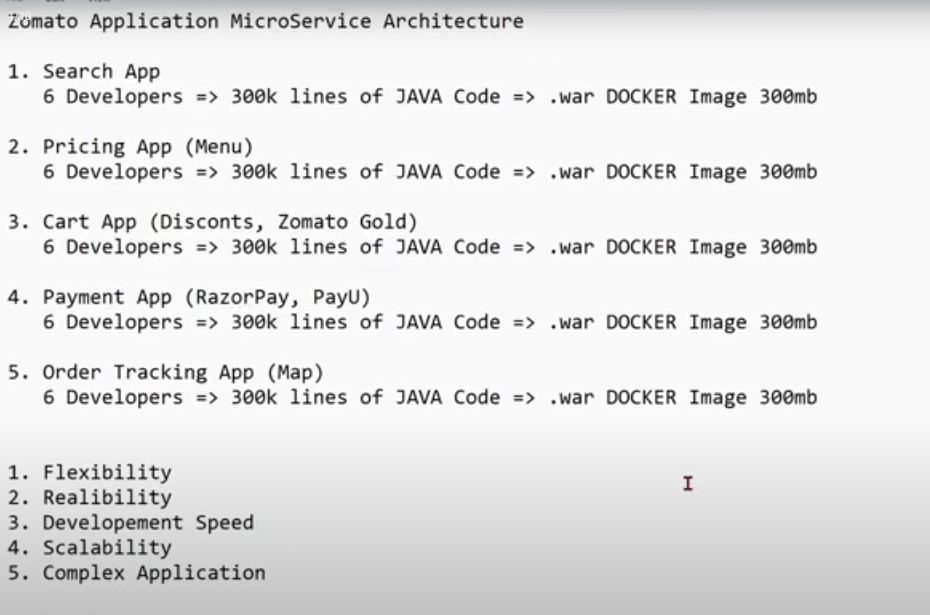
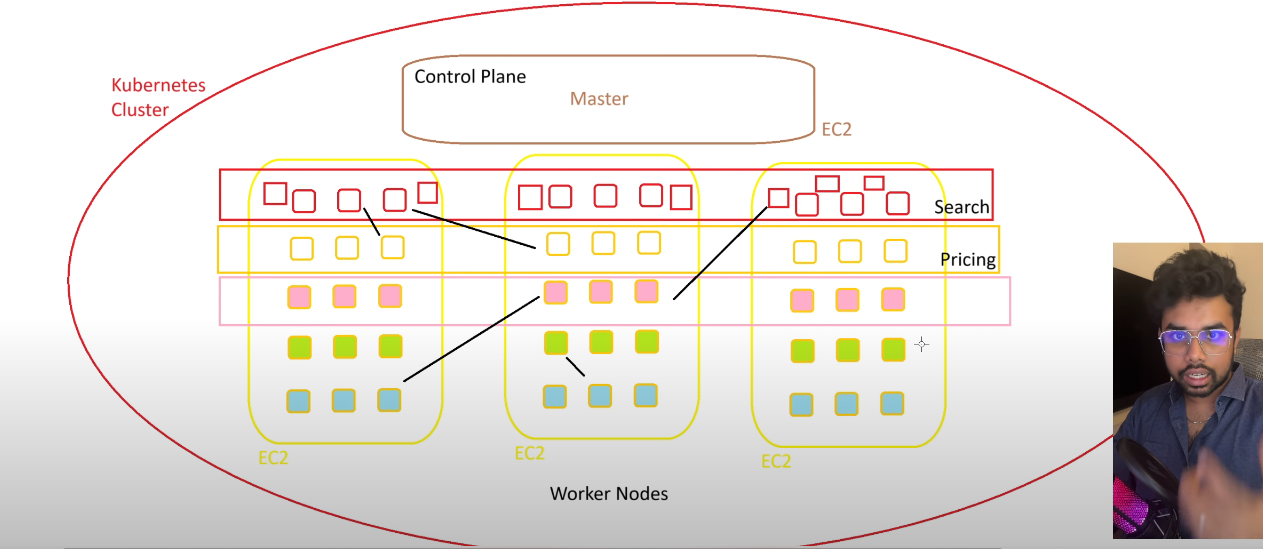
# PART-1 --> Kubernetes from scarch - Introduction and Installation

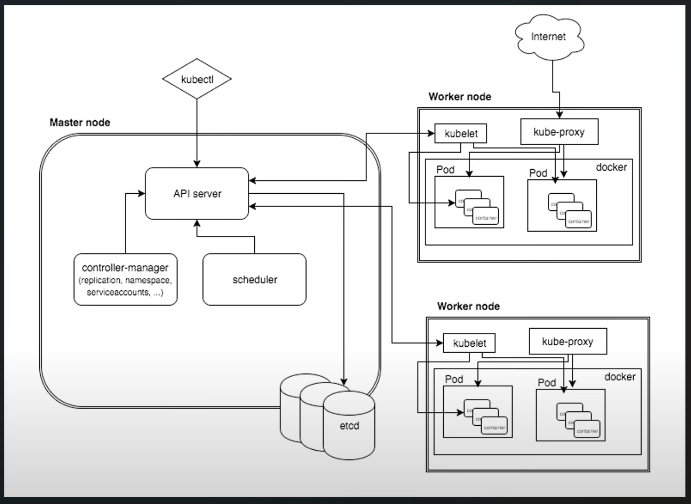


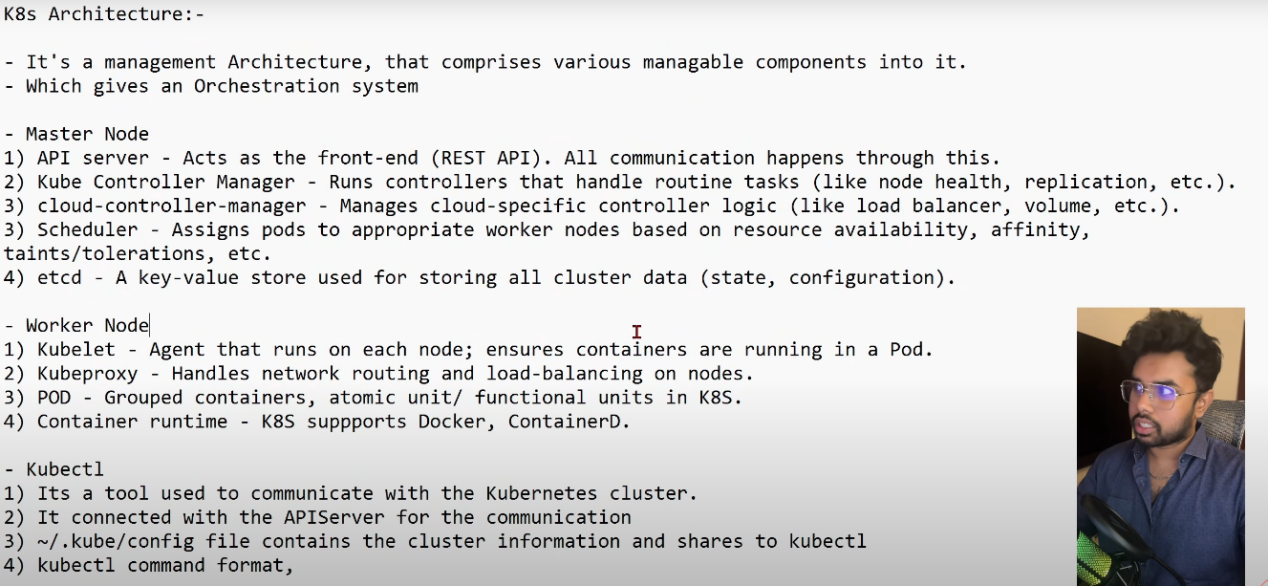


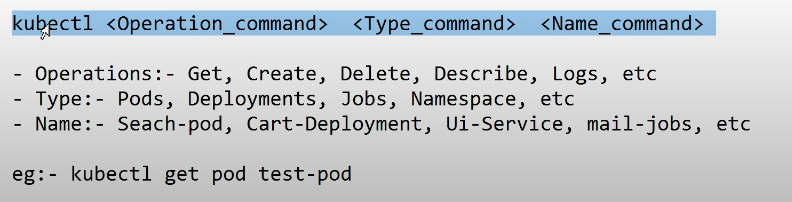


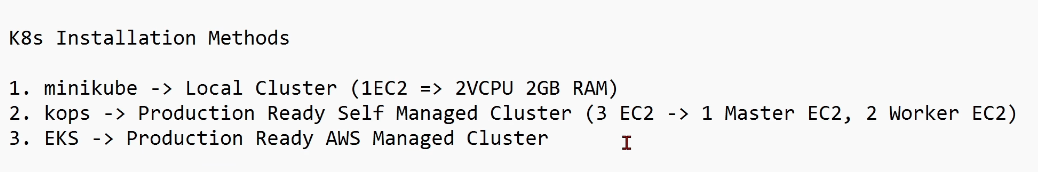


* control plane or master components both are same in kubernetes









* kubernetes install methods --> in google --> kubernetes documentation

## In aws console

* Create IAM user and access key for CLI access
* Attach policies - AdministratorAccess
* create bucket in s3

## Launch EC2 instance with ALL TCP

* aws configure
* access key
* secret access key
* region --> ap-southeast-1
* format --> json
* aws iam list-users --> testing and listing of iam users
* printenv --> to see environment variables
* export name=mitran
* echo “Hello my name is $name”
* unset name --> environment variable deleted
* export AWS\_ACCESS\_KEY\_ID=access key
* export AWS\_SECRET\_ACCESS\_KEY=secret access key
* printenv

Deploying to AWS --> **Make sure you have [installed kOps](https://kops.sigs.k8s.io/install/) and [installed kubectl](https://kops.sigs.k8s.io/install/).**

### Installed kOps

curl -Lo 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/

* kops

**Installed kubectl**

curl -Lo kubectl https://dl.k8s.io/release/$(curl -s -L https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl chmod +x ./kubectl sudo mv ./kubectl /usr/local/bin/kubectl

* kubectl
* sudo rm -rf /usr/local/bin/kubectl --> to remove

### Prepare local environment :- Gossip-based cluster

* export NAME=myfirstcluster.k8s.local
* export KOPS\_STATE\_STORE=s3://bucketname

### Create public and private key

* ssh-keygen

### Creating cluster

kops create cluster \

--name=${NAME} \

--cloud=aws \

--zones=ap-southeast-1a

kops edit ig --name=mithran.k8s.local nodes-ap-southeast-1a

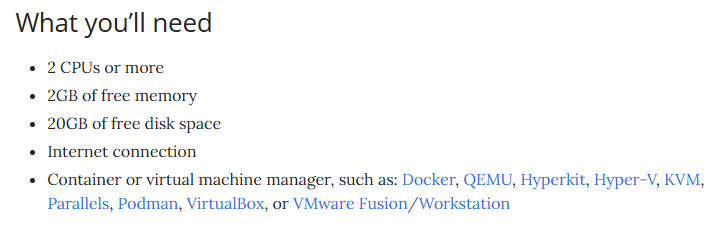
kops update cluster --name mithran.k8s.local --yes --admin

kops validate cluster

kubectl get pods

kops delete cluster --name ${NAME} --yes

### Minikube



* sudo yum update -y
* sudo yum -y install docker
* sudo service docker start
* sudo usermod -a -G docker ec2-user
* sudo chmod 666 /var/run/docker.sock
* docker version

curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

* minikube kubectl -- get pods
* alias kubectl=”minikube kubectl --”
* kubectl get pods
* kubectl get services

# PART-2 --> Kubernetes Namespaces, Pods, ReplicaSets & DaemonSets

* kops validate cluster
* permenently set env variables in linux --> own user
  + vim ~/.bashrc
  + export MYVAR="somevalue"
  + :wq
  + source ~/.bashrc
* permenently set env variables in linux --> all user
  + sudo vim /etc/profile.d/java.sh
  + export JAVA\_HOME=/usr/lib/jvm/java-17-openjdk
  + export PATH=$PATH:$JAVA\_HOME/bin
  + sudo chmod +x /etc/profile.d/java.sh
  + source /etc/profile

## Namespace

* kubectl get namespace
* kubectl create namespace
* kubernetes api docs in goggle for reference

### app.yaml

apiVersion: v1

kind: Namespace

metadata:

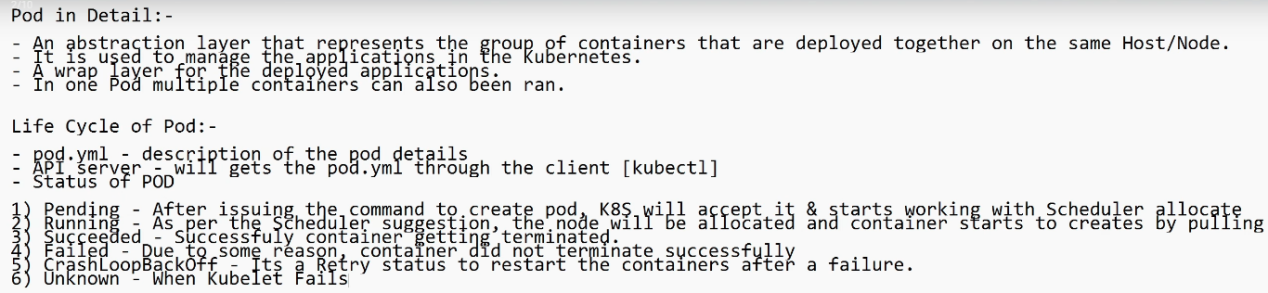
name: qa

* kubectl appy -f app.yaml
* kubectl get namespace

### Imperative way - kubectl cli commands

### Declarative way - yaml

## Pod



* kubectl get pods
* kubectl get pods -n dev
* kubectl get pods --all-namespaces

### pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: myfirstpod

spec:

containers:

- name: cont1

image: httpd

ports:

- containerPort: 80

* kubectl appy -f pod.yaml
* kubectl get pod
* kubectl get pod -o wide --> wide means extra information
* kubectl describe