Kubernetes
==========

Menions: This is an individual node used in kubernetes
Combination of these minions is called as Kubernetes cluster

Master is the main machine which triggers the container orchestraion
It distributes the work load to the Slaves

Slaves are the nodes that accept the work load from the master
and handle activites load balancing,autoscalling,high availability etc

Kubernetes uses various of types of Object

1 Pod: This is a layer of abstraction on top of a container.This is the samallest object that kubernetes can work on.In the Pod we have a container.
The advantage of using a Pod is that kubectl commands will work on the Pod and the
Pod communicates these instructions to the container.In this way we can use the same kubectl irresepective of which technology containers are in the Pod.

2 Service: This is used for port mapping and network load balancing

3 NameSpace: This is used for creating partitions in the cluster.Pods running in a namespace cannot communicate with other pods running in other namespace

4 Secrets: This is used for passing encrypted data to the Pods

5 ReplicationController: This is used for managing multiple replicas of PODs

and also perfroming saclling

6 ReplicaSet: This is similar to replicationcontroller but it is more advanced where features like selector can be implemented

7 Deployment: This used for perfroming all activites that a Replicaset can do
it can also handle rolling update

8 PersistantVolume: Used to specify the section of storage that should be used for volumes

9 PersistantVolumeClaims: Used to reserver a certain amout of storage for a pod from the
persistant volume.

10 Statefulsets: These are used to handle stateful application like data bases
where consistency in read write operations has to be maintained.

11 HorrizontalPodAutScaller: Used for auto scalling of pods depending on the load

Kubernetes Architecture
==========
Master Componentes
==========
Container runtime: This can be docker or anyother container technology

apiServer: Users interact with the apiServer using some clinet like ui,command line tool like kubelet.It is the apiServer which is the gateway to the cluster
It works as a gatekeeper for authentication and it validates if a specific user is having permissions to execute a specific command.Example if we want to deploy a pod or a deployment first apiServers validates if the user is authorised to perform that action and if so it passes to the next process

ie the "Scheduler"

Scheduler: This process accepts the instructions from apiServer after validation and starts an application on a sepcific node or set of nodes.It estimates how much amount of h/w is required for an application and then checks which slave have the necessary h/w resources and instructs the kubelet to deploy the application

kubelet: This is the actual process that takes the orders from scheduler and deploy an application on a slave.This kubelet is present on both master and slave

controller manager: This check if the desired state of the cluster is always maintained.If a pod dies it recreates that pod to maintain the desired state

etcd: Here the cluster state is maintained in key value pairs.
It maintains info about the slaves and the h/w resources available on the slaves and also the pods running on the slaves
The scheduler and the control manager read the info from this etcd and schedule the pods and maintain the desired state

==========
===
Worker components
==========
containerrun time: Docker or some other container technology

kubelet: This process interacts with container run time and the node and it start a pod with a container in it

kubeproxy: This will take the request from
services to pod
It has the intellegence to forward a request to
a near by pod.Eg If an application pod wants to communicate with a db pod then kubeproxy will take that request to the nearby pod

==========
==
Kubernetes setup can be done in2 ways
1 Managed Kubernetes setup
  On AWS (EKS : Elastic Kubernetes service)
  On GCP (GKE : Google Kubernetes Engine)
  On Azure (AKS :Azure Kubernetes Service)
2 Unmanaged Kubernetes setup
  1 KOPS
  2 Kubeadm

==========
KOPS installations
==========
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 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, lets create s3 bucket using aws cli Note: Make sure you choose bucket name that is uniqe accross all aws accounts

aws s3 mb s3://project.in.k8s --region us-west-2
6. Create private hosted zone in AWS Route53
Head over to aws Route53 and create hostedzone
Choose name for example (sai.in)
Choose type as privated hosted zone for VPC
Select default vpc in the region you are setting up your cluster
Hit create
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=intelliqq.in
export KOPS_STATE_STORE=s3://intelliqq.in.k8s
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=${KOPS_STATE_STORE} \
--node-count=2 \
--master-size=t3.medium \
--node-size=t3.medium \
--zones=us-west-2a \
--name=${KOPS_CLUSTER_NAME} \
--dns private \
--master-count 1
10. 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 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.

11. To connect to the master
ssh admin@api.javahome.in
Destroy the kubernetes cluster
kops delete cluster --yes
Update Nodes and Master in the cluster
We can change numner 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

==========
=