Kubernetes works based on combination of Master Node and Worker Node

Components of Master Node:

1. API Server
2. Scheduler
3. Control Manager
4. Etcd

Components of Slave Node:

1. Kubelet
2. Kube-proxy
3. Container Runtime

Main Components Of K8S:

1) Pod: A pod is the smallest deployable unit that can run and hold containers. It represents a single instance of a running process in a cluster and encapsulates one or more containers. Containers within the same pod share the same network namespace, allowing them to communicate with each other using localhost. Pod works under Worker Node

2) Service: A Service is an abstraction that defines a logical set of pods and a policy by which to access them. In simpler terms, a Service enables network communication to a group of pods, providing a stable endpoint that other applications can use to access the functionality offered by those pods.

Creation of Kubernetes Engine :

* Kubernetes Controller – kubectl
* Create Cluster
* EC2 Small
* Boot Size 10
* Create and Connect

Creation of Namespace:

* Kubectl get ns
* Kubectl create ns cap-dev

Creation of Pod:

* Kubectl run podname –-image imagename –-port=portnumber
* Kubectl get pods
* Kubectl get nodes
* kubectl get pod -o wide
* kubectl describe pod podnmae

Creation of Deployment:

In Kubernetes, a deployment is a resource object that provides declarative updates to applications It allows you to describe an application’s life cycle, such as which images to use for the app, the number of pods there should be, and the way in which they should be updated. A deployment provides declarative updates for pods and replica sets.

* kubectl create deployment deploymentname –image=imagename
* kubectl get deployments

Creation of Replica Set:

* kubectl scale deployment deployment name –replicas=no of replicas

Creation of Service:

In Kubernetes, a Service is an abstraction that defines a logical set of Pods and a policy by which to access them. It enables other applications in the cluster to communicate with your application or microservice.

Components of Service:

1. Cluster IP
2. Node Port
3. Load Balancer

* Kubectl expose deployment deploymentname –-type=typeofservice –port=portnumber
* Kubectl get svc
* Kubectl get svc -w

Creation of YAML File:

Creating a YAML file for Kubernetes involves specifying the configuration details for a particular Kubernetes resource, such as a Deployment, Service, Pod, ConfigMap, etc. Here's a basic example of how you can create a YAML file for a simple Nginx Deployment:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80