Architute

Master --🡪 Assign works to the clusters

API service,etcd, scheduler , controller manager.

Etcd – storage, tokens, Key values -DB

Controller manager – node controller, endpoint controller, name space controller

Weather the state is maintained or not any one node killed again it will maintain the replica state.

scheduler – it will watch the worker nodes, new arrived pods

API server – UI

Node(worker) ----- node is none as minion, Which maintain the pods

Kubeadm --- **kubeadm** is a new tool that is part of the Kubernetes distribution as of 1.4.0 which helps you to install and set up a Kubernetes cluster

Kubectl -- The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs

Kubelet – to interface b/n master and worker nodes, it gives to the master weather the cluster state is running or not

The kubelet is the primary “node agent” that runs on each node. The kubelet works in terms of a PodSpec. A PodSpec is a YAML or JSON object that describes a pod

Docker –

It is using for pulling the imgaes

Pod—consists of one or more container

Kubeproxy– This reflects services as defined in the Kubernetes API on each node and can do simple TCP, UDP, and SCTP stream forwarding or round robin TCP, UDP, and SCTP forwarding across a set of backends. Service cluster IPs and ports are currently found through Docker-links-compatible environment variables specifying ports opened by the service proxy. There is an optional addon that provides cluster DNS for these cluster IPs. The user must create a service with the apiserver API to configure the proxy.

CAdvisor--- cAdvisor is integrated into kubelet binary this will discovers all containers in the machines and collect the cpu and memory and file systemsz and network usage and statistics, it will provide the overall machine usage by analyzing the root container on the machine.

Installation:

<https://www.youtube.com/watch?v=ndNKI5GBdd0>

<https://github.com/narendrasingamaneni91/DevOpsDemos/blob/master/Kubernetes/k8s-setup.md>

Deployment:

Create the node.yml (nginx.yml) file

|  |
| --- |
| --- |
|  | apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2 |
|  | kind: Deployment |
|  | metadata: |
|  | name: nginx |
|  | spec: |
|  | strategy: |
|  | type: Recreate |
|  | selector: |
|  | matchLabels: |
|  | app: nginx |
|  | replicas: 3# tells deployment to run 1 pods matching the template |
|  | template: # create pods using pod definition in this template |
|  | metadata: |
|  | labels: |
|  | app: nginx |
|  | spec: |
|  | containers: |
|  | - name: nginx |
|  | image: nginx |
|  | ports: |
|  | - containerPort: 80 |

Create service.yml

|  |
| --- |
| --- |
|  | apiVersion: v1 |
|  | kind: Service |
|  | metadata: |
|  | name: nginx |
|  | namespace: default |
|  | labels: |
|  | app: nginx |
|  | annotations: |
|  | service.beta.kubernetes.io/aws-load-balancer-type: "nlb" |
|  | spec: |
|  | externalTrafficPolicy: Local |
|  | ports: |
|  | - name: http |
|  | port: 80 |
|  | protocol: TCP |
|  | targetPort: 80 |
|  | selector: |
|  | app: nginx |
|  | type: LoadBalancer  to activate the service yml file  **kubectl apply -f ./service.yml** |

**kubectl apply -f node.yml**

ingress:

n Kubernetes, an Ingress is an object that allows access to your Kubernetes services from outside the Kubernetes cluster. You configure access by creating a collection of rules that define which inbound connections reach which services. This lets you consolidate your routing rules into a single resource.

Replication:

The Replication Controller is the original form of replication in Kubernetes. ... If a pod does crash, the Replication Controller replaces it. Replication Controllers also provide other benefits, such as the ability to scale the number of pods, and to update or delete multiple pods with a single command

Replica Set:

Replica Set ensures how many replica of pod should be running. It can be considered as a replacement of replication controller. The key difference between the replica set and the replication controller is, the replication controller only supports equality-based selector whereas the replica set supports set-based selector

Namespace:

Namespaces cannot be nested inside one another and each Kubernetes resource can only be in one namespace. Namespaces are a way to divide cluster resources between multiple users (via resource quota). In future versions of Kubernetes, objects in the same namespace will have the same access control policies by default.

Creating the node with abc.Yml

apiVersion: v1

kind: Pod

metadata:

name: Tomcat

spec:

containers:

- name: Tomcat

image: tomcat: 8.0

ports:

containerPort: 7500

imagePullPolicy: Always

Heapsetter: Monitoring Tool for Kubernetes ikt gives the information about the CPU, Memory,file systems, workload status ,network usage monitoring