Kubernetes

Role in

Devops

Table of Contents

- ✓ Introduce Docker and Docker Swarm
- ✓ Introduce Kubernetes as a Orchestration Concept and Architecture
- ✓ Core Concept Kubernetes Objects
- ✓ Swarm vs Kubernetes
- ✓ Kubernetes Installation Methods
- ✓ kubectl Commands
- ✓ Introduce Manifest and YAML and Write it
- ✓ Kubernetes Volume Storge
  - ✓ StorageClass (CephFS , Glusterfs , NFS , iSCSI , Cinder , Local , …)
  - ✓ Persistent Volumes
  - ✓ Persistent Volumes Claim
- ✓ Kubernetes Network About Container Network Interface (CNI)
  - ✓ (WeaveNet , Flannel , Contiv , Cilium , Calico , …)
- ✓ Kubernetes Scaling
  - ✓ Horizontal Infrastructure Scaling
  - ✓ Vertical Infrastructure Scaling
  - ✓ Auto-scaling
- ✓ Implementing Deployment With Kubernetes
  - ✓ Deployment Nginx With Scale On kubernetes
- ✓ Introduce Kubernetes Exam Certified Kubernetes Administrator (CKA) Details

Docker and Docker Swarm is

Docker is :

A Tool designed to make ,

Easier to create
Deploy
And Run

Applications by using **Containers**

Docker Swarm is :

An open-source container Orchestration platform and
The Native Clustering Engine by **Docker**

Containers is :

A Package from applications ,
Can be run, with its dependencies, isolated from other processes .

Banner Docker :

- **Build Any App**
- **Ship Any Where**
- **Run Any Where**

Docker Swarm Overview

Docker Swarm Components and Architecture

Kubernetes is …

Kubernetes is :

Desigen by Google be name K8s.

Automated Container Deployment , Scaling and Management Platform

Kubernetes is :

An open source system for managing containerized application in a clustered environment.

Kubernetes help to :

- ✓ DevOps as a Service
- ✓ Automatically scale up-down the application and update
- ✓ With the zero downtime

kubernetes

Kubernetes Version

Original author(s): Google

Developer(s) : Cloud Native Computing Foundation

Initial release : 7 June 2014; 5 years ago

Stable release : 1.16 / September 18, 2019; 2 months ago

kubernetes

Repository : github.com/kubernetes/kubernetes

Written in : Go

Type : Cluster Management Software

License : Apache License 2.0

Website : kubernetes.io

Kubernetes History

kubernetes

Kubernetes Version

Features and Preferences Of Kubernetes :

- ✓ **Service Discovery and Networks , Load Balancing**
- ✓ **Storage Orchestration**
  - NFS ،iSCSI ،Gluster ،Ceph ،Cinde
  - AWS Cloud
- ✓ **Automated Rollouts and Rollbacks , Rolling Update**
- ✓ **Scaling And Monitoring**
  - Horizontal
  - Vertical
  - Auto Scale
  - Manual Scale

Kubernetes Version

Glossary Of Kubernetes :

- ✓ **Cluster**
  - ✓ Kubernetes Engine With Serveral Server (Physical Or Virtual)For Support All Components
- ✓ **Node**
- ✓ **Pod**
- ✓ **Replication Controllers**
- ✓ **Replica Sets**

  } Run a specified number of Pod
- ✓ **Selector**
- ✓ **Label**

  } Filter for Label of Pod
- ✓ **Name**
- ✓ **Volume**
- ✓ **Namespace** ➔ Domain Name
- ✓ **Service** ➔ include (IP , DNS Name)
- ✓ **Kind**

Kubernetes Overview

Kubernetes Components and Architecture

Kubernetes Components and Architecture

Kubernetes Components :

- ✓ **Kubernetes - Master Machine Components**
  - Etcd
  - API Server
  - Controller Manager
  - Scheduler
- ✓ **Kubernetes - Worker Node Components**
  - Docker
  - Kubelet Service
  - Kubernetes Proxy Service
- ✓ **Kubernetes Objects and Workloads**
  - Pods
  - Replication Controllers
  - Replication Sets
  - Deployments
  - Stateful Sets
  - Daemon Sets
  - Services

Kubernetes Components and Architecture

Kubernetes - Master Machine Components

- ✓ The Kubernetes Master is Responsible for Maintaining the Good State for your Cluster.
- ✓ Interact with Kubernetes, by kubectl command-line interface, you're communicating with your cluster's Kubernetes master.
- ✓ Typically all these processes run on a single node in the cluster, and this node is also referred to as the master.
- ✓ The master can also be replicated for Availability and Redundancy
- ✓ The master server acts as the **Primary Control Plane** for Kubernetes Clusters.

Kubernetes Components and Architecture

Kubernetes - Master Machine Components

❖ Etcd :

- ✓ **Create By CoreOS**
- ✓ **Very Small And Thin And Distributed**
- ✓ **Display the overall status of the cluster**

- ✓ Consistent and **Highly-Available key value store** used as Kubernetes' **backing store** for all cluster data.
- ✓ If your Kubernetes cluster uses etcd as its backing store, make sure you have a **back up plan** for those data.

Kubernetes Components and Architecture

Kubernetes Multi Master Architecture

Kubernetes Components and Architecture

Kubernetes - Master Machine Components

❖ API Server :

- ✓ Exposes the Kubernetes API. It is the **Front-End** for the Kubernetes **Control Plane**.
- ✓ The main implementation of a Kubernetes API server is kube-apiserver.
- ✓ kube-apiserver is designed to scale horizontally.
- ✓ That is, it scales by deploying more instances.
- ✓ You can run several instances of kube-apiserver and balance traffic between those instances.

Kubernetes Components and Architecture

Kubernetes - Master Machine Components

❖ Controller Manager :

- ✓ For Manage Workload life cycles, and Perform Routine Tasks.
- ✓ Component on the master that runs Controllers and each Controllers is a separate process .
- ✓ These controllers include:
  - ➢ Node Controller : For Hosts Checking
  - ➢ Replication Controller: For Pods Checking
  - ➢ Endpoints Controller: Join Services and Pods
  - ➢ Service Account & Token Controllers : Create default accounts and API access tokens for new namespaces.

Kubernetes Components and Architecture

Kubernetes - Master Machine Components

❖ **Scheduler :**

✓ Component on the master that watches **Newly Created Pods** that **have NO Node Assigned**, and select a node for them to **run on**.

✓ **Scheduling Factors :**

individual and collective resource requirements
hardware/software/policy constraints
data locality
inter-workload

Kubernetes Components and Architecture

Kubernetes - Worker Node Components

- ✓ **A Node is a worker machine in Kubernetes, Previously known as a Minion.**
- ✓ **A Node may be a VM or physical machine, depending on the cluster.**
- ✓ **Each node contains the services necessary to run pods and is managed by the master components.**
- ✓ **The services on a node include :**
  - ❑ kubelet
  - ❑ The container runtime (Docker)
  - ❑ kube-proxy

Finaly , The servers that perform work by Running Containers are known as node

Figure 1.9 The components that make up a Kubernetes cluster

Kubernetes Components and Architecture

Kubernetes - Worker Node Components

- **Kubelet Service :**
  - ✓ **The kubelet service communicates with the master components to authenticate to the cluster and receive commands and work.**
  - ✓ **This service is responsible for relaying information , and from the control plane services.**
  - ✓ **Interacting with the etcd store to read configuration details or write new values.**
  - ✓ **It controls the container runtime to launch or destroy containers as needed.**

.

Kubernetes Components and Architecture

Kubernetes - Worker Node Components

❖ A Container Runtime Interface (CRI):

- ✓ For run containers in Pods, Kubernetes uses a Container Runtime (CRI)
- ✓ Here are the installation instructions for various runtimes:
  - ➢ Docker
  - ➢ CRI-O
  - ➢ Containerd
  - ➢ Other CRI : rkt , Runc

Container Runtime (CRI) Flow :

Pull Image
Create Container ID
Assign Network ID to Container
Assign file system and Cgroup to Container
Run Container
Connect To Container

Kubernetes Components and Architecture

Kubernetes - Worker Node Components

A Container Runtime Interface (CRI):

Cgroup Drivers :

- ✓ When systemd is chosen as the init system for a Linux distribution.
- ✓ The init process generates and consumes a root control group (cgroup) and acts as a cgroup manager.
- ✓ Systemd has a tight integration with cgroups and will allocate cgroups per process
- ✓ It's possible to configure your container runtime and the kubelet to use cgroupfs

Kubernetes Components and Architecture

Kubernetes - Worker Node Components

❖Kubernetes Proxy Service :

- ✓ **Create and Manage networking environment .**
- ✓ **kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service A way to expose an application running on a set of Pods as a network service concept.**
- ✓ **kube-proxy maintains network rules on nodes.**
- ✓ **These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.**
- ✓ **kube-proxy forwards the traffic itself.**

Kubernetes Components and Architecture

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

- **Pods** **kind: Pod**

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

❖ Pods

- ✓ A Pod is a group of one or more containers. (such as Docker containers), with shared storage/network, and a specification for how to run the containers.
- ✓ A **pod** is the most Basic Unit that Kubernetes deals with.
- ✓ Containers themselves are not Assigned to hosts.
- ✓ Instead, one or more tightly coupled containers are encapsulated in an object called a pod.
- ✓ Containers within a Pod **share an IP address and port space**, and **can find each other via localhost (Pod Name)**
- ✓ Containers in different Pods have distinct IP addresses and can not communicate by IPC without special configuration. These containers usually communicate with each other via Pod IP addresses.
- ✓ In terms of Docker constructs, a Pod is modelled as a group of Docker containers with shared namespaces and shared filesystem volumes.

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

Termination of Pods:

Because Pods represent Running Processes on nodes in the cluster.

Then Nice Terminate Beter Kill Terminate

Default grace period (30s) in Nice Terminate

When a user requests a pod to be removed :

1. **the system sending a TERM signal to the main process in each container.**
2. **After the grace period ends, the KILL signal is sent to those processes,**
3. **And then the Pod is removed from the API server**

Note : Default grace period (0s) in Kill Terminate

--grace-period=0 --force remove pod

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

Termination of Pods Flow:

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

Termination of Pods Flow:

Container hooks :

There are Two hooks that are exposed to Containers:

- PostStart
- PreStop

- **PostStart**
  - ✓ This hook executes immediately after a container is created.
  - ✓ However, there is no guarantee that the hook will execute before the container ENTRYPOINT. No parameters are passed to the handler.

- **PreStop**
  - ✓ This hook is called immediately before a container is terminated due to
    - ➢ An API request or management event such as liveness probe failure, resource contention and others
    - ➢ A call to the preStop hook fails if the container is already in terminated or completed state

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

❖ **Replication Controllers :** **kind: ReplicationController**

Is Defines

A pod template and control parameters

to scale Number of replicas = Number of pod horizontally

IF Change Replica :

Increasing = Copies and Running Pods

Decreasing = Teminate and Kill pods

If a pod or a host fails , the controller will start new pods to compensate

Replication controllers can also perform rolling updates

Kubernetes Components and Architecture

Kubernetes Objects and Workloads

❖ Replication Sets : kind: ReplicaSet

- ReplicaSet is the Next-Generation ReplicationController
- Replication sets are Replace Replication Controllers
- Because , greater flexibility to manage pods and supports the new set-based label selector.
- It's mainly used by Deployment as a mechanism to orchestrate pod creation, deletion and updates.
- Note that we recommend using Deployments instead of directly using Replica Sets

```
replicas: 3
selector:
  matchLabels:
    tier: frontend
template:
  metadata:
    labels:
      tier: frontend
```