



1 Why is Kubernetes most widely used?


Top reasons why Kubernetes is popular:




Largest open-source project




Great Community support




Robust container deployment



Effective persistent storage



Multi-cloud support




Container health monitoring



MORE VIDEOS

2 What is Google Container Engine?

Google Container Engine is an open-source management platform for Docker containers and clusters








kubernetes

It is based on Kubernetes and provides support for the clusters that run in public cloud services of Google

MORE VIDEOS

Features	Kubernetes	Docker
Deployment 	Applications are deployed as a combination of pods, deployments, and services	Applications are deployed in the form of services
Autoscaling 	Available	Not available
Health checks 	The health checks are of two kinds: liveness and readiness	The health checks are limited to services
Setup 	Kubernetes is hard to setup and configure	Docker's setup and installation is easy
Tolerance ratio 	High fault tolerance	Low fault tolerance

4

What are the notable features of Kubernetes?



Container
balancing

Kubernetes always knows where to place containers



Services

Manages containers, offers security, networking and storage services



Self-monitoring

It monitors and continuously checks the health of nodes and containers



Scaling

MORE VIDEOS

Kubernetes allows you scaling resources not only vertically but also horizontally



Open-source

Kubernetes is an open source platform and can run in any environment



Storage
orchestration

Kubernetes mounts and adds a storage system to run apps

5

What are the main advantages of Kubernetes?



Automated rollback for changes that go wrong



Kubernetes allows you scaling resources not only vertically but also horizontally, easily and quickly



Automates various manual processes and controls server hosting and launching



Saves money by optimizing resources

MORE VIDEOS

6

Explain the architecture Layer of Kubernetes.

Kubernetes is built in three layers with each higher layer hiding the complexity found in lower layer



Application Layer
Pool and Services



Kubernetes Layer



Infrastructure Layer

MORE VIDEOS

6

Explain the architecture layer of Kubernetes.



Base/Infrastructure Layer

- ❖ This layer makes a cluster which is a collection of storage and networking resources that Kubernetes uses to run all the workloads in the system
- ❖ Cluster groups many machines into a single unit for it to be consumed

CORE VIDEOS

6

Explain the architecture layer of Kubernetes.



Mid/Kubernetes Layer

- ❖ The machines in the cluster are given a role in the Kubernetes system
- ❖ The Master is the control plane of Kubernetes having functionalities like Authorization and authentication, RESTful API entry point. It has 4 components namely - etcd, API server, scheduler, and controller manager
- ❖ The nodes are other machines in the cluster and are managed by Kubernetes master. They do all the processing and controlling of Kubernetes

CORE VIDEOS

6

Explain the architecture Layer of Kubernetes.



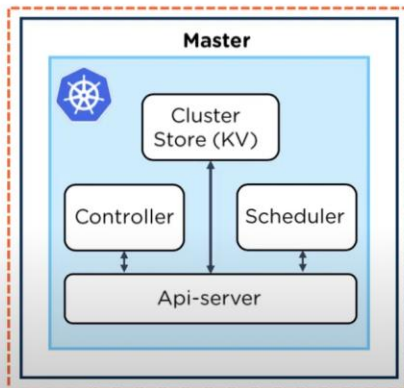
Final/Application Layer

- ❖ Kubernetes runs on an API approach
- ❖ To run an application, a plan is submitted in yaml or json
- ❖ The master server runs the plan and examines the requirements
- ❖ All user interact with the cluster by the help of API ecosystem
- ❖ Next comes the scheduler and controller-manager components that keeps the cluster functioning correctly
- ❖ Finally, the workers provides the output for the job

CORE VIDEOS

7

Explain Master node and list its components.



The Master node is the most vital component responsible for Kubernetes architecture. It is the central controlling unit of Kubernetes and manages workload and communications across the clusters.

The master node has four components:

1. ETCD
2. Controller Manager
3. Scheduler
4. API-Server

[MORE VIDEOS](#)

8

Explain clusters in Kubernetes.



Nodes combine their resources to form a powerful machine known as cluster.

Kubernetes does not work with individual nodes; it works with the cluster as a whole.

When a node is added or removed, the cluster shifts around the work as necessary.

[MORE VIDEOS](#)

9

What are the different types of Controllers in Kubernetes?

The controllers are used for many services in Kubernetes. There are five types of controller manager:



Node controller
Controls and handles the node in the system.



Service account
Enables access controls in the system.



Token controller
Cleans up any tokens for non-existent service accounts.



Endpoints controller
Joins services & pods to the endpoint objects.



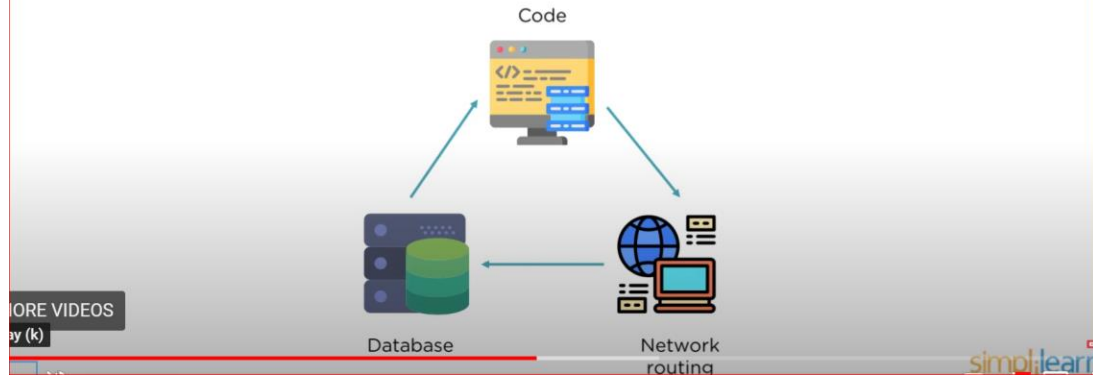
Replication controller
Manages the pod lifecycle.

[MORE VIDEOS](#)

10

What is the role of Cloud Controller Manager?

Cloud controller manager is an essential component for persistent storage, the abstraction of cloud-specific code from core Kubernetes code, and network routing.

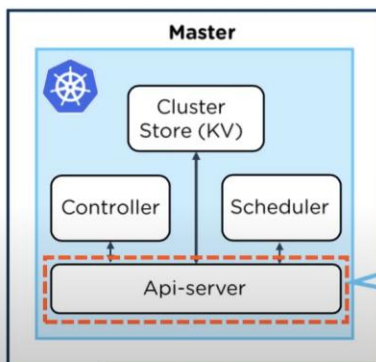


MORE VIDEOS

by (k)

11

What is the role of the Kube API Server and the Kube scheduler?

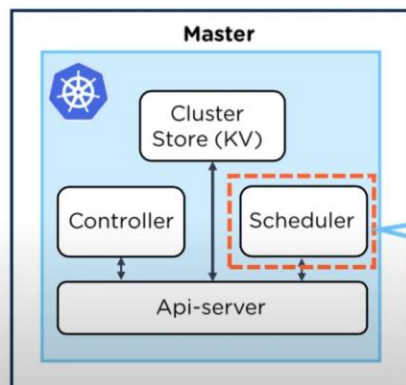


- ❖ Kube API Server is responsible for establishment of communication between the Kubernetes node and the Kubernetes master components and exposes all APIs of Kubernetes master node components
- ❖ It is a central management entity that receives all REST requests for modifications, serving as a frontend to the cluster
- ❖ It implements an interface which means different tools and libraries can communicate effectively

MORE VIDEOS

11

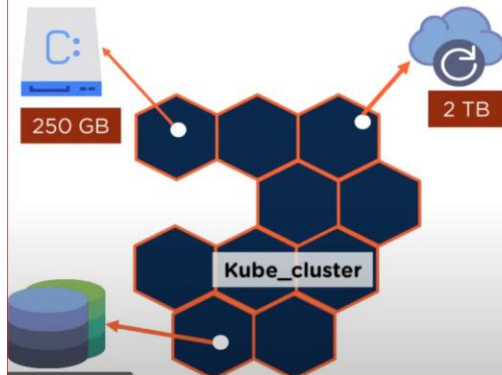
What is the role of the Kube API Server and the Kube scheduler?



- ❖ Kube scheduler distributes workload on the worker node and makes sure that no workload is scheduled to a node that is already full
- ❖ The scheduler is responsible for workload utilization and allocating pod to a new node
- ❖ The scheduler should have an idea of the total resources available as well as resources allocated to existing workloads on each node

12

Explain Persistent volumes in Kubernetes.



We use persistent volumes in Kubernetes to store data permanently

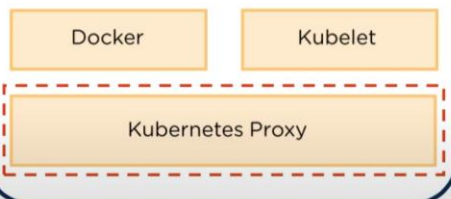
Local or cloud drives can be attached to the cluster as Persistent Volume

Persistent Volumes provide a file system that mounts to the cluster, without being associated with any node

13

What is a Kubelet and kubeproxy?

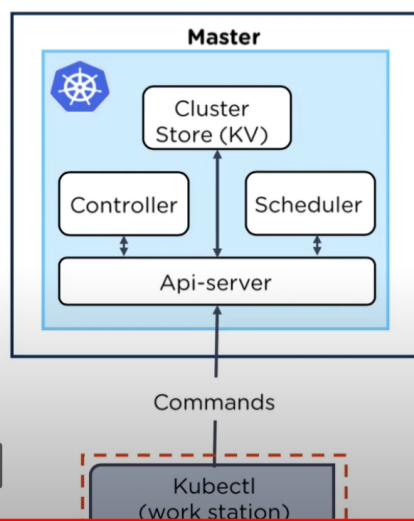
Kubernetes Node



- ❖ Kube-proxy is a proxy service which runs on each node and helps in making services available to the external host
- ❖ Every node in the cluster runs a simple network proxy, and Kube-proxy routes request to the correct container in a node
- ❖ It performs primitive load balancing and manages pods on node, volumes, secrets, creating new containers' health checkup, etc.

14

What is Kubectl?



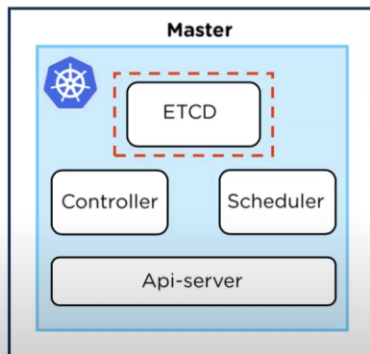
- ❖ Kubectl is a command line interface used for running commands for Kubernetes clusters
- ❖ It also controls the Kubernetes cluster manager

MORE VIDEOS

simple

15

What is ETCD?



- ❖ ETCD stores the configuration details and essential values. Key-value is available and distributed across multiple nodes
- ❖ It communicates with all other components to receive the commands and work in order to perform an action
- ❖ It also manages network rules and port forwarding activity

16

What are the disadvantages of Kubernetes?



It is hard to install and configure



It takes time to start running and gain traction



It is not simple to manage the services



Kubernetes is not an easy platform to learn, it takes a lot of time and effort

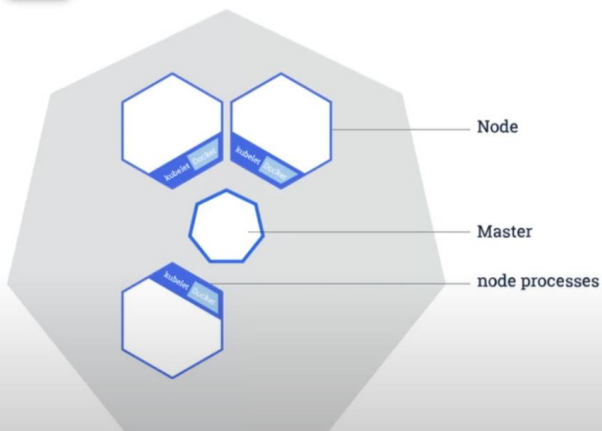


Kubernetes is expensive platform as highly skilled people are required to run it

MORE VIDEOS

17

What happens if a master node fails? What happens if a worker node fails?



Whenever master node under Kubernetes fails, the cluster still remains in an operational mode. It doesn't affect pod.

Whenever worker node under Kubernetes fails, master stop receiving updates from worker node

MORE VIDEOS

Kubernetes cluster

18

What is a service role in Kubernetes?

A service is an abstraction for pods. It provides a virtual IP (VIP) address. It allows clients connect to the containers running in the pods, using the Virtual IP address.

The command used to display services under Kubernetes is `kubectl get services`

19

How do I rollback a deployment?

Applying changes to a Deployment process with the `--record` flag then Kubernetes by default saves the previous Deployment activities in its history

To display all the prior Deployments:

```
kubectl rollout history deployment <deployment>
```

To restore last deployment:

```
kubectl rollout undo deployment <deployment>
```

20

What is an Ingress Controller?



An Ingress Controller is a pod that can act as an inbound traffic handler. Prominent features are HTTP path and service-based routing and SSL termination

22

How do you package Kubernetes Applications?

Helm is a package manager which allow users to package, configure, and deploy applications and services to the Kubernetes cluster

COMMANDS:

```
helm search redis # searches for a specific
application
helm install stable/redis # installs the
application
helm ls # list the applications
```

23

What are init containers?

A pod can have many containers. Init container gets executed before any other containers run in the pod

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
  annotations:
    pod.beta.kubernetes.io/init-containers: '[
  {
    "name": "init-myservice",
    "image": "busybox",
    "command": ["sh", "-c", "until nslookup
myservice;
sleep 2; done;"]
  }
]'
```

```
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', 'echo The app is running! &&
sleep 3600']
```

24

What is the difference between configmap and secret?

Config maps ideally stores application configuration in a plain text

COMMAND:

```
kubectl create configmap myconfigmap
--from-literal=env=dev
```

Secrets store sensitive data like password in an encrypted format

COMMAND:

```
❖ echo -n 'admin' > ./username.txt
❖ echo -n 'abcd1234'
./password.txt
❖ kubectl create secret generic mysecret --
from-file=./username.txt --from-
file=./password.txt
```

26

How do you deploy a feature with zero downtime in Kubernetes?

By default Deployment in Kubernetes using RollingUpdate as a strategy

- ❖ Update the nginx image
`kubectl set image deployment nginx nginx=nginx:1.15`
- ❖ Check the replica sets
`kubectl get replicaset`
- ❖ Check the status of a deployment rollout
`kubectl rollout history deployment nginx`
- ❖ Check the revisions in a deployment
`kubectl rollout history deployment nginx`

27

How to monitor a pod is always running?

A liveness probe always checks if an application in a pod is running, if this check fails the container gets restarted

```
spec:
  containers:
  - name: liveness
    image: k8s.gcr.io/liveness
    args:
    - /server
    livenessProbe:
      httpGet:
        path: /healthz
```

[MORE VIDEOS](#)

28

How to drain traffic from a pod during maintenance?

Pods inside the nodes also take a hit when a node is put into maintenance. However, we can avoid it by using the below command:

```
kubectl drain <nodename>
```

Once the node is up and running after maintenance and it should be added to rotation, execute the following command:

```
kubectl uncordon <nodename>
```

28

How do you tie services to a pod or a set of pods?

By declaring pods with the label(s) and by having a selector in the service which acts as a glue to stick the service to the pods

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
  - protocol: TCP
    port: 80
```

29

How to get all pods on a node?

The following command is used to get all the pods on a node in Kubernetes Cluster

```
$ kubectl get po --all-namespaces -o
jsonpath='{range .items[?(@.spec.nodeName
=="nodename")]}{.metadata.name}{"\n"}{end}'
```

30

How do pods mount NFS volume?

Configuring NFS Server

Define NFS server pod and NFS service:

- ❖ \$ kubectl create -f nfs-server-pod.yaml
- ❖ \$ kubectl create -f nfs-server-service.yaml

Define the pod:

```
$ kubectl create -f web-pod.yaml
Now the pod serves index.html from the NFS server:
$ curl http://<the container IP address>/
Hello World!
```