Top 36 Kubernetes Interview Questions in 2023

Description

Introduction

In this particular blog post you will get familiar with the most important kubernetes interview questions asked in the DevOps space, I have tried to cover the most basic questions to the more advanced questions on Kubernetes. Even if you are not a DevOps engineer, this post will give adequate knowledge if you are an application developer.

What is Kubernetes?

Kubernetes is one of the most famous open source container orchestration tool. We have tools like Docker to create containers, but in order to manage these containers we would need a container orchestration tools. These container orchestration tools are used to automate the tasks such as management, monitoring, scaling and deployment of the containers.

What is K8s ?

K8s is another name or alias that is used for Kubernetes, 8 stands for the number of letters between 'K' and 's' in the word Kubernetes.

What are the feature of Kubernetes?

Horizontal Scaling – This feature uses a HorizontalPodAutoscalar which automatically increases or decreases the number of pods in a deployment, replication controller, replica set, or stateful set based on the CPU utilization. Scaling can be controlled via configuration as well.

Self Healing – Whenever the container fails, Kubernetes tries to restart them automatically. Whenever a node dies it reschedules or replaces a container. It will automatically kill the old containers and remove it out of the network if they are not responding to the user requests. All these decision are taken based on the health checks done by Kubernetes which is done internally.

Load balancing & service discovery – This is one of the key features where Kubernetes assigns IP addresses and a DNS Name for a set of containers. Kubernetes can also take care of load balancing and service discovery like any other cloud platforms.

Automated Rollouts and Rolls backs – Kubernetes makes the changes progressively. If there are any issues in the deployment, Kubernetes will automatically rollback the deployments.

Storage Orchestration – Kubernetes can automatically mount the storage system according to your

choice, be it local storage or cloud storage like AWS etc. Kubernetes provides an essential feature called 'persistent storage' for storing the data, which cannot be lost after the pod is killed or rescheduled. Kubernetes supports various storage systems for storing the data, such as Google Compute Engine's Persistent Disks (GCE PD) or Amazon Elastic Block Storage (EBS). It also provides the distributed file systems: NFS or GFS.

Batch Execution – Everything that runs on Kubernetes is a workload. A workload can be a single component or several. There are several built-in workload resources available. Kubernetes provides two workload resources to create batch transactions i.e. a Job object and a CronJob object.

Configuration Management – Kubernetes provides a feature to change the configuration without rebuilding the image, so there is no need to redeploy the application when we change the configuration. Configuration will be outside the image.

What is container orchestration in Devops

Orchestration means coordination of multiple services and stringing them together in order to execute a workflow or a process. If you have seen an orchestra where there are multiple people playing their own instrument. So orchestration means the amalgamation of all instruments playing together in harmony. Similarly container orchestration means all the services in individual containers working together to fulfill the needs of a single server.

What is the need for Container Orchestration

Container orchestration is needed to achieve complete workflow of an application. Lets say we have 6-7 application docker containers. Every application needs to communicate with each other. So in order to manage these containers we would need a container orchestration tools. These tools are used to automate the tasks such as management, monitoring, scaling and deployment of the containers.

Difference between Kubernetes and Docker

From the trends it seems like both of these are competing technologies, but they are not alternatives to each other. In fact they go hand in hand together. Docker can run without kubernetes and kubernetes can run without docker. Good projects and architectures will use both of these wisely as they complement each other.

Docker	Kubernetes
Docker is a container technology which creates an isolated environment for the application.	While kubernetes is an Infrastructure for managing multiple containers.
Docker is mostly used in the CI CD process for automating, building and deploying applications.	It is used for automated scheduling and management of the deployed application containers.

Docker	Kubernetes
Docker is a container platform forconfiguring, building and distributingcontainers.	Kubernetes is an ecosystem for managing a cluster of docker containers.

Difference between Kubernetes and Docker Swarm

Kubernetes and Docker Swarm are both container orchestration tools and alternatives to each other.

Kubernetes	Docker Swarm
Kubernetes is a portable, open-source, cloud-native infrastructure tool initially designed by Google to manage their clusters. Being a container orchestration tool, it automates the scaling, deployment, and management of containerized applications.	Docker Swarm is an open-source container orchestration platform that is native to Docker. It supports orchestrating clusters of Docker engines.
Complex installation.	Easier installation.
More complex with high learning curve, but more powerful.	More lightweight and easier to use , but limited functionality.
Supports auto scaling.	Supports manual scaling.
Kubernetes has built in monitoring.	Docker swarm needs third party tools for monitoring.
Manual setup of load balancer is needed.	Docker swarm supports auto load balancing.
Kubernetes has Kubectl CLI tool.	Docker swarm integrates with docker CLI.

What is a node in Kubernetes

Node in Kubernetes is like a single machine in a cluster, which can be a physical machine in a data center or a virtual machine from a cloud provider. There is a two type of nodes in Kubernetes i.e. master node and worker nodes. The master in Kubernetes controls the nodes that have containers. Each machine can substitute any other machine in a Kubernetes cluster.

Master node has 4 processes running inside it i.e. Api server, scheduler, controller manager, etcd.

What is a pod in Kubernetes

- Pod is a basic and smallest unit of Kubernetes.
- Pod is an abstraction over a container.
- Pod is like a layer or running environment for the container.
- Usually only one application is run inside a single Pod and each pod gets its own IP address.
- Pods communicate with each other using this IP address. For some reason if a Pod dies, and a new Pod is created in its place, then a new IP address is assigned to it.

- The main reason for Kubernetes to introduce Pods is to be independent of the container technology so that it can be replaced if required. Instead of working with the containers directly, you interact with the pods

What is a service in Kubernetes

In Kubernetes, every pod has its own IP address. The IP address keeps on changing whenever a pod dies and a new pod is created. A service in Kubernetes stands in front of the pods and it has a stable IP address, hence the clients can call service IP address to access to pods instead of calling the pods directly whose IP address keeps changing frequently. One of the best thing about kubernetes service is that, the lifecycle of the service and the pod is not connected. Even if the pod dies, the service and its IP address will stay.

Service can also act as a load balancer to forward the request to a pod, if there are multiple pods of the same application.

What are the various Service types available in Kubernetes?

- **Cluster IP –** Cluster IP is the default service which has a static IP address. Accessible within the cluster.
- **NodePort –** Nodeport service makes external traffic access to fixed port on each worker node. This Service type is not secure. Not for production use case.
- **LoadBalancer –** Load balancer can be thought of an extension to NodePort service type and NodePort is an extension of ClusterIP service type. Each cloud provider has its own native loadbalancer implementation. Whenever a loadbalancer service is created, the NodePort and ClusterIP service is created automatically.
- **Headless –** When clients or internal application want to communicate with specific pods (like stateful database pods) directly.

What is ingress in Kubernetes

An ingress allows users to access Kubernetes services from outside the Kubernetes cluster. We can configure the access rules that define which inbound connections reach which services.

In order to access the kubernetes service, we have IP address of the node we want to talk to, so in the browser it will look like http://123.78.101.5:8081. Obviously for the end product you don't want the users to type the IP address for accessing the application. So we would need a proper domain name such as https://your-app.com . For this purpose we have a Kubernetes component called Ingress. So instead of service, the request first goes to ingress, and then the ingress will forward the request to the service.

# What is the difference between Deploying applications on host and container

A host will have an operating system with a kernel that holds various libraries installed on the operating system that is needed for an application. In this host based architecture there will be n number of applications, and each application will share the libraries present in the operating system.

In container based architecture, there is only one common component i.e. kernel. Every application will be containerized and it will be isolated from other applications. So, the applications have the necessary libraries and binaries isolated from the rest of the system, and cannot be encroached by any other application.

What are the different components of Kubernetes architecture

Footer Tagline
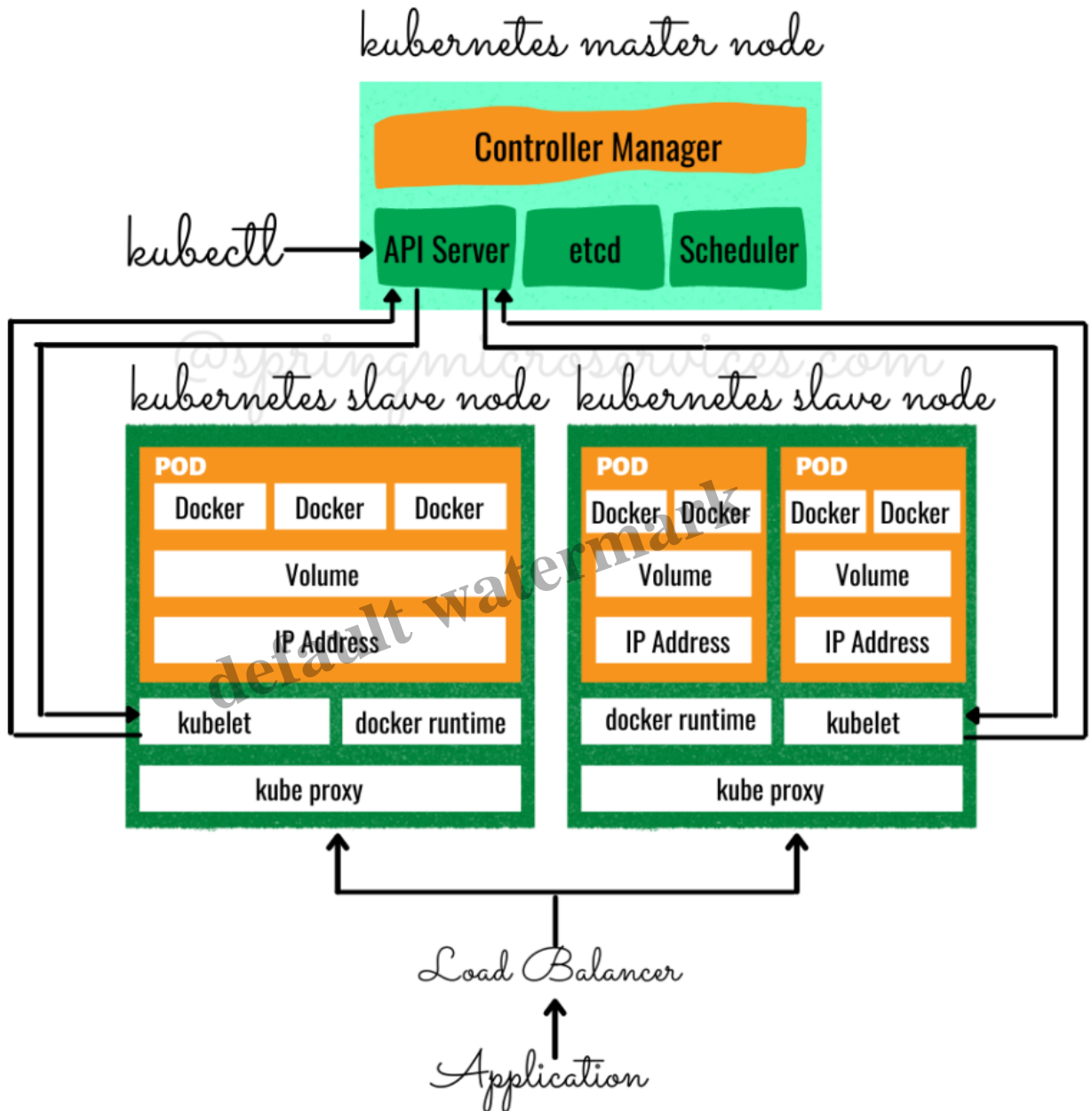
Kubernetes Architecture

Click here to see an entire blog post on kubernetes architecture.

What is MiniKube

MiniKube is used to run single node Kubernetes cluster on personal computers containing Windows, MacOS etc. If you want to try out Kubernetes for learning purpose or for daily development work then you can install MiniKube.

What is ConfigMap and Secret

ConfigMap is used to store configuration information like URLs of other dependent services that your service depends on. It can also be used to store the database configuration information. We just need to connect the configMap to the pod and the pod will get the required configuration. The traditional way of storing such information was either in application.properties or application.yaml or in the form for environment variables. The properties or yaml files would be part of the application build. One of the drawback of this approach was that, if any of the value changed then we would have to rebuild the application and redo other steps to reflect the new change. In case of ConfigMap if there is any change in values then we don't need to rebuild the image, we just change the ConfigMap and the changes will be reflected.

Secret is just like the ConfigMap. The only difference is that Secret is used to store secret data such as credentials, certificates etc. It would be risky to store the passwords in ConfigMap, hence secrets are the ultimate choice. The data is stored in base64 encoded format. Just like ConfigMap, it is also connected to the Pod so that it can read the information. You can read the information from the ConfigMap or Secret inside of your application pod using the environment variable or even as a properties file.

What are Volumes in Kubernetes

Volumes in Kubernetes are used to persist the data in the database reliably and for the long term period. It basically attaches a physical storage or your hard drive to your pod. The storage could be either at your local machine where the pod is running or it could be at the remote storage which would be outside your kubernetes cluster. It could be either cloud storage or on premise storage which is not part of your kubernetes cluster.

If there were no kubernetes volumes and if the application or the database container or the pod gets restarted, then data would be gone which would be problematic and inconvenient. Since the introduction of volumes if your kubernetes pod or container got restarted, your data would be safe.

Kubernetes does not manage the data persistence or data backup, you as a kubernetes user have to manage it.

What is deployment in kubernetes

Deployment in kubernetes is like an abstraction over a pod, which makes it more convenient to interact with the pods, replicate them and do some other configuration. So in practice you would mostly work with deployments and not with the pods.

What is statefulset in kubernetes

Statefulset is meant for applications like databases such as mongodb, postgres etc, as these are stateful applications. If you want to replicate the databases, then you need to use the statefulset rather than the deployments. Statefulset just like deployment would take care of replicating the pods and scaling them up and down, but making sure that database reads, writes are synchronized so that no database inconsistencies occur. However deploying databases using statefuset in kubernetes cluster can be somewhat tedious. One of the common practice is to host the database outside the kubernetes cluster.

What is etcd in kubernetes

Etcd in kubernetes is one of the processes present in the master node. Etcd is a key-value store which contains the cluster state information. You can think of it as a cluster brain. Every change in the cluster e.g. when a new pod gets created in the cluster or when when the pod dies, all these changes are saved or updated into this key-value store. Other processes of the master node such as api server, scheduler, controller manager are dependent on the information present in the etcd.

Etcd contains the information such as whether the cluster is healthy or not? how many resources are available? etc.

What is ClusterIP in kubernetes

The ClusterIP is the default Kubernetes service that provides a service inside a cluster (with no external access) that other apps inside your cluster can access.

What is a LoadBalancer in Kubernetes

The Load Balancer service is used to expose services in the cluster to the internet. A Network load balancer creates a single IP address that forwards all traffic to your service.

As shown in the below figure, all the incoming traffic from outside the internet comes to a single IP address on the load balancer which routes the incoming traffic to a particular pod (via service) using an algorithm known as round-robin. If any of the pod goes down load balances are notified so that the traffic is not routed to that unavailable node. Thus load balancers in Kubernetes are responsible for distributing incoming traffic to the pods

What is Kubelet in Kubernetes

Kubelet is one of the process in kubernetes which runs on each worker node and enables the worker node to communicate with the master. So, Kubelet works based on the configuration of containers present in the PodSpec and makes sure that the containers described in the PodSpec are healthy and running.

Kubelet interacts with both the container and the node.

What is Kubectl in Kubernetes

Kubectl is the platform where you can pass commands to the cluster. Kubectl basically provides the CLI to run commands against the Kubernetes cluster with various ways to create and manage the Kubernetes component.

What is Heapster

Heapster is similar to other pods in the Kubernetes cluster which is used to query usage data from all the nodes within the Kubernetes cluster. Heapster is like an aggregator which monitors the performance and collects all the metrics from the data collected by the Kublet.

What is a Kube-Proxy in Kuberntes

Kube-proxy is used for directing traffic to the right container based on IP and the port number of incoming requests. Kube-proxy is an implementation of a load balancer and network proxy used to support service abstraction with other networking operations. Kube-proxy must be installed on every worker node.

To summarize, Kube-Proxy is responsible for the communication of pods within the cluster and to the outside network, which runs on every node. This service is responsible to maintain network protocols when your pod establishes a network communication.

What is Kube-apiserver

Kube-apiserver is a master API service which acts as an entry point to K8 cluster. API server is the heart of the entire cluster. The **API Server** exposes RESTful APIs so that we can interact with the master node. The way we can fire the calls is using **kubectl**. It is basically a command line utility tool which takes yml or json files that we provide, and it in turn hands it over to the API server, the API server will take the request, validate and process by performing the task against the worker nodes. **kubectl** is just the wrapper around the curl command which makes the http call.

What does node status contain

Address, Condition, Capacity, and Info are the main components of node status.

What is a kube-scheduler in Kubernetes

The kube-scheduler is responsible for assigning nodes to newly created pods. It schedule's POD's according to available resources on executor nodes.

What do you mean by namespace in Kubernetes

The namespaces allow applications to be organized into groups that fall under same domain, such as a separate namespace for all finance applications, for monitoring applications and another for all security applications.

Namespaces can also be used to divide cluster resources between multiple users. They are meant for environments where there are multiple users spanning multiple projects or teams and provide a scope of resources.

What do you mean by Kubernetes controller manager

Kubernetes controller manager enables the running of multiple processes on the master node even though they are compiled to run as a single process.

Control manager runs in the background. The Control manager is responsible for making sure the cluster is in the desired state. For example, If we want 5 pods to be up and running, and within those 5 pods we always need to have 2 containers in each of those pods, then it is the responsibility of the control manager to make sure that the cluster is in the desired state as requested.

What are the types of controller managers

The primary controller managers that can run on the master node are the service accounts controller, namespace controller, endpoints controller, token controller, node controller, and replication controller.

What do you mean by headless service

Headless service enables you to access the pods directly without a proxy. It is similar to normal service, but the only difference is that it does not have Cluster IP.

What is the difference between Replication controller and Replica set ?

Replication Controller monitors the pods and automatically restarts them if there is any failure. If the node fails, the the controller will respawn all the pods of that particular node on another node. It is responsible for updating or deleting multiple pods with a single command.

Replication controller and Replica set are almost the same thing. Replica set can be said as an advanced version of Replication controller. The only difference is in the usage of selectors to replicate pods. Replica Set uses Set-Based selectors while replication controllers use Equity-Based selectors.

What are the ways to establish security in Kubernetes

- We can introduce RBAC (Role-based access control).
- We can make use of namespaces to establish security boundaries.
- Set the admission control policies to avoid running the privileged containers.
- Enable audit logging

Category

1. Devops
2. Interview
3. Kubernetes

Date Created
April 5, 2023
Author
kk-ravi144gmail-com

Footer Tagline