

Kill Your Tech Interview

3877 Full-Stack, Coding & System Design Interview Questions

Answered To Get Your Next Six-Figure Job Offer

[See All Questions](#)
[Data Science & ML QAs](#)
 **Full-Stack, Web & Mobile**
 **Coding & Data Structures**
 **System I**
 **Full-Stack**
 **Coding**
 **System Design**
 **.NET Core** 67

 **ADO.NET** 33

 **ASP.NET** 54

 **ASP.NET MVC** 36

 **ASP.NET Web API** 33

 **Agile & Scrum** 47

 **Android** 113

 **Angular** 120

 **AngularJS** 61

 **Azure** 35

 **C#** 125

 **CSS** 50

 **Dependency Injection** 17

 **Design Patterns** 68

 **DevOps** 44

 **Entity Framework** 57

 **Flutter** 67

 **Git** 36

 **Golang** 49

 **GraphQL** 25

 **HTML5** 55

 **Ionic** 29

 **Java** 165

 **JavaScript** 179

 **Kotlin** 68

 **LINQ** 38

 **Laravel** 41

 **MongoDB** 83

 **MySQL** 60

 **Node.js** 126

 **OOP** 58

 **Objective-C** 43

 **PHP** 82

 **PWA** 22

 **Python** 112

 **React** 130

 **React Hooks** 46

 **React Native** 72

 **Reactive Programming** 12

 **Redis** 25

 **Redux** 35

 **Ruby** 84

 **Ruby on Rails** 72

 **SQL** 42

 **Spring** 87

 **Swift** 72

 **T-SQL** 51

 **TypeScript** 91

 **UX Design** 78

 **Unit Testing** 24



Having Data Science & ML Interview? Check [MLStack.Cafe](#) - 1704 Data Science & ML Interview Questions & Answers! Having ML & DS Interview? Check [MLStack.Cafe](#) - 1704 ML & DS Interview Questions and Answers

21 Kubernetes Interview Questions (ANSWERED) For Senior and DevOps Developers

[Kubernetes 27](#)



Kubernetes has become the go-to orchestration platform since it was launched in 2014. According Cloud Native Computing Foundation (CNCF) survey 84% of companies are using containers in production and the vast majority (78%) are using Kubernetes. Follow along and check 21 most common Kubernetes interview questions every DevOps and senior developer shall know.



instead! Claim your \$100 credit.
ADS VIA CARBON

Q1: What is Kubernetes? Why organizations are using it?

Entry



Kubernetes 27

Answer

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

To understand what Kubernetes is good for, let's look at some examples:

- You would like to run a certain application in a container on multiple different locations. Sure, if it's 2-3 servers/locations, you can do it by yourself but it can be challenging to scale it up to additional multiple location.
- Performing updates and changes across hundreds of containers
- Handle cases where the current load requires to scale up (or down)

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: github.com

Q2: How can containers within a pod communicate with each other?

Junior



Kubernetes 27

Answer

Containers within a pod share networking space and can reach other on `localhost`. For instance, if you have two containers within a pod, a MySQL container running on port `3306`, and a PHP container running on port `80`, the PHP container could access the MySQL one through `localhost:3306`.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: github.com

Q3: What does a Pod do?

Junior



Kubernetes 27

Answer

Pods represent the processes running on a cluster. By limiting pods to a single process, Kubernetes can report on the health of each process running in the cluster. Pods have:

- configuration information that determine how a container should run.

Although most pods contain a single container, many will have a few containers that work closely together to execute a desired function.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: www.vmware.com

Q4: What is Kubernetes, exactly?

Junior



Kubernetes 27

Answer

The purpose of **Kubernetes** is to make it easier to organize and schedule your application across a fleet of machines. At a high level it is an operating system for your cluster.

Basically, it allows you to not worry about what specific machine in your datacenter each application runs on. Additionally it provides generic primitives for health checking and replicating your application across these machines, as well as services for wiring your application into micro-services so that each layer in your application is decoupled from other layers so that you can scale/update/maintain them independently.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: stackoverflow.com

Q5: Explain what are some Pods usage patterns?

Mid



Kubernetes 27

Answer

Pods can be used in two main ways:

- **Pods that run a single container.** The simplest and most common Pod pattern is a single container per pod, where the single container represents an entire application. In this case, you can think of a Pod as a wrapper.
- **Pods that run multiple containers that need to work together.** Pods with multiple containers are primarily used to support colocated, co-managed programs that need to share resources. These colocated containers might form a single cohesive unit of service—one container serving files from a shared volume while another container refreshes or updates those files. The Pod wraps these containers and storage resources together as a single manageable entity.

Each Pod is meant to run a single instance of a given application. If you want to run multiple instances, you should use one Pod for each instance of the application. This is generally referred to as *replication*. Replicated Pods are created and managed as a group by a controller, such as a Deployment.

37 Advanced iOS Developer Interview Questions (SOLVED and EXPLAINED)

© Objective-C 43

Swift 72

iOS 36

Q6: Explain what is a Master Node and what component does it consist of?

Mid



Kubernetes 27

Answer

- 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 various **components**, each having its process. They are:

- ETCD
- Controller Manager
- Scheduler
- API Server

ETCD (Cluster store)

- This component stores the configuration details and essential values
- It communicates with all other components to receive the commands and work in order to perform an action
- It also manages network rules and posts forwarding activity

Controller Manager

- It is responsible for most of the controllers and performs a task
- It is a daemon which runs in a continuous loop and is responsible for collecting and sending information to API server
- The key controllers handle nodes, endpoints, etc.

Scheduler

- It is one of the key components of the master node associated with the distribution of workload
- The scheduler is responsible for workload utilization and allocating pod to a new node

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: www.quora.com

Q7: Explain when to use Docker vs Docker Compose vs Docker Swarm vs Kubernetes

Mid



Kubernetes 27

Answer

- **Docker** is a container engine, it makes you build and run usually no more than one container at most, locally on your PC for development purposes.
- **Docker Compose** is a Docker utility to run multiple containers and let them share volumes and networking via the docker engine features, runs locally to emulate service composition and remotely on clusters. Docker Compose is mostly used as a helper when you want to start multiple Docker containers and don't want to start each one separately using `docker run ...`.
- **Docker Swarm** is for running and connecting containers on *multiple* hosts. It does things like scaling, starting a new container when one crashes, networking containers.
- **Kubernetes** is a container orchestration platform, it takes care of running containers and enhancing the engine features so that containers can be composed and scaled to serve complex applications (sort of PaaS, managed by you or cloud provider). Kubernetes' goal is very similar to that for Docker Swarm but it's developer by Google.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: stackoverflow.com

Q8: What are namespaces? What is the problem with using one default namespace?

Mid



Kubernetes 27

Answer

Namespaces allow you split your cluster into virtual clusters where you can group your applications in a way that makes sense and is completely separated from the other groups (so you can for example create an app with the same name in two different namespaces).

- When using the default namespace alone, it becomes hard over time to get an overview of all the applications you manage in your cluster. Namespaces make it easier to organize the applications into groups that makes sense, like a namespace of all the monitoring applications and a namespace for all the security applications, etc.

- Another use case for namespaces is one cluster, multiple teams. When multiple teams use the same cluster, they might end up stepping on each others toes. For example if they end up creating an app with the same name it means one of the teams overidden the app of the other team because there can't be two apps in Kubernetes with the same name (in the same namespace).

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: github.com

 **Full-Stack, Web & Mobile**

 **Coding & Data Structures**

 **Sy**

 **Full-Stack**

 **Coding**

 **System Design**

 **TypeScript 91**

 **PWA 22**

 **Reactive Programming 12**

 **CSS 50**

 **Azure 35**

 **Unit Testing 24**

 **JavaScript 179**

 **Objective-C 43**

 **Swift 72**

 **MySQL 60**

 Having Data Science & ML Interview? Check  **MLStack.Cafe - 1704 Data**

Science & ML Interview Questions & Answers! Having ML & DS Interview? Check  **MLStack.Cafe - 1704 ML & DS Interview Questions and Answers**

Q9: What does it mean that "pods are ephemeral"?

Mid

 **Kubernetes 27**

Answer

Pods are *ephemeral*. They are not designed to run forever, and when a Pod is terminated it cannot be brought back. In general, Pods do not disappear until they are deleted by a user or by a controller.

Pods do not "heal" or repair themselves. For example, if a Pod is scheduled on a node which later fails, the Pod is deleted. Similarly, if a Pod is evicted from a node for any reason, the Pod does not replace itself.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: github.com

Q10: What happens when a master fails? What happens when a worker fails?

Mid



Kubernetes 27

Answer

Kubernetes is designed to be resilient to any individual node failure, master or worker. When a master fails the nodes of the cluster will keep operating, but there can be no changes including pod creation or service member changes until the master is available. When a worker fails, the master stops receiving messages from the worker. If the master does not receive status updates from the worker the node will be marked as `NotReady`. If a node is `NotReady` for 5 minutes, the master reschedules all pods that were running on the dead node to other available nodes.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: raw.githubusercontent.com

Q11: What is a StatefulSet in Kubernetes?

Mid



Kubernetes 27

Answer

When using Kubernetes, most of the time you don't care how your pods are scheduled, but sometimes you care that pods are deployed in order, that they have a persistent storage volume, or that they have a unique, stable network identifier across restarts and reschedules. In those cases, **StatefulSets** can help you accomplish your objective. It manages the deployment and scaling of a set of Pods, and provides guarantees about the **ordering** and **uniqueness** of these Pods.

StatefulSets are valuable for applications that require one or more of the following.

- Stable, unique network identifiers.
- Stable, persistent storage.
- Ordered, graceful deployment and scaling.
- Ordered, automated rolling updates.

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: kubernetes.io

Q12: What is a DaemonSet?

Mid



Kubernetes 27

DaemonSets are used in Kubernetes when you need to run one or more pods on all (or a subset of) the nodes in a cluster. The typical use case for a DaemonSet is logging and monitoring for the hosts. For example, a node needs a service (daemon) that collects health or log data and pushes them to a central system or database.

As the name suggests you can use daemon sets for running daemons (and other tools) that need to run on all nodes of a cluster. These can be things like cluster storage daemons (e.g. Quobyte, glusterd, ceph, etc.), log collectors (e.g. fluentd or logstash), or monitoring daemons (e.g. Prometheus Node Exporter, collectd, New Relic agent, etc.)

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: kubernetes.io

Q13: What is the difference between Kubernetes and Docker?

Mid

 Docker 64

Problem

And what are they used for?

Answer

Docker and Kubernetes are complementary.

- **Docker** provides an open standard for packaging and distributing containerized applications, while
- **Kubernetes** provides for the orchestration and management of distributed, containerized applications created with Docker.

In other words, Kubernetes provides the infrastructure needed to deploy and run applications built with Docker.

Having Tech or Coding Interview? Check  64 Docker Interview Questions

Source: stackoverflow.com

5 Salary Negotiation Rules for Software Developers

\$ Career 1

Q14: When to use StatefulSet?

Mid

 Kubernetes 27

Answer

Some examples of reasons you'd use a StatefulSet include:

- A webapp that needs to communicate with its replicas using known predefined network identifiers

Having Tech or Coding Interview? Check  27 Kubernetes Interview Questions

Source: www.bluematador.com

Q15: Which problems does a Container Orchestration solve?

Mid

 Docker 64

Answer

Containers run in an isolated process (usually in its own namespace). This means that by default the container will not be aware of other containers. Additionally, it will not be aware of the system files, network interfaces, and processes. While this can greatly help with portability of the software it does not solve several production issues such as microservices, container discovery, scalability, disaster recovery, or upgrades.

Adding a container orchestrator can greatly reduce the complexity in production as these tools are designed to resolve the issues outlined above. For example, Kubernetes is built to allow containers to be linked together, deploy containers across an entire network, scale and load balance the network based on container resource consumption, and allow upgrades of individual containers with no downtime.

If you are only running a single container or two containers together you are correct in that an orchestrator may be unnecessary and add unneeded complexity.

Having Tech or Coding Interview? Check  64 Docker Interview Questions

Source: devops.stackexchange.com



Full-Stack, Web & Mobile



Coding & Data Structures



Sy



Full-Stack



Coding



System Design



DevOps 44



Azure 35



Entity Framework 57



Unit Testing 24



GraphQL 25



.NET Core 67



WebSockets 24



Spring 87



Java 165



Objective-C 43



Having Data Science & ML Interview? Check  MLStack.Cafe - 1704 Data

Science & ML Interview Questions & Answers! Having ML & DS Interview? Check 

Q16: How does Kubernetes use etcd?

Senior

 Kubernetes 27

Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

 Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

Q17: Why do we need Kubernetes (and other orchestrators) above containers?

Senior

 Kubernetes 27

Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

 Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

60 .NET Interview Questions Devs Must Focus On (ANSWERED)

 .NET Core 67 ADO.NET 33 ASP.NET 54

Q18: Can you explain a relationship between *container runtime* and *container orchestration*?

Expert

 Docker 64

Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to open Expert question!

Q19: Explain what are Taints in Kubernetes?

Expert

 Kubernetes 27

Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to **open Expert question!**



Q20: How do I build a High Availability (HA) cluster?

Expert

 Kubernetes 27

Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to **open Expert question!**



Q21: How does *StatefulSets* use differ from the use of "stateless" Pods with Persistent Volumes?

Expert

 Kubernetes 27

Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to **open Expert question!**



Top 60+ MEAN Stack Developer Interview Questions And Answers

 API Design 46 Angular 120 CSS 50

24 Unit Testing Interview Questions (ANSWERED) For Software Developers

 Software Testing 29

Unit Tests and Test Driven Development (TDD) help you really understand the design of the code you are working on. Instead of writing code to do something, you are starting by outlining all the conditions you are subjecting the code to and what output...



35 Domain-Driven Design Interview Questions (ANSWERED) for Software Devs and Architects

 DDD 37

Domain-Driven Design is nothing magical but it is crucial to understand the importance of Ubiquitous Language, Domain Modeling, Context Mapping, extracting the Bounded Contexts correctly, designing efficient Aggregates and etc. before your next DDD p...



 Azure 35

At its core, Microsoft Azure is a public cloud computing platform - with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be used for services such as analytics, virtual c...



52 Node.js Interview Questions (ANSWERED) for JavaScript Developers

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. Follow along to refresh your knowledge and explore the 52 most frequently asked and advanced Node JS Interview Questions and Answers every...

 Node.js 126

21 Dependency Injection Interview Questions (ANSWERED) For Developers and Software Architects

Dependency Injection is most useful when you're aiming for code reuse, versatility and robustness to changes in your problem domain. DI is also useful for decoupling your system. DI also allows easier unit testing without having to hit a database and...

 Dependency Injection 17 Design Patterns 68 Software Architecture 85

27 Advanced React Hooks Interview Questions (ANSWERED) For ReactJS Developers



hooks, you can now do all these without switching, using functional com...

React Hooks 46

13 Repository Pattern and UoW Interview Questions (ANSWERED)

The main advantage of the repository pattern is that it abstracts the database behind it. Think of it as a tech-agnostic way of fetching and storing data in a data store. Follow along and refresh the 13 most common and advanced Repository Pattern and...

Design Patterns 68

36 Advanced TypeScript Interview Questions (ANSWERED) For JavaScript Developers

TypeScript starts from the same syntax and semantics that millions of JavaScript developers know today. Don't miss that advanced list of 36 TypeScript interview questions and answers and nail your next web developer tech interview in style....

TypeScript 91

50+ Mobile Developer Interview Questions (ANSWERED) to Know

Mobile app developers are responsible for developing the applications both on Android and iOS and using all sort of tech including PWA, React Native, Ionic, Xamarin and etc. iOS developers can expect to earn, on average, over \$113,000, with some jobs...

Android 113

Design Patterns 68

Ionic 29

27 Objective-C Interview Questions (ANSWERED) Every iOS Developer Should Know

Objective-C is a nice middle ground that gets close to the performance of compiled languages (like C++) while still maintaining some dynamism that you find in interpreted languages (like JavaScript or Ruby). There is still some Objective-C developmen...

Objective-C 43

iOS 36

FullStack.Cafe is a biggest hand-picked collection of top Full-Stack, Coding, Data Structures & System Design Interview Questions to land 6-figure job offer in no time.

Coded with ❤️ using React in Australia AU

by @aershov24, Full Stack Cafe Pty Ltd 🎉, 2018-2022

[Privacy](#) • [Terms of Service](#) • [Guest Posts](#) • [Contacts](#) • [MLStack.Cafe](#)

Great  based on 86 Trustpilot  Reviews

