

# 12 Kubernetes interview questions and answers to score that job

By **Trenton Romph** on July 1, 2020

---

[Home](#) › [Blog](#) › [Kubernetes](#) › 12 Kubernetes interview questions and answers to score that job

You've put in the time applying for tech jobs and have made it to the daunting interview.

You know they are going to ask you about Kubernetes, so make sure you are prepared. If you are completely new to Kubernetes we recommend you first read our [Kubernetes pronunciation post](#), so you don't *really* embarrass yourself, and then check out the [Kubernetes glossary](#) we created to help give you a decent Kubernetes foundation.

Once you have a solid understanding of Kubernetes, then you should move on to the interview questions below.

Unless you are interviewing specifically for a Kubernetes-related job, you'll probably be given very basic Kubernetes questions first, just to see if you know anything about it. If you answer those well or say you have experience with it, most likely your interviewer will dig a little deeper to gauge your actual Kubernetes skill level. So we'll start with the basics first and then ramp up to more advanced Kubernetes questions as we go.



You after reading this article

## Question 1. What is Kubernetes?

Kubernetes is an open-source container orchestration platform. It was developed by Google and was donated to the Cloud Native Computing Foundation (CNCF) in 2015.

## Question 2. How does Kubernetes relate to Docker?

Docker is a container runtime, which is a software that runs containerized applications. When Kubernetes schedules a pod to a node, the kubelet running on that node instructs Docker to launch the containers.

## Question 3. What is container orchestration?

Container orchestration is the automation of components and processes related to running containers. It includes things like configuring and scheduling containers, the availability of containers, allocation of resources between containers, and securing the interaction between

## Question 4. What do you know about Kubernetes clusters?

A Kubernetes cluster is a set of nodes that containerized applications run on. These nodes can be physical machines or virtual machines.

## Question 5. What is kubectl?

Kubectl is the command-line configuration tool for Kubernetes that communicates with a Kubernetes API server. Using kubectl allows you to create, inspect, update, and delete Kubernetes objects.

To learn all the stuff you can do with kubectl, check out our [kubectl cheatsheet](#).

## Question 6. What is a pod?

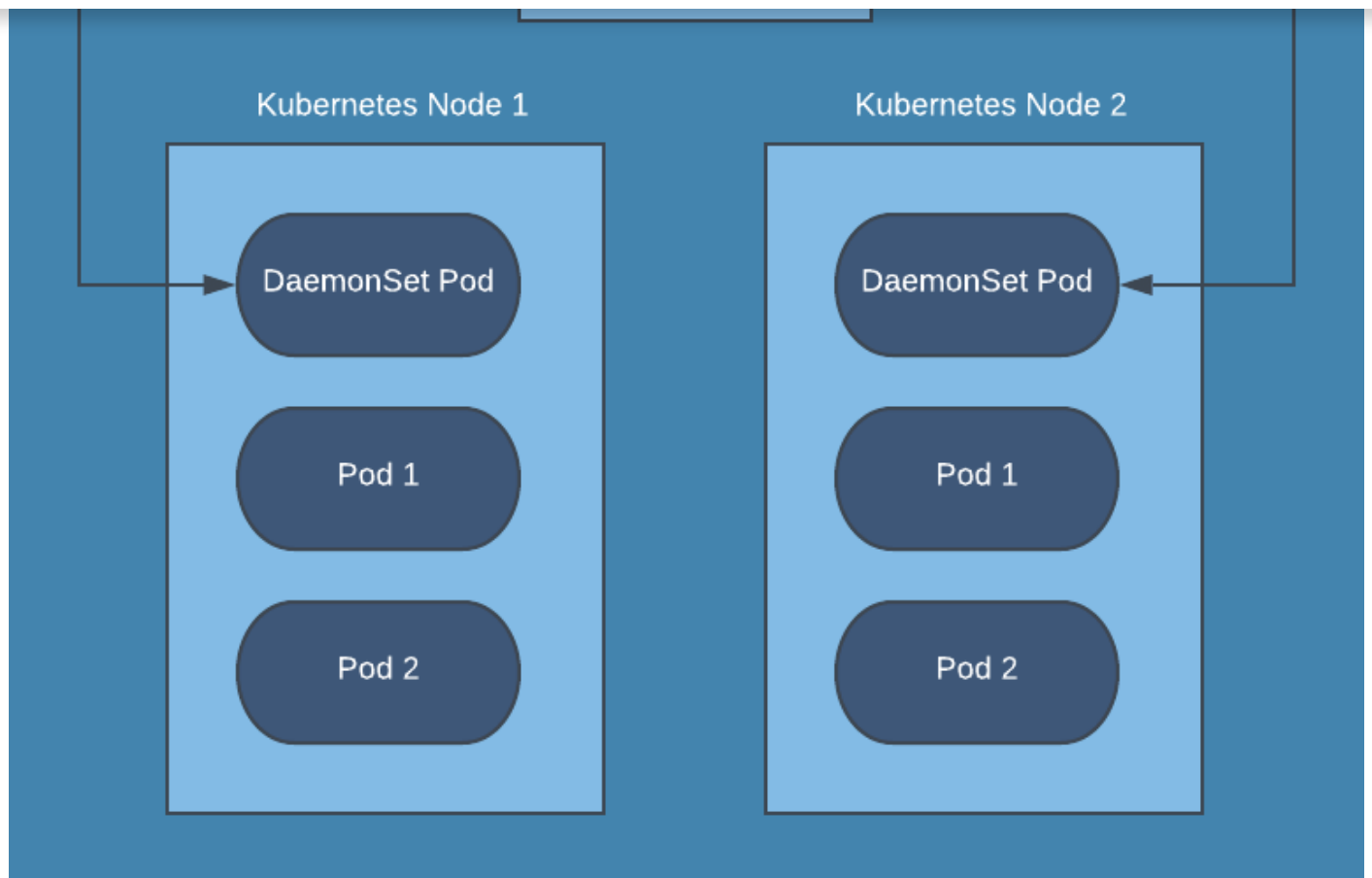
A pod is the most basic Kubernetes object. A pod consists of a group of containers running in your cluster. Most commonly, a pod runs a single primary container.

## Question 7. Can you explain the different components of Kubernetes architecture?

Kubernetes is composed of two layers: a **control plane** and a **data plane**. The **control plane** is the container orchestration layer that includes 1. Kubernetes objects that control the cluster, and 2. the data about the cluster's state and configuration. The **data plane** is the layer that processes the data requests and is managed by the control plane.

## Question 8. What is the difference between a daemonset, a deployment, and a replication controller?

A [daemonset](#) ensures that all nodes you select are running exactly one copy of a pod.



A deployment is a resource object in Kubernetes that provides declarative updates to applications. It manages the scheduling and lifecycle of pods. It provides several key features for managing pods, including pod health checks, rolling updates of pods, the ability to roll back, and the ability to easily scale pods horizontally.

The replication controller specifies how many exact copies of a pod should be running in a cluster. It differs from a deployment in that it does not offer pod health checks, and the rolling update process is not as robust.

## Question 9. Do all of the nodes have to be the same size in your cluster?

No, they don't. The Kubernetes components, like kubelet, will take up resources on your nodes, and you'll still need more capacity for the node to do any work. In a larger cluster, it often makes sense to create a mix of different instance sizes. That way, pods that require a lot of memory with intensive compute workloads can be scheduled by Kubernetes on large nodes, and smaller nodes can handle smaller pods.

A sidecar container is a utility container that is used to extend support for a main container in a Pod. Sidecar containers can be paired with one or more main containers, and they enhance the functionality of those main containers. An example would be using a sidecar container specifically to process system logs or for monitoring.

## Question 11. How do logs work for pods?

With a traditional server setup, application logs are written to a file and then viewed either on each server or collected by a logging agent and sent to a centralized location. In Kubernetes, however, writing logs to disk from a pod is discouraged since you would then have to manage log files for pods. The better way is to have your application output logs to **stdout** and **stderr**. The kubelet on each node collects stdout and stderr on the running pods and then combines them into a log file managed by Kubernetes. Then you can use different [kubectl commands](#) to view the logs.

## Question 12. How can you separate resources?

You can separate resources by using namespaces. These can be created either using `kubectl` or applying a YAML file. After you have created the namespace you can then place resources, or create new resources, within that namespace. Some people think of namespaces in Kubernetes like a virtual cluster in your actual Kubernetes cluster.

## Conclusion

If you study these questions and commit the answers to memory, you'll be well on your way to landing that sweet gig.

Want to make sure you stay informed about Kubernetes, AWS, and all the other stuff you'll have to know for your new job? Sign up for Blue Matador updates below [↓](#). We'll keep you sounding smart.

## What to Read Next

[How to Delete Pods from a Kubernetes Node](#)

[Building and deploying a Docker image to a Kubernetes cluster](#)



MONITORING

WHY US

PRIC

Subscribe to getting updates in your inbox.

Email\*

SUBSCRIBE

**kalyan kumar choudhury** 8/10/2020, 8:48:26 PM

nice blog to know about docker tool

[↩](#) Reply to *kalyan kumar choudhury***priyanka dasari** 4/1/2021, 4:24:00 PM

Thanks for sharing such a great Information! Waiting for more updates like this.  
<https://www.gologica.com/course/kubernetes-administrator-training/>

[↩](#) Reply to *priyanka dasari*

First Name\*

Last Name

Email\*

Website



MONITORING

WHY US

PRIC

protected by reCAPTCHA

[Privacy](#) - [Terms](#)

SUBMIT COMMENT

# Get automated cloud monitoring for instant insights.

## START MONITORING FREE

### MONITORING

[AWS monitoring](#)

[Kubernetes monitoring](#)

[Serverless monitoring](#)

[Azure monitoring](#)

### WHY BLUE MATADOR

[Blog](#)[eBooks](#) [Kubernetes](#) [Cloudwatch](#)[Docs](#)[Integrations](#)[Developers](#)

**COMPANY**

[About us](#)[Careers](#)[Contact](#)

**SUPPORT**



START FREE

REQUEST DEMO