

This DevOps interview questions and answers pdf file is a repackaging of a popular article written by Cameron McKenzie ([@cameronmcnz](#)) and published on [TheServerSide](#).

DevOps interview questions and answers

Looking for [GitHub, Jenkins and DevOps interview questions](#) that experienced programming experts planning to get leverage over other employment aspirants and candidates will find challenging? Hoping to land that DevOps position and you require some assistance to prep? Or on the other hand are you are a DevOps manager interviewing potential candidates? These basic DevOps interview questions and answers for experienced programming professionals will test the grit of even the most exceptional CI/CD specialists.

Tough DevOps interview questions

An ever increasing number of companies hope to see a reference to DevOps on the resumes of candidates applying for DevOps positions. These sample DevOps interview questions are the ones candidates get asked most often.

Not so basic DevOps interview questions

The accompanying is the unanswered rundown of the most generally asked DevOps interview questions:

1. Name five DevOps tools that companies should consider embracing while performing a DevOps transition.
2. What is continuous integration (CI), and what is its motivation?
3. What does it mean to shift left in DevOps?
4. The acronym CAMS is regularly used to portray the center principles of DevOps. What does it mean?
5. Is a culture change a prerequisite or an aftereffect of doing DevOps?
6. Name three vital DevOps KPIs.
7. Describe three DevOps antipatterns.
8. What are advantages of executing DevOps automation?
9. Name two prominent Java frameworks for creating microservices.
10. What is constant delivery (CD)?

Sample DevOps interview questions answered

Despite whether you concur or not with my answers to these sample DevOps interview questions, be ready to answer them in your own specific manner and in your own voice.

1. Name five important DevOps tools that organizations should consider adopting when undergoing a DevOps transition.

I would answer the first of these sample DevOps interview questions with an assertion that there are five key arenas in which tools can assist in a DevOps transition:

- configuration management
- source code management
- CI
- containerization
- collaboration

Chef, Puppet and Ansible are all highly capable configuration management tools.

GitHub and Jenkins DevOps interview questions

[Git](#) and a cloud-based platform, such as [GitHub or GitLab](#), make the management of distributed source code easier.

Jenkins is perhaps the most popular [CI server](#) on the market, although there are some highly competent competitors, such as Concourse CI and Atlassian's Bamboo.

For containerization, Docker tends to get the most attention, but Rkt and LXD have a significant following as well.

In terms of collaboration, Jira is an excellent tool for Agile team collaboration.

2. What is CI, and what is its purpose?

The key to answering the second of these sample DevOps interview questions is concentrating on CI and not falling into a description of CD or continuous deployment.

CI is the process of compiling the entire code base every time a member of the software development team checks code into the shared source code repository.

If a team member checks in code that causes compilation to fail, then the developer has broken the build. If this is the case, other developers won't be able to synchronize with the shared source code repository without introducing compilation errors into their own local workspace. This means collaborative and shared software development cannot go forward. As such, when a [CI build](#) breaks, it is imperative that the problem is corrected immediately.

A CI process often includes a suite of unit, integration and regression tests that run every time compilation succeeds. If any of these tests fail, the build is considered *unstable* but not broken. An unstable build isn't uncommon, especially during an Agile sprint, where development is ongoing.

3. What does it mean to *shift left* in DevOps?

If you were to graph the traditional software development lifecycle on a piece of paper, the left side of the graph would likely include tasks such as design and development, while the right side would likely include user acceptance, stress testing and production staging. To *shift left* in DevOps implies a desire to take many of those tasks that often happen toward the end of the application development process and move them into earlier stages.

In some cases, this might mean to incorporate [static code analysis](#) routines in every build. Another way to perform a DevOps shift left is to create production-ready artifacts at the end of every Agile sprint so that users and stakeholders can get incremental updates on how development is progressing. Proper DevOps means shifting left as much as possible.

4. The acronym CAMS is often used to describe the core tenets of DevOps. What does it stand for?

The CAMS acronym stands for culture, automation, measurement and sharing.

5. Is a culture change a requirement or a result of doing DevOps?

The term *culture* describes the processes and practices people use. And culture can only change when people change the processes and practices they routinely employ. The introduction of [distributed version control system tools](#), such as Git, deployments with Docker instead of the use of monolithic application servers, and the use of CI tools, such as [Jenkins or Maven](#), in the software development lifecycle will result in a culture change. But these changes are the result of the introduction of new DevOps tools, not a prerequisite.

Any [cultural change](#) that DevOps precipitates is an output of using new tools and adopting new processes. It is not an input.

It should be noted that many [DevOps evangelists](#) would take serious umbrage with this sample DevOps interview question and answer. Be prepared to argue and defend both sides of the culture argument.

6. Name three important DevOps KPIs.

There are many KPIs that can be used to measure DevOps success. Three of the most common DevOps KPIs are:

- mean time to failure recovery
- deployment frequency
- percentage of failed deployments

7. Describe three DevOps anti-patterns

One DevOps anti-pattern is to reduce the silo-based isolation of development and operations with a new DevOps team that silos itself from other parts of the organization.

The first DevOps anti-pattern mentioned leads to the second one, which is failing to include all aspects of the organization in an ongoing DevOps transition. Organizations embarking upon a DevOps transition should have some type of DevOps center of excellence that makes sure all tiers of the organization -- from IT executives to project managers to members of the security and hardware provisioning teams -- embrace and employ DevOps practices.

Another common anti-pattern is not defining KPIs at the start of a DevOps transition. Without identifying important DevOps KPIs and creating a baseline at the start of the journey, it is impossible to evaluate success and identify failures.

8. What are benefits of implementing DevOps automation?

The core benefit to DevOps automation is that it takes the possibility for human error out of the CD equation.

Furthermore, when automation replaces manual tasks, those tasks become repeatable and predictable. As such, when something goes wrong in an automated process, it is easy to identify and correct the problem. These types of refinements are said to harden the DevOps process, leading to more reliable and robust systems.

Another benefit to DevOps automation is that it can remove bottlenecks from the CI pipeline. This increases deployment frequency and reduces the number of failed deployments, both of which are important DevOps KPIs.

9. Name two popular Java development frameworks for creating microservices.

There are a number of Java frameworks for microservices development on the market today; two of the most popular are [Spring Boot](#) and Eclipse MicroProfile. Don't fall for the [Java microservice myth](#) that Spring Boot and the Eclipse MicroProfile are the only players in the game though. There are many competitors.

10. What is CD?

CD expands upon the ideas of CI. With CI, if you can compile a build, it's in its optimum state. In contrast to CI, CD asserts that there should be a main branch of ongoing source code integration that you can package, deploy and release into production at any given point.

Author Martin Fowler [contends that](#) you are doing CD if:

- Your software is deployable throughout its lifecycle.
- Your team prioritizes keeping the software deployable over working on new features.
- Anyone can get fast, automated feedback on the production readiness of their systems whenever somebody makes a change to them.
- You can perform push-button deployments of any version of the software to any environment on demand.

CI is a part of CD, but so is the creation of deployable [container-packaged executables](#). So, too, is the use of a comprehensive suite of tests that evaluates software quality, performance under load and user acceptance.

Sample DevOps interview questions and answers summary

This certainly isn't an exhaustive list of every question you might encounter, but many of these sample DevOps interview questions and answers are likely to be encountered in a technical interview -- especially if DevOps is part of the job requirement. If you understand the answers provided here and you can respond in your own words, you are well on your way to landing that new job as a DevOps

engineer.