

**1. DevOps has become a new phenomenon for many top software companies such as Google, Amazon, and Microsoft to name a few. However, not all product architectures lend themselves to being DevOps friendly. Discuss some of the challenges in adopting DevOps and possible remedies to those problems.**

Since DevOps' achievable cycle time depends on the environmental constraints and deployment model, it can be really hard to actually get the software delivered to use as the real product. And working with safety-critical system can be difficult too.

One of the solution is designing the architecture to have two parts. One for regular daily operation and another half for DevOps updates. DevOps will try to tie legacy code and architecture with regular delivery.

**2. Build tool is one of the key components in achieving a good DevOps environment. Many organizations have already moved from using Ant to Maven and in recent times to use Gradle/Rake. Discuss the limitations of Ant and Maven and explain how Gradle/Rake fix those problems.**

Apache Ant uses XML, which isn't a procedural programming language. Maven has inflexibility when custom operations must be defined. And there is a problem due to the external automatic dependencies.

Rake and Gradle solve above problems using a programming-language-based tool to build applications. With Rake, code lines are written in Ruby; with Gradle, configuration employs a Groovy-based domain-specific language (DSL).

**3. Jenkins, TeamCity, and Bamboo are the three most widely used Continuous Integration servers. Discuss what are some of the key features these CI infrastructures offer and how they are different.**

Jenkins is an open source, Java based system that contains a lot of plug-in options. It's easy to find support. But the UI is outdated and ugly.

TeamCity is based on Java and has good Java Support. And it also offers .NET support. But it's licensed, it does have a free edition for small projects.

Bamboo is developed from Atlassian and can integrate with other Atlassian tools.

**4. Chef, Puppet, and Ansible are the three most popular tools for achieving Infrastructure as Code. Explain what Infrastructure as Code means and then discuss the similarities and differences among these three tools.**

Infrastructure as Code is a process of maintaining and provisioning infrastructure and configuration through definitions files, which allows developers to code and push to server.

Similarities: They all aim for improving the application life cycle and to achieve "Infrastructure as Code".

Differences: Puppet is based on Ruby but uses a DSL similar to JSON, it describes the system's desired end state. Chef writes a recipe on Ruby based DSL, it can be used as a client-server tool or an isolated installation.

**5. What are the key differences between Virtualization and Containerization? When would you choose Containerization and when would you opt for Virtualization? Please investigate more on your own on this topic before answering this question.**

Virtualization is used to recreate production environments on development machines, it focuses on security and multitenancy.

Containerization is focuses more on migration between cloud providers.

Containerization should be used when it requires compatibility between different providers. Virtualization should be used when the exact production development environment need to reused.

**6. The authors discuss why logging is important for DevOps and identify Loggly, New Relic, and GrayLog2 as some of the key tools that help build intelligence around log data. There is one more widely used tool called Splunk. Find out what Splunk does in terms of log intelligence/management and how it helps doing DevOps.**

Splunk analyzes the machine data and collects the index log. It will help DevOps to keep track of the developing process and figure out the status of development.

**7. Discuss some of the key features of Nagios, New Relic, and Cacti as monitoring tools.**

Nagios is a open source tool for monitoring IT infrastructures such as end-user stations, IT services, and active network components. It has an easy navigating web interface with dashboard that includes a high-level overview of hosts, services, and network devices. It also provides trending and capacity-planning graphs.

New Relic is an interface for Web apps and monitors native mobile applications for iOS and Android. It allows to monitor apps in different states and customizes dashboards to integrate extra monitor features.

Cacti is a Web-based system with an easy navigating interface. It allows high level customization for configuration. Users can build a tree view mode graph.

**8. Explain a typical DevOps workflow if your application uses AWS tools. [Hint: Figure 2 ]**

Development team uses Rake to develop and commit the code, then push to cookbook and app repository. Then Jenkins notifies whether the new version is a successfully build or failure. All the new update also go AWS and follow Chef Amazon OpsWorks. Then Nagios monitor cloud.

**9. What are the four key related challenges that organizations face in transitioning to DevOps culture?**

- Separate big architectures and features into produce and deploy independently multiple small chunks.
- Maintaining a constant working configuration and build environment that works with different versions and dependencies.
- Changing from legacy application life-cycle management or product life-cycle management environment to purpose-built development and production environment.
- Combine traditionally siloed cultures of development with operations.

**10. We have traditionally believed that knowing how to write algorithms in a mainstream language, designing functional aspects of a system, refactoring and testing developed features are some of the key skills that makes up a good software engineer. This belief is not a reality anymore. What are other skills that a software engineer must possess to be competitive in the market in the current time and why?**

Software engineers must all take stack developer approach, in which they all take responsibility for the testing and developing environment. Good database administration and testing are required too. Most importantly, excellent communication skill is needed since collaboration with other team members are required.