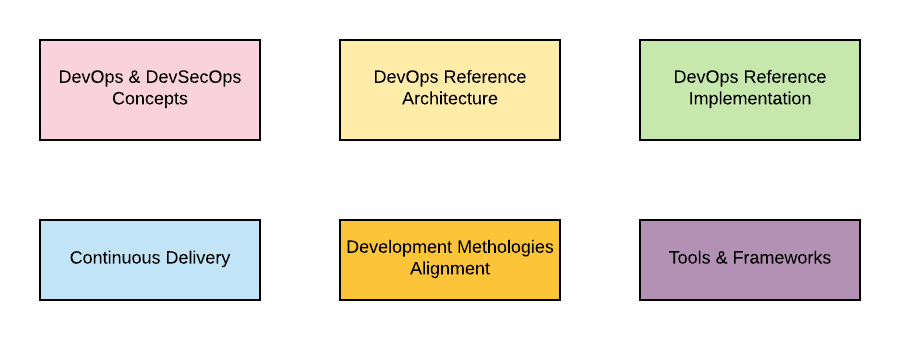
# 15 Tricky DevOps Architect Interview Questions & Answers



DevOps and DevSecOps are two sides of the same coin. They both share some goals, but they also have their differences. It is important to understand what each one means so that you can implement them properly in your organization. In this post, you will learn about some of the **questions (and answers)** which could be asked in the [**DevOps**](https://vitalflux.com/category/devops/)**Architect interview**. The following are some of the topics you might want to cover for doing**interview preparation for DevOps Architect** position:

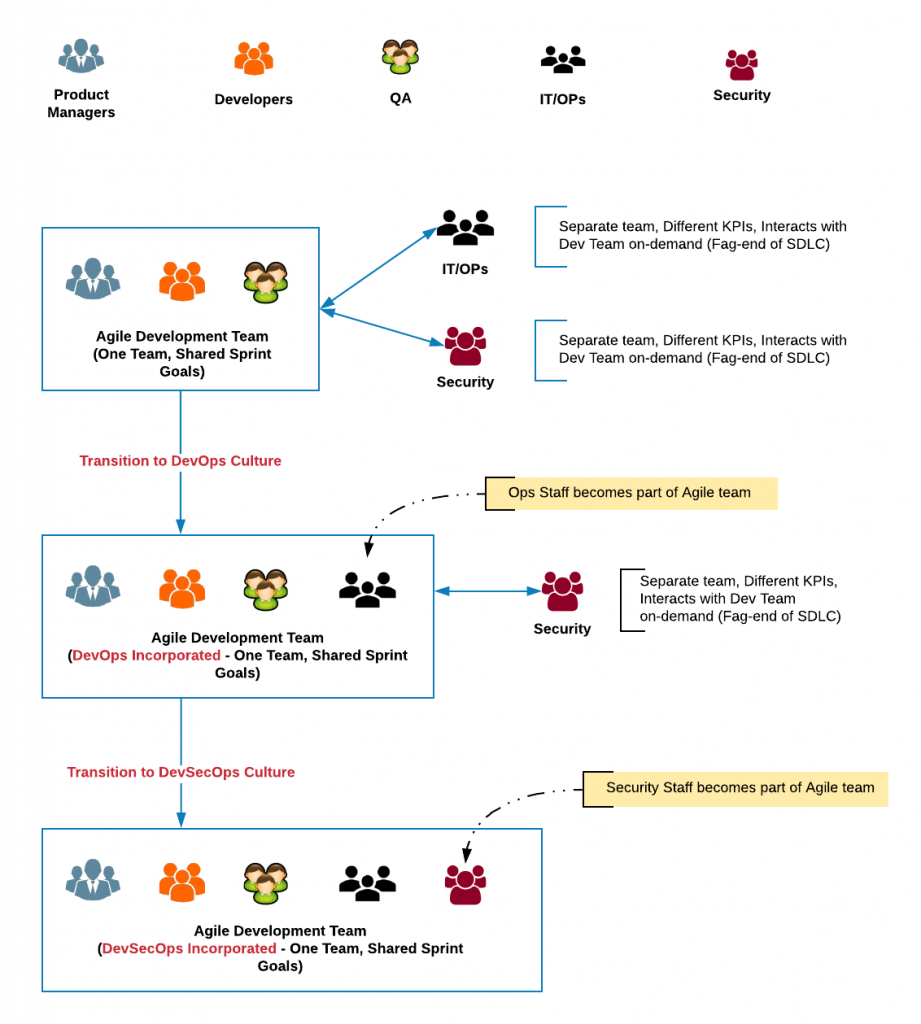
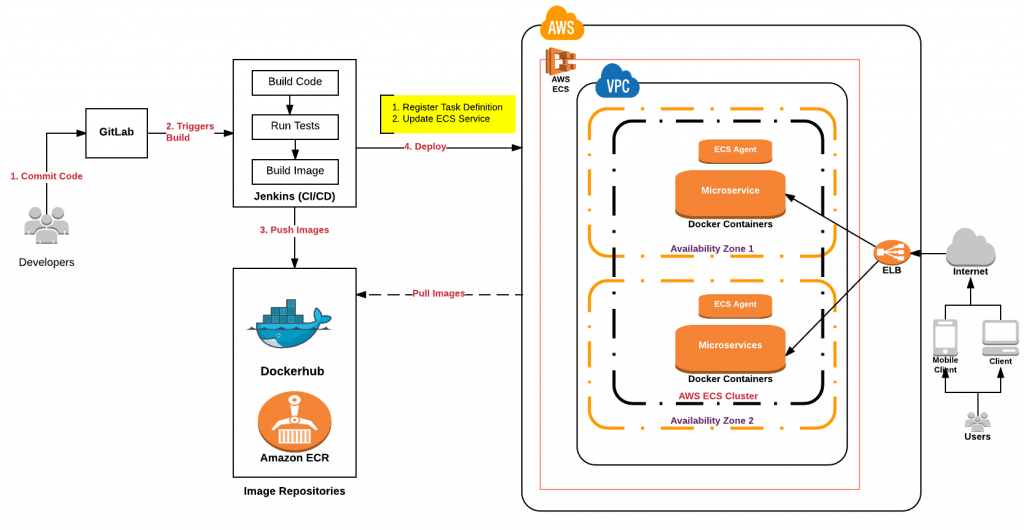
1. DevOps & DevSecOps concepts
2. Setting up DevOps implementation
3. DevOps reference architecture
4. DevOps reference implementation
5. Continuous delivery concepts & reference architecture
6. Technologies (Tools s& Frameworks)

Here is a related diagram on topics that are useful for DevOps architect interview preparation.



## DevOps Architect Interview Q&A

The following are some of the interview questions and answers for DevOps Architect:

1. **What is DevOps or DevSecOps?**  
   **Ans**: DevOps or DevSecOps can be defined as a **philosophy for software development** with the primary goal of **reducing mean-time-to-change** (MTTC) and **mean-time-to-recovery** (MTTR) for moving new features or bug fixes into production for incorporating customers’ feedback in a faster manner.
2. **What is the difference between DevOps and DevSecOps?**  
   **Ans**: The primary difference is that DevOps requires Developers & IT/OPs staff members to work together as one team with shared sprint goals while DevSecOps requires Developers, IT/Ops, and Security staff to work together as one team with shared sprint goals. In simpler words, DevOps is the marriage of development and operations, whereas DevSecOps is an extension of DevOps that includes security practices. The goal of both is to optimize collaboration between IT professionals so they can deliver value faster. Another key difference is the inclusion of security-related tools at the different stages of SDLC for the automation of security testing.
3. **How to get started with DevOps initiatives?**  
   **Ans**: There are different steps such as assessment, gaps analysis, maturity model, project implementation roadmap, etc which are required to get started with DevOps or DevSecOps initiatives in your organization. One of the key aspects of getting started with DevOps or DevSecOps  is assembling the right team. This team should have expertise in both development and operations, as well as experience with DevOps tools and processes. If you don’t have all of the required skills in-house, you may need to bring in outside consultants. Another important aspect of getting started with DevOps or DevSecOps is understanding your current infrastructure. This means knowing things like how many servers you have, what operating systems they’re running, and who manages the configuration of each one. You’ll need this information for your plan, as well as for testing any new processes or tools that you develop. The greater details could be found on this page: [How to get started with DevOps or DevSecOps in your organization?](https://vitalflux.com/steps-start-devops-devsecops-initiative/)
4. **How would you go about measuring the success of DevOps adoption across the organization?**  
   **Ans**: The primary method of measuring DevOps success is to measure some of the following:  
   A. **Mean-Time-to-Change (MTTC)**: MTTC is used to measure how quickly an organization can adapt to changes. MTTC is calculated by dividing the time it takes to make a change by the number of changes made.  
   B. **Mean-Time-to-Recovery (MTTR)**: MTTR (Mean Time to Recovery) is a metric that measures how quickly problems are resolved. MTTR can be used to measure the success of DevOps initiatives by comparing the pre-DevOps and post-DevOps MTTR values.  
   C. Frequency of release  
   D. **Mean-time-to-incident (MTTI)**: MTTI is a metric that measures how quickly problems occur.
5. **Could you define a sample DevOps maturity model?**  
   **Ans**: The following could be some of the different stages across which organization could plan to progress in order to achieve success in DevOps or DevSecOps rollout:  
   A. **Ad-Hoc**: Perform ad-hoc tasks such as build automation, environment provisioning automation on-demand.  
   B. **Organized or Planned**: Plan the DevOps project roadmap; As part of planning, choose development methodologies, team, tools & frameworks to be used, required teams’ training/mentoring, etc.  
   C. **Align & Assess**: Implement proof-of-concept for one or two teams covering key aspects of DevOps implementation. Learn lessons and change the plan appropriately to be used for major rollout across different teams in the organization.  
   D. **Institutionalization**: Roll out the DevOps/DevSecOps implementation across different teams in the organization.  
   E. **Continuous Improvement**: Learn the lessons and continuously improve the DevOps processes.
6. **Many talk about setting up of DevOps team for implementing DevOps in their organization. What are your thoughts in this relation?**  
   **Ans**: A separate DevOps team can’t be set up. This is because DevOps is a software development philosophy requiring Developers & IT/OPs staff members to work as a single team with shared Sprint (Agile SCRUM) goals. **Thus, the product team gets transformed to work on DevOps or DevSecOps philosophy**. Here is a diagram representing the aspects related to product teams getting transformed to work based on DevOps philosophy.
7. **What could be some of the responsibilities of DevOps/DevSecOps Architect?**  
   **Ans**: The following could be some of the responsibilities of a DevOps Architect:  
   A. Define the architecture of the continuous delivery pipeline  
   B. Design and implement self-service provisioning capabilities  
   C. Define and deploy configuration management processes  
   D. Put up a plan for DevOps implementation across different teams  
   E. Oversee the DevOps implementation. Work with developers and operations staff to optimize the application delivery process  
   F. Lay down the technologies to be used in DevOps implementation. These technologies could be related to a different aspect of automation including build, deployment, environment provisioning etc.  
   G. Implement security controls throughout the software development life cycle  
   H. Integrate with monitoring tools to provide visibility into system health  
   I. Manage and monitor cloud infrastructure deployments
8. **Could DevOps be achieved without setting up systems and processes for continuous delivery?**  
   **Ans:**Yes. DevOps with the primary goal of reducing MTTC and MTTR could be achieved using ad-hoc automation of build, deployment, and environment provisioning processes.
9. **Many times, DevOps is related to Docker technology? What are your thoughts in this relation?**  
   **Ans**: DevOps does not require Docker technology to be implemented to achieve its primary objective. However, the very fact that cloud-native apps with Docker as underlying technology could be seamlessly used for automated provisioning of environments, automated deployment across different environments including cloud services makes it a very useful technology for DevOps adoption.
10. **What are some of the technologies used for DevOps implementation?**  
    Ans: The following represents some of the technologies which could be used for DevOps implementation:  
    A. Configuration management tools such as Chef, Puppet, Ansible, etc for automated provisioning of environments  
    B. Continuous integration tools such as Jenkins  
    C. Code repositories such as Git  
    D. Container technologies such as Docker, Kubernetes etc  
    E. Artifacts repository such as Nexus Repository Manager, JFrog Artifactory etc.
11. **Describe different stages of the continuous software delivery model?**  
    Ans: The following are some of the stages of the continuous delivery model:  
    A. **Code check-in** by developers leading to activation of WebHooks triggering CI Builds  
    B. **App compilation & building** of deployment artifacts  
    C. **Deployment artifacts upload** in artifactories  
    D. **Test environment provisioning** and  
    E. **App artifacts deployment** across different test environments  
    F. **App artifacts production deployment**
12. **How would you go about architecting a continuous delivery solution using Docker?**  
    **Ans**: The following are some of the key points in relation to architecting a continuous delivery solution using Docker:  
    A. Align the apps to be dockerized or containerized. This may require consideration of microservices architecture.  
    B. Consideration of development and test environment provisioning using Docker  
    C. Consideration for setting up Docker container registry on-premise or cloud (for example, AWS ECR)
13. **Could you architect a continuous delivery solution for delivering software changes on to AWS cloud?**  
    **Ans**: A detailed write-up on continuous delivery on AWS ECS and AWS EC2 could be found on one of my Github pages, [Continuous Delivery of Microservices on AWS.](https://github.com/eajitesh/Continuous-Delivery-Microservices-AWS) Here is the sample diagram representing the continuous delivery reference architecture for AWS.
14. **What is the difference between Agile and DevOps or DevSecOps way of software development?**  
    **Ans**: The primary difference between traditional Agile and DevOps/DevSecOps team is the inclusion of IT/OPs and Security staff members in the Agile teams thereby having sprint goals comprising of stories for both IT/OPs and security staff members. More importantly, the definition of DONE changes. In Agile SCRUM development, the **definition of DONE is creating demonstrable software/artifacts at the end of the sprint**. In Agile SCRUM team working based on DevOps/DevSecOps philosophy, the **definition of DONE changes to creating demonstrable software/artifacts in an environment resembling production** at the end of the sprint
15. **Describe different stages of DevSecOps and related tools which could be used at each of these stages?  
    Ans**: The following are different stages of SDLC where security tools could be used for security-related automation in the continuous delivery cycle:  
    A. Source code scanning: [CheckMarx](https://www.checkmarx.com/), [Gitrob](https://github.com/michenriksen/gitrob)  
    B. Test & scans: [Evident.IO](https://www.paloaltonetworks.com/products/secure-the-cloud/evident.html), [GauntLT](https://gauntlt.org/), [Contrast](https://www.contrastsecurity.com/)  
    C. Artifacts: Sonatype, [Nessus](https://www.tenable.com/products/nessus/nessus-professional), [Tanium](https://www.tanium.com/)  
    D. Deploy: [INSPEC](https://www.inspec.io/), Beaker  
    E. Monitoring: [Splunk](https://www.splunk.com/), [Evident.io](https://www.paloaltonetworks.com/products/secure-the-cloud/evident.html), [FireEye](https://www.fireeye.com/), [Metasploit](https://www.metasploit.com/)