Here is the list of DevOps Interview Questions which are recently asked in Infosys company. These questions are included for both Freshers and Experienced professionals.

**1. What is DevOps?**

Hey there! DevOps stands for Development and Operations. It's a set of practices that combines software development and IT operations to shorten the systems development life cycle and provide continuous delivery of high-quality software. DevOps aims to increase the speed of software delivery while maintaining quality. It involves collaboration between development teams (Dev) and IT operations teams (Ops) to automate and streamline the processes of building, testing, and deploying software. This approach helps organizations to deliver applications and services at a faster pace. Let me know if you need more details on this!

So, if you are developing whatever code, you develop today it should be build and deploy. That is called DevOps.

Devops is a process of improving the application delivery by ensuring this proper automation with a code quality ensuring there is a continuous monitoring and continuous testing in place.

**Devops Tools: -**

Devops is a procedure. If you have to follow this procedure we have to do

1. Develop
2. Build
3. Deploy & Scanning
4. Test

All the above points we are doing manually. This manual process is difficult. So, to automate this process we are using DevOps tools.

Devops tools are used because of to reduce human errors.

Multiple environments in DevOps:

1. Dev
2. QA
3. Preprod
4. Prod

Some popular DevOps tools that are commonly used in the industry include:

1. \*Continuous Integration/Continuous Deployment (CI/CD) Tools:\*

- Jenkins

- GitLab CI/CD

- CircleCI

2. \*Configuration Management Tools:\*

- Ansible

- Chef

- Puppet

3. \*Containerization Tools:\*

- Docker

- Kubernetes

4. \*Monitoring and Logging Tools:\*

- Prometheus

- ELK Stack (Elasticsearch, Logstash, Kibana)

- Grafana

5. \*Collaboration Tools:\*

- Slack

- Microsoft Teams

These tools help automate various stages of the software development lifecycle, improve collaboration between teams, and ensure the smooth delivery of software products. Let me know if you need more information about any specific tool or aspect of DevOps!

**2. Why do we need DevOps?**

Hey! We need DevOps because it helps organizations to deliver software and services more quickly, efficiently, and reliably. By implementing DevOps practices, teams can automate manual tasks, reduce errors, and increase collaboration between development and operations teams. This leads to faster development cycles, quicker resolution of issues, and ultimately, improved customer satisfaction. DevOps also promotes a culture of continuous improvement and innovation within an organization. Overall, DevOps is essential in today's fast-paced digital world to stay competitive and deliver high-quality software products efficiently. Let me know if you have any more questions about this!

1. Faster releases
2. Less defects

Devops addresses the above two features, by

1. Number of testing cycles are more &
2. Daily development, Building and testing happens.
3. Daily process and co-ordination.

**3. Mention the key aspects or principle behind**[DevOps](https://www.credosystemz.com/training-in-chennai/devops-training-chennai/)**?**

To fuel a DevOps culture, we have to build robust tooling, also. That means investing up front in five key areas: abstraction, composability, automation, orchestration, and idempotency. Together, these concepts allow the sharing work at every level of the pipeline.

**The key aspects of the principles behind DevOps include:**

Culture, Automation, CI, CD, Monitoring and Feedback, Infrastructure as a code Iac

1. \*Culture:\* Encouraging collaboration and communication between development and operations teams to faster a shared responsibility for software delivery.

2. \*Automation:\* Automating manual tasks in the software development lifecycle to increase efficiency, reduce errors, and accelerate the delivery process.

3. \*Continuous Integration (CI):\* Integrating code changes frequently to a shared repository, allowing early detection of issues and ensuring that the codebase is always in a deployable state.

4. \*Continuous Deployment (CD):\* Automating the deployment process to deliver code changes to production quickly and reliably.

5. \*Monitoring and Feedback:\* Implementing monitoring tools to track the performance of applications in real-time and gather feedback for continuous improvement.

6. \*Infrastructure as Code (IaC):\* Managing and provisioning infrastructure through code to ensure consistency and scalability.

By focusing on these key aspects, organizations can embrace the DevOps culture and practices to enhance collaboration, streamline processes, and deliver high-quality software efficiently. Let me know if you have any more questions about DevOps principles!

**What is CI/CD:**

Whenever developers write code, we integrate all the code of all developers at any point of time and we build, test and deliver/deploy to client. This is called CI/CD.

Code-build-Integration-QA or UAT: This process is done with manually that is called Continuous Delivery.

Code-build-Integration-QA or UAT: This process is done with Automation that is called Continuous Deployment.

**4. List out some of the popular tools for DevOps?**

The popular tools for DevOps are:

* Git. Code, Build.
* Gradle. Build.
* Selenium. Test.
* Jenkins. Build, Test, Deploy.
* Puppet. Deploy, Operate.
* Chef. Deploy, Operate.
* Docker. Build, Deploy, Operate.
* Kubernetes. Build, Deploy, Operate.

**5. What is a version control system?**

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that the help software teams manage changes to source code over time.

**6. What is Git and explain the difference between Git and SVN?**

The difference between Git and SVN version control systems is that Git is a distributed version control system, whereas SVN is a centralized version control system. Git uses multiple repositories including a centralized the repository and server, as well as some local repositories.

**7. What language is used in Git?**

JavaScript continues to be the most popular programming language on GitHub, while Python is now the second most popular, followed by Java and the fast-growing the TypeScript community.

**8. What is SubGit?**

SubGit is stress-free tool for migrating from SVN to Git. It can be used with any Git server whether it is Github, Gitlab, Gerrit, or Bitbucket.It is developed by Tmate software. Answered by devquora. It is software for the migrating SVN to GIT.

**Free PDF : Get our updated DevOps Course Content pdf**[Download Now](https://www.credosystemz.com/wp-content/uploads/2023/07/Devops-Training-Course-Content.pdf)

**9. How can you clone a Git repository via Jenkins?**

There are two ways to clone the project from the Github. Create a new Jenkins job called ‘Clone-with-https’, move to the “Source Control Management” setting and choose “Git” options if you can’t see the Git options that mean ‘GitHub’ plugin wasn’t installed in the Jenkins machine.

**10. What are the Advantages of Ansible?**

The primary benefit of Ansible is it allows IT administrators to automate away the drudgery from their daily tasks. That frees them to focus on the efforts that help deliver more value to the business by spending time on more important tasks.

**11. What’s the difference between Ansible Playbook and Roles?**

Ansible, the role is the primary mechanism for the breaking a playbook into multiple files. This simplifies writing complex playbooks, and it makes them easier to reuse. The breaking of playbook allows you to logically break the playbook into reusable components.

**12. How do I see a list of all the ansible\_ variables?**

There are Three sources of variables in Ansible are:

* Variables gathered from facts.
* You can get them by running command: ansible -m setup hostname.
* But obviously you know what they are.

**13. What is Docker?**

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure within the same ways you manage your applications.

**14. What is Docker image?**

A Docker image is a file used to execute code in a Docker container. Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run.

**15. What is Docker Container?**

Docker streamlines the development lifecycle by allowing developers to work in the standardized environments using local containers which provide your applications and services. Containers are great for continuous integration and continuous delivery workflows.