1. What does DevOps stand for?

DevOps is a Combination of two words Development and Operation. It basically signifies the methodology used for faster integration, development and deployment of any application.

1. Can you name three popular DevOps tools? (easy)

Three Tools that I use the most are

1. Git: A version control system that helps teams collaborate on software development, track changes in the software after each commit and to automate the software delivery process. I personally have used Git and GitHub repository in my CI/CD (continuous integration and Continuous delivery) pipeline as a storage space for all the files.
2. Jenkins: I have used Jenkins (as it is an open-source automation server) for creating a pipeline. It allows me to automate the process of continuous integration.
3. Docker: A containerization platform that allows developers to build, package and deploy application as containers.

1. What is the primary purpose of implementing DevOps? (easy)

In the real-world production of a software mainly two kinds of people work together the development personals and the deployment personals. Usually, all the errors or difficulties faced by the client is reported to the deployment/operational team but as they were not there while development process, they cannot actually fix the bug themselves and needs help from the development team. This process usually requires a lot of time and thus DevOps is used to reduce this time.

1. Which scripting language is commonly used in DevOps? (easy)

Python.

1. What is the primary objective of Continuous Integration (CI)? (easy)

Continuous Integration particularly deals with updating the software after each commit in the shared repository.

1. What are the key components of a typical CI/CD pipeline? (medium)

According to me key components of a CI/CD pipeline are:

1. Version control of software
2. Component to build the application
3. Continuous integration after each commit to the shared software.
4. Automatic testing
5. Creation of an artifact repository
6. Deployment of the software (deployment is usually not recommended)
7. What is the difference between blue-green deployment and canary deployment? (medium)

In blue green development two identical development environments are made namely green and blue. we keep our currently deployed application running on the green part and create new space for deployment of new application in the blue part so that the already running application is not damaged. It ensures zero downtime keeping userbase at risk.

In canary development pattern we roll out new application as an initial testing phase in a small percentage of a single running environment. It ensures that user base is not affected but the application may go down if anything unfriendly happens.

1. How does infrastructure as code (IaC) help in DevOps practices? (medium)

It is a process of managing infrastructure as a code which would otherwise be done manually. IcA basically provides configuration files that contain infrastructure specification of the client that makes it easier to edit and distribute configuration.

1. What are the advantages of using Docker in a DevOps environment? (medium)

There are multiple advantages of docker like

1. It makes out application portable.
2. It provides application environment (all the required libraries and other requirements are stored in a single docker image only)
3. We don’t need to worry about the hardware specification and the presence of required software on a particular computer
4. We can deploy different versions of application in a container.

1. How does monitoring and observability play a role in DevOps? (medium)

By monitoring an application, we can detect the errors in it and can resolve them before they cause a major damage. Observability measures the internal states of a system by observing its output. Thus we can predict the bugs in a system.

1. What is the role of version control systems in a DevOps workflow? (medium)

If ever we roll out a new version of a system and there occurs an error in its working the version control system plays a very important role by providing the previous version (the correct version of the system). It also helps developers to maintain the agility of development process.

1. Explain the importance of communication and collaboration in DevOps. (medium)

It is only communication and collaboration between the Development and the operational team that helps a company get better results of it projects. Through better communication both the teams can get better feedbacks and problems can be solved effectively.

1. What is the purpose of a build system in the context of DevOps? (medium)

Build systems create a compiled version of the program. They make an artifact out of the program inserted into them. Programmers can test their programs with the help of build systems. Build systems automates the systems of building and packaging software and thus ensures that code changes are integrated and tested frequently.

1. Can you describe the role of configuration management in a DevOps pipeline? (medium)
2. How does a microservices architecture benefit from DevOps practices? (medium)
3. What are some common challenges faced when implementing DevOps in an organization? (medium)
4. What is the purpose of using a staging environment in a DevOps pipeline? (medium)
5. How do feature toggles contribute to the continuous delivery process? (medium)
6. What are the benefits of using a distributed version control system in DevOps? (medium)
7. Explain the difference between continuous deployment and continuous delivery. (medium)
8. How do you ensure security is maintained throughout the DevOps pipeline? (hard)
9. Describe the importance of measuring and improving key performance indicators (KPIs) in a DevOps environment. (hard)
10. How do you manage dependencies and avoid conflicts in a multi-team DevOps environment? (hard)
11. Explain the concept of "shift-left" in terms of security and testing in a DevOps pipeline. (hard)
12. Discuss the role of chaos engineering in improving the resiliency of a DevOps pipeline. (hard)