**1. What is meant by DevOps, and what is its purpose?**

Sample answer:

DevOps is a set of practices, tools, and philosophies used to improve communication and collaboration across different teams when delivering new software. In short, it bridges the gap between developers and IT staff.

The goal of DevOps is to integrate automation and monitoring at all stages of the software development lifecycle. This improves the speed at which new systems and applications can be delivered.

2. What are the main types of DevOps tools?

Sample answer:

Organizations use several different types of DevOps tools, including:

Version control tools, such as Git

Continuous integration tools, such as Jenkins

Configuration management tools, such as Puppet

Containerization tools, such as Docker

Cloud tools, such as AWS

Monitoring tools, such as Nagios

3. What are some of the key skills of a DevOps engineer?

Sample answer:

DevOps encompasses several different processes within the software development lifecycle. As such, DevOps engineers must have a wide range of soft and hard skills, including:

A solid understanding of DevOps

Communication and teamwork skills

Programming skills

Software security skills

Cloud technology skills

4. What are the differences between DevOps and Agile?

Sample answer:

DevOps arose largely as an alternative to Agile workflows like Scrum. The key differences include:

DevOps brings together all teams involved in software development and maintenance, whereas Agile typically siloes specific tasks to different teams

DevOps emphasizes consistency, stability, and automation, whereas Agile focuses on adaptability and iterative development

DevOps encompasses the full flow of software from ideation to delivery and maintenance, whereas Agile generally ends with the completion of code

5. What is the role of configuration management?

Sample answer:

Configuration management (CM) is the process in which software systems are automated, updated, monitored, and managed. It helps reduce the risk of unexpected system failures and offers greater agility for personnel working across the DevOps strategy.

6. What is the role of continuous integration?

Sample answer:

Continuous integration (CI) is the process in which the integration of code changes into a software project is automated. It allows developers to merge code changes into a single repository whenever a new change is made, thus improving collaboration and transparency.

7. What is the role of continuous testing?

Sample answer:

Continuous testing involves regular testing of software at every stage of the development lifecycle to ensure any bugs are fixed rapidly. It provides continuous feedback so that developers can evaluate software quality throughout the delivery process.

8. What is the role of automation testing?

Sample answer:

Automation testing is the process of automatically reviewing and validating scripts to verify the functionality of the program. It occurs with minimal human intervention so that it can be performed in a more time-sensitive way.

9. What is the role of continuous monitoring?

Sample answer:

Continuous monitoring refers to the regular monitoring of software at every stage of the development lifecycle to ensure the performance, reliability, and compliance of the application and infrastructure. This provides transparency and facilitates the early detection of any issues.

10. What is meant by branching?

Sample answer:

Branching is a technique used by developers within version control systems, whereby the source code is copied to make two versions that are developed separately. These branches can then be developed independently without affecting the code base, thus promoting collaboration.

11. What do you know about Jenkins?

Sample answer:

Jenkins is the most popular continuous integration and continuous delivery (CI/CD) tool on the market today. Pretty much all DevOps teams rely on Jenkins to automate parts of software development related to the build, test, and deployment of applications.

12. How are application development and infrastructure different?

Sample answer:

Application development consists of the following core operations:

Coding

Unit testing

Packaging

Deployment

Infrastructure consists of:

Provisioning

Configuration

Monitoring

Deployment

13. What do you know about Git?

Sample answer:

Git is a popular open-source version control system used to track changes in the source code, create multiple scripts, and, ultimately, improve collaboration between developers. It’s a critical tool in the coding stage of the software development lifecycle.

14. What is meant by ‘SSH’?

Sample answer:

SSH, or Secure Shell, is a network protocol for safely encrypting any data shared from a computer over a network. It creates a separate secure channel for communication and data sharing that overlays unsecured networks.

15. What are the benefits of version control?

Sample answer:

Version control systems improve the efficiency of coding. This is achieved through:

Traceability: Version control tracks changes to code from all developers, providing a clear history that helps improve the functionality of the software

Branching: Version control allows developers to work on code independently without impacting contributions from other collaborators

Error reduction: Version control helps detect the root cause of software bugs as well as any duplications for easy removal

16. What are anti-patterns? Can you name a couple?

Sample answer:

Anti-patterns are ideas that are counter-productive to the DevOps philosophy. They’ll often fix a short-term problem at the expense of a long-term goal. Some examples include:

DevOps is a process rather than a culture

DevOps is driven either by development or IT operations, rather than a combination of both

17. What is the role of cloud computing in DevOps?

Sample answer:

Cloud computing provides a centralized, scalable communication platform for DevOps teams at each stage of the software development lifecycle. It allows team members to collaborate more quickly and closely.

17 intermediate DevOps interview questions

These intermediate DevOps interview questions are perfect for mid-level roles, where candidates already have some professional experience with the DevOps framework.

intermediate DevOps interview questions

18. What are the main phases in the DevOps lifecycle?

Sample answer:

The software development lifecycle consists of planning, coding, build, testing, release, deployment, and monitoring. When applied to DevOps, there are four key phases:

Continuous Integration, which includes the coding and build stages

Continuous Delivery, consisting of the testing and release stages

Continuous Deployment, which includes the release and deployment stages

Continuous Feedback, consisting of the monitoring stage

19. What are some of the business benefits of DevOps?

Sample answer:

When implemented effectively, DevOps transforms businesses. Key benefits include:

Faster deployment times

More collaborative work environment

Improved customer experience

Earlier defect detection

Better defect resolution

Time saved by automation

20. What are the key differences between continuous delivery and continuous deployment?

Sample answer:

Continuous delivery and continuous deployment are two parts of continuous integration:

Continuous delivery deploys all code changes to a testing or production environment

Continuous deployment automatically releases new changes to customers

21. What do you know about Docker?

Sample answer:

Docker is a containerization tool used by DevOps teams during the continuous deployment stage. It packages applications and all of their constituent parts inside software containers, which are then ready for deployment across different operating systems.

22. What is meant by CAMS?

Sample answer:

CAMS is an acronym used to describe four of the key DevOps principles. These include:

Culture: DevOps is upheld by a culture of collaboration and transparency

Automation: DevOps establishes repeatable (or automated) systems to reduce errors and save time

Measurement: DevOps relies on continuous performance tracking and feedback to improve efficiency

Sharing: DevOps teams share all information and feedback, including problems

23. What are the two main types of branching strategies?

Sample answer:

There two main branching strategies available to developers are:

Release branching, which creates a branch for a potential new release

Feature branching, which creates a branch for specific features or tasks

24. What are some of the best KPIs for evaluating DevOps performance?

Sample answer:

There’s a wide range of KPIs in the DevOps field. Some of the most effective ones include:

Deployment frequency: Measures how often new features are launched

Change volume: Measures the extent to which the code is changed in new deployments

Deployment failure rate: Measures how often new deployments lead to outages or other issues

Deployment time: Measures the time taken to roll out new deployments

Mean time to recovery: Measures the average time taken to recover from a system failure

25. What do you know about Puppet?

Sample answer:

Puppet is a widely used open-source software configuration management tool that supports automated testing, continuous integration, and continuous delivery. It’s specifically designed for Windows and Linux systems.

Puppet’s infrastructure consists of the main server environment, which stores all codes, and the client environment, from which clients communicate with the main server.

26. What are the key differences between continuous testing and automation testing?

Sample answer:

Automated testing is the process of automating a set of tasks to improve speed and reduce error.

Continuous testing, on the other hand, encompasses a wider scope of applications. It focuses on business risks and sets out to achieve continuous improvements that mitigate those risks.

27. What do you know about Nagios?

Sample answer:

Nagios is a widely used open-source monitoring system that runs periodic checks on the critical parameters of servers, networks, and applications.

It keeps DevOps teams updated on things like memory, disk usage, log files, and microprocessor load. Supporting continuous monitoring, Nagios alerts technical staff of issues before they materialize and impact the end user.

28. What are the key differences between asset management and configuration management?

Sample answer:

Configuration management refers to the management of assets as entities within the software development pipeline. This involves reviewing the accuracy and reliability of different configuration items, as well as the relationships they have with each other.

Asset management, on the other hand, reviews these assets from a financial perspective. It tracks items from the point of acquisition to disposal, to determine whether they offer sufficient economic value to the organization.

29. What do you know about Ansible?

Sample answer:

Ansible is a popular DevOps automation tool used during the build, configuration, and management phases. By automating processes like testing and deployment, Ansible helps DevOps teams save time, reduce errors, and scale in pace with growing demand.

30. What is pair programming?

Sample answer:

Pair programming is a popular programming technique whereby two developers work together on the same task, sharing a single computer. Typically, one developer will write the code, while the other will review each line of code as it is typed in.

31. What do you know about Chef?

Sample answer:

Chef is a popular configuration management tool that supports continuous delivery by automating processes across several DevOps stages. Using the Ruby programming language, Chef translates system tasks into repeatable actions, known as recipes and cookbooks.

32. When are post-mortem meetings used?

Sample answer:

Post-mortem meetings are used by DevOps teams typically between the release of a new iteration and the planning stage of the next one. Team members discuss the successes and failures of the previous project, reflecting on what can be improved next time.

33.What do you know about Selenium?

Sample answer:

Selenium is a popular open-source framework for testing web applications. It’s widely used by DevOps teams, allowing them to implement automated testing without needing to use a formal test scripting language.

34. What are the benefits of cloud tools in DevOps?

Sample answer:

Cloud computing tools like AWS and Azure support the CI/CD (continuous integration and continuous development) phases of the DevOps lifecycle. They allow DevOps teams to share code, track work, and deploy software remotely, across any platform.

16 advanced DevOps interview questions

Our advanced DevOps interview questions have been hand-picked for senior DevOps roles, where candidates already have significant professional experience in the field.

advanced DevOps interview questions

35. What are the best strategies for improving DevOps performance?

Sample answer:

The performance of a DevOps initiative can be improved in the following ways:

Design a clear roadmap during the planning stage

Use efficiency-based KPIs

Use centralized storage

Perform regression testing

Focus on culture rather than processes

Upgrade the DevOps infrastructure

36. How can DevOps be implemented securely?

Sample answer:

Security practices should be incorporated into every step of the DevOps lifecycle. To implement them, teams can:

Formalize cybersecurity policies within the team

Use privileged access management

Use secrets management

Segment network access

Automate security processes

37. What is the “shift left to reduce failure” concept?

Sample answer:

The term “shift left to reduce failure”, or simply “shift left”, refers to efforts within DevOps teams to identify defects as early as possible in the software development lifecycle. Essentially, it involves regular testing at the start of the development pipeline rather than at the end.

38. How does the blue-green deployment pattern work?

Sample answer:

The blue-green deployment pattern is a deployment strategy that aims to reduce new release risks. It involves running two separate production environments (blue and green), one for the current production system and the other for staging the new release.

Traffic is gradually shifted from the old production environment to the new release once testing is complete and software is deployed. The new release becomes live and, once the traffic has been completely transferred, the old production environment becomes idle.

39. How does the canary deployment pattern work?

Sample answer:

The canary deployment pattern is a deployment strategy that aims to minimize the impact of potential defects in a new software release. It involves rolling out updates to a small subset of users before making them universally available.

The development team uses a router or load balancer to target individual routes with the new release. After launch, metrics are collected to evaluate the performance of the update, and a decision is made on whether the release is ready to be rolled out on a larger scale.

40. What is infrastructure as code, and how is it implemented?

Sample answer:

Infrastructure as Code (IaC) is the management of an application’s infrastructure through software (namely code) rather than manual processes.

The infrastructure as code concept can be implemented at various stages of the DevOps lifecycle—version control, continuous integration, and automated testing—by writing code. This process facilitates more efficient, reliable, and secure changes to applications.

41. What are the key differences between git fetch and git pull?

Sample answer:

Git fetch and git pull are two important commands in GitHub that are often confused with each other.

Git fetch retrieves the latest data from the remote repository, but without integrating this data into the working branches. Git pull, on the other hand, retrieves and updates local branches with the new information from their corresponding remote branches.

42. What are the key differences between git merge and git rebase?

Sample answer:

Git merge and git rebase are two commands in GitHub used to integrate changes from one branch into another. However, they integrate these changes in different ways.

A git merge creates a new commit from the head branch, whereas a git rebase rewrites the changes of one branch onto another without creating a new commit.

43. What are the key differences between containerization and virtualization?

Sample answer:

Containerization and virtualization are the two most popular methods for hosting applications in a computer system.

Virtualization allows developers to run multiple operating systems on the hardware of a single physical server. Containerization enables developers to deploy multiple applications under the same operating system on a single virtual machine or server.

44. How is regression testing implemented?

Sample answer:

Regression testing assesses how an application behaves after a new change has been implemented. It should be performed between integration testing and user testing.

Regression testing can be implemented by checking the original code after new changes have been made or by evaluating how updates affect performance. It can also be automated.

45. What are the key differences between cloud and on-premises services in DevOps?

Sample answer:

Cloud and on-premises services are the two main data hosting pathways available to DevOps teams. With on-premises services, data is stored on in-house servers. With cloud services, data is hosted remotely by a third-party provider.

There are strengths and weaknesses to both strategies, which can be summarized as:

Cloud services offer less security control over data and infrastructure, but they scale better, offer extra services, and typically incur lower costs

On-premises services come with large maintenance costs and unique security threats, but they provide greater control and customization scope

46. What is load balancing?

Sample answer:

Load balancing is the process of evenly distributing incoming network traffic across a group of backend servers. It occurs during the deployment phase of the DevOps lifecycle and is usually automated by developers.

47. How can the DevOps troubleshooting process be optimized?

Sample answer:

Every DevOps team should have an established troubleshooting framework. Some of the best practices for troubleshooting are to:

Set troubleshooting responsibilities within the team

Gather relevant information including metrics, logs, and errors

Hypothesize potential causes and tick them off one by one

Closely monitor behavior after the issue has been fixed

Report on issue resolution

48. What are the three main types of cloud computing services?

Sample answer:

The three main types of cloud services are:

Infrastructure-as-a-Service (IaaS), which provides internet-based access to storage and computing power, while the developer is responsible for managing everything else

Platforms-as-a-Service (PaaS), which helps developers build web applications without needing to manage the underlying infrastructure, such as networks, servers, and storage

Software-as-a-Service (SaaS), which hosts all components of the software application, including servers, storage, data, and the application itself

49. What is the difference between monitoring and observability?

Sample answer:

Monitoring is a process of tracking and data collection that helps DevOps teams better understand the current state of their systems. Observability, on the other hand, uses these findings to diagnose problems and actively resolve them through debugging.

50. What are the key differences between centralized version control and distributed version control?

Sample answer:

With centralized version control, the server’s remote repository provides the latest code to client machines. With decentralized, or distributed, version control, the complete codebase is made available on each computer’s own local repository.