Top 10 Frequently Asked DevOps Interview Questions in 2024

1. [**What Is DevOps?**](https://mindmajix.com/devops-interview-questions#devops)
2. [**Why DevOps Is Important?**](https://mindmajix.com/devops-interview-questions#important)
3. [**What Is the Difference Between Agile and DevOps?**](https://mindmajix.com/devops-interview-questions#agile-and-devops)
4. [**Which Are the Most Popular DevOps Tools?**](https://mindmajix.com/devops-interview-questions#tools)
5. [**What Are DevOps Key Performance Indicators?**](https://mindmajix.com/devops-interview-questions#kpi)
6. [**What Is the DevOps Toolchain?**](https://mindmajix.com/devops-interview-questions#toolchain)
7. [**Explain Puppet Codedir**](https://mindmajix.com/devops-interview-questions#codedir)
8. [**How Chef Cookbook Is Different From Chef Recipe?**](https://mindmajix.com/devops-interview-questions#cookbook)
9. [**How to Turn Off the Auto-deployment Feature?**](https://mindmajix.com/devops-interview-questions#auto-deployment)
10. [**What Is a Docker Container and How Do You Create It?**](https://mindmajix.com/devops-interview-questions#docker-container)

Basic DevOps Interview Questions and Answers

1. What is DevOps?

**Ans:**DevOps can be defined as a combination of software development practices and tools used for increasing organizations' ability while delivering applications, services, and more in close alignment with business objectives.

|  |
| --- |
| *If you want to enrich your career and become a****professional in DevOps****, then enroll in "*[***DevOps Online Training***](https://mindmajix.com/devops-training)*" - This course will help you to achieve excellence in this domain.* |

2. Why DevOps is Important?

**Ans:**DevOps is more about a set of processes that correlate to bring development teams and processes to support software development. The important reason behind the DevOps popularity is that it helps enterprises to build and enhance products at a quicker pace than traditional software development methods.

**The major reasons to adopt DevOps are listed below:**

* Faster innovations
* Shorter development cycles
* Reduced deployment failures
* Improved communication and collaboration
* More stable operating environments
* Increased efficiencies
* Reduced Costs and IT Headcount

3. What is the difference between Agile and DevOps?

**Ans:**The major [differences between Agile and DevOps](https://mindmajix.com/agile-vs-devops) are listed below:

|  |  |  |
| --- | --- | --- |
| **Description** | **Agile** | **DevOps** |
| **Definition** | It's an iterative approach that focuses on development. | It's a practice of both development and operations. |
| **Purpose** | Manages complex projects. | Manages end-to-end engineering processes. |
| **Target areas** | Software development. | End-to-end business solutions and faster deliveries. |
| **Tools** | Kanboard, JIRA, and Bugzilla are popular Agile tools. | AWS, Puppet, Chef are some popular DevOps tools. |
| **Release cycles** | Supports Agile release Cycles | Shorter release cycles and also supports defect detection. |
| **Feedback source** | Self-feedback | Feedback from customers. |

4. List the core operations of DevOps for application development and infrastructure.

**Ans:**The core operations of DevOps for application development and infrastructure are listed below:

**Application development consists of the following core operations:**

* Code building
* Code coverage
* Unit testing
* Packaging
* Deployment

**Infrastructure consists of the following core operations:**

* Provisioning
* Configuration
* Orchestration
* Deployment

5. Which are the most popular DevOps tools?

**Ans:**The most popular DevOps tools are listed below:

* **Git**: Version Control System tool
* **Jenkins:** Continuous Integration tool
* **Docker:** Containerization tool
* **Puppet:** Configuration Management and Deployment tools
* **Ansible:** Configuration Management and Deployment tool
* **Nagios:** Continuous Monitoring tool

|  |
| --- |
| ***Related Article:***[***DevOps Tutorial for Beginners***](https://mindmajix.com/devops-tutorial) |

6. What are DevOps key performance indicators?

**Ans:**The following are the DevOps key performance indicators (KPIs):

* Deployment Frequency
* Deployment Failure
* Change Lead Time
* Change Volume
* Mean Time To Detection
* Mean Time Between Failures
* Mean Time To Recovery
* Change Failure Rate
* Efficiency
* Performance
* Pipeline Adoption

7. List the key components of DevOps.

**Ans:**The following are the key components of DevOps:

* Continuous Integration
* Continuous Delivery
* Microservices
* Infrastructure as Code
* Monitoring and Logging
* Communication and Collaboration

8. What is the DevOps toolchain?

**Ans:**A stack of tools combine to form a DevOps toolchain, it automates the tasks like developing and deploying your application. DevOps can be performed manually with simple steps, but the need for automation quickly increases with the increase in its complexity, and toolchain automation is essential for continuous delivery. GitHub a Version Control Repository is the core component of a DevOps toolchain. More tools may contain backlog tracking, delivery pipelines, etc.

|  |
| --- |
| ***Related Article:***[***Introduction To DevOps Tools***](https://mindmajix.com/introduction-to-devops-tools) |

9. What is AWS DevOps?

**Ans:**AWS facilitates essential services that help you implement DevOps at your company and that are built to use in collaboration with AWS. These services automate manual actions, help teams manage complex environments at scale, and keep engineers stable of the high velocity generated by DevOps.

**Related Article:**[**DevOps Vs SysOps**](https://mindmajix.com/devops-vs-sysops)

10. What is the significance of NRPE in Nagios?

**Ans:**“Nagios Remote Plugin executor”  popularly known as NERP enables us to execute the Nagios plugins remotely. With the help of this mechanism, we can check the performance parameters of the remote Machine.

11. Explain Nagios working?

**Ans:**Nagios runs on a server either as a background process or as a service. Nagios will run the plugins regularly with the help of the hosts or servers present in your Network. We can check the status information by using the web interface. It will execute the scripts based on a schedule.

12. What is VCS?

**Ans:**Version control systems are a kind of software tool which reports the changes in the code and integrates these changes with the existing code. As the developer makes changes in the code frequently, these types of tools are helpful in integrating the new code smoothly without disturbing the work of other team members. Along with integration, it will test the new code so that we can avoid the code leading to bugs)

13. What are the types of Version Control Systems?

**Ans:**Primarily there are three types of Version control systems they are:

* Local Version Control Systems
* Centralized Version Control Systems
* Distributed Version Control Systems

14. What are the benefits of version control?

**Ans:**The primary benefits you can expect from a version control system are the following:

* Complete long-term change history of every file is available.
* All the past versions and variants are kept independent from each other inside the VCS through branching, and whenever required, you can merge back together with the file’s content to verify the changes.

15. What is a branching strategy in DevOps?

**Ans:**Branching is a technique employed for code isolation. In simple terms, it makes a copy of the source code to create two versions that are developed separately. There are various types of branching available. Therefore, the DevOps team must make a choice depending on application requirements. This choice is called a branching strategy.

16. What is Git?

**Ans:**Git is a distributed version control system particularly used for recording the changes in the source code during software development. It manages a set of files or a project that change over time. It stores the information in a data structure called the repository.

**Let's understand the importance of Git through its benefits to organizations:**

* Feature branch workflow
* Distributed development
* Pull requests
* Data redundancy and replication
* High availability
* Superior disk utilization and network performance
* Collaboration friendly

|  |
| --- |
| ***Related Article:***[***Git Tutorial Online***](https://mindmajix.com/git-tutorial) |

17. What’s the main purpose of Continuous Integration (CI)?

**Ans:**Continuous integration is a development practice of automating the integration of code changes from several contributors to a single software project. By regularly integrating, you can detect errors quickly and locate them easily. The source code version control is the crux of the CI process.

18. What are the benefits of Continuous Integration?

**Ans:**The major benefits of Continuous Integration are listed below:

* Faster development cycles
* Smarter risk mitigation
* Stable codes
* Team Communication
* Reduced Overhead
* Flexibility
* Consistency of Build Process

19. What is Trunk-Based development?

**Ans:**A Trunk-Based development is a source control branching model for software development where developers associate on code in a single branch called trunk and employ documented techniques to create long-lived development branches. This process is called Trunk-Based development. It is a key enabler of continuous integration and by extension continuous delivery.

20. How to create a backup and copy files in Jenkins?

**Ans:**The following steps will help you understand how to create a backup and copy files in Jenkins:

* Periodically backup your JENKINS\_HOME directory, which contains job configurations, slave node configurations, and all your build history.
* To create a backup of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

DevOps Interview Questions for 3 Years Experienced

21. How to move or copy Jenkins from one server to another?

**Ans:**There are multiple ways to copy or move Jenkins from one server to another:

1. Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
2. Make a copy of an existing job by making a clone of a job directory by a different name.
3. Rename an existing job by renaming a directory. Note that if you change a job name you will need to change any other job that tries to call the renamed job.

22. How to create a new build job in Jenkins?

**Ans:**The following steps will help you understand how to create a Jenkins job:

* **Step1:** Logon to the Jenkins dashboard and click on the new item at the top left side of the dashboard.
* **Step2:** Enter the item name and click on the freestyle project.
* **Step3:** Enter the details of the project you want to test.
* **Step4:** Enter your repository URL under source code management.
* **Step5:** Under the build, click on add build step, and click on execute Windows batch command.
* **Step6:** After entering the data, click on apply and save the project.
* **Step7:** Click on Build now button to build the source code.
* **Step8:** To see the status of the build you run, click on the build number and click on the console output.

23. List the top Jenkins plugins.

**Ans:**There are many useful plugins in Jenkins. Here, I have listed a few of the top plugins used in Jenkins.

* Dashboard view plugin
* Performance plugin
* Kubernetes plugin
* Amazon ECS Container Service
* Git plugin
* Monitoring plugin

|  |
| --- |
| ***Related Article:***[***What is Jenkins***](https://mindmajix.com/jenkins-tutorial) |

24. Can we build multiple jobs at a time using Jenkins?

**Ans:**Yes. we can build multiple jobs or projects at a time using a Jenkins plugin. After the parent job is implemented, other jobs are implemented automatically. A pipeline multibranch plugin is used for creating a job automatically.

25. What is Continuous Testing? Explain its benefits.

**Ans:**Continuous Testing is defined as a process of executing automated tests as part of the software delivery lifecycle to obtain feedback on business risks associated with the software release. The objective of continuous testing is to test early and test often to prevent the problems from progressing to the next stage of the SDLC.

**The benefits of Continuous Testing are listed below:**

* Mitigates business risks.
* Improves code quality.
* Agile and reliable process.
* Provides test environment access with service virtualization.
* Maintains consistency through the same configuration for all the tests.

26. What do you mean by Automation Testing?

**Ans:**Automation Testing or Test Automation is a software testing technique. It is used to automate the testing tasks and repetitive tasks that are difficult to perform manually. It involves the use of separate testing tools which lets you create test scripts to test and compare the actual and expected outcomes.

27. What are the benefits of Automation Testing?

**Ans:**The major benefits of automation testing are listed below:

* Supports wider test coverage of application features
* Ensures consistency
* Allows parallel execution
* Improves efficiency
* Reusable test scripts
* Saves money and time
* Reliable results

28. List the best Continuous Testing tools?

**Ans:**The following are the best Continuous Testing tools:

* Selenium
* Katalon Studio
* Eggplant
* Watir
* Tosca

29. Mention the testing types supported by Selenium?

**Ans:**Selenium supports functional testing and regression testing.

* **Functional Testing:** It verifies each function of the software application against the functional specifications/requirements.
* **Regression Testing:** In this, test cases are re-executed to verify the previous functionality of the application.

30. How to launch the Browser using WebDriver?

**Ans:**

* For Firefox:

WebDriver driver = new FirefoxDriver();

* For Chrome:

WebDriver driver = new ChromeDriver();

* For Internet Explorer (IE):

WebDriver driver = new InternetExplorerDriver();

31. What’s the difference between Continuous Delivery and Continuous Deployment?

**Ans:**

**Continuous Delivery:** It is a process in which continuous integration, automated testing, and automated deployment capabilities develop, build, test, and release high-quality software rapidly and reliably with minimal manual overhead.

**Continuous Deployment:** It is a process in which qualified changes in the architecture or software code are deployed automatically to production as soon as they are ready and without human intervention.

32. Explain the implementation of Continuous Testing in DevOps?

**Ans:**By following the below-mentioned steps we can implement continuous testing in DevOps:

* In DevOps continuous testing starts in the development phase, as the developer tests the functionality of the code by using tools like selenium.
* Tools like GitHub store these tests and versions along with the software code. DevOps team uses these tests to perform testing on the new build of the software code.
* when the code reaches pre-production, the professional QA team uses these tests by making some changes to the test specifications.
* the operations team can reuse these tests for user acceptance testing and for resolving the post-delivery issues.

33. What is the significance of Continuous Testing in DevOps?

**Ans:**The continuous testing process is done in DevOps to avoid testing the entire code at a time. In traditional SDLC, we will test the code after the whole code is developed but in DevOps, we will test instantly every change made in the code. This kind of testing avoids delays in the product release, and it will also help to achieve better quality in the product.

34. What is the objective of Configuration Management?

**Ans:**

* The objective of Continuous Management is to manage all the configurations of the SDLC by making the development/deployment process controllable and reliable to provide high-quality software.
* There are various components in a configuration management system such as servers, networking, storage, and software. The main purpose of configuration management is to make sure the target systems and software are in the desired state.

35. Which are the best configuration management tools?

**Ans:**The best Configuration Management tools are mentioned below:

* **Chef:** Automation platform that transforms infrastructure into code.
* **Ansible:** Automates entire IT infrastructure.
* **Puppet:** Open-source configuration management tool.
* **Saltstack:** It is based on the Python programming language and allows scalable and efficient configuration.
* **CFEngine:** Open-source configuration management tool, which automates large-scale and complex IT infrastructure.

You can also mention any other tools if you have real-time experience in your previous job and explain how it improves the software development process.

|  |
| --- |
| ***Visit here to learn***[***DevOps Training in Bangalore***](https://mindmajix.com/devops-training-bangalore) |

36. Why is Infrastructure as code important?

**Ans:**

* Infrastructure as code (IaC) is a method to manage and provision IT infrastructure (networks, databases, connection topology, etc.) through source code, rather than manual process or interactive configuration tools.
* It helps you to automate the infrastructure deployment process easily, consistently, and reliably.

37. How does configuration management differ from provisioning infrastructure?

**Ans:**Configuration management and provisioning infrastructure, both are important for the DevOps toolchain. While configuration management is best when it comes to employing desired configurations for target machines or groups of machines, provisioning helps you to create, modify, delete, and track infrastructure using APIs or code.

38. What is Puppet?

**Ans:**Puppet is an open-source configuration management tool used for deploying, configuring, and managing servers. It follows a client-server architecture, in which the client is an agent, and the server is known as the master.

Puppet agent and master communicate through a secure encrypted channel with the help of SSL.

39. What is a Puppet Manifest?

**Ans:**Puppet Manifest is a base component for the Puppet configuration management policy. In Puppet Master, each Puppet node or Puppet Agent has its configuration details written in the native Puppet language. The details that are written in a language that puppets can understand and describe how resources should be configured are termed as Puppet manifests.

Puppet Manifests declares resources that define a state to be enforced on a node. They are considered to be building blocks for complex Puppet modules.

DevOps Interview Questions for 6 Years Experienced

40. How Puppet Module is different from Puppet Manifests?

**Ans:**Puppet Module is a bundle of manifests and data. They have a specific directory structure that allows Puppet to automatically load classes, facts, custom types, defined types, and tasks. Modules must have a valid name and are installed in Puppet’s module path.

Puppet Manifests are nothing but Puppet programs that are composed of Puppet code. It uses.PP extension.

41. Explain Puppet Codedir.

**Ans:**Puppet Codedir is the main directory for Puppet code and data and is mostly used by Puppet master and Puppetapply. It consists of a global modules directory, Hiera data, and environments (which consists of your manifests and modules).

**The Codedir will be located in one of the following locations:**

* Unix:

/etc/puppetlabs/code

Unix non-root users:

~/.puppetlabs/etc/code

* Windows:

%PROGRAMDATA%PuppetLabscode

42. How can you configure systems with Puppet?

**Ans:**You can configure systems with Puppet in two ways:

* In client or server architecture, you should use the Puppet Agent and Puppet Master.
* In stand-alone architecture, you should use the Puppet application.

43. What is Factor in Puppet? How does it work?

**Ans:**The factor is Puppet’s cross-platform system profiling library. Puppet uses factors to gather information during the Puppet run.

Factor discovers and reports basic information of Puppet Agent including network settings, IP addresses, hardware details, etc., and makes available in Puppet manifests as variables.

44. How does Ansible work?

**Ans:**Ansible is an open-source automation tool. It operates by connecting to your nodes and pushing out small programs called Ansible modules to them. It executes these modules through SSH by default and removes them when finished.

It handles many nodes from a single system over an SSH connection by using Ansible playbooks. These Playbooks are capable to execute multiple tasks and represented in YAML format.

45. How Ansible playbook is different from ad-hoc commands?

**Ans:**

* Ansible playbook is a structured unit of scripts that describes work for server configuration. It is used for repeated actions.
* An ad-hoc command is used to do something quicker, mostly one-time use.

|  |
| --- |
| ***Related Article:***[***DevOps Configuration Tools***](https://mindmajix.com/9-configuration-management-tools-for-devops) |

46. How to get a list of Ansible predefined variables?

**Ans:**Ansible stores facts about machines under management by default and these can be accessed in playbooks and templates. To get a list of all the facts that are available about a machine, run a setup module as an ad-hoc action:

Ansible -m setup hostname

This will present all the facts that are available under that particular host.

47. What are Ansible Handlers?

**Ans:**Handlers are exactly like regular tasks inside an Ansible playbook but run only when the task contains notify directive and also additional information if it changes something.

**Eg:** When a config file was changed, the task referencing that config file notifies the service restart handler.

|  |
| --- |
| ***Related Article:***[***What is Site Reliability Engineering***](https://mindmajix.com/what-is-site-reliability-engineering) |

48. How does Chef work?

**Ans:**Start your answer by defining Chef. It is an automation platform that is particularly used for transforming infrastructure into code. It uses pure-Ruby domain-specific language to write system configurations.

Now you can explain the architecture of Chef and how it works.

**Chef architecture consists of three core components:**

Chef Workstation, Chef Node, and Chef Servers.

* **Chef Server:** The chef Server is the core point for infrastructure configuration data. It stores, configures, manages, and presents the configuration data to all components.
* **Chef Workstation:** Workstations are the hosts where you can create, test, and modify your cookbooks and configuration data.
* **Chef Node:** These are the servers managed by the Chef. Chef client is installed on each node which is under management by Chef. Chef clients run on the nodes to contact the chef server for the information required to configure the node.

49. Explain Chef Resources.

**Ans:**First, begin with the definition of a Chef Resource. A Chef Resource describes a piece of an operating system at its desired state. It is a configuration policy statement that is used for representing the desired state of a node to which you want to take the current configuration for using the resource providers.

**The functions of a Chef Resource are listed below:**

* It helps to know the present status of the target machine.
* Defines the steps needed to perform for getting the target machine to the desired state.
* Specifies resource types such as template, package, file, or service.

50. How Chef CookBook is different from Chef Recipe?

**Ans:**

* A Chef Recipe is a set of resources that primarily configures a software package or some piece of infrastructure. Recipes are written in the Ruby language.
* While a Chef Cookbook consists of recipes that a node desired state.

51. What happens when you don’t specify a Resource’s action in Chef

**Ans:**In case, if you don't specify a resource’s action, then Chef applies the default action.

For example, in resource 1, the action is not specified, still, it will take a default action.

file 'C:UsersAdministratorchef-reposettings.ini' do

content 'greeting=hello world'

end

In resource 2, when you define the action with create command, it is also used to create the default action.

file 'C:UsersAdministratorchef-reposettings.ini' do

action :create

content 'greeting=hello world'

end

52. How to refresh static components of a deployed application without redeploying the entire application?

**Ans:**By using Weblogic.Deployer you can define a component and target a server through the following syntax:

java weblogic.Deployer -adminurl http://admin:7001 -name appname -targets server1,server2 -deploy jsps/\*.jsp

53. How to turn off the auto-deployment feature?

**Ans:**The auto-deployment feature is used for determining whether there are any new applications or changes in existing applications and dynamically deploy them.

It is enabled for servers that run in development mode.

To turn off the auto-deployment feature, follow one of the methods to place servers in production mode:

* In the Administration Console, click the name of the domain in the left pane and select the Production Mode checkbox in the right pane.
* At the command line, include the following argument when starting the domain’s Administration Server:

-Dweblogic.ProductionModeEnabled=true

* Production mode is set for all WebLogic Server instances in a given domain.

54. Why is Continuous Monitoring important?

**Ans:**Continuous Monitoring helps to detect and measure the security implications for planned and unexpected changes and assesses the vulnerabilities in a threat space.

It delivers information on the application’s performance and usage patterns.

55. Which are the best Continuous Monitoring tools?

**Ans:**The following are the best Continuous Monitoring tools:

* Nagios
* Tenable
* Solarwinds
* Lansweeper
* Spiceworks
* Snort

|  |
| --- |
| ***Related Article:***[***DevOps Monitoring Tools***](https://mindmajix.com/list-of-bi-or-monitoring-devops-tools) |

56.  what is containerization in DevOps and list their benefits?

**Ans:**Containerization is defined as a process of binding the application and its required environment. Binding makes the application run in any computational environment. DevOps main goal is to bridge the gap between the development team and the operations team.

To bridge the gap between them, it should make both sides work on an identical environment. Containerization helps in setting an identical environment quickly, and it will provide easy access to operating system resources. Docker tool is widely used for implementing containerization in DevOps.

They are a streamlined way to create, test, deploy, and redeploy applications in multiple environments.

**Benefits of Containers are listed below:**

* Less overhead
* Improved productivity
* More consistent operation
* Better application deployment
* Greater efficiency

57. What is a Docker Container and how do you create it?

**Ans:**

* A Docker container is an open-source software development platform that stores the code and all of its dependencies and runs the application quickly and reliably from one computing environment to the other.
* Docker containers are not specified to any particular infrastructure; they can run on any infrastructure, on any computer, and in any cloud.
* A Docker container image is a standalone, lightweight, and executable package of software that has everything to run the application such as code, system tools, runtime, system libraries, and settings.

**Docker Containers can be created with the Docker image using the following command:**

docker run -t -i <image name> <command name>

This will create and start the container.

If you want to check the list of all running containers with status on the host, use the following command:

docker ps -a

If you have any additional DevOps questions and are unable to find the answers, please do mention them in the comment section below. We’ll get back to you at the earliest.

**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**](https://www.testgorilla.com/test-library/situational-judgment-tests/communication-test/) and teamwork skills
* [**Programming skills**](https://www.testgorilla.com/test-library/test-type/programming-skills-tests)
* 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.

**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.

**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.

#### 1.

What benefits does DevOps have in business?

Hide Answer

DevOps can bring several benefits to a business, such as:

* **Faster time to market**: DevOps practices can help to streamline the development and deployment process, allowing for faster delivery of new products and features.
* **Increased collaboration**: DevOps promotes collaboration between development and operations teams, resulting in better communication, more efficient problem-solving, and higher-quality software.
* **Improved agility**: DevOps allows for more rapid and flexible responses to changing business needs and customer demands.
* **Increased reliability**: DevOps practices such as continuous testing, monitoring, and automated deployment can help to improve the reliability and stability of software systems.
* **Greater scalability**: DevOps practices can help to make it easier to scale systems to meet growing business needs and user demand.
* **Cost saving**s: DevOps can help to reduce the costs associated with the development, deployment, and maintenance of software systems by automating many manual processes and reducing downtime.
* **Better security**: DevOps practices such as continuous testing and monitoring can help to improve the security of software systems.

#### 2.

What are the key components of a successful DevOps workflow?

Hide Answer

The key components include Continuous Integration (CI), Continuous Delivery (CD), Automated testing, Infrastructure as Code (IaC), Configuration Management, Monitoring & Logging, and Collaboration & Communication.

#### 3.

What are the different phases of the DevOps lifecycle?

Hide Answer

The DevOps lifecycle is designed to streamline the development process, minimize errors and defects, and ensure that software is delivered to end-users quickly and reliably. The different phases of the DevOps lifecycle are:

* **Plan**: Define project goals, requirements, and resources
* **Code**: Develop and write code
* **Build**: Compile code into executable software
* **Test**: Verify and validate software functionality
* **Release**: Deploy code to the production environment
* **Deploy**: Automated deployment and scaling of software
* **Operate**: Monitor and maintain the software in production
* **Monitor**: Collect and analyze software performance data
* **Optimize**: Continuously improve and evolve the software system

#### 4.

What are the best programming and scripting languages for DevOps engineers?

Hide Answer

The [best programming and scripting languages DevOps](https://www.turing.com/blog/programming-and-scripting-languages-devops-engineers-should-learn-in-2022/) engineers must know are as follows:

**Programming languages**:-

* Bash
* SQL
* Go
* Terraform (Infrastructure as Code)
* Ansible (Automation and Configuration Management)
* Puppet (Automation and Configuration Management)

**Scripting languages**:-

* JavaScript
* Python
* Ruby
* Perl
* Groovy

#### 5.

Explain configuration management in DevOps.

Hide Answer

Configuration Management (CM) is a practice in DevOps that involves organizing and maintaining the configuration of software systems and infrastructure. It includes version control, monitoring, and change management of software systems, configurations, and dependencies.

The goal of CM is to ensure that software systems are consistent and reliable to make tracking and managing changes to these systems easier. This helps to minimize downtime, increase efficiency, and ensure that software systems remain up-to-date and secure.

Configuration Management is often performed using tools such as Ansible, Puppet, Chef, and SaltStack, which automate the process and make it easier to manage complex software systems at scale.

#### 6.

Name and explain trending DevOps tools.

Hide Answer

**Docker**: A platform for creating, deploying, and running containers, which provides a way to package and isolate applications and their dependencies.

**Kubernetes**: An open-source platform for automating containers' deployment, scaling, and management.

**Ansible**: An open-source tool for automating configuration management and provisioning infrastructure.

**Jenkins**: An open-source tool to automate software development, testing, and deployment.

**Terraform**: An open-source tool for managing and provisioning infrastructure as code.

**GitLab**: An open-source tool that provides source code management, continuous integration, and deployment pipelines in a single application.

**Nagios**: An open-source tool for monitoring and alerting on the performance and availability of software systems.

**Grafana**: An open-source platform for creating and managing interactive, reusable dashboards for monitoring and alerting.

**ELK Stack**: A collection of open-source tools for collecting, analyzing, and visualizing log data from software systems.

**New Relic**: A SaaS-based tool for monitoring, troubleshooting, and optimizing software performance.

#### 7.

How would you strategize for a successful DevOps implementation?

Hide Answer

For a successful DevOps implementation, I will follow the following steps:

* Define the business objectives
* Build cross-functional teams
* Adopt agile practices
* Automate manual tasks
* Implement continuous integration and continuous delivery
* Use infrastructure as code
* Monitor and measure
* Continuously improve
* Foster a culture of learning to encourage experimentation and innovation

#### 8.

What role does AWS play in DevOps?

Hide Answer

AWS provides a highly scalable and flexible cloud infrastructure for hosting and deploying applications, making it easier for DevOps teams to manage and scale their software systems. Moreover, it offers a range of tools and services to support continuous delivery, such as AWS CodePipeline and AWS CodeDeploy, which automate the software release process.

AWS CloudFormation and AWS OpsWorks allow automation of the management and provisioning of infrastructure and applications. Then we have Amazon CloudWatch and Amazon CloudTrail, which enable the teams to monitor and log the performance and behavior of their software systems, ensuring reliability and security.

AWS also supports containerization through Amazon Elastic Container Service and Amazon Elastic Kubernetes Service. It also provides serverless computing capabilities through services such as AWS Lambda. In conclusion, AWS offers a range of DevOps tools for efficient and successful DevOps implementation.

#### 9.

DevOps vs. Agile: How are they different?

Hide Answer

DevOps and Agile are both methodologies used to improve software development and delivery, but they have different focuses and goals:

**Focus**: Agile is focused primarily on the development process and the delivery of high-quality software, while DevOps is focused on the entire software delivery process, from development to operations.

**Goals**: The goal of Agile is to deliver software in small, incremental updates, with a focus on collaboration, flexibility, and rapid feedback. DevOps aims to streamline the software delivery process, automate manual tasks, and improve collaboration between development and operations teams.

**Teams**: Agile teams mainly focus on software development, while DevOps teams are cross-functional and their job include both development and operations.

**Processes**: Agile uses iterative development processes, such as Scrum or Kanban, to develop software, while DevOps uses a continuous delivery process that integrates code changes, testing, and deployment into a single, automated pipeline.

**Culture**: Agile emphasizes a culture of collaboration, continuous improvement, and flexible responses to change, while DevOps emphasizes a culture of automation, collaboration, and continuous improvement across the entire software delivery process.

To summarize, DevOps is a natural extension of Agile that incorporates the principles of Agile and applies them to the entire software delivery process, not just the development phase.

#### 10.

What is a container, and how does it relate to DevOps?

Hide Answer

A container is a standalone executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, environment variables, and system tools. Containers are related to DevOps because they enable faster, more consistent, and more efficient software delivery.

#### 11.

Explain Component-based development in DevOps.

Hide Answer

Component-based development, also known as CBD, is a unique approach to product development. In this, developers search for pre-existing well-defined, verified, and tested code components instead of developing from scratch.

#### 12.

How is version control crucial in DevOps?

Hide Answer

Version control is crucial in DevOps because it allows teams to manage and save code changes and track the evolution of their software systems over time. Some key benefits include collaboration, traceability, reversibility, branching, and release management.

#### 13.

Describe continuous integration.

Hide Answer

Continuous integration (CI) is a software development practice that automatically builds, tests, and integrates code changes into a shared repository. The goal of CI is to detect and fix integration problems early in the development process, reducing the risk of bugs and improving the quality of the software.

#### 14.

What is continuous delivery?

Hide Answer

Continuous delivery (CD) is a software development practice that aims to automate the entire software delivery process, from code commit to deployment. The goal of CD is to make it possible to release software to production at any time by ensuring that the software is always in a releasable state.

Learn more about CI/CD [here](https://www.turing.com/kb/ci-cd-pipeline).

#### 15.

Explain continuous testing.

Hide Answer

Continuous testing is a software testing practice that involves automating the testing process and integrating it into the continuous delivery pipeline. The goal of continuous testing is to catch and fix issues as early as possible in the development process before they reach production.

#### 16.

What is continuous monitoring?

Hide Answer

Continuous monitoring is a software development practice that involves monitoring applications' performance, availability, and security in production environments. The goal is to detect and resolve issues quickly and efficiently to ensure that the application remains operational and secure.

#### 17.

What key metrics should you focus on for DevOps success?

Hide Answer

Focusing on the right key metrics can provide valuable insights into your DevOps processes and help you identify areas for improvement. Here are some key metrics to consider:

**Deployment frequency**: Measures how often new builds or features are deployed to production. Frequent deployments can indicate effective CI/CD processes, while rare deployments can hint at bottlenecks or inefficiencies.

**Change lead time**: The time it takes for code changes to move from initial commit to deployment in a production environment. A low change lead time can indicate agile processes that allow for quick adaptation and innovation.

**Mean time to recovery (MTTR)**: The average time it takes to restore a system or service after an incident or failure. A low MTTR indicates that the DevOps team can quickly identify, diagnose, and resolve issues, minimizing service downtime.

**Change failure rate**: The percentage of deployments that result in a failure or require a rollback or hotfix. A low change failure rate suggests effective testing and deployment strategies, reducing the risk of introducing new issues.

**Cycle time**: The total time it takes for work to progress from start to finish, including development, testing, and deployment. A short cycle time indicates an efficient process and faster delivery of value to customers.

**Automation percentage**: The proportion of tasks that are automated within the CI/CD pipeline. High automation levels can accelerate processes, reduce human error, and improve consistency and reliability.

**Test coverage**: Measures the percentage of code or functionality covered by tests, which offers insight into how thoroughly your applications are being tested before deployment. High test coverage helps ensure code quality and reduces the likelihood of production issues.

**System uptime and availability**: Monitors the overall reliability and stability of your applications, services, and infrastructure. A high uptime percentage indicates more resilient and reliable systems.

**Customer feedback**: Collects quantitative and qualitative data on user experience, satisfaction, and suggestions for improvement. This metric can reveal how well the application or service is aligning with business objectives and meeting customer needs.

**Team collaboration and satisfaction**: Measures the effectiveness of communication, efficiency, and morale within the DevOps teams. High satisfaction levels can translate to more productive and successful DevOps practices.

#### 18.

List down the types of HTTP requests.

Hide Answer

HTTP requests (methods) play a crucial role in DevOps when interacting with APIs, automation, webhooks, and monitoring systems. Here are the main HTTP methods used in a DevOps context:

**GET**: Retrieves information or resources from a server. Commonly used to fetch data or obtain status details in monitoring systems or APIs.

**POST**: Submits data to a server to create a new resource or initiate an action. Often used in APIs to create new items, trigger builds, or start deployments.

**PUT**: Updates a resource or data on the server. Used in APIs and automation to edit existing information or re-configure existing resources.

**PATCH**: Applies partial updates to a resource on the server. Utilized when only a certain part of the data needs an update, rather than the entire resource.

**DELETE**: Deletes a specific resource from the server. Use this method to remove data, stop running processes, or delete existing resources within automation and APIs.

**HEAD**: Identical to GET but only retrieves the headers and not the body of the response. Useful for checking if a resource exists or obtaining metadata without actually transferring the resource data.

**OPTIONS**: Retrieves the communication options available for a specific resource or URL. Use this method to identify the allowed HTTP methods for a resource, or to test the communication capabilities of an API.

**CONNECT**: Establishes a network connection between the client and a specified resource for use with a network proxy.

**TRACE**: Retrieves a diagnostic representation of the request and response messages for a resource. It is mainly used for testing and debugging purposes.

#### 19.

What is the role of automation in DevOps?

Hide Answer

Automation plays a critical role in DevOps, allowing teams to develop, test, and deploy software more efficiently by reducing manual intervention, increasing consistency, and accelerating processes. Key aspects of automation in DevOps include Continuous Integration (CI), Continuous Deployment (CD), Infrastructure as Code (IaC), Configuration Management, Automated Testing, Monitoring and Logging, Automated Security, among others. By automating these aspects of the software development lifecycle, DevOps teams can streamline their workflows, maximize efficiency, reduce errors, and ultimately deliver higher-quality software faster.

#### 20.

What is the difference between a service and a microservice?

Hide Answer

A service and a microservice are both architectural patterns for building and deploying software applications, but there are some key differences between them:

#### 21.

How do you secure a CI/CD pipeline?

Hide Answer

To secure a CI/CD pipeline, follow these steps:

* Ensure all tools and dependencies are up to date
* Implement strong access controls and authentication
* Scan code for vulnerabilities (e.g., SonarQube, OWASP Dependency-Check)
* Cloud provider managed private build environments (e.g., AWS CodeBuild)
* Store sensitive data like keys, tokens, and passwords in a secret management tool (e.g., HashiCorp Vault, AWS Secrets Manager)
* Regularly audit infrastructure and system logs for anomalies

#### 22.

How does incident management fit into the DevOps workflow?

Hide Answer

Incident management is a crucial component of the DevOps workflow, as it helps quickly resolve issues in the production environment and prevent them from becoming bigger problems.

#### 23.

What is the difference between a git pull and a git fetch?

Hide Answer

git pull and git fetch are two distinct commands in Git that serve different purposes, primarily related to updating a local repository with changes from a remote repository

git pull is a combination of git fetch and git merge. It retrieves data from the remote repository and automatically merges it into the local branch.

git fetch is used to retrieve data from remote repositories, but it does not automatically merge the data into the local branch. It only downloads the data and stores it in the local repository as a separate branch, which means the developer must manually merge the fetched data with the remote branch.

#### 24.

What is the difference between a container and a virtual machine?

Hide Answer

A container and a virtual machine are both technologies used for application virtualization. However, there are some key differences between the two.

A virtual machine runs an entire operating system, which can be resource-intensive, while a container shares the host operating system and only includes the necessary libraries and dependencies to run an application, making it lighter and more efficient.

Containers provide isolation between applications, while virtual machines provide complete isolation from the host operating system and other virtual machines.

### Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

[Apply Now](https://developers.turing.com/?referrer=https://www.google.com/)

### INTERMEDIATE DEVOPS TECHNICAL INTERVIEW QUESTIONS AND ANSWERS

#### 1.

Explain how you will handle merge conflicts in Git.

Hide Answer

The following three steps can help resolve merge conflicts in Git: -

* Understand the problem, then merge conflict can arise due to different problems, for example, same line edit on the same file, deleting some files, or files with the same file names. You can understand what caused the conflict by checking the git status.
* The next step is to mark and clean up the conflict. For this, open the file with mergetool. Git will mark the conflict portion as ‘<<<<>>>>[other/branch/name]’ -
* Now run commit again, and merge the current branch with the master branch.

When answering these DevOps interview questions, please include all the steps in your answer. The more details you provide, the better your chances of moving through the interview process.

#### 2.

Mention some advantages of Forking workflow over other Git workflows.

Hide Answer

This type of DevOps interview question warrants a detailed answer. Below are some advantages of Forking workflow over other Git workflows.

There is a fundamental difference between Forking workflow and other Git workflows. Unlike other Git workflows that have a single central code repository on the server side, in Forking workflow, every developer gets their own server-side repositories.

The Forking workflow finds use in public open-source projects leading to the integration of individual contributions without the need for all users pushing to a central repository for clean project history. Only the project maintainer pushes to the central repository, while the individual developers can use their personal server-side repositories.

Once the developers complete their local commits and are ready to publish, they push their commits to their respective public repositories. After that, they send a pull request to the central repository. This notifies the project maintainer to integrate the update with the central repository.

#### 3.

Is it possible to move or copy Jenkins from one server to another? How?

Hide Answer

Yes, one can copy the Jenkins jobs directory from the old server to the new one. To move a job from one Jenkins installation to another, one can simply copy the required job directory.

Another method is to create a clone of an existing job directory with a different name. Another way is to rename an existing job by renaming the directory. However, in this method, one needs to change any other job calling the renamed job.

#### 4.

What are automation testing and continuous testing?

Hide Answer

[Automation testing](https://www.turing.com/blog/automation-testing-trends-in-2022/) is a process that automates the manual testing process. Different testing tools allow developers to generate test scripts that can be executed continually without the need for human intervention.

In continuous testing, the automated tests are executed as part of the DevOps software delivery pipeline. Each build is continuously tested so that the development stays ahead of the problems and prevents them from moving on to the next software delivery lifecycle stage. This process speeds up the workflow of the developer. This is because the developers do not need to run all the tests once they make changes.

The above type of DevOps interview question only asks for a definition of the processes and not a differentiation between the processes. However, if you want you can also differentiate between the two processes.

#### 5.

Mention the technical challenges with Selenium.

Hide Answer

Mentioned below are some of the technical challenges with Selenium: -

* It only works with web-based applications
* It does not work with bitmap comparison
* While commercial tools such as HP UFT have vendor support, Selenium does not
* Selenium does not have an object repository, thus storing and maintaining objects is complex

For a DevOps interview question asking about technical challenges of a tool or component, apart from highlighting the challenges, you can also recount your experience with such challenges and how you overcame them. Giving a personal experience for such a question shows that you haven’t simply mugged up answers.

#### 6.

What are Puppet Manifests?

Hide Answer

While this is a rather simple DevOps interview question, knowing the answer to such questions shows you are serious about your work. Puppet manifests are programs written in the native Puppet language and saved with the .pp extension.

As such, any Puppet program built to create or manage a target host machine is referred to as a manifest. These manifests are made of Puppet code. The configuration details of Puppet nodes and Puppet agents are contained in the Puppet Master.

#### 7.

Explain the working of Ansible.

Hide Answer

As an open-source tool used for automation, Ansible is divided into two server types - nodes and controlling machines. The installation of Ansible happens on the controlling machine, and this machine, along with SSH, helps manage the nodes.

The controlling machine has inventories that specify the node’s location. Ansible does not have an agent as a tool, which precludes the need for any mandatory installations on the nodes. Therefore, no background programs need to be executed when Ansible manages the nodes.

Ansible Playbooks help Ansible manage multiple nodes from one system with an SSH connection. This is because Playbooks exist in the YAML format and can perform many tasks simultaneously.

In a DevOps interview question like the above, you should include all the details. Moreover, in such a DevOps interview question, you may expect follow-up questions, such as, “Have you used Ansible? Take us through any interesting or weird experience you had while using it.”

#### 8.

Explain the Sudo concept in Linux.

Hide Answer

The sudo (superuser do) command in Linux is a powerful utility that allows users to execute commands with the privileges of another user, usually the superuser or root. The sudo concept provides a controlled way of managing which users can perform administrative tasks without granting them unrestricted root access.

#### 9.

What is the purpose of SSH?

Hide Answer

SSH is the abbreviation of “Secure Shell.” The SSH protocol was designed to provide a secure protocol when connecting with unsecured remote computers. SSH uses a client-server paradigm, where the communication between the client and server happens over a secure channel. There are three layers of the SSH protocol: -

**Transport layer**: This layer ensures that the communication between the client and the server is secure. It monitors the encryption and decryption of data and protects the connection’s integrity. Data caching and compression are also their functions.

**Authentication layer**: This layer is responsible for conducting client authentication.

**Connection layer**: This layer comes into play after authentication and manages the communication channels. Communication channels created by SSH use public-key cryptography for client authentication. Once the secure connection is in place, the exchange of information through SSH happens in a safe and encrypted way, irrespective of the network infrastructure being used. With SSH, tunneling, forwarding TCP, and transferring files can be done securely.

#### 10.

Talk about Nagios Log server.

Hide Answer

The purpose of the Nagios Log server is to simplify the search for log data. Thus, it is best suited for tasks such as alert set-up, notifications for potential threats, log data querying, and quick system auditing. Using Nagios Log server can place all log data at a single location with high availability.

#### 11.

Explain how you will handle sensitive data in DevOps.

Hide Answer

Handling sensitive data in DevOps requires a robust approach to ensure the confidentiality, integrity, and availability of data. Here are some steps that can be taken to handle sensitive data in DevOps:

* **Identify and classify sensitive data**: The first step is to identify what data is sensitive and then classify it based on its level of sensitivity. This will help determine the appropriate measures to be taken to protect it.
* **Implement access controls**: Access controls should be put in place to ensure that only authorized personnel have access to sensitive data. This includes implementing strong passwords, two-factor authentication, and limiting access to sensitive data on a need-to-know basis.
* **Encrypt data**: Sensitive data should be encrypted in transit and at rest. This helps protect the data from being intercepted or accessed by unauthorized parties.
* **Use secure communication channels**: Communication channels to transfer sensitive data should be secured using encryption protocols such as SSL/TLS.
* **Implement auditing and logging**: Audit logs should be kept to monitor who has accessed sensitive data and what actions were taken. This helps detect and respond to any unauthorized access or suspicious activity.
* **Conduct regular security assessments**: Regular security assessments should be conducted to identify vulnerabilities and potential security risks. This helps ensure that the security measures put in place are effective and up to date.

#### 12.

What is high availability, and how can you achieve it in your infrastructure?

Hide Answer

High availability, abbreviated as HA, refers to removing singular failure points to let applications continue operating even if a server or another IT part it depends on fails. To achieve HA in an infrastructure, these steps are crucial: -

**Capacity planning**: It’s key to anticipate the number of requests and users at various dates and times to avoid capacity bottlenecks. For this, regular reviews of event logs and traffic loads must be conducted to establish a utilization baseline to predict and analyze future trends.

A vital step here is to determine the infrastructure’s resources, like memory, network bandwidth, processors, etc., measure their performance, and compare that to their maximum capacities. This way, their capacity can be identified to take the necessary steps to achieve HA.

**Redundancy planning**: This involves duplicating the infrastructure’s system components so that not a single one’s failure can power down the entire application.

**Failure protection**: Multiple issues can hinder achieving high availability, which is why anticipating system issues beforehand is key. Incorrect cluster configuration, mismatching of cluster resources to physical resources, networked storage access problems, etc., are just a few of the many issues that can occur. Paying close attention to these issues unique to the infrastructure and understanding their weak points will help determine the response method for each.

When answering this DevOps interview question, list all the prominent and common issues/bottlenecks/problems with proper examples to show a firm grasp of the concept to move ahead in the interview round.

#### 13.

Describe Blue-Green deployments in DevOps. How does Blue/Green and Rolling deployment differ in Kubernetes?

Hide Answer

By definition, blue-green deployment is a code release model comprising two different yet identical environments simultaneously existing. Here, the traffic is moved from one environment to the other to let the updated environment go into production, while the older one can be retired via a continuous cycle.

Blue-green deployment is a widely-used technique in DevOps that companies adopt to roll out new software updates or designs without causing downtime. It is usually implemented for web app maintenance and requires two identically running applications with the same hardware environments established for a single application. The active version is the blue one, which serves the end users, and the inactive one is green.

Blue/Green deployment uses two environments with the new version in one environment while the current version runs in the other. Traffic is switched when the new version is ready. Rolling deployment updates pods incrementally, replacing old versions with new while maintaining availability.

#### 14.

Explain how Nagios works.

Hide Answer

Nagios operates on a server, typically as a service or demon. It periodically runs plugins housed on the same server, which contacts servers or hosts on the internet or your network. You can use the web interface to view the status information and receive SMS or email notifications if something happens.

The Nagios daemon acts like a scheduler that runs specific scripts at particular moments. It then stores the script results and runs other scripts to check if the results change.

#### 15.

List the trending Jenkins plugins.

Hide Answer

This is a common DevOps interview question, but you must list the most trending plugins and as many as possible.  
Git plugin: It facilitates Git functions critical for a Jenkins project and provides multiple Git operations like fetching, pulling, branching, checking out, merging, listing, pushing, and tagging repositories. The Git plugin also serves as a DVCD (Distributed Version Control DevOps) to assist distributed non-linear workflows via data assurance to develop high-quality software.

**Jira plugin**: This open-source plugin integrates Jenkins with Jira (both Server and Cloud versions), allowing DevOps teams to gain more visibility into their development pipelines.

**Kubernetes plugin**: This plugin integrates Jenkins with Kubernetes to provide developers with scaling automation when running Jenkins slaves in a Kubernetes environment. This plugin also creates Kubernetes Pods dynamically for every agent the Docker image defines. It runs and terminates each agent after build completion.

**Docker plugin**: This plugin helps developers spawn Docker containers and automatically run builds on them. The Docker plugin lets DevOps teams use Docker hosts to provision docker containers as Jenkins agent nodes running single builds. They can terminate the nodes without the build processes requiring any Docker awareness.

**SonarQube plugin**: This plugin seamlessly integrates SonarQube with Jenkins to help DevOps teams detect bugs, duplication, and vulnerabilities and ensure high code quality before creating code automatically via Jenkins.

**Maven integration plugin**: While Jenkins doesn’t have in-built Maven support, this plugin provides advanced integration of Maven 2 or 3 projects with Jenkins. The Maven integration plugin offers multiple functionalities like incremental builds, binary post-build deployments, automatic configurations of Junit, Findbugs, and other reporting plugins.

#### 16.

Explain Docker Swarms.

Hide Answer

A Docker Swarm is a native clustering that turns a group of Docker hosts into a virtual, single Docker host. The Swarm serves the standard Docker API, and any tool already communicating with the Docker Daemon can utilize Swarm to scale transparently to various hosts. Supported tools include Dokku, Docker Machine, Docker Compose, and Jenkins.

#### 17.

Describe a multi-stage Dockerfile, and why is it useful?

Hide Answer

A multi-stage Dockerfile allows multiple build stages within a single Dockerfile. Each stage can use a different base image, and only required artifacts are carried forward. It reduces the final image size, improves build time, and enhances security.

#### 18.

How do you ensure compliance adherence in a DevOps environment?

Hide Answer

Ensuring compliance adherence in a DevOps environment requires a comprehensive approach that involves implementing policies, procedures, and tools that support compliance throughout the software development lifecycle. Here are some key steps that can help:

**Establish clear policies and procedures**: Develop clear policies and procedures that define the compliance requirements for your organization. This may include standards for security, data privacy, and regulatory compliance.

**Implement automated testing**: Automated testing can help identify potential compliance issues early in the development process. This includes security testing, vulnerability scanning, and code analysis.

**Implement change management processes**: Change management processes help ensure that changes are properly tested and approved before they are deployed. This helps reduce the risk of introducing compliance issues into the production environment.

**Use version control**: Version control systems allow you to track changes to code and configurations, which can help with auditing and compliance reporting.

**Monitor and log all activitie**s: Monitoring and logging all activities in the DevOps environment can help identify compliance issues and provide an audit trail for regulatory reporting.

#### 19.

Can you explain the concept of "Infrastructure as Code"?

Hide Answer

Infrastructure as Code is an approach to data center server, networking, and storage infrastructure management. This approach is designed to simplify large-scale management and configuration majorly.

Traditional data center management involved manual action for every configuration change, using system administrators and operators. In comparison, infrastructure as code facilitates housing infrastructure configurations in standardized files, readable by software that maintains the infrastructure’s state.

This approach is popular in DevOps as it helps to improve productivity as operations and administrators don’t need to conduct manual configuration for data center infrastructure changes. Moreover, IaC also offers better reliability as the infrastructure configuration is stored in digital files, reducing human error chances

#### 20.

Explain Ansible Playbooks. Write a simple Ansible playbook to install and start Nginx.

Hide Answer

Ansible’s Playbooks are its language for configuration, deployment, and orchestration. These Playbooks can define a policy that a team would want its remote systems to establish, or a group of steps in a general IT procedure.

Playbooks are human-readable and follow a basic text language. At the basic level, these can also be used for the configuration and deployment management of remote machines.

#### 21.

Explain the concept of the term "cloud-native".

Hide Answer

The term ‘[cloud native](https://www.turing.com/kb/what-is-cloud-native)’ is used to describe any application built to reside in the cloud from its very beginning. Cloud-native comprises cloud technologies such as container  
orchestrators, auto-scaling, and microservices.

#### 22.

How can you get a list of every ansible\_variable?

Hide Answer

By default, Ansible collects ‘facts’ about machines under management. You can access these facts in templates and Playbooks. To check the full list of all the facts available about a particular machine, run the setup module as an ad-hoc action using this:  
Ansible -m setup hostname  
Using this will print out a complete list of all facts available for a particular host.

#### 23.

Write a simple Bash script to check if a service is running.

Hide Answer

#### 24.

Name the platforms Docker currently runs on.

Hide Answer

Currently, Docker only runs on Cloud, Linux, Windows, and Mac platforms

**Linux:**

* RHEL 6.5+
* Ubuntu 12.04, 13.04
* CentOS 6+
* Fedora 19/20+
* ArchLinux
* Gentoo
* openSUSE 12.3+
* CRUX 3.0+

**Cloud:**

* Rackspace
* Amazon EC2
* Microsoft Azure
* Google Compute Engine

**Mac:**  
Docker runs on macOS 10.13 and newer versions.

**Windows**:  
Docker runs on Windows 10 and Windows Server 2016 and newer versions.

#### 25.

What is the usage of a Dockerfile?

Hide Answer

A Dockerfile is used to provide instructions to Docker, allowing it to build images automatically. This Dockerfile is a text document containing all the user commands that can be called on the command line to create an image. Using Docker build, users can assemble an automated build to execute numerous command-line instructions one after the other.

#### 26.

List the top configuration management tools.

Hide Answer

**Ansible**: This configuration management tool helps to automate the entire IT infrastructure.

**Chef**: This tool acts as an automation platform to transform infrastructure into code.

**Saltstack:** This tool is based on Python and seamlessly allows efficient and scalable configuration.

**Puppet**: This open-source configuration management tool helps implement automation to handle complex software systems.

**CFEngine**: This is another open-source configuration management tool that helps teams automate complex and large-scale IT infrastructure.

#### 27.

How do you use Docker for containerization?

Hide Answer

Docker is a popular tool for containerization, which allows you to create lightweight, portable, and isolated environments for your applications. Here's a high-level overview of how to use Docker for containerization:

**Install Docker**: First, you need to install Docker on your system. You can download Docker Desktop from the official website, which provides an easy-to-use interface for managing Docker containers.

**Create a Dockerfile**: A Dockerfile is a text file that contains instructions for building a Docker image. You can create a Dockerfile in the root directory of your application, and it should specify the base image, environment variables, dependencies, and commands to run your application.

**Build a Docker image**: Once you have a Dockerfile, you can use the docker build command to build a Docker image. The command will read the Dockerfile and create a new image that includes your application and all its dependencies.

**Run a Docker container**: After you've built a Docker image, you can use the docker run command to create and start a new container. You can specify options like port forwarding, environment variables, and volumes to customize the container.

#### 28.

What are Anti-Patterns in DevOps, and how to avoid them?

Hide Answer

In DevOps, and the overall software development process, a pattern refers to the path of solving a problem. Contrary to that, an anti-pattern is a pattern a DevOps team uses to fix its short-term problem, risking long-term goals as an anti-pattern is typically ineffective and results in counterproductiveness.

Some examples of these anti-patterns include:

**RCA** - Root cause analysis is the process of determining the root cause of an issue and the appropriate action needed to avoid its recurrence.

**Blame culture** - This involves blaming and punishing those responsible when a mistake occurs.

**Silos** - A departmental or organizational silo states the mentality of a team that doesn’t share their expertise with another team within the organization.

Apart from these, various anti-patterns can exist in DevOps. You can take these steps to avoid them:

* Structuring teams correctly and adding the vital processes needed for their success. This also includes offering the required resources, information security, and technology to help them attain the best results.
* Clearly defining roles and responsibilities for every team member. It ensures each member follows the plans and strategies in place so that managers can effectively monitor timelines and make the right decisions.
* Implementing continuous integration, including security scanning, automated regression testing, code reviews, open-source license and compliance monitoring, and continuous deployment, including quality control, development control, and production. This is an effective solution to fix underlying processes and avoid anti-patterns.
* Establishing the right culture by enforcing important DevOps culture principles, such as openness to failure, continual improvement, and collaboration.

#### 29.

Clarify the usage of Selenium for automated testing

Hide Answer

Selenium is a popular open-source tool used for automating web browsers. It can be used for automated testing to simulate user interactions with a web application, such as clicking buttons, filling in forms, and navigating between pages. Selenium provides a suite of tools for web browser automation, such as Selenium WebDriver, Selenium Grid, among others.

Selenium WebDriver is the most commonly used tool in the Selenium suite. It allows developers to write code in a variety of programming languages, such as Java, Python, or C#, to control a web browser and interact with web elements on a web page. This allows for the creation of automated tests that can run repeatedly and quickly without requiring manual intervention.

#### 30.

What is auto-scaling in Kubernetes?

Hide Answer

Autoscaling is one of the vital features of Kubernetes clusters. Autoscaling enables a cluster to increase the number of nodes as the service response demand increases and decreases the number when the requirement decreases. This Kubernetes feature is only supported in Google Container Engine (GKE) and Google Cloud Engine (GCE).

#### 31.

Explain EUCALYPTUS.

Hide Answer

It is another common DevOps interview question that you can answer briefly. However, answering it correctly can show your firm grasp of the overall DevOps domain. EUCALYPTUS stands for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems. EUCALYPTUS is typically used with DevOps tools like Chef and Puppet.

#### 32.

List the three key variables that affect inheritance and recursion in Nagios.

Hide Answer

* Name - This is the placeholder used by other objects.
* Use - This defines a parent object whose properties are to be used.
* Register - This can have a 0 or 1 value. 0 indicates that it's just a template, while 1 indicates that it’s an actual object.

#### 33.

How to use Istio for service mesh?

Hide Answer

Istio comes with two distinct components to be used for service mesh: the data plane and the control plane. Here’s how it works:

**Data plane**: The data plane refers to the communication between each service. Without the service mesh, the network fails to understand the traffic it receives and can’t make decisions based on the traffic type it has, who it is from, or who it’s being sent to.

Service mesh employs a proxy to intercept the network traffic, enabling various application-aware features based on pre-set configurations. Then, an Envoy proxy is deployed with every service started in the cluster, which can also run alongside a VM service.

**Control plane**: The control plane then interprets the desired configuration, its service views, and programs the proxy servers. It also updates them as the environment or configuration changes.

#### 34.

List the branching strategies you’ve used previously.

Hide Answer

This DevOps interview question is asked to check your branching experience. You can explain how you’ve used branching in previous roles. Below are a few points you can refer to:

**Feature branching** - Feature branch models maintain all the changes for specific features inside a branch. Once automated tests are used to fully test and validate a feature, the branch is then merged with master.

**Release branching** - After a develop branch has enough features for deployment, one can clone that branch to create a release branch that starts the next release cycle. Hence, no new features can be introduced after this; only documentation generation, bug fixes, and other release-associated tasks can enter this branch. Once it’s ready for shipping, the release branch merges with the master and receives a version number.

**Task branching** - This model involves implementing each task on its respective branch using the task key in the branch name, which makes it easier to check which code performs which task.

#### 35.

Explain the usage of Grafana for data visualization.

Hide Answer

Grafana is a popular open-source data visualization and monitoring tool used to create interactive dashboards and visualizations for analyzing data. It can connect to a wide variety of data sources, including databases, cloud services, and third-party APIs, and allows users to create customized visualizations using a range of built-in and community-contributed plugins.

Some primary Grafana features include integrating data sources, creating dashboards and data visualizations, and generating notifications.

#### 36.

How do you use Elasticsearch for log analysis?

Hide Answer

Elasticsearch is a powerful tool for log analysis that allows you to easily search, analyze, and visualize large volumes of log data. Here are the basic steps for using Elasticsearch for log analysis:

**Install and configure Elasticsearch**: First, you must install Elasticsearch on your machine or server. Once installed, you need to configure Elasticsearch by specifying the location where log data will be stored.

**Index your log data**: To use Elasticsearch for log analysis, you must index your log data. You can do this by using Elasticsearch APIs or by using third-party tools such as Logstash or Fluentd.

**Search and analyze your log data**: Once your log data is indexed, you can search and analyze it using Elasticsearch's query language. Elasticsearch provides a powerful query language that allows you to search for specific log entries based on various criteria such as time range, severity level, and keyword.

**Visualize your log data**: Elasticsearch also provides built-in visualization tools that allow you to create charts, graphs, and other visual representations of your log data. You can use these tools to identify patterns and trends in your log data and to monitor the health and performance of your systems.

#### 37.

How do you use Kubernetes for rolling updates?

Hide Answer

The following steps can be followed to use Kubernetes for rolling updates:

* Create a yaml file containing deployment specifications through a text editor, like Nano.
* Save the file and exit.
* Next, use ‘kubect1 create’ command and the yaml file to create the deployment.
* Use the ‘kubect1 get deployment’ command to check the deployment. The output should indicate that the deployment is good to go.
* Next, run the ‘kubect1 get rs’ command to check the ReplicaSets.
* Lastly, check if the pods are ready. Use the ‘kubect1 get pod’ command for this.

#### 38.

What are the benefits of A/B testing?

Hide Answer

Following are the major benefits of implementing A/B testing:

**Improved user engagement**: Various elements of a website, ad, app, or platform can be a/b tested, including headlines, fonts, buttons, colors, and more. These changes can help understand which ones are increasing user responses and can be further implemented to move towards business success.

**Decreased bounce rates**: A/b testing can also help understand what needs to be optimized to keep visitors on the app or website for as long as possible. Various elements such as the images, texts, CTAs, etc. can be tested to assess which changes help lower bounce rates.

**Risk minimization**: If the team is unaware of how a new element or feature will perform, a/b tests can be performed to check how it affects the system and what user reactions it gets. This way, a/b testing also helps to minimize risk and roll back features/elements/code if they have a larger negative impact.

**Better content**: Using a/b testing, the content of a website or application can be tested to check if its getting the desired responses or if there’s anything that is ineffective and needs to be eliminated. This helps in creating final versions that comprise effective content for end users.

**Increased conversion rates**: Ultimately, making changes and running a/b tests to see which ones work best help in creating the best possible final version of a product that gets more purchases, sign-ups, or other conversion-related numbers.

#### 39.

Give a complete overview of Jenkins architecture.

Hide Answer

Jenkins uses a Master/Slave architecture for distributed build management. This has two components: the Jenkins server, and the Jenkins node/build/slave server. Here’s what the architecture looks like:

* The Jenkins server is a war file-powered web dashboard. Using this, you can configure projects/jobs. However, the build takes place in the slave/nodes. By default, only one nodes/slave runs on the Jenkins server, but you can add more using a username, IP address, and password through the ssh/jnlp/webstart.
* The key Jenkins server is the master, whose job is to manage the scheduling of build jobs, deploying builds to slaves for execution, monitoring slaves, and recording and presenting build results. In a distributed architecture, a Jenkins master can also execute build jobs by itself.
* As for the slaves, their task is to do as per their configurations in the Jenkins server, which includes executing build jobs sent by the master. Teams can also configure projects to continuously run on specific slave machines or a particular type of slave machines or just let Jenkins select the next available slave.

### Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

[Apply Now](https://developers.turing.com/?referrer=https://www.google.com/)

### ADVANCED DEVOPS INTERVIEW QUESTIONS AND ANSWERS

#### 1.

Explain the Shift Left to Reduce Failure concept in DevOps.

Hide Answer

In the SDLC, the left side implicates planning, design, and development, while the right indicates production staging, stress testing, and user acceptance. Talking about DevOps, shifting left means undertaking as many tasks that occur at the end of the SDLC as possible into the earlier stages. In doing so, the chances of facing errors during the later stages of software development is greatly reduced as they’re identified and rectified in the earlier stages.

In this approach, the operations and development team work side by side when building the test case and deployment automation. This is done as failures within the production environment aren’t observed earlier quite often. Both the operations and development teams are expected to assume ownership in the development and maintenance of standard deployment procedures by using cloud and pattern capabilities. This helps to ensure that production deployments are successful.

#### 2.

What are ‘post mortem’ meetings in DevOps?

Hide Answer

Post mortem meetings refer to those that are scheduled for discussing the things that have gone wrong while adopting the DevOps methodology. During such meetings, the team is expected to discuss steps that need to be taken to avoid the recurrence of such failures in the future.

#### 3.

Explain the concept of ‘pair programming’.

Hide Answer

This is an advanced DevOps interview question that recruiters often ask to check a candidate’s expertise. Hence, knowing the proper answer to this question can help you advance further in the interview.

Pair programming is a common engineering practice wherein two programmers operate on the same design, system, and code. The two follow the ‘Extreme Programming’ rules, where one programmer is the ‘driver,’ and the other is the ‘observer’, who thoroughly monitors the project progress to determine further bottlenecks for immediate rectification.

#### 4.

What is the dogpile effect, and how can it be avoided?

Hide Answer

The dogpile effect is also known as cache stampede. This usually occurs when massive parallel computing systems using caching strategies face extremely high load. Dogpile effect is referred to as the event that happens when the cache invalidates or expires, and various requests hit the website at the same time.

The most common approach to avoid dogpiling is by putting semaphore locks in the cache so that when it expires, the first process to get the lock will create the new value to the cache.

#### 5.

How can you ensure a script runs successfully every time the repository receives new commits through Git push?

Hide Answer

There are three ways of setting up a script to get executed when the destination repository receives new Git push commits. These are called hooks and their three types include:

**Pre-receive hook** - This is invoked before references are updated while commits are pushed. The pre-receive hook helps ensure the scripts required for enforcing development policies are executed.

**Update hook** - This triggers script running before updates are actually deployed. This hook is used once for every commit pushed to the destination repository.

**Post-receive hook** - This triggers the script after the changes or updates have been sent to and accepted by the repository. The post-receive hook is ideal for configuring email notification processes, continuous integration-based scripts, deployment scripts, etc.

#### 6.

Explain how the canary deployment works.

Hide Answer

In DevOps, canary deployment refers to the deployment strategy pattern that focuses on minimizing the impact of potential bugs in a new software update or release. This sort of deployment involves releasing updates to only a small number of users before making them available universally.

In this, developers use a load balancer or router to target singular routes with the new release. After deployment, they collect the metrics to assess the update’s performance and make a decision on whether it’s ready to be rolled out for a larger audience.

#### 7.

List the major differences between on-premises and cloud services in DevOps.

Hide Answer

On-premises and cloud services are the two primary data hosting pathways used by DevOps teams. With cloud services, the team hosts data remotely using a third-party provider, whereas on-premises services involve data storage in the organization’s in-house servers. The key differences between the two include:

* Cloud services usually offer less security control over infrastructure and data. However, they provide extra services, scale better, and incur lower expenses.
* On-premises services entail unique security threats and massive maintenance costs, but they offer a bigger customization scope and better control.

#### 8.

How can you ensure minimum or zero downtime when updating a live heavy-traffic site?

Hide Answer

This is an uncommon Devops interview question but can be asked by managers gauging your DevOps expertise at the advanced level. Here are the best practices to maintain minimum or zero downtime when deploying a live website’s newer version involving heavy traffic:

**Before deploying on a production environment**

* Rigorously testing the new changes and ensuring they work in a test environment almost similar to the production system.
* Running automation of test cases, if possible.
* Building an automated sanity testing script that can be run on production without impacting real data. These are usually read-only test cases, and depending on the application needs, developers can add more cases here.
* Creating scripts for manual tasks, if possible, avoiding human errors during the day of deployment.
* Testing the scripts to ensure they work properly within a non-production environment.
* Keeping the build artifacts ready, such as the database scripts, application deployment files, configuration files, etc.
* Rehearsing deployment, where the developers deploy in a non-production environment almost identical to the production environment.
* Creating and maintaining a checklist of to-do tasks on deployment day.

**During deployment**

* Using the green-blue deployment approach to avoid down-time risk.
* Maintaining a backup of current data/site to rollback when necessary.
* Implementing sanity test cases before running depth testing.

#### 9.

Which one would you use to create an isolated environment: Docker or Vagrant?

Hide Answer

In a nutshell, Docker is the ideal option for building and running an application environment, even if it's isolated. Vagrant is a tool for virtual machine management, whereas Docker is used to create and deploy applications.

It does so by packaging an application into a lightweight container, which can hold almost any software component and its dependencies, such as configuration files, libraries, executables, etc. The container can then execute it in a repeatable and guaranteed runtime environment.

#### 10.

What is ‘session affinity’?

Hide Answer

The session affinity technique, also known as a sticky session, is a popular load-balancing technique requiring an allocated machine always to serve a user session. When user information is stored in a load balancer server app during a session, the session data will be needed to be available to all machines.

However, this can be avoided by continuously serving a user session request from a single machine, which is associated with the session as soon as it’s created. Every request in the particular session redirects to the associated machine, ensuring that the user data is housed at a single machine and the load is shared as well.

Teams typically do this through a SessionId cookie, which is sent to the client upon the first request, and every subsequent client request must contain the same cookie for session identification.

#### 11.

Write a simple Python script to fetch Git repository information using the GitHub API.

Hide Answer

#### 12.

How are SOA, monolithic, and microservices architecture different?

Hide Answer

The monolithic, microservices architecture and service-oriented architecture (SOA) are quite different from one another. Here’s how:

* The monolithic architecture is like a large container where an application’s software components are assembled and properly packaged.
* The microservices architecture is a popular architectural style that structures applications as a group of small autonomous services modeled as per a business domain.
* The SOA is a group of services that can communicate with each other, and this typically involves either data passing or two or more services coordinating activities.

#### 13.

How can you create a backup and copy file in Jenkins?

Hide Answer

This is a rather simple DevOps interview question. To create a backup, you would need to periodically backup the JENKINS\_HOME directory. This directory houses all the build configurations, the slave configurations, and the build history.

To backup the Jenkins setup, you would simply need to copy the directory. Furthermore, a job directory can also be copied to replicate or clone a job or rename the directory itself.

Additionally, you can also use the "Thin Backup" plugin or other backup plugins to automate the backup process and ensure that you have the latest backup available in case of any failure.

#### 14.

What are Puppet Modules, and how are they different from Puppet Manifests?

Hide Answer

A Puppet Module is just a data collection (facts, templates, files, etc.) and manifests. These Modules come with a particular directory structure and help organize Puppet codes, as they can be used to split the code into various manifests. Using Puppet Modules to organize almost all the Puppet Manifests is considered the best practice. Puppet Modules differ from Manifests as the latter is just Puppet programs comprising Puppet code

#### 15.

Write a simple Terraform configuration to create an AWS S3 bucket.

Hide Answer

#### 16.

Why should Nagios be used for HTTP monitoring?

Hide Answer

Nagios can provide the full monitoring service for HTTP servers and protocols. Some of the key benefits of conducting efficient HTTP monitoring using Nagios are as follows:

* Application, services, and server availability can be increased drastically.
* User experience can be monitored well.
* Protocol failures and network outages can be detected as quickly as possible.
* Web transactions and server performance can be monitored.
* URLs can be monitored as well

#### 17.

Explain the various components of Selenium.

Hide Answer

**IDE** - The Selenium IDE (integrated development environment) comprises a simple framework and a Firefox plugin that you can easily install. This component is typically used for prototyping.

**RC** - The Selenium RC (remote control) is a testing framework that quality analysts and developers use. This component supports coding in almost any programming language and helps to automate UI testing processes of web apps against a HTTP website.

**WebDriver** - The Selenium WebDriver offers a better approach to automating the testing of web-based apps and doesn’t rely on JavaScript. This web framework also allows the performance of cross-browser tests.

**Grid** - The Selenium Grid is a proxy server that operates alongside the Selenium RC, and using browsers, it can run parallel tests on various machines or nodes.

#### 18.

Is Docker better than virtual machines? Explain why.

Hide Answer

Docker has several advantages over virtual machines, making it the better option between the two. These include:

**Boot-up time** - Docker comes with a quicker boot-up time than a virtual machine.

**Memory space** - Docker occupies much lesser space than virtual machines.

**Performance** - A Docker container offers better performance as it is hosted in a single Docker engine. Contrarily, performance becomes unstable when multiple virtual machines are run simultaneously.

**Efficiency** - Docker’s efficiency is much higher than that of a virtual machine.

**Scaling** - Docker is simpler to scale up when compared to virtual machines.

**Space allocation** - One can share and use data volumes repeatedly across various Docker containers, unlike a virtual machine that cannot share data volumes.

**Portability** - Virtual machines are known to have cross-platform compatibility bottlenecks that Docker doesn’t.

#### 19.

Explain the usage of SSL certificates in Chef.

Hide Answer

SSL (Secure Sockets Layer) certificates are used to establish a secure communication channel between a Chef server and its client nodes. Chef uses SSL certificates to encrypt the data that is transmitted between the server and the clients, ensuring that sensitive data, such as passwords and configuration data, are protected from unauthorized access.

An SSL certificate is needed between the Chef server and the client to ensure that each node can access the proper data. When SSL certificates are sent to the server, the Chef server stores the public key pair of every node. The server then compares this against the public key to identify the node and provide access to the required data.

#### 20.

What is ‘state stalking’ in Nagios?

Hide Answer

State stalking is a process used in Nagios for logging purposes. If stalking is enabled for a specific service or host, Nagios will “stalk” or watch that service or host carefully. It will log any change it observes in the check result output, which ultimately helps to analyze log files.

#### 21.

List the ways a build can be run or scheduled in Jenkins.

Hide Answer

There are four primary ways of scheduling or running a build in Jenkins. These are:

* Using source code management commits
* After completing other builds in Jenkins
* Scheduling the build to run at a specified time in Jenkins
* Sending manual build requests

#### 22.

Write a simple Jenkins pipeline script to build and deploy a Docker container.

Hide Answer

#### 23.

List the steps to deploy the custom build of a core plugin.

Hide Answer

The following steps can be followed to deploy a core plugin’s custom build effectively:

* Start by copying the .hpi file to the $JENKINS\_HOME/plugins.
* Next, delete the plugin’s development directory.
* Create an empty file and name it .hpi.pinned.
* Restart Jenkins, and your custom build for the core plugin will be ready.

#### 24.

How can you use WebDriver to launch the Browser?

Hide Answer

he WebDriver can be used to launch the three different Browsers by using the following commands:

* For Chrome - WebDriver driver = new ChromeDriver();
* For Firefox - WebDriver driver = new FirefoxDriver();
* For Internet Explorer - Webdriver driver = new InternetExplorerDriver();

It is worth noting that the specific code required to launch a browser using WebDriver may vary depending on the programming language being used and the specific environment setup. Additionally, the appropriate driver executable for the specific browser being used must be installed and configured correctly.

#### 25.

How can you turn off auto-deployment?

Hide Answer

Auto-deployment is a feature that helps to determine if there are any new changes or applications in existing applications and dynamically releases them. It is typically enabled for servers running in development mode. The following method can be use to turn it off.

* Click the domain name in the Administration Console (located in the left pane) and tick the Production Mode checkbox.
* Include the following argument at the command line when you start the domain’s Administration Server: -Dweblogic.ProductionModelEnabled=true
* Production mode will be set for the WebLogic Server instance in the domain.

#### 26.

How are distributed and centralized version control systems different?

Hide Answer

#### 27.

Write a sample GitLab CI/CD YAML configuration to build, test, and deploy a Node.js application.

Hide Answer

#### 28.

Explain the process of setting up a Jenkins job

Hide Answer

A Jenkins job can be set up by going to Jenkins’ top page and selecting ‘New Job.’ Then, we can select ‘Build a free-style software project,’ where we can choose the elements for this job:

* Optional triggers to control when Jenkins performs a build.
* Optional SCM, like Subversion or CVS, where the source code will be housed.
* A build script that will perform the build (maven, ant, batch file, shell script, etc.)
* Optional steps to inform other systems/people about the build result, like sending emails, updating issue trackers, IMs, etc.
* Optional steps to gather data out of the build, like recording javadoc and/or archiving artifacts and test results.

Depending on your Jenkins configuration, you may need to configure Jenkins agents/slaves to run your build. These agents can run on the same machine as Jenkins or on a different machine altogether. Once you have configured your job, you can save it and trigger it manually or based on the selected trigger.

#### 29.

How to ensure security in Jenkins? What are the three security mechanisms Jenkins can use to authenticate a user?

Hide Answer

The following steps can be taken to secure Jenkins:

* Ensuring that ‘global security’ is switched on.
* Ensuring that Jenkins is integrated with the organization’s user directory using the appropriate plugins.
* Automating the process of setting privileges or rights in Jenkins using custom version-controlled scripts.
* Ensuring that the matrix or Project matrix is enabled for fine-tuning access.
* Periodically running security audits on Jenkins folders or data and limiting physical access to them.

The three security mechanisms Jenkins can use to authenticate a user are:

* Jenkins utilizes internal databases to store user credentials and data responsible for authentication.
* Jenkins can use the LDAP (lightweight directory access protocol) server to authenticate users as well.
* Teams can also configure Jenkins to use the authentication mechanism used by the deployed application server.

#### 30.

Write a Python script to create an AWS Lambda function using Boto3.

Hide Answer

#### 31.

How do the Verify and Assert commands differ in Selenium?

Hide Answer

* The Assert command in Selenium helps check if a provided condition is true or false. For instance, we assert whether a given element is existent on the web page or not. If the condition says true, then the program control will run the next test step, but if it’s false, the execution will come to a halt, and no test will be executed further.
* The Verify command also evaluates if a given condition is false or true, but irrespective of the condition, the program execution doesn’t stop. This means any verification failure won’t halt the execution, and test steps will continue to be executed.

#### 32.

Write a sample Packer configuration to build an AWS AMI.

Hide Answer

#### 33.

How does Nagios assist in distributed monitoring? How does Flap Detection work in Nagios?

Hide Answer

In this DevOps interview question, the interviewer may expect an answer explaining Nagios's distributed architecture. Using Nagios, teams can monitor the entire enterprise by employing a distributed monitoring system where local Nagios slave instances run monitoring tasks and report results to a single master.

The team can manage all the notifications, configurations, and master reporting, while the slaves perform all the work. This particular design leverages Nagios’ ability to use passive checks (external processes or applications that revert results to Nagios).

Flapping happens when a host or service changes state frequently, causing numerous problems and recovery notifications. Whenever Nagios checks a service or host status, it will try to see if it has begun or stopped flapping. The procedure mentioned below is what Nagios follows:

* Storing the last 21 check results of the service or host, assessing the historical check results, and identifying where state transitions/changes occur.
* Using the state transitions to find the percent state change value for the service or host.
* Comparing the percent state change value against high or low flapping thresholds.

#### 34.

What is the main configuration file of Nagios and where is it located?

Hide Answer

Nagios’ main configuration file comprises various directives that impact how a Nagios daemon runs. This configuration file is ready by the daemon and the CGIs (they specify the main config file’s location). You can define how the file is created and where it is located.

When you run a configure script, a sample main config file is created in the Nagios distribution’s base directory. The file’s default name is nagios.cfg and it’s usually located in the etc/ subdirectory of the Nagios installation (/usr/local/bagios/etc/).

#### 35.

Explain the role of a service mesh in the DevOps context and provide an example using Istio.

Hide Answer

A [service mesh](https://www.turing.com/blog/service-mesh-how-to-overcome-deployment-challenges/) is a configurable infrastructure layer for the microservices application that makes communication between service instances flexible, reliable, and fast. The purpose is to handle the network communication between microservices, including load balancing, service discovery, encryption, and authentication.

In DevOps, it simplifies network management, promotes security, and enables advanced deployment strategies. Istio is a popular service mesh that integrates with Kubernetes.

#### 36.

Write a DaemonSet configuration manifest in Kubernetes.

Hide Answer

#### 37.

Write an example of AWS CloudFormation YAML template to create an S3 bucket and an EC2 instance.

Hide Answer