Innovative Senior DevOps Engineer with a strong Linux background and 15+ years of experience designing, implementing, and managing cutting-edge deployment automation of cloud resources.

Q #1) Explain DevOps.

DevOps promotes communication and collaboration between business, development & operations teams. In a way, we are talking about unification.

DevOps is not only about tools but also about involving people and processes.

DevOps is inspired by the Agile process.

DevOps is about automating the development, release and operation processes.

DevOps helps in the speed of delivering applications to the end-users.

DevOps is also about continuous improvement which helps in learning through feedback.

Q #2) Explain a few prerequisites that are useful for DevOps implementation.

Answer: Pre-requisites includes:

Commitment at the senior level in the organization.

Need for change to be communicated across the organization.

Version control software.

Automated tools for compliance to process.

Automated Testing

Automated Deployment

Q #3) What are the best practices for DevOps implementation?

Answer: DevOps implementation varies from one organization to another organization. Today organizations are looking to deliver the software faster and from that point of view here are my thoughts on how DevOps can bring about efficient communication with all the stakeholders including the operations team.

Every organization has certain business goals and DevOps implementation should align with it. There should be a certain need for change.

Encouraging communication and collaboration especially between development and operations.

Automation is the key and should be done and carried out wherever possible within the SDLC stages to facilitate the working of DevOps.

The tools used in DevOps should integrate very well. From the point of view of any commercial tools like IBM Rational CLM or Microsoft TFS, the implementation of DevOps is good as it is from a single vendor. Similarly, one needs to look at open-source tools that integrate to give similar desired results.

DevOps implementation is not successful without Continuous Integration and Continuous Delivery practices. So CI of your code with rigorous testing and doing CD is the key to it.

It is very important for the Ops team to ensure that the applications are working very well at appropriate levels. If required they will need to work with the development teams to build any tools that would help to build the right monitoring capabilities into the applications.

Finally, encourage feedback from end-users to enable continuous improvement which provides the key for improving the process and delivering quality software.

Q #4) What are the key components of DevOps?

Continuous Integration

Continuous Testing

Continuous Delivery

Continuous Monitoring

Q #5) Explain Continuous Integration.

Answer: Continuous Integration is a very important component of the Agile process. Typically, developers work on features or user stories within a sprint and commit their changes to the version control repository.

Once the code is committed, then the entire work of developers is well integrated and the build is performed on a regular basis based on every check-in or schedule. Hence, Continuous Integration as a practice forces the developer to integrate their changes with the others so as to get early feedback.

Q #6) Explain Continuous Delivery.

Answer: Continuous Delivery is an extension of Continuous Integration which primarily helps to get the features that the developers are developing out to the end-users as soon as possible. During this process, it goes through various stages of QA, Staging, etc., and then for delivery to the PRODUCTION system.

Q #7) Explain Continuous Testing.

Answer: From the above goal of Continuous Integration which is to get the application out to end-users are primarily enabling continuous delivery. This cannot be completed without a sufficient amount of unit testing and automation testing.

Hence, we need to validate that the code produced and integrated with all the developers who perform as required.

Q #8) Explain Continuous Monitoring.

Answer: As the application is developed and deployed, we do need to monitor its performance. Monitoring is also very important as it might help to uncover the defects which might not have been detected earlier.

Q #9) How will you approach when a project needs to implement DevOps?

Answer: Following approaches can be used but that will vary based on the actual project scenarios or from organization to organization

Stage 1: For any DevOps implementation, an assessment of the existing process and implementation for around 2 to 3 weeks taking an average of 5 applications is a must to gauge the AS-IS process, identifies the improvement areas and provides a roadmap for the implementation.

Typically a senior profile like a DevOps architect should be involved to conduct this assessment.

Stage 2: A pilot POC can be done to showcase the end to end working of DevOps. Once this has been accepted and approved by the end-users only then the actual implementation, handover and rolls out the plan for the projects involved should be carried out.

Any pieces of training required will also need to be imparted to the process or tools. DevOps architects, engineers, and testers would need to be involved at this stage.

Stage 3: The projects should now be in DevOps mode with all the components of continuous integration/delivery/testing and monitoring that is being followed.

Q #10) Can DevOps be applied to a Waterfall process? Explain the significance of the Agile process in DevOps implementation.

Answer: In the waterfall process, as all of us are aware initially complete Requirements are gathered, next the System is designed, Implementation of the System is then done followed by System testing and deployed to the end-users. In this process, the problem was that there was a huge waiting time for build and deployment which made it very difficult to get the feedback.

The solution to the above problem was that the Agile process has to bring in agility in both development and operations. The agile process could be the principal or a certain pre-requisite may be required for DevOps implementation. DevOps goes hand in hand with the Agile process.

The focus area is to release the software in a very timely manner with shorter release cycles and quick feedback. So, the agile process focus will mainly be on speed and in DevOps, it works well with the automation of various tools

Q #11) What is the difference between Continuous Delivery and Continuous Deployment?

Answer: In an Agile Sprint, For Example, there are many features or user stories that are developed, tested and ready for deploying. But based on customer scenarios and priorities not all would be deployed. So, here in continuous delivery, it is very important to keep the code readily available for deployment.

In Continous Deployment, all the changes developed by the developer goes through various stages to be deployed to the PRODUCTION environment in an automated way.

Q #12) What is your expertise on the DevOps projects?

Answer: Explain your role as a DevOps Engineer and how you were working as a part of the 24\*7 environment and maybe in shifts, the projects involved in automating the CI and CD pipeline and providing support to the project teams.

Hence, taking complete responsibility for maintaining and extending the environments for DevOps automation to more and more projects and different technologies (Example: .NET, J2EE projects) involved within the organization.

Also, explain the process (Example Agile) and tools that were involved in an end to end automation. You could also talk about your experience, if any, in DevOps support over the Cloud environment.

Q #13) What are the top 10 DevOps tools that are used in the industry today?

Jira

GIT/SVN

Bitbucket

Jenkins

Bamboo

SonarQube

Artifactory/Nexus

Docker

Chef / Puppet /Ansible

IBM Urbancode Deploy / CA-RA

Nagios / Splunk

Q #14) Can you explain the uses of the tools mentioned in the above question and how they connect to give a DevOps model (CI/CD)?

Planning

Jira – Used for Project Planning and Issue management

Continuous Integration

Git – Version Control

Jenkins – Open Source Continuous Integration tool which can also help in Continuous Delivery.

SonarQube – Code Analysis

JFrog Artifactory – Binary Repository Manager

Continuous Delivery

Chef / Puppet / Ansible – Configuration Management and Application Deployment

IBM Urbancode Deploy / CA RA – Continuous Delivery

Continuous Monitoring

Nagios / Splunk

Sample DevOps Workflow:

Sample Devops workflow

Typically in an Agile process user stories, tasks, defects, etc., are all stored in JIRA and assigned to the Product Owners and Developers.

Developers pick up the tasks assigned to them and work on the development. The source code is version controlled and stored in GIT. The developers commit their changes to the source code in GIT. Eventually, the code is shared among the developers using GitHub.

Jenkins which is the Continuous Integration tool pulls the code and on every check-in or based on a schedule the build is done using build tools like Maven or ANT.

As the J2EE WAR files are produced, they are also version controlled and stored in a binary repository manager like Artifactory or Nexus.

Unit Testing using JUnit and Code Analysis with SonarQube is also done and automated

Once the above process is completed the Continuous Delivery is performed to different environments based on approvals using tools like IBM UrbanCode Deploy / CA RA Continuous Testing (Functional and Acceptance Testing) is invoked in the appropriate test environments using tools like Selenium

Continuous Monitoring would be an ongoing activity in the PROD environment

Q #15) What is configuration management in terms of infrastructure and mention a few popular tools used?

Answer: Configuration management consists of practices and the various tools involved to automate the delivery and infrastructure operations. It is all about keeping the server ready (Example Installing system packages, network configuration settings) for application deployment once the application is developed.

So the Ops or the system admin needs to ensure parity in different environments (Dev, QA, PROD, etc…) by provisioning the systems.

Tools Used in this space to automate the tasks of configuration management described above are Chef / Puppet / Ansible

Q #16) Which scripting tools are used in DevOps?

Answer: Python, Bash

Q #17) Explain the typical roles involved in DevOps.

DevOps Architect: The leader who is responsible for the entire DevOps process.

DevOps Engineer: The person should be experienced with Agile, SCM or Version Control, CI /CD and setting up automation tools for the same, Infrastructure automation and Database management skills. Any developer who has skills in coding or scripting and has the acumen to get into deployment or system admin can qualify for the role of a DevOps engineer.

Q #18) Have you been involved in DevOps implementation in the cloud? If yes which cloud computing platform?

Answer: Examples of popular Cloud Computing Platforms:

AWS – Amazon Web Services

Microsoft Azure

Google Cloud

Q #19) Explain some of the metrics that were followed for DevOps success.

Some of the examples are as follows:

The first and most important factor is the speed of delivery which means time taken for any work item to get into the production environment.

Next would be the deployment and how much time it would take once this process is automated.

It is almost necessary to track how many defects are found in different environments with respect to the PRODUCTION environment. This is very important in considering the features that need to be released faster. The use of Agile methodologies helps a lot and the prime goal is to reduce PRODUCTION level defects.

Normally deployments do not fail but it is very important to keep a track on this aspect and have a mechanism to roll back to the previous stable version.

In any DevOps implementation, unit testing is the key as well as functional testing. Based on the code changes done, often we need to look at whether these test breaks and to what extent. It is imperative that the automated test is robust enough to sustain any code changes.

It is very important to measure the actual or the average time that it takes to recover in case of a failure in the PRODUCTION environment. This is termed as Mean Time To Recover (MTTR) and it should be short. This also means that one needs to have proper monitoring tools to keep recovery time short.

Performance of the application is another key metric that should be monitored especially after any deployments are done.

A very important factor for success is the number of bugs being reported by the customers which primarily depends on the quality of the application.

Q #20) What are your expectations from a career perspective of DevOps?

Answer: To be involved in the end to end delivery process and the most important aspect of helping to improve the process so as to enable the development and operations teams to work together and understand each other’s point of view.

Top DevOps Interview Questions and Answers

The following section features the top DevOps Interview Questions and Answers that will prepare you for your next DevOps interview:

**Question: What are the different phases in DevOps?**

**Answer:**DevOps is mainly classified into 6 phases. Its phases are in a particular cycle. However, all the phases are not separated by boundaries, and no phase begins even if the previous one has ended completely. Now, let's study the DevOps cycle phase in more detail.

**1. Planning**

Planning and software development is the first phase of the DevOps lifecycle. This phase involves understanding the project properly for the ultimate work goal of its participants. This also feeds the various phases of development and operations. It's also important that organizations are trained on tools and metrics to have enough clarity of the project's management.

**2. Development**

In this phase, the project gets built by designing infrastructure, writing codes, defining tests, or by automation process. Evidence is important in this phase. In managing applications, operations with data, developers store codes in code manager, which allows viewing, versioning, and much more.

**3. Continuous Integration**

This phase automates the mechanism of validation, testing. This has a unique feature that ensures the development environment is properly and then published in a service that integrates it with the remaining applications.

**4. Automated Deployment**

DevOps stimulates the automation of deployments by tools and scripts with the ultimate goal of solving the whole process with the activation of a feature. The most important aspect of this phase is the arrival of the cloud, as a code that forces a change from finite infrastructure management to permanent cost optimization management.

**5. Operations**

Usually, all operations related to DevOps happen continuously throughout the life of software, as there is a dynamic change in the infrastructure. This platform provides opportunities for transformation, availability, and scalability.

**6. Monitoring**

This phase is a permanent phase of the DevOps process. DevOps monitors and analyzes information that displays the current status of the application.

**Question: Why has DevOps gained popularity over the past few years?**

**Answer:** Nowadays, DevOps are in great demand in the current industry, and many businesses are eagerly wanting to invest in DevOps talent. Some of the huge multi-national companies such as Facebook and Netflix are investing their money and time in DevOps for automation and pacing up application deployment as every large industry wants to see some automation in the coming years. It helps the organizations to grow and expand their businesses to generate large revenues. Its popularity continues to grow in demand as tech competition increases as most companies start adopting DevOps practices; then, it becomes even more important for the competitors to invest in similar or better development practices, increasing demand.

DevOps implementation has given provable results in businesses which contend higher efficiency, with its new technology standards; tech workers can implement codes faster than ever before, and with lesser errors. As now, more consumers and businesses rely on cloud software as it requires fast deployments to meet the consumer needs without interrupting services; this increases user adoption of cloud software like DevOps over the years.

**Question: What is the difference between Ansible and Puppet?**

**Answer:**Today, DevOps professionals have to manage and control a huge number of servers hosting, so for this, they need exponential growth in computing as well as new technology such as virtualization and cloud computing. Thus Puppet and Ansible are the tools that are used for managing a large number of servers.

These are also called Remote Execution and Configuration Management tools, and it allows the admin to perform or execute the commands on many servers simultaneously. Its main feature is generally to maintain and configure thousands of servers at a single time. Apart from this, Ansible and Puppet has major differences right from the moment and can be differentiated concerning many mechanisms as shown below:

**Category**

1. Ansible
2. Puppet

**Scalability**

1. Scalability in Ansible is very convenient and simple.
2. Puppet also offers scalability but somewhere lacks as compared to ansible

**Management and Scheduling**

1. In ansible, configuration gets pushed to the nodes from the server for better employment of code.
2. In puppet, configuration gets pulled from the selected server.

**Language**

1. It is fully published in Python and uses YAML syntax to convey or write configurations.
2. It is written in Ruby and uses declarative language to form the configurations.

**Availability in case of failures**

1. In this case, availability will have lesser worries as the subordinate node is present in case of any nodal failure.
2. In this case, many multiple master servers are present so that if the original master fails, it does not stop the ongoing task.

**Repository**

1. The repository of Ansible is Ansible Galaxy, where all the information is stored.
2. A storehouse of the puppet is Puppet forge, which has 6000 modules.

**Setting up and Usage**

1. Ansible has a master that runs the client machine and writes the configuration to manage tasks.
2. Puppet uses a client-server architecture to manage complex tasks.

**Question: What are the benefits of using the Version Control System (VCS)?**

**Answer:**The key benefits of Version Control are as follows:

1. With the Version Control System (VCS), all the workers are allowed to access the file freely at any time. It also allows merging all the changes that are made in a common version.
2. It is designed to help multiple people by collaboratively edit text files, which makes sharing comparatively easy between multiple computers.
3. It is important for documents that require a lot of redrafting and revision as they provide an audit trail for redrafting and updating final versions.
4. It permits all the team members to have access to the complete history of the project so that in case of any breakdown in the central server, we can use any teammate's storehouse.
5. All the previous versions and variants are smartly packed up inside the VCS. Any version is requested at any time to get information about the previous complete projects.

**Question: What are the different components of Selenium?**

**Answer:** Selenium is an open-source tool that is used for automating different web applications. It has mainly four major components that help to run multiple test cases and provides services of using various browsers and languages for automation. The components of Selenium are as follows:

**1. Selenium IDE**

Selenium IDE (Integrated Development Environment) is one of the simplest frameworks in the selenium suite. It has an easy record and playback function, which helps in figuring the tool that provides easy learning. If the tester is aware of the basics of HTML, JavaScript, and DOM, it is easier to utilize Selenium IDE. It enters the commands quickly and reduces the possibility of entering invalid commands.

**2. Selenium RC**

Selenium RC (Remote Control) is a tool that helps in understanding test scripts and providing support for different programming languages like Ruby, PHP, Java, etc. It mainly depends upon JavaScript for automation and does not support record and playback features.

**3. Selenium WebDriver**

It is mainly the extension of selenium RC, but it supports all the latest browsers and various platforms. It is created to support vital web pages in which elements present on the page can change without reloading the page, and it directly calls the browser for automation.

**4. Selenium GRID**

Selenium GRID is a tool that runs multiple test cases against different browsers and machines in parallel. Several nodes are not fixed in the grid, and it can be launched on various browsers and platforms. It is used together with selenium RC.

**Question: What is the purpose of configuration management in DevOps?**

**Answer:**Configuration management helps in automating tasks that are otherwise time-consuming and tedious and enhances an organization's agility. It brings consistency and improves the process of a product/service by streamlining design, documentation, control, and implementation of changes during various phases of the project.

**Question: What is the purpose of AWS in DevOps?**

**Answer:**AWS services support the automation of manual tasks and processes that help developers build applications faster and more efficiently. These processes can be deployment, development, test workflows, configuration management and container management.

**Question: What is the difference between a centralized and distributed version control system (VCS)?**

**Answer:** In a centralized repository system, the repository is located in a central location, and clients access this system when they need something. In such a version control system, the repository is always updated with the latest changes as the changes are directly committed to the central system; therefore, all the clients always have access to the latest code. CVS and SVN are examples of centralized VCS.

In a distributed VCS, everyone in the team has their repository, which is a mirror of the central repository. It provides flexibility, as one can work offline. Only when the changes have to be committed to the central system, you need to be online. This makes distributed VCS faster. Git and Mercurial are distributed VCS.

**Question: Explain the differences between git pull and Git fetch?**

**Answer:**

|  |  |
| --- | --- |
| **git fetch** | **git pull** |
| the command to use this feature is: git fetch <remote> | command to use: git pull <remote> <branch> |
| Fetches the changes from the remote repository but doesn't merge them with the local repository. | Fetches the changes of the branch from the remote repository and merges them with the local repository. pull = fetch + merge |
| Done at latter stages, since no merging is involved, there are no conflicts to be resolved. | There are chances of merge conflicts if two or more people are working on different copies of the same code/file. |
| The local repository is unchanged, but the central repository is updated. | The changes from the central repository are updated in the local repository. |
| Developers can see the changes made by others before they push their changes for integration. | Developers can first bring the latest files to their local, and then start updating the same. |

**Question: What is git stash?**

**Answer:**Git stash command is used to save the changes temporarily in the working directory. This gives developers a clean directory to work on. They can later merge the changes in the git workflow. If this command is used, the changes in the tracked files are merged in the working directory. Git stash command can be used many times in the git directory. It is used as git stash

**Question: What is a merge conflict in Git, and how can it be resolved?**

**Answer:**Merge conflicts occur when changes are made to a single file by multiple people at the same time. Due to this, Git won't be able to tell which of the multiple versions is the correct version. To resolve the conflicts, we should create a new Git repo, add a file, create a branch, make the edits and commit the changes. The next step is to merge the new branch into the master. Once this is done, Git clearly shows the differences in the different versions of the file and where the edits need to be made to remove the conflicts.

**Question**: **Can you tell us the fundamental differences between DevOps & Agile?**

**Answer**: Although DevOps shares some similarities with the Agile methodology, which is one of the most [popular SDLC methodologies](https://hackr.io/blog/sdlc-methodologies), both are fundamentally different approaches to software development. Following are the various fundamental differences between the two:

* **Agile Approach –**The agile approach is only meant for development in Agile while the agile approach is meant for both development and operations in DevOps.
* **Practices and Processes –** While agile involves practices such as Agile Scrum and Agile Kanban, DevOps involves processes such as CD (Continuous Delivery), CI (Continuous Integration), and CT (Continuous Testing).
* **Priority –** Agile prioritizes timeliness whereas, DevOps gives equal priority to timeliness and quality.
* **Release Cycles –** DevOps offers smaller release cycles with immediate feedback while Agile offers only smaller release cycles without immediate feedback.
* **Feedback Source –** Agile relies on feedback from customers while feedback from self (monitoring tools) is involved in DevOps.
* **Scope of Work –** For Agile, the scope of work is agility only but for DevOps, it is agility and the need for automation.

**Question**: **Why do we need DevOps?**

**Answer**: Organizations these days are trying to transport small features to customers via a series of release trains instead of releasing big feature sets. There are several benefits of doing so, including better software quality and quick customer feedback.

All such benefits lead to a higher level of customer satisfaction, which is the most important goal for any product development project. To do so, companies need to:

* Increase deployment frequency
* Lessen lead time between fixes
* The lower failure rate of new releases
* In case of new release crashing, have a faster mean time to recovery

DevOps helps in fulfilling all these requirements and thus, achieving seamless software delivery. Full-fledged organizations like Amazon, Etsy, and Google have adopted DevOps methodology resulting in achieving performance levels that were previously uncharted.

With the adoption of DevOps methodology, organizations are able to accomplish tens to thousands of deployments in a single day. Moreover, doing so while offering first-rate reliability, security, and stability.

**Question**: **What are the important business and technical benefits of using DevOps?**

**Answer**: DevOps brings a lot of business and technical benefits to the table. Some of the most important ones are listed down as follows:

**Business benefits**

* Enhanced operating environ ment stability
* Faster delivery of features
* More time for adding value to the product

**Technical benefits**

* Continuous software delivery
* Faster problem resolution
* Lesser complex problems

**Question**: **Can you name some of the most-used DevOps tools?**

**Answer**: Following is a list of some of the most widely used DevOps tools:

* [Ansible](https://hackr.io/tutorials/learn-ansible?ref=blog-post) – A configuration management and application deployment tool
* Chef – A configuration management and application deployment tool
* Docker – A containerization tool
* Git – A version control system (VCS) tool
* [Jenkins](https://hackr.io/tutorials/learn-jenkins?ref=blog-post) – A continuous integration (CI) tool
* Jira – An agile team collaboration tool
* Nagios – A continuous monitoring tool
* Puppet – A configuration management and application deployment tool
* Selenium – A continuous testing (CT) tool

**Question: What is Selenium used for?**

**Answer**: [Selenium](https://hackr.io/blog/what-is-selenium) is used for continuous testing in DevOps. The tool specializes in functional and regression forms of testing.

**Question**: **What do you understand by Puppet in DevOps?**

**Answer**: It is a configuration management tool that is used for automating administration tasks. Puppet makes use of the Master-Slave architecture in which the two entities communicate via an encrypted channel.

System admins need to perform a lot of repetitive tasks, notably installing and configuring servers. Writing scripts for automating such tasks is an option but it becomes hectic when the infrastructure is large. Configuration management is a great workaround for this.

Puppet helps in configuring, deploying, and managing servers. Not only does it make such redundant tasks easier but it also cuts a significant portion of the total work time. The mature configuration management tool:

* Continuously checks whether the needed configuration for a host is in place or not. If altered, the configuration is automatically reverted back
* Defines distinct configurations for every host
* Does dynamic scaling (up and down) of machines
* Provides control over all the configured machines so that a centralized change can automatically get propagated to all of them

**Question: What do you understand by anti-patterns of DevOps?**

**Answer**: When a DevOps pattern commonly adopted by other organizations doesn’t work in a specific context and still the organization continues using it, it leads to the adoption of an anti-pattern. In other words, anti-patterns are myths about DevOps. Some of the notable anti-patterns are:

* An organization needs to have a separate DevOps group
* Agile equals DevOps
* DevOps is a process
* DevOps is development-driven release management
* DevOps is not possible because the organization is unique
* DevOps is not possible because the people available are unsuitable
* DevOps means Developers Managing Production
* DevOps will solve all problems
* Failing to include all aspects of the organization in an ongoing DevOps transition
* Not defining KPIs at the start of a DevOps transition
* Reduce the silo-based isolation of development and operations with a new DevOps team that silos itself from other parts of the organization

**Question**: **DevOps has something called CI. What is it and what is its purpose?**

**Answer**: CI in DevOps stands for Continuous Integration. CI is a development practice in which developers integrate code into a shared repository multiple times in a single day.

Continuous Integration of development and testing enhances the quality of the software as well as reducing the total time required for delivery.

The developer has broken the build if a team member checking in code runs into a compilation failure. As such, other developers are not able to sync with the shared source code repository without introducing compilation errors into their own workspaces.

This disrupts the collaborative and shared development process. Hence, as soon as a CI build breaks, it’s important to identify and correct the problem immediately.

Typically, a CI process includes a suite of unit, integration, and regression tests that run each time the compilation succeeds. In case any of the aforesaid tests fail, the CI build is considered unstable (which is common during an Agile sprint when development is ongoing) and not broken.

**Question**: **More often than not we hear shift left in DevOps. What is it?**

**Answer**: The traditional [software development lifecycle](https://hackr.io/blog/sdlc-methodologies#popular-sdlc-methodologies) when graphed on a paper has two sides, left and right. While the left side of the graph includes design and development, the right side includes production staging, stress testing, and user acceptance.

To shift left in DevOps simply means the necessity of taking as many tasks on the right i.e. that typically happens toward the end of the application development process and incorporates them into earlier stages of a DevOps methodology.

There are several ways of accomplishing a shit left in DevOps, most notably:

* Create production-ready artifacts at the end of every Agile sprint
* Incorporating static code analysis routines in every build

The level of doing the DevOps the right way is directly dependent on the degree of shifting left as much as possible.

**Question**: **What does CAMS in DevOps stand for?**

**Answer**: The acronym CAMS is usually used for describing the core creeds of DevOps methodology. It stands for:

* **C**ulture
* **A**utomation
* **M**easurement
* **S**haring

**Question**: **What are the several KPIs used to gauge DevOps success?**

**Answer**: KPIs is a contracted form of Key Performance Indicators. In order to measure the success of a DevOps process, several KPIs can be used. Some of the most popular ones are:

* Application performance
* Application usage and traffic
* The automated test pass percentage
* Availability
* Change volume
* Customer tickets
* Defect escape rate
* Deployment frequency
* Deployment time
* Error rates
* Failed deployments
* Lead time
* Mean time to detection (MTTD)
* Meantime to recovery (MTTR)

**Question**: **In your opinion, what are the major benefits of implementing DevOps automation?**

**Answer**: Following are the major benefits of implementing DevOps automation:

* Removal of the possibility of human error from the CD equation (Core benefit).
* As tasks become more predictable and repeatable, it is easy to identify and correct when something goes wrong. Hence, it results in producing more reliable and robust systems.
* Removes bottlenecks from the CI pipeline. It results in increased deployment frequency and decreased number of failed deployments. Both of them are important DevOps KPIs.

**Question**: **What do you understand by containers?**

**Answer**: Containers are a form of lightweight virtualization that help in providing isolation among processes. Containers are heavier than a [chroot](https://en.wikipedia.org/wiki/Chroot) but lighter than a [hypervisor](https://en.wikipedia.org/wiki/Hypervisor).

**Question**: **Microservices are a core part of DevOps. Can you name any two popular Java development frameworks for creating microservices?**

**Answer**: There are several Java frameworks that allow creating microservices. However, Eclipse MicroProfile and Spring Boot stand out from the herd as the two leading Java development frameworks used in DevOps for creating microservices.

**Question**: **What do you understand by a Version Control System (VCS)? Define its uses.**

**Answer**: A Version Control System or VCS is a system that is capable of recording changes made to a file or a group of files over time. Git and Mercurial are two of the most popular version control systems. Important uses of a VCS are:

* Check what was the last modification that caused a problem
* Compare the changes made over time
* Identifying who introduced a new issue and at what time
* Revert a file or files to some earlier state
* Revert the complete project to a previous state

**Question**: **Git is a popular DevOps tool. Tell us how you will revert a commit that has already been pushed and made public.**

**Answer**: There are two ways of doing so:

* By creating a new commit to undo all changes made by the commit that has already been pushed and made public. Following command is used for doing so:  
  git revert
* By fixing or removing the bad file in a new commit and then pushing it to the remote repository. After making necessary changes to the file, commit it to the remote repository using the command:  
  git commit -m “commit message”

**Question**: **What are post mortem meetings?**

**Answer**: Many times there is a need to discuss what went wrong during a DevOps process. For this, post mortem meetings are arranged. These meetings yield steps that should be taken to avoid the same failure or set of failures in the future for which the meeting was arranged in the first place.

**Question**: **Draw a comparison between Asset Management and Configuration Management.**

**Answer**: The process of monitoring as well as maintaining things of value to an entity or group is called an Asset Management.

Configuration Management refers to the process of controlling, identifying, planning for, and verifying the configuration items within service in support of Change Management.

**Question: Can you state and explain various key elements of continuous testing?**

**Answer**: Various key elements of continuous testing are:

* Advanced analysis – Used for forecasting and predicting unknown future events
* Policy analysis – Meant for improving the testing process
* Requirement traceability – Refers to the ability to describe as well as follow the life of a requirement, from its origin to deployment
* Risk assessment – The method or process of identifying hazards and risk factors that can cause potential damage
* Service virtualization – Allows using virtual services instead of production services. Emulates software components for simple testing
* Test optimization – Improve the overall testing process

**Question**: **Please explain the core operations of DevOps in terms of development and infrastructure.**

**Answer**: Core operations of DevOps in terms of development and infrastructure are:

* **Application development –** Developing a product that is able to meet all customer requirements and offers a remarkable level of quality
* **Code coverage –** a measurement of the total number of blocks or lines or arcs of the code executed while the automated tests are running
* **Code developing –** Prepare the codebase required for the product development
* **Configuration –** Allowing the product to be used in an optimum way
* **Deployment –** Installing the software to be used by the end-user
* **Orchestration –** Arrangement of several automated tasks
* **Packaging –** Activities involved when the release is ready for deployment
* **Provisioning –** Ensuring that the infrastructure changes arrive just-in-time with the code that requires it
* **Unit testing –** Meant for testing individual units or components

**Question: What is DevOps?**

**Answer:**DevOps is the value-added practice wherein the development, as well as operation engineers, work together from the initial point of designing a product to deployment in the markets.

**Question: What is the basic premise of DevOps?**

**Answer:**DevOps is a cultural shift wherein a collaboration and operation teams work together throughout the product or service life cycle.

**Question: Which methodology is DevOps related to?**

**Answer:**DevOps is related to Agile methodology.

**Question: What are the priorities in DevOps?**

**Answer:**The priorities in DevOps include resource management, teamwork, and communication.

**Question: What are the benefits of DevOps?**

**Answer:**The varied benefits of DevOps include innovation, stability, functionality, and speed.

**Question: What are the different advantages of Git?**

**Answer:**Git has the following advantages:

1. It helps in data redundancy and replication.
2. It is highly available.
3. It supports collaboration.
4. It can be used for a variety of projects.
5. It uses only one Git directory per repository.
6. It supports disk utilization.
7. It offers higher network performance.

**Question: What is the difference between Git fetch and Git pull?**

**Answer:**Git pull is common that pulls any new commits from a branch of the central repository to update the target branch. Git fetch pulls any new commits from only the desired branch and then stores them into a new brand of the local repository.

**Question: Can we handle merge conflict in Git?**

**Answer:**Yes, we can handle merge conflict in the following three steps:

* Step 1: Developing a clear understanding by checking everything using Git status.
* Step 2: Mark and clean up the conflict by applying the merge tool.
* Step 3: Performing commit and merging with the current branch along with the master branch.

**Question: What is Forking Workflow?**

**Answer:**Forking Workflow gives every developer with their service side repositories, thereby supporting open source projects.

**Question:How is Forking Workflow better than Git Workflow?**

**Answer:**Forking Workflow is better than Git Workflow because it helps in integrating the contributions of different developers without needing everyone to push to a single central repository for cleaning project history. Thus the developers are allowed to push their server-side repositories, and thereby on the project maintainer will push to the official repository.

**Question: What is Git rebase?**

**Answer:**Git rebase is a command which is designed to integrate the changes from one branch to another brand.

**Question: How is Git rebase different from Git merge?**

**Answer:**Git rebase is different from Git merge because, in the case of Git rebase, the feature branch is transferred to the master branch's ending point. However, in the case of Git merge, the merging adds to a new commit in the history wherein history does not change, but the master branch is changed.

**Question: Can we move or copy Jenkins from one server to another?**

**Answer:**Yes, we can move or copy the Jenkins from one server to other. For instance, by copying the Jenkins jobs directory can be moved from the older server to the new server. This way, installation can be moved from one installation to another by copying in the corresponding job directory.

**Question: Can we make a new copy of an existing Jenkins job?**

**Answer:**Yes, we can make a new copy of an existing Jenkins job by creating a clone of the directory in a different name.

**Question: What is the difference between continuous testing and automation testing?**

**Answer:**In continuous testing, the process of executing the automated test is part of the software delivery process. In contrast, automation testing is a process wherein the manual process of testing is applied wherein the separate testing tool helps the developers in creating test scripts that can be executed again and again without any kind of manual intervention.

**Question: What is the role of a Selenium Grid?**

**Answer:**The role of a Selenium Grid is to execute the same or different test scripts and that too on different platforms and browsers so that the distributed test execution can be made. It helps in testing under various environments and offers an ability to save execution time.

**Question:Can we secure Jenkins?**

**Answer:**Yes, we can secure Jenkins in the following ways:

1. Ensuring that global security is on
2. Checking if Jenkins is integrated
3. Making sure that the project matrix is enabled
4. Automating the process of setting up rights and privileges
5. Limiting physical access to Jenkins data
6. Applying security audits regularly

**Question:What is Jenkins Pipeline?**

**Answer:**Jenkins pipeline is a suite of plugins that supports the implementation and integration of Jenkins's continuous delivery pipeline.

**Question:What is continuous integration?**

**Answer:**Continuous integration, also known as a continuous delivery pipeline, includes functions of build, deploy, test, and release. It is a time-saving feature and builds jobs that are integrated and chain into a particular sequence.

**Question: What is a Puppet Module?**

**Answer:**Puppet Module is a collection of manifest and data, including files, templates, and facts that have a specific directory structure.

**Question: How is the Puppet Module different from Puppet Manifest?**

**Answer:**Puppet Manifests use the .pp extension. They are Puppet programs, which consist of the Puppet Code. On the other hand, Puppet Modules organizes various kinds of Puppet Manifests that can split the Puppet code with it.

**Question: What is Ansible work in DevOps?**

**Answer:**Ansible work is an open-source automation tool used in DevOps.

**Question: What are the categories of Ansible in DevOps?**

**Answer:**There are two categories of Ansible in DevOps which include,

1. Controlling machines.
2. Nodes.

**Question: Can we install Ansible on the controlling machines?**

**Answer:**Yes, we can install Ansible on the controlling machine by using the machine nodes that are managed with the help of SSH.

**Question: Are Ansible agentless tool? What are its benefits?**

**Answer:**Yes, Ansible is an agentless tool because it does not require any kind of mandatory installations on the remote nodes. The benefits of the Ansible tool include the following:

1. Task automation
2. Configuration management
3. Application deployment

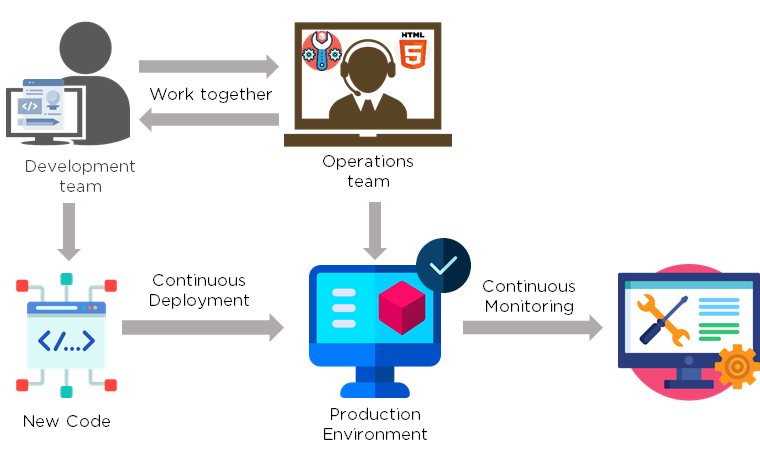
General DevOps Interview Questions

1. What do you know about DevOps?

Your answer must be simple and straightforward. Begin by explaining the growing importance of DevOps in the IT industry. Discuss how such an approach aims to synergize the efforts of the development and operations teams to accelerate the delivery of software products, with a minimal failure rate. Include how DevOps is a value-added practice, where development and operations engineers join hands throughout the product or service lifecycle, right from the design stage to the point of deployment.

2. How is DevOps different from agile methodology?

[DevOps](https://www.simplilearn.com/tutorials/devops-tutorial/what-is-devops) is a culture that allows the development and the operations team to work together. This results in continuous development, testing, integration, deployment, and monitoring of the software throughout the lifecycle.



[Agile](https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile) is a software development methodology that focuses on iterative, incremental, small, and rapid releases of software, along with customer feedback. It addresses gaps and conflicts between the customer and developers.



DevOps addresses gaps and conflicts between the Developers and IT Operations.



3. Which are some of the most popular DevOps tools?

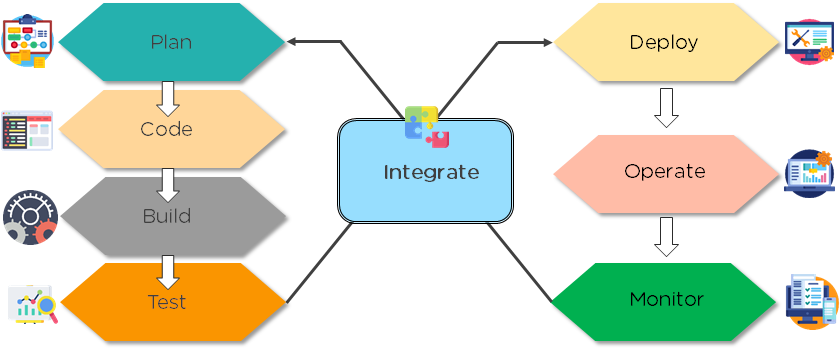
The most popular [DevOps tools](https://www.simplilearn.com/tutorials/devops-tutorial/devops-tools) include:

1. Selenium
2. Puppet
3. Chef
4. Git
5. Jenkins
6. Ansible
7. Docker

4. What are the different phases in DevOps?

The various phases of the DevOps lifecycle are as follows:

* Plan - Initially, there should be a plan for the type of application that needs to be developed. Getting a rough picture of the development process is always a good idea.
* Code - The application is coded as per the end-user requirements.
* Build - Build the application by integrating various codes formed in the previous steps.
* Test - This is the most crucial step of the application development. Test the application and rebuild, if necessary.
* Integrate - Multiple codes from different programmers are integrated into one.
* Deploy - Code is deployed into a cloud environment for further usage. It is ensured that any new changes do not affect the functioning of a high traffic website.
* Operate - Operations are performed on the code if required.
* Monitor - Application performance is monitored. Changes are made to meet the end-user requirements.



The above figure indicates the DevOps lifecycle.

5. Mention some of the core benefits of DevOps.

The core benefits of DevOps are as follows:

Technical benefits

* Continuous software delivery
* Less complex problems to manage
* Early detection and faster correction of defects

Business benefits

* Faster delivery of features
* Stable operating environments
* Improved communication and collaboration between the teams

6. How will you approach a project that needs to implement DevOps?

The following standard approaches can be used to implement DevOps in a specific project:

Stage 1

An assessment of the existing process and implementation for about two to three weeks to identify areas of improvement so that the team can create a road map for the implementation.

Stage 2

Create a proof of concept (PoC). Once it is accepted and approved, the team can start on the actual implementation and roll-out of the project plan.

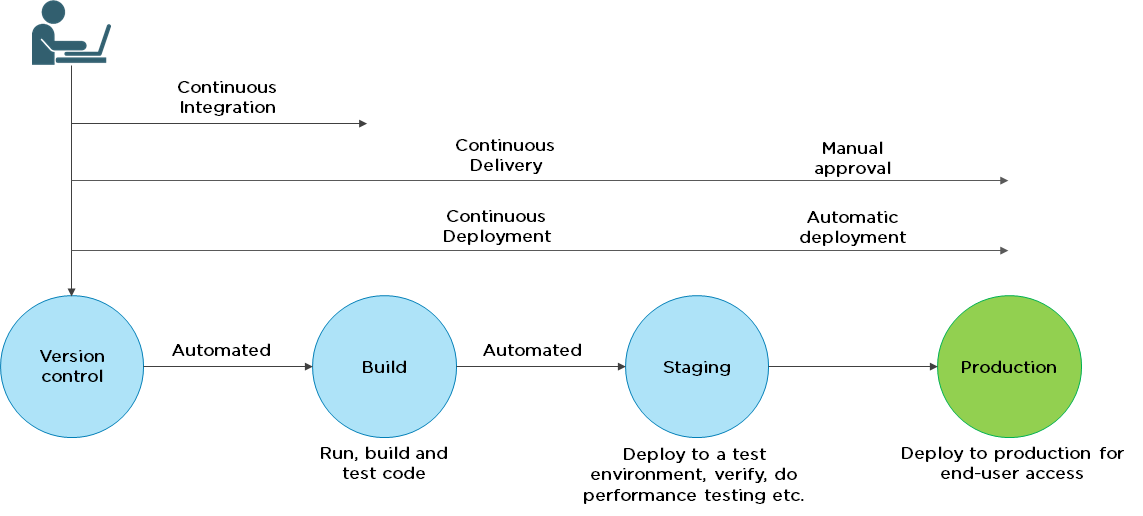
Stage 3

The project is now ready for implementing DevOps by using version control/integration/testing/deployment/delivery and monitoring followed step by step.

By following the proper steps for version control, integration, testing, deployment, delivery, and monitoring, the project is now ready for DevOps implementation.

7. What is the difference between continuous delivery and continuous deployment?

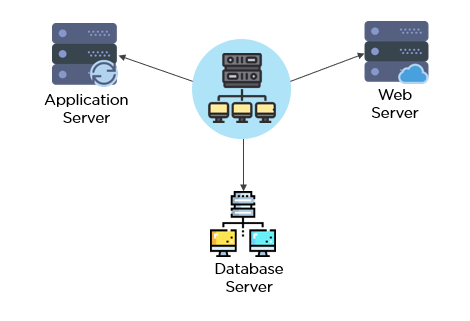
|  |  |
| --- | --- |
| Continuous Delivery | Continuous Deployment |
| Ensures code can be safely deployed on to production | Every change that passes the automated tests is deployed to production automatically |
| Ensures business applications and services function as expected | Makes software development and the release process faster and more robust |
| Delivers every change to a production-like environment through rigorous automated testing | There is no explicit approval from a developer and requires a developed culture of monitoring |



8. What is the role of configuration management in DevOps?

* Enables management of and changes to multiple systems.
* Standardizes resource configurations, which in turn, manage IT infrastructure.
* It helps with the administration and management of multiple servers and maintains the integrity of the entire infrastructure.

9. How does continuous monitoring help you maintain the entire architecture of the system?



Continuous monitoring in DevOps is a process of detecting, identifying, and reporting any faults or threats in the entire infrastructure of the system.

* Ensures that all services, applications, and resources are running on the servers properly.
* Monitors the status of servers and determines if applications are working correctly or not.
* Enables continuous audit, transaction inspection, and controlled monitoring.

10. What is the role of AWS in DevOps?

AWS has the following role in DevOps:

* Flexible services - Provides ready-to-use, flexible services without the need to install or set up the software.
* Built for scale - You can manage a single instance or scale to thousands using AWS services.
* Automation - AWS lets you automate tasks and processes, giving you more time to innovate
* Secure - Using AWS Identity and Access Management (IAM), you can set user permissions and policies.
* Large partner ecosystem - AWS supports a large ecosystem of partners that integrate with and extend AWS services.

11. Name three important DevOps KPIs.

The three important KPIs are as follows:

* Meantime to failure recovery - This is the average time taken to recover from a failure.
* Deployment frequency - The frequency in which the deployment occurs.
* Percentage of failed deployments - The number of times the deployment fails.

12. Explain the term "Infrastructure as Code" (IaC) as it relates to configuration management.

* Writing code to manage configuration, deployment, and automatic provisioning.
* Managing data centers with machine-readable definition files, rather than physical hardware configuration.
* Ensuring all your servers and other infrastructure components are provisioned consistently and effortlessly.
* Administering cloud computing environments, also known as infrastructure as a service (IaaS).

13. How is IaC implemented using AWS?

Start by talking about the age-old mechanisms of writing commands onto script files and testing them in a separate environment before deployment and how this approach is being replaced by IaC. Similar to the codes written for other services, with the help of AWS, IaC allows developers to write, test, and maintain infrastructure entities in a descriptive manner, using formats such as JSON or YAML. This enables easier development and faster deployment of infrastructure changes.

14. Why Has DevOps Gained Prominence over the Last Few Years?

Before talking about the growing popularity of [DevOps](https://www.simplilearn.com/tutorials/devops-tutorial/what-is-devops), discuss the current industry scenario. Begin with some examples of how big players such as [Netflix and Facebook](https://techbeacon.com/10-companies-killing-it-devops) are investing in DevOps to automate and accelerate application deployment and how this has helped them grow their business. Using Facebook as an example, you would point to Facebook’s continuous deployment and code ownership models and how these have helped it scale up but ensure the quality of experience at the same time. Hundreds of lines of code are implemented without affecting quality, stability, and security.

Your next use case should be Netflix. This streaming and on-demand video company follow similar practices with fully automated processes and systems. Mention the user base of these two organizations: Facebook has 2 billion users while Netflix streams online content to more than 100 million users worldwide.

These are great examples of how DevOps can help organizations to ensure higher success rates for releases, reduce the lead time between bug fixes, streamline and continuous delivery through automation, and an overall reduction in manpower costs.

DevOps Interview Guide

Here's How You Crack the Interview in the First Go[DOWNLOAD NOW](https://www.simplilearn.com/tutorials/devops-tutorial/devops-interview-questions)

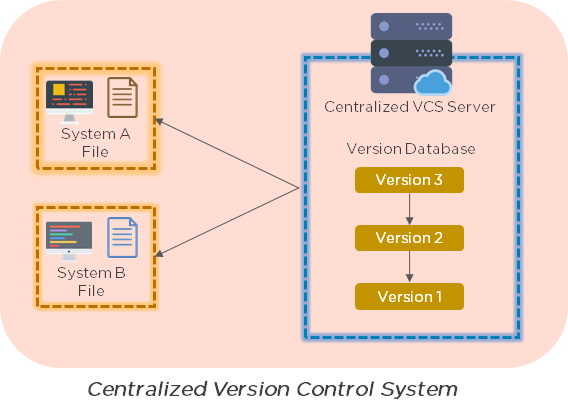


DevOps Interview Questions for Source Code Management — Git

15. Explain the difference between a centralized and distributed version control system (VCS).

Centralized Version Control System

* All file versions are stored on a central server
* No developer has a copy of all files on a local system
* If the central server crashes, all data from the project will be lost



Distributed Control System

* Every developer has a copy of all versions of the code on their systems
* Enables team members to work offline and does not rely on a single location for backups
* There is no threat, even if the server crashes

16. What is the git command that downloads any repository from GitHub to your computer?



The [git command](https://www.simplilearn.com/tutorials/git-tutorial/git-commands) that downloads any repository from GitHub to your computer is git clone.

17. How do you push a file from your local system to the GitHub repository using Git?

First, connect the local repository to your remote repository:

git remote add origin [copied web address]

// Ex: git remote add origin [**https://github.com/Simplilearn-github/test.git**](https://github.com/Simplilearn-github/test.git)

Second, push your file to the remote repository:

git push origin master

18. How is a bare repository different from the standard way of initializing a Git repository?

Using the standard method:

git init

* You create a working directory with git init
* A .git subfolder is created with all the git-related revision history

Using the bare way

git init --bare

* It does not contain any working or checked out a copy of source files
* Bare repositories store git revision history in the root folder of your repository, instead of the .git subfolder

19. Which of the following CLI commands can be used to rename files?

1. git rm
2. git mv
3. git rm -r
4. None of the above

The correct answer is B) git mv

20. What is the process for reverting a commit that has already been pushed and made public?

There are two ways that you can revert a commit:

1. Remove or fix the bad file in a new commit and push it to the remote repository. Then commit it to the remote repository using:  
     
   git commit –m "commit message"
2. Create a new commit that undoes all the changes that were made in the bad commit. Use the following command:  
     
   git revert <commit id>

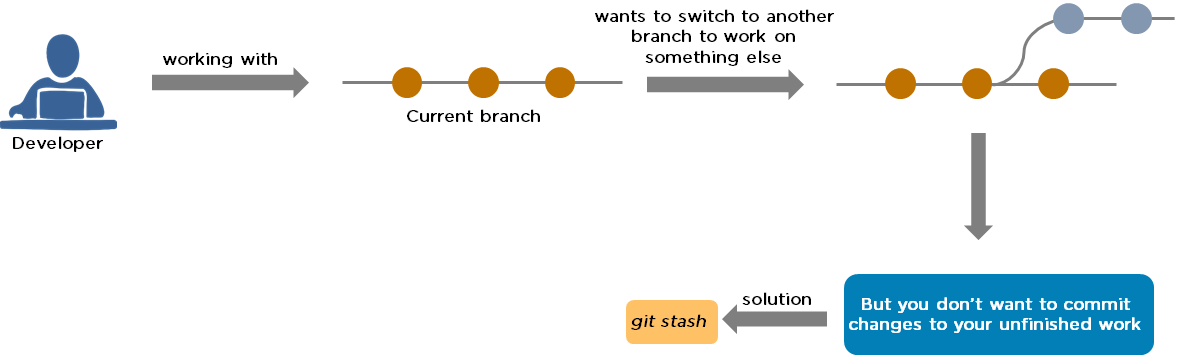
Example: git revert 56de0938f

21. Explain the difference between git fetch and git pull.

|  |  |
| --- | --- |
| Git fetch | Git pull |
| Git fetch only downloads new data from a remote repository | Git pull updates the current HEAD branch with the latest changes from the remote server |
| Does not integrate any new data into your working files | Downloads new data and integrate it with the current working files |
| Users can run a Git fetch at any time to update the remote-tracking branches | Tries to merge remote changes with your local ones |
| Command - git fetch origin                    git fetch –-all | Command - git pull origin master |

22. What is Git stash?

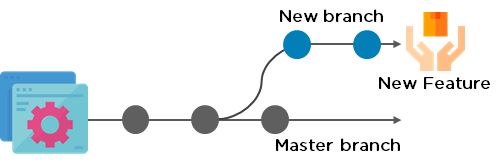
A developer working with a current branch wants to switch to another branch to work on something else, but the developer doesn't want to commit changes to your unfinished work. The solution to this issue is Git stash. Git stash takes your modified tracked files and saves them on a stack of unfinished changes that you can reapply at any time.



23. Explain the concept of branching in Git.

Suppose you are working on an application, and you want to add a new feature to the app. You can create a new branch and build the new feature on that branch.

* By default, you always work on the master branch
* The circles on the branch represent various commits made on the branch
* After you are done with all the changes, you can merge it with the master branch



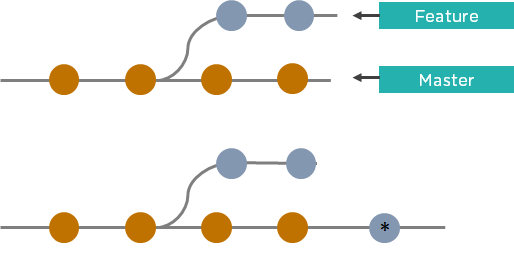
24. What is the difference between Git Merge and Git Rebase?

Suppose you are working on a new feature in a dedicated branch, and another team member updates the master branch with new commits. You can use these two functions:

Git Merge

To incorporate the new commits into your feature branch, use Git merge.

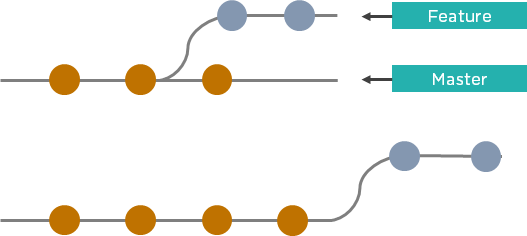
* Creates an extra merge commit every time you need to incorporate changes
* But, it pollutes your feature branch history



Git Rebase

As an alternative to merging, you can rebase the feature branch on to master.

* Incorporates all the new commits in the master branch
* It creates new commits for every commit in the original branch and rewrites project history



25. How do you find a list of files that have been changed in a particular commit?

The command to get a list of files that have been changed in a particular commit is:

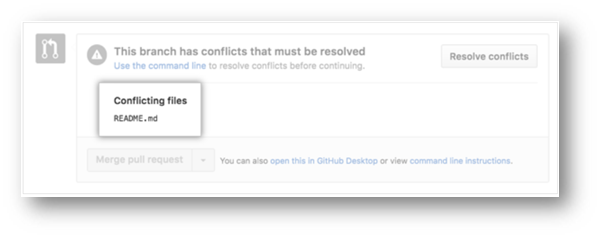
git diff-tree –r {commit hash}

Example: git diff-tree –r 87e673f21b

* -r flag instructs the command to list individual files
* commit hash will list all the files that were changed or added in that commit

26. What is a merge conflict in Git, and how can it be resolved?

A [Git merge conflict](https://www.simplilearn.com/tutorials/git-tutorial/merge-conflicts-in-git) happens when you have merge branches with competing for commits, and Git needs your help to decide which changes to incorporate in the final merge.

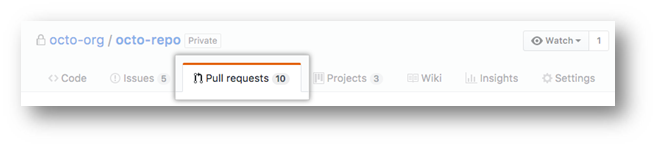


Manually edit the conflicted file to select the changes that you want to keep in the final merge.

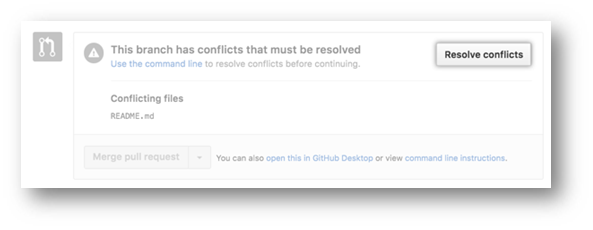
Resolve using GitHub conflict editor

This is done when a merge conflict is caused after competing for line changes. For example, this may occur when people make different changes to the same line of the same file on different branches in your Git repository.

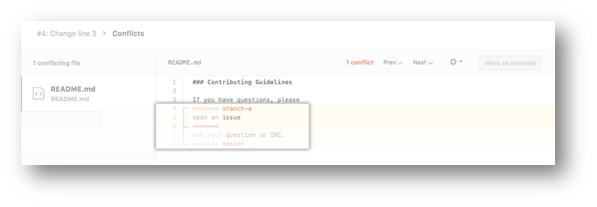
* Resolving a merge conflict using conflict editor:
* Under your repository name, click "Pull requests."



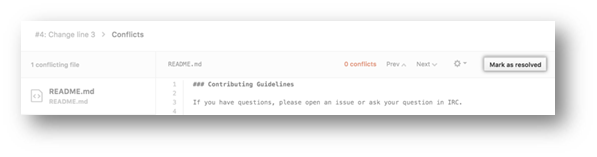
* In the "Pull requests" drop-down, click the pull request with a merge conflict that you'd like to resolve
* Near the bottom of your pull request, click "Resolve conflicts."



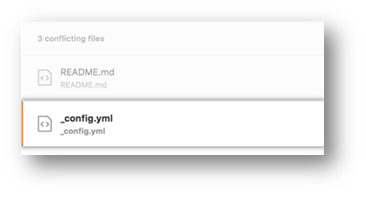
* Decide if you only want to keep your branch's changes, the other branch's changes, or make a brand new change, which may incorporate changes from both branches.
* Delete the conflict markers <<<<<<<, =======, >>>>>>> and make changes you want in the final merge.



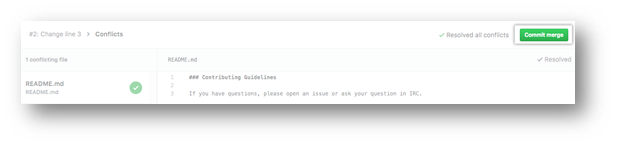
* If you have more than one merge conflict in your file, scroll down to the next set of conflict markers and repeat steps four and five to resolve your merge conflict.
* Once you have resolved all the conflicts in the file, click Mark as resolved.



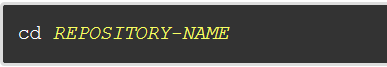
* If you have more than one file with a conflict, select the next file you want to edit on the left side of the page under "conflicting files" and repeat steps four to seven until you've resolved all of your pull request's merge conflicts.



* Once you've resolved your merge conflicts, click Commit merge. This merges the entire base branch into your head branch.



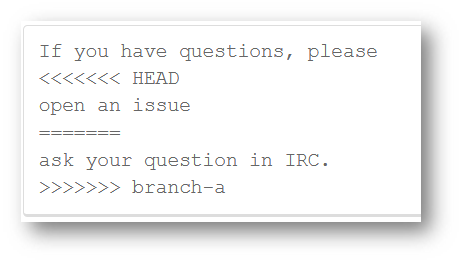
* To merge your pull request, click Merge pull request.
* A merge conflict is resolved using the command line.
* Open Git Bash.
* Navigate into the local Git repository that contains the merge conflict.



* Generate a list of the files that the merge conflict affects. In this example, the file styleguide.md has a merge conflict.

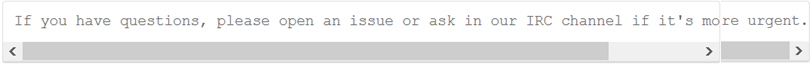


* Open any text editor, such as Sublime Text or Atom, and navigate to the file that has merge conflicts.
* To see the beginning of the merge conflict in your file, search the file for the conflict marker "<<<<<<<. " Open it, and you'll see the changes from the base branch after the line "<<<<<<< HEAD."
* Next, you'll see "=======", which divides your changes from the changes in the other branch, followed by ">>>>>>> BRANCH-NAME".



* Decide if you only want to keep your branch's changes, the other branch's changes, or make a brand new change, which may incorporate changes from both branches.
* Delete the conflict markers "<<<<<<<", "=======", ">>>>>>>" and make the changes you want in the final merge.

        In this example, both the changes are incorporated into the final merge:



* Add or stage your changes.



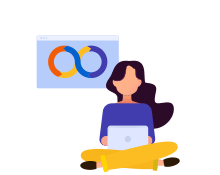
* Commit your changes with a comment.



Now you can merge the branches on the command line, or push your changes to your remote repository on GitHub and merge your changes in a pull request.

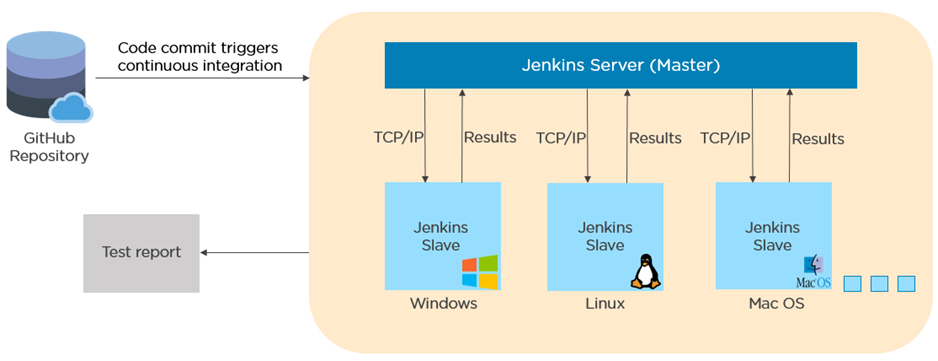
Post Graduate Program in DevOps

Designed in collaboration with Caltech CTME[EXPLORE COURSE](https://www.simplilearn.com/pgp-devops?source=GhPreviewCTABanner)



DevOps Interview Questions for Continuous Integration - Jenkins

27. Explain the master-slave architecture of Jenkins.



* Jenkins master pulls the code from the remote GitHub repository every time there is a code commit.
* It distributes the workload to all the Jenkins slaves.
* On request from the Jenkins master, the slaves carry out, builds, test, and produce test reports.

28. What is Jenkinsfile?

Jenkinsfile contains the definition of a Jenkins pipeline and is checked into the source control repository. It is a text file.

* It allows code review and iteration on the pipeline.
* It permits an audit trail for the pipeline.
* There is a single source of truth for the pipeline, which can be viewed and edited.

29. Which of the following commands runs Jenkins from the command line?

1. java –jar Jenkins.war
2. java –war Jenkins.jar
3. java –jar Jenkins.jar
4. java –war Jenkins.war

The correct answer is A) java –jar Jenkins.war

30. What concepts are key aspects of the Jenkins pipeline?

* Pipeline: User-defined model of a CD pipeline. The pipeline's code defines the entire build process, which includes building, testing and delivering an application
* Node: A machine that is part of the Jenkins environment and capable of executing a pipeline
* Step: A single task that tells Jenkins what to do at a particular point in time
* Stage: Defines a conceptually distinct subset of tasks performed through the entire pipeline (build, test, deploy stages)

31. Which file is used to define dependency in Maven?

1. build.xml
2. pom.xml
3. dependency.xml
4. Version.xml

The correct answer is B) pom.xml

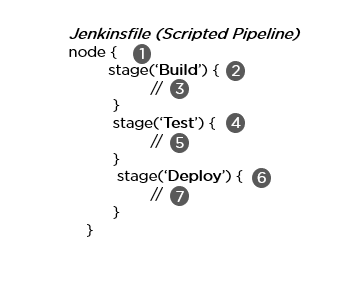
32. Explain the two types of pipeline in Jenkins, along with their syntax.

Jenkins provides two ways of developing a pipeline code: Scripted and Declarative.

A. Scripted Pipeline: It is based on Groovy script as their Domain Specific Language. One or more node blocks do the core work throughout the entire pipeline.

Syntax:

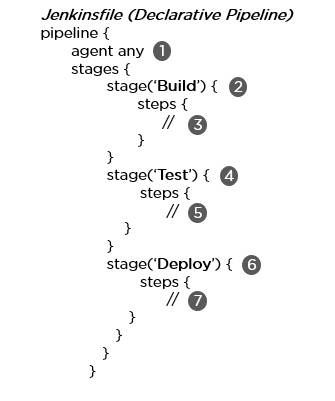
1. Executes the pipeline or any of its stages on any available agent
2. Defines the build stage
3. Performs steps related to building stage
4. Defines the test stage
5. Performs steps related to the test stage
6. Defines the deploy stage
7. Performs steps related to the deploy stage



B. Declarative Pipeline: It provides a simple and friendly syntax to define a pipeline. Here, the pipeline block defines the work done throughout the pipeline.

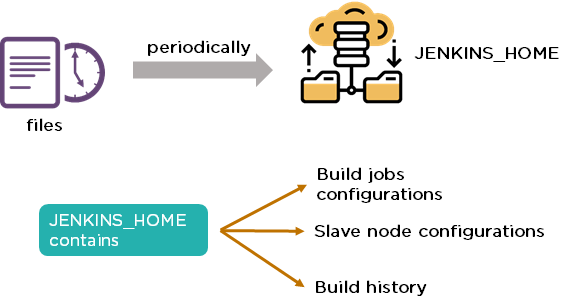
Syntax:

1. Executes the pipeline or any of its stages on any available agent
2. Defines the build stage
3. Performs steps related to building stage
4. Defines the test stage
5. Performs steps related to the test stage
6. Defines the deploy stage
7. Performs steps related to the deploy stage



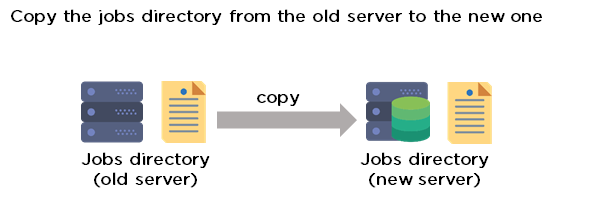
33. How do you create a backup and copy files in Jenkins?

In order to create a backup file, periodically back up your JENKINS\_HOME directory.



In order to create a backup of Jenkins setup, copy the JENKINS\_HOME directory. You can also copy a job directory to clone or replicate a job or rename the directory.

34. How can you copy Jenkins from one server to another?



* Move the job from one Jenkins installation to another by copying the corresponding job directory.
* Create a copy of an existing job by making a clone of a job directory with a different name.
* Rename an existing job by renaming a directory.

35. Name three security mechanisms Jenkins uses to authenticate users.

* Jenkins uses an internal database to store user data and credentials.
* Jenkins can use the Lightweight Directory Access Protocol (LDAP) server to authenticate users.
* Jenkins can be configured to employ the authentication mechanism that the deployed application server uses.

36. How is a custom build of a core plugin deployed?

Steps to deploy a custom build of a core plugin:

* Copy the .hpi file to $JENKINS\_HOME/plugins
* Remove the plugin's development directory
* Create an empty file called <plugin>.hpi.pinned
* Restart Jenkins and use your custom build of a core plugin

37. How can you temporarily turn off Jenkins security if the administrative users have locked themselves out of the admin console?



* When security is enabled, the Config file contains an XML element named useSecurity that will be set to true.
* By changing this setting to false, security will be disabled the next time Jenkins is restarted.

38. What are the ways in which a build can be scheduled/run in Jenkins?

* By source code management commits.
* After completion of other builds.
* Scheduled to run at a specified time.
* Manual build requests.

39. What are the commands that you can use to restart Jenkins manually?

Two ways to manually restart Jenkins:

1. (Jenkins\_url)/restart            // Forces a restart without waiting for builds to complete
2. (Jenkins\_url)/safeRestart    // Allows all running builds to complete before it restarts

DevOps Engineer Exam Practice Test

Assess your understanding of the DevOps concepts[TRY ANSWERING](https://www.simplilearn.com/devops-engineer-mcqs-free-practice-test?source=GhPreviewCTABanner)



DevOps Interview Questions for Continuous Testing - Selenium

40. What are the different Selenium components?

[Selenium](https://www.simplilearn.com/tutorials/selenium-tutorial/what-is-selenium) has the following components:

[Selenium Integrated Development Environment (IDE)](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-ide)

* It has a simple framework and should be used for prototyping.
* It has an easy-to-install Firefox plug-in.

Selenium Remote Control (RC)

* Testing framework for a developer to write code in any programming language (Java, PHP, Perl, C#, etc.).

[**Selenium WebDriver**](https://www.simplilearn.com/tutorials/selenium-tutorial/what-is-selenium-webdriver)

* Applies a better approach to automate browser activities.
* It does not rely on JavaScript.

Selenium Grid

* Works with Selenium RC and runs tests on different nodes using browsers.

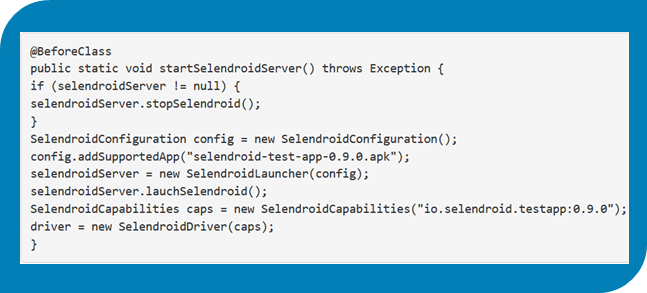
41. What are the different exceptions in Selenium WebDriver?

Exceptions are events that occur during the execution of a program and disrupt the normal flow of a program's instructions. Selenium has the following exceptions:

* TimeoutException - It is thrown when a command performing an operation does not complete in the stipulated time.
* NoSuchElementException - It is thrown when an element with specific attributes is not found on the web page.
* ElementNotVisibleException - It is thrown when an element is present in Document Object Model (DOM) but is not visible. Ex: Hidden Elements defined in HTML using type=“hidden”.
* SessionNotFoundException - The WebDriver is performing the action immediately after quitting the browser.

42. Can Selenium test an application on an Android browser?

Selenium is capable of testing an application on an Android browser using an Android driver. You can use the Selendroid or Appium framework to test native apps or web apps in the Android browser. The following is a sample code:



43. What are the different test types that Selenium supports?

Functional - This is a type of black-box testing in which the test cases are based on the software specification.

Regression - This testing helps to find new errors, regressions, etc. in different functional and non-functional areas of code after the alteration.

Load Testing - This testing seeks to monitor the response of a device after putting a load on it. It is carried out to study the behavior of the system under certain conditions.

44. How can you access the text of a web element?

Get command is used to retrieve the text of a specified web element. The command does not return any parameter but returns a string value.

Used for:

* Verification of messages
* Labels
* Errors displayed on the web page

Syntax:

String Text=driver.findElement(By.id(“text”)).getText();

45. How can you handle keyboard and mouse actions using Selenium?

You can handle keyboard and mouse events with the advanced user interaction API. The advanced user interactions API contains actions and action classes.

|  |  |
| --- | --- |
| Method | Description |
| clickAndHold() | Clicks without releasing the current mouse location |
| dragAndDrop() | Performs click-and-hold at the location of the source element |
| keyDown(modifier\_key) | Performs a modifier key press (ctrl, shift, Fn, etc.) |
| keyUp(modifier\_key) | Performs a key release |

46. Which of these options is not a WebElement method?

1. getText()
2. size()
3. getTagName()
4. sendKeys()

The correct answer is B) size()

47. When do we use findElement() and findElements()?

A. findElement()

It finds the first element in the current web page that matches the specified locator value.

Syntax:

WebElement element=driver.findElements(By.xpath(“//div[@id=‘example’]//ul//li”));

B. findElements()

It finds all the elements in the current web page that matches the specified locator value.

Syntax:

List elementList=driver.findElements(By.xpath(“//div[@id=‘example’]//ul//li”));

48. What are driver.close() and driver.quit() in WebDriver?

These are two different methods used to close the browser session in Selenium WebDriver:

* driver.close() - This is used to close the current browser window on which the focus is set. In this case, there is only one browser open.
* driver.quit() - It closes all the browser windows and ends the WebDriver session using the driver.dispose method.

49. How can you submit a form using Selenium?

The following lines of code will let you submit a form using Selenium:

WebElement el = driver.findElement(By.id(“ElementID”));

el.submit();

DevOps Career Guide

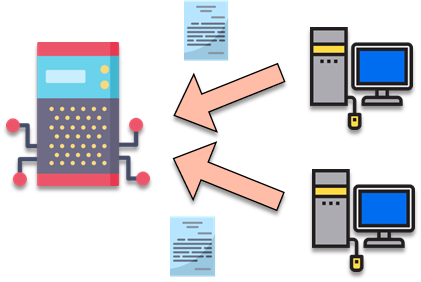
A Guide to Becoming A DevOps Engineer[DOWNLOAD NOW](https://www.simplilearn.com/tutorials/devops-tutorial/devops-interview-questions)



DevOps Interview Questions for Configuration Management — Chef, Puppet, Ansible

50. Why are SSL certificates used in Chef?

* SSL certificates are used between the Chef server and the client to ensure that each node has access to the right data.
* Every node has a private and public key pair. The public key is stored at the Chef server.
* When an SSL certificate is sent to the server, it will contain the private key of the node.
* The server compares this against the public key in order to identify the node and give the node access to the required data.



51. Which of the following commands would you use to stop or disable the 'httpd' service when the system boots?

1. # systemctl disable httpd.service
2. # system disable httpd.service
3. # system disable httpd
4. # systemctl disable httpd.service

The correct answer is A) # systemctl disable httpd.service

52. What is Test Kitchen in Chef?

Test Kitchen is a command-line tool in Chef that spins up an instance and tests the cookbook on it before deploying it on the actual nodes.

Here are the most commonly used kitchen commands:



53. How does chef-apply differ from chef-client?

* chef-apply is run on the client system.  
    
  chef-apply applies the recipe mentioned in the command on the client system.  
    
  $ chef-apply recipe\_name.rb
* chef-client is also run on the client system.  
    
  chef-client applies all the cookbooks in your server's run list to the client system.  
    
  $ knife chef-client

54. What is the command to sign the requested certificates?

* For Puppet version 2.7:  
    
  # puppetca –sign hostname-of-agent  
    
  Example:  
    
  # puppetca –sign ChefAgent  
    
  # puppetca sign hostname-of-agent  
    
  Example:  
    
  # puppetca sign ChefAgent
* For Puppet version 2.7:  
    
  # puppetca –sign hostname-of-agent  
    
  Example:  
    
  # puppetca –sign ChefAgent  
    
  # puppetca sign hostname-of-agent  
    
  Example:  
    
  # puppetca sign ChefAgent

55. Which open source or community tools do you use to make Puppet more powerful?

* Changes in the configuration are tracked using Jira, and further maintenance is done through internal procedures.
* Version control takes the support of Git and Puppet's code manager app.
* The changes are also passed through Jenkin's continuous integration pipeline.

56. What are the resources in Puppet?

* Resources are the basic units of any configuration management tool.
* These are the features of a node, like their software packages or services.
* A resource declaration, written in a catalog, describes the action to be performed on or with the resource.
* When the catalog is executed, it sets the node to the desired state.

57. What is a class in Puppet?

Classes are named blocks in your manifest that configure various functionalities of the node, such as services, files, and packages.

The classes are added to a node's catalog and are executed only when explicitly invoked.

Class apache (String $version = ‘latest’) {

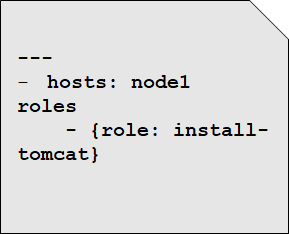
package{

‘httpd’: ensure => $version,

before => File[‘/etc/httpd.conf’],}

58. What is an Ansible role?

An Ansible role is an independent block of tasks, variables, files, and templates embedded inside a playbook.



This playbook installs tomcat on node1.

59. When should I use '{{ }}'?

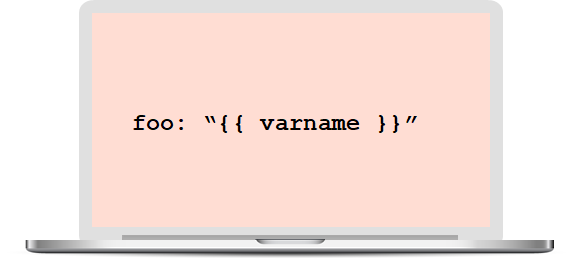
Always use {{}} for variables, unless you have a conditional statement, such as "when: …". This is because conditional statements are run through Jinja, which resolves the expressions.

 For example:

      echo “This prints the value of {{foo}}”

      when : foo is defined

Using brackets makes it simpler to distinguish between strings and undefined variables.



This also ensures that Ansible doesn't recognize the line as a dictionary declaration.

60. What is the best way to make content reusable/redistributable?

There are three ways to make content reusable or redistributable in Ansible:

* Roles are used to managing tasks in a playbook. They can be easily shared via Ansible Galaxy.
* "include" is used to add a submodule or another file to a playbook. This means a code written once can be added to multiple playbooks.
* "import" is an improvement of "include," which ensures that a file is added only once. This is helpful when a line is run recursively.

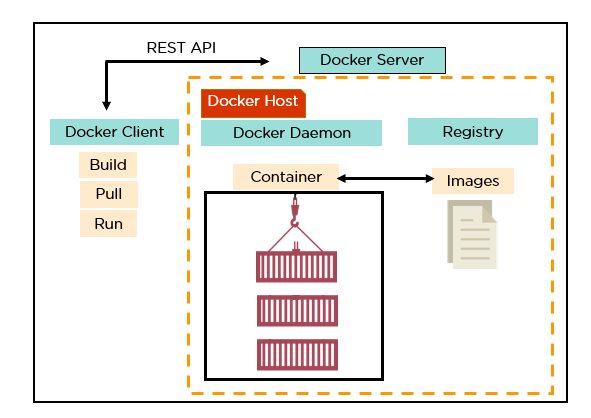
61. How is Ansible different from Puppet?

|  |  |
| --- | --- |
| Ansible | Puppet |
| Easy agentless installation | Agent-based installation |
| Based on Python | Based on Ruby |
| Configuration files are written in YAML | Configuration files are written in DSL |
| No support for Windows | Support for all popular OS's |

DevOps Interview Questions on Containerization

62. Explain the architecture of Docker.

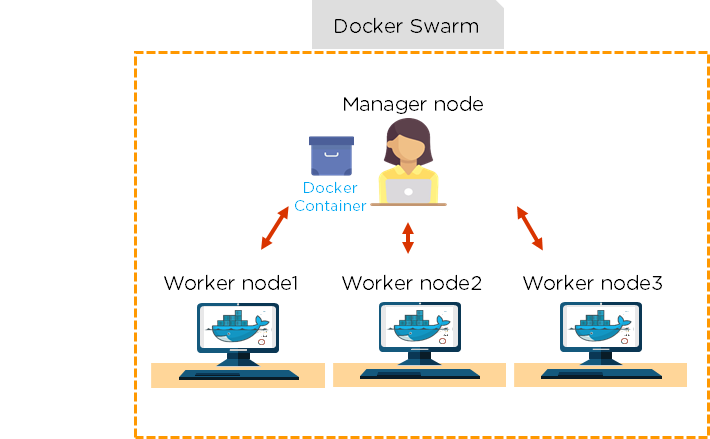
* [Docker](https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docke) uses a client-server architecture.
* Docker Client is a service that runs a command. The command is translated using the REST API and is sent to the Docker Daemon (server).
* Docker Daemon accepts the request and interacts with the operating system to build Docker images and run Docker containers.
* A Docker image is a template of instructions, which is used to create containers.
* [Docker container](https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docker-container) is an executable package of an application and its dependencies together.
* Docker registry is a service to host and distribute Docker images among users.



63. What are the advantages of Docker over virtual machines?

|  |  |  |
| --- | --- | --- |
| Criteria | Virtual Machine | Docker |
| Memory space | Occupies a lot of memory space | Docker containers occupy less space |
| Boot-up time | Long boot-up time | Short boot-up time |
| Performance | Running multiple virtual machines leads to unstable performance | Containers have a better performance, as they are hosted in a single Docker engine |
| Scaling | Difficult to scale up | Easy to scale up |
| Efficiency | Low efficiency | High efficiency |
| Portability | Compatibility issues while porting across different platforms | Easily portable across different platforms |
| Space allocation | Data volumes cannot be shared | Data volumes are shared and used again across multiple containers |

64. How do we share Docker containers with different nodes?



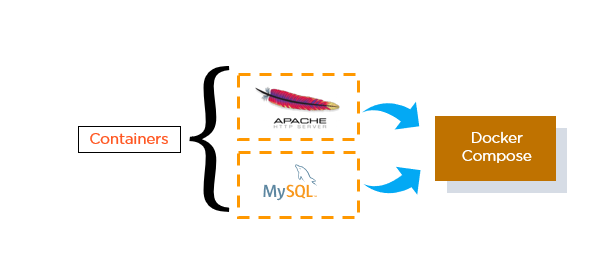
* It is possible to share Docker containers on different nodes with [Docker Swarm](https://www.simplilearn.com/tutorials/docker-tutorial/docker-swarm).
* Docker Swarm is a tool that allows IT administrators and developers to create and manage a cluster of swarm nodes within the Docker platform.
* A swarm consists of two types of nodes: a manager node and a worker node.

65. What are the commands used to create a Docker swarm?

* Create a swarm where you want to run your manager node.  
    
  Docker swarm init --advertise-addr <MANAGER-IP>
* Once you've created a swarm on your manager node, you can add worker nodes to your swarm.
* When a node is initialized as a manager, it immediately creates a token. In order to create a worker node, the following command (token) should be executed on the host machine of a worker node.  
    
  docker swarm join \ --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \ 192.168.99.100:2377

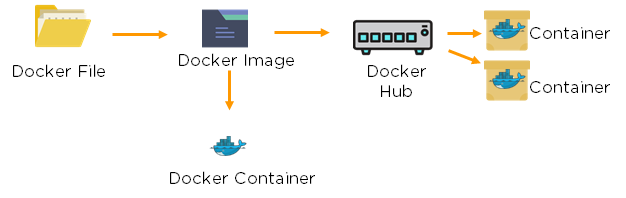
66. How do you run multiple containers using a single service?

* It is possible to run multiple containers as a single service with Docker Compose.
* Here, each container runs in isolation but can interact with each other.
* All Docker Compose files are YAML files.



67. What is a Dockerfile used for?

* A Dockerfile is used for creating Docker images using the build command.
* With a Docker image, any user can run the code to create Docker containers.
* Once a Docker image is built, it's uploaded in a Docker registry.
* From the Docker registry, users can get the Docker image and build new containers whenever they want.



68. Explain the differences between Docker images and Docker containers.

|  |  |
| --- | --- |
| Docker Images | Docker Container |
| Docker images are templates of Docker containers | Containers are runtime instances of a Docker image |
| An image is built using a Dockerfile | Containers are created using Docker images |
| It is stored in a Docker repository or a Docker hub | They are stored in the Docker daemon |
| The image layer is a read-only filesystem | Every container layer is a read-write filesystem |

69. Instead of YAML, what can you use as an alternate file for building Docker compose?

To build a Docker compose, a user can use a JSON file instead of YAML. In case a user wants to use a JSON file, he/she should specify the filename as given:

Docker-compose -f Docker-compose.json up

70. How do you create a Docker container?

Task: Create a MySQL Docker container

A user can either build a Docker image or pull an existing Docker image (like MySQL) from Docker Hub.

Now, Docker creates a new container MySQL from the existing Docker image. Simultaneously, the container layer of the read-write filesystem is also created on top of the image layer.

* Command to create a Docker container: Docker run -t –i MySQL
* Command to list down the running containers: Docker ps

71. What is the difference between a registry and a repository?

|  |  |
| --- | --- |
| Registry | Repository |
| A Docker registry is an open-source server-side service used for hosting and distributing Docker images | The repository is a collection of multiple versions of Docker images |
| In a registry, a user can distinguish between Docker images with their tag names | It is stored in a Docker registry |
| Docker also has its own default registry called Docker Hub | It has two types: public and private repositories |

72. What are the cloud platforms that support Docker?

The following are the cloud platforms that Docker runs on:

* [Amazon Web Services](https://www.simplilearn.com/tutorials/aws-tutorial/what-is-aws)
* [Microsoft Azure](https://www.simplilearn.com/tutorials/azure-tutorial/what-is-azure)
* Google Cloud Platform
* Rackspace

73. What is the purpose of the expose and publish commands in Docker?

Expose

* Expose is an instruction used in Dockerfile.
* It is used to expose ports within a Docker network.
* It is a documenting instruction used at the time of building an image and running a container.
* Expose is the command used in Docker.
* Example: Expose 8080

Publish

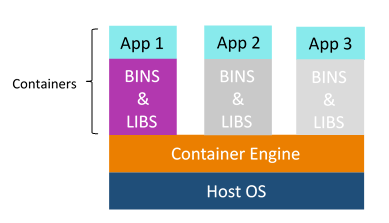
* Publish is used in a Docker run command.
* It can be used outside a Docker environment.
* It is used to map a host port to a running container port.
* --publish or –p is the command used in Docker.
* Example: docker run –d –p 0.0.0.80:80

**Containerization and Virtualization Interview Questions**

Let’s see how much you know about containers and VMs.

**Q1. What are containers?**

My suggestion is to explain the need for containerization first, containers are used to provide consistent computing environment from a developer’s laptop to a test environment, from a staging environment into production.  
Now give a definition of containers, a container consists of an entire runtime environment: an application, plus all its dependencies, libraries and other binaries, and configuration files needed to run it, bundled into one package. Containerizing the application platform and its dependencies removes the differences in OS distributions and underlying infrastructure.



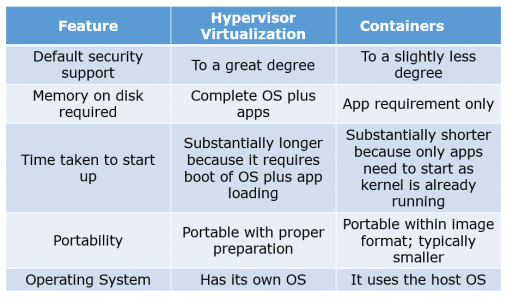
**Q2. What are the advantages that Containerization provides over virtualization?**

Below are the advantages of containerization over virtualization:

* Containers provide real-time provisioning and scalability but VMs provide slow provisioning
* Containers are lightweight when compared to VMs
* VMs have limited performance when compared to containers
* Containers have better resource utilization compared to VMs

**Q3. How exactly are containers (Docker in our case) different from hypervisor virtualization (vSphere)? What are the benefits?**

Given below are some differences. Make sure you include these differences in your answer:



**Q4. What is Docker image?**

I suggest that you go with the below mentioned flow:  
Docker image is the source of Docker container. In other words, Docker images are used to create containers. Images are created with the build command, and they’ll produce a container when started with run. Images are stored in a Docker registry such as registry.hub.docker.com because they can become quite large, images are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.  
**Tip: Be aware of Dockerhub in order to answer questions on pre-available images.**

**Q5. What is Docker container?**

This is a very important question so just make sure you don’t deviate from the topic. I advise you to follow the below mentioned format:  
Docker containers include the application and all of its dependencies but share the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud.  
Now explain how to create a Docker container, Docker containers can be created by either creating a Docker image and then running it or you can use Docker images that are present on the Dockerhub.  
Docker containers are basically runtime instances of Docker images.

**Q6. What is Docker hub?**

Answer to this question is pretty direct. Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

**Q7.** **How is Docker different from other container technologies?**

According to me, below points should be there in your answer:  
Docker containers are easy to deploy in a cloud. It can get more applications running on the same hardware than other technologies, it makes it easy for developers to quickly create, ready-to-run containerized applications and it makes managing and deploying applications much easier. You can even share containers with your applications.  
If you have some more points to add you can do that but make sure the above the above explanation is there in your answer.

**Q8.** **What is Docker Swarm?**

You should start this answer by explaining Docker Swarn. It is native clustering for Docker which turns a pool of Docker hosts into a single, virtual Docker host. Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts.  
I will also suggest you to include some supported tools:

* Dokku
* Docker Compose
* Docker Machine
* Jenkins

**Q9. What is Dockerfile used for?**

This answer according to me should begin by explaining the use of Dockerfile. Docker can build images automatically by reading the instructions from a Dockerfile.  
Now I suggest you to give a small definition of Dockerfle. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

Now expect a few questions to test your experience with Docker.

**Q10.** **Can I use json instead of yaml for my compose file in Docker?**

You can use json instead of yaml for your compose file, to use json file with compose, specify the filename to use for eg:  
**docker-compose -f docker-compose.json up**

**Q11.** **Tell us how you have used Docker in your past position?**

Explain how you have used Docker to help rapid deployment. Explain how you have scripted Docker and used Docker with other tools like Puppet, Chef or Jenkins. If you have no past practical experience in Docker and have past experience with other tools in similar space, be honest and explain the same. In this case, it makes sense if you can compare other tools to Docker in terms of functionality.

**Q12. How to create Docker container?**

I will suggest you to give a direct answer to this. We can use Docker image to create Docker container by using the below command:  
**docker run -t -i <image name> <command name>**This command will create and start container.  
You should also add, If you want to check the list of all running container with status on a host use the below command:  
**docker ps -a**

**Q13. How to stop and restart the Docker container?**

In order to stop the Docker container you can use the below command:  
**docker stop <container ID>**  
Now to restart the Docker container you can use:  
**docker restart <container ID>**

Q14. How far do Docker containers scale?

Large web deployments like Google and Twitter, and platform providers such as Heroku and dotCloud all run on container technology, at a scale of hundreds of thousands or even millions of containers running in parallel.

**[DevOps Certification Training](https://www.edureka.co/devops-certification-training" \t "_blank)**

Q15. What platforms does Docker run on?

I will start this answer by saying Docker runs on only Linux and Cloud platforms and then I will mention the below vendors of Linux:

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

Cloud:

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

**Note that Docker does not run on Windows or Mac.**

**Q16. Do I lose my data when the Docker container exits?**

You can answer this by saying, no I won’t loose my data when Dcoker container exits. Any data that your application writes to disk gets preserved in its container until you explicitly delete the container. The file system for the container persists even after the container halts.