Last updated on Mar 8, 2024

Share this

If a career in development & operations is what you want, then becoming a DevOps developer is your answer. However, going through the DevOps technical interview questions isn't exactly a breezy affair. Moreover, if you're a hiring manager looking to recruit top DevOps developers, gauging each candidate's skills can be tedious if you don't ask the right questions.

That's why we've created this list of 100 DevOps interview questions that are most commonly asked during DevOps interviews. Whether you want a DevOps job or want to hire a skilled DevOps engineer, these DevOps interview questions and answers will help you out.

Table of contents

BASIC DEVOPS INTERVIEW QUESTIONS AND ANSWERS

1. What benefits does DevOps have in business?

Hide Answer

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

- **Faster time to market**: DevOps practices can help to streamline the development and deployment process, allowing for faster delivery of new products and features.
- **Increased collaboration**: DevOps promotes collaboration between development and operations teams, resulting in better communication, more efficient problem-solving, and higher-quality software.
- **Improved agility**: DevOps allows for more rapid and flexible responses to changing business needs and customer demands.

- **Increased reliability**: DevOps practices such as continuous testing, monitoring, and automated deployment can help to improve the reliability and stability of software systems.
- **Greater scalability:** DevOps practices can help to make it easier to scale systems to meet growing business needs and user demand.
- **Cost savings**: DevOps can help to reduce the costs associated with the development, deployment, and maintenance of software systems by automating many manual processes and reducing downtime.
- **Better security**: DevOps practices such as continuous testing and monitoring can help to improve the security of software systems.

2. What are the key components of a successful DevOps workflow?

Hide Answer

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

3. What are the different phases of the DevOps lifecycle?

Hide Answer

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

- **Plan**: Define project goals, requirements, and resources
- **Code**: Develop and write code
- **Build**: Compile code into executable software
- **Test**: Verify and validate software functionality
- **Release**: Deploy code to the production environment
- **Deploy**: Automated deployment and scaling of software
- **Operate:** Monitor and maintain the software in production

- **Monitor:** Collect and analyze software performance data
- **Optimize:** Continuously improve and evolve the software system

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

Hide Answer

The best programming and scripting languages DevOps engineers must know are as follows:

Programming languages:-

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

Scripting languages:-

- JavaScript
- Python
- Ruby
- Perl
- Groovy

5. Explain configuration management in DevOps.

Hide Answer

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

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

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

---

6. Name and explain trending DevOps tools.

Hide Answer

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

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

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

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

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

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

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

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

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

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

7. How would you strategize for a successful DevOps implementation?

Hide Answer

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

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

8. What role does AWS play in DevOps?

Hide Answer

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

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

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

9. DevOps vs. Agile: How are they different?

Hide Answer

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

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

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

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

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

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

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

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

Hide Answer

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

11. **Explain Component-based development in DevOps.**

Hide Answer

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

12. **How is version control crucial in DevOps?**

Hide Answer

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

13. **Describe continuous integration.**

Hide Answer

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

14. **What is continuous delivery?**

Hide Answer

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

Learn more about CI/CD here.

---

15. **Explain continuous testing.**

Hide Answer

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

---

16. **What is continuous monitoring?**

Hide Answer

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

---

17. **What key metrics should you focus on for DevOps success?**

Hide Answer

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

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

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

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

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

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

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

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

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

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

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

18. List down the types of HTTP requests.

Hide Answer

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

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

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

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

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

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

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

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

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

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

19. What is the role of automation in DevOps?

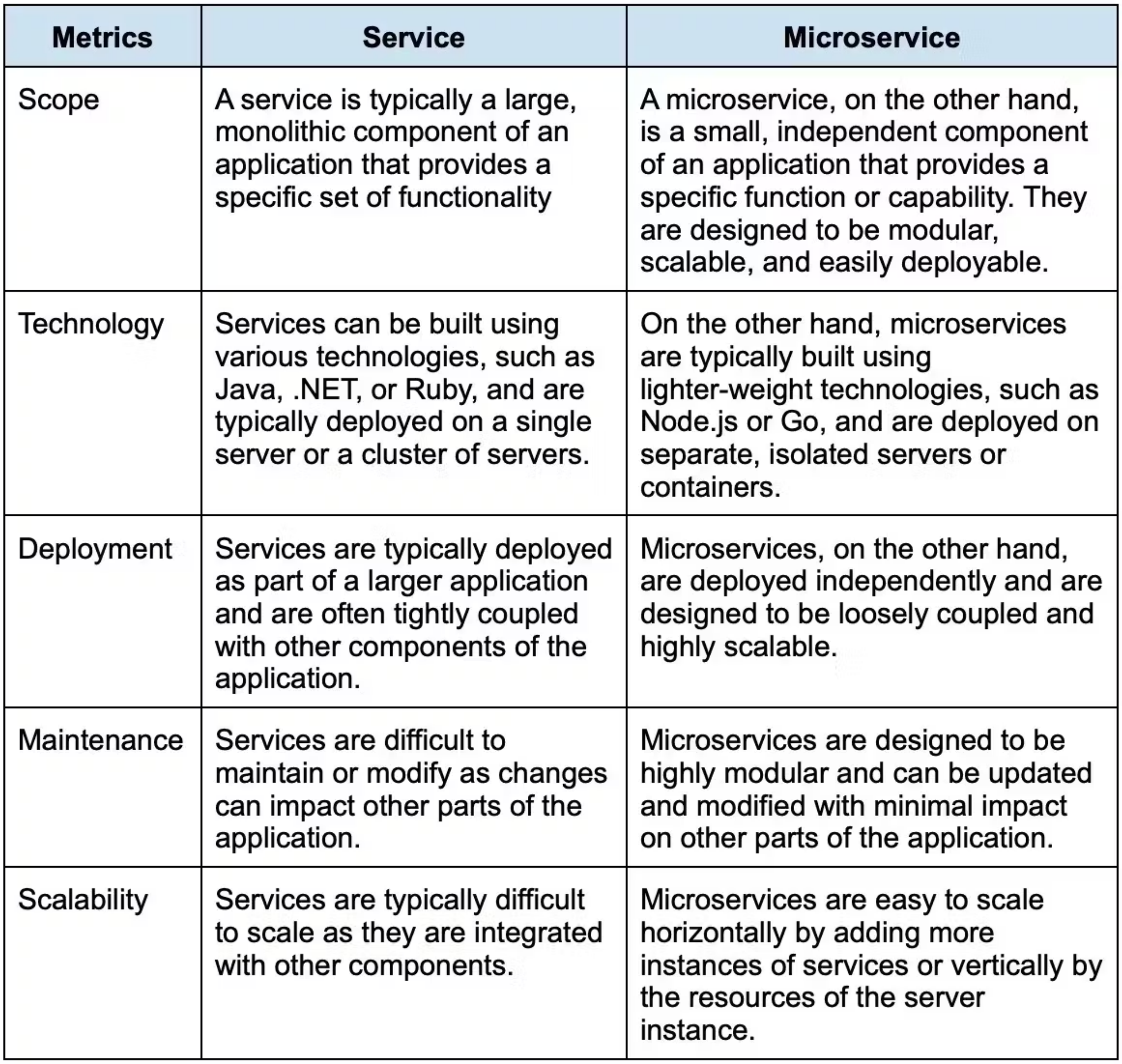
Hide Answer

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

20. What is the difference between a service and a microservice?

Hide Answer

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



Metrics	Service	Microservice
Scope	A service is typically a large, monolithic component of an application that provides a specific set of functionality	A microservice, on the other hand, is a small, independent component of an application that provides a specific function or capability. They are designed to be modular, scalable, and easily deployable.
Technology	Services can be built using various technologies, such as Java, .NET, or Ruby, and are typically deployed on a single server or a cluster of servers.	On the other hand, microservices are typically built using lighter-weight technologies, such as Node.js or Go, and are deployed on separate, isolated servers or containers.
Deployment	Services are typically deployed as part of a larger application and are often tightly coupled with other components of the application.	Microservices, on the other hand, are deployed independently and are designed to be loosely coupled and highly scalable.
Maintenance	Services are difficult to maintain or modify as changes can impact other parts of the application.	Microservices are designed to be highly modular and can be updated and modified with minimal impact on other parts of the application.
Scalability	Services are typically difficult to scale as they are integrated with other components.	Microservices are easy to scale horizontally by adding more instances of services or vertically by the resources of the server instance.

---

21. How do you secure a CI/CD pipeline?

Hide Answer

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

- Ensure all tools and dependencies are up to date
- Implement strong access controls and authentication
- Scan code for vulnerabilities (e.g., SonarQube, OWASP Dependency-Check)
- Cloud provider managed private build environments (e.g., AWS CodeBuild)

- Store sensitive data like keys, tokens, and passwords in a secret management tool (e.g., HashiCorp Vault, AWS Secrets Manager)
- Regularly audit infrastructure and system logs for anomalies

22. How does incident management fit into the DevOps workflow?

Hide Answer

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

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

Hide Answer

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

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

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

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

Hide Answer

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

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

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

INTERMEDIATE DEVOPS TECHNICAL INTERVIEW QUESTIONS AND ANSWERS

1. Explain how you will handle merge conflicts in Git.

Hide Answer

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

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

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

2. Mention some advantages of Forking workflow over other Git workflows.

Hide Answer

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

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

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

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

---

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

Hide Answer

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

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

---

4. What are automation testing and continuous testing?

Hide Answer

Automation testing is a process that automates the manual testing process. Different testing tools allow developers to generate test scripts that can be executed continually without the need for human intervention.

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

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

5. Mention the technical challenges with Selenium.

Hide Answer

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

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

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

6. What are Puppet Manifests?

Hide Answer

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

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

7. Explain the working of Ansible.

Hide Answer

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

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

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

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

8. Explain the Sudo concept in Linux.

Hide Answer

The sudo (superuser do) command in Linux is a powerful utility that allows users to execute commands with the privileges of another user, usually the superuser or root.

The sudo concept provides a controlled way of managing which users can perform administrative tasks without granting them unrestricted root access.

9. What is the purpose of SSH?

Hide Answer

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

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

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

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

10. Talk about Nagios Log server.

Hide Answer

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

11. Explain how you will handle sensitive data in DevOps.

Hide Answer

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

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

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

Hide Answer

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

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

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

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

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

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

---

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

Hide Answer

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

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

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

14. **Explain how Nagios works.**

Hide Answer

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

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

15. **List the trending Jenkins plugins.**

Hide Answer

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

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

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

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

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

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

16. Explain Docker Swarms.

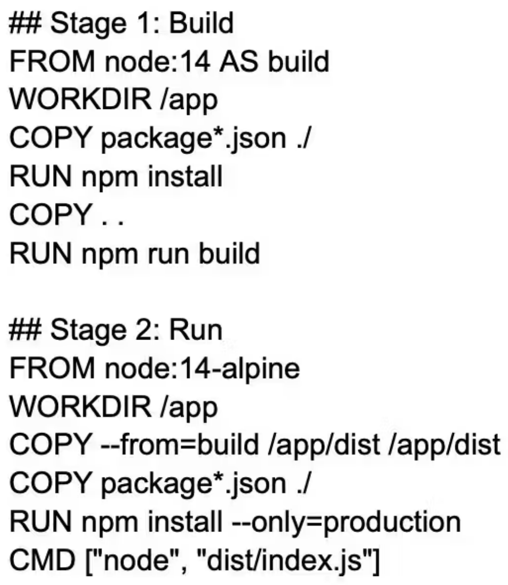
Hide Answer

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

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

Hide Answer

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



```
## Stage 1: Build
FROM node:14 AS build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

## Stage 2: Run
FROM node:14-alpine
WORKDIR /app
COPY --from=build /app/dist /app/dist
COPY package*.json ./
RUN npm install --only=production
CMD ["node", "dist/index.js"]
```

---

18. How do you ensure compliance adherence in a DevOps environment?

Hide Answer

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

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

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

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

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

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

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

Hide Answer

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

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

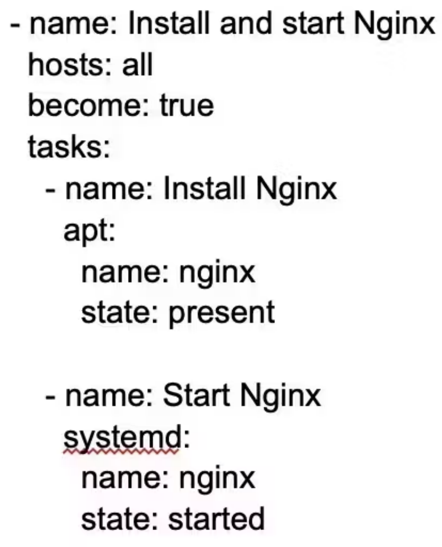
This approach is popular in DevOps as it helps to improve productivity as operations and administrators don't need to conduct manual configuration for data center infrastructure changes. Moreover, IaC also offers better reliability as the infrastructure configuration is stored in digital files, reducing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

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

Hide Answer

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

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



```
- name: Install and start Nginx
  hosts: all
  become: true
  tasks:
    - name: Install Nginx
      apt:
        name: nginx
        state: present

    - name: Start Nginx
      systemd:
        name: nginx
        state: started
```

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

Hide Answer

The term ‘cloud native’ is used to describe any application built to reside in the cloud from its very beginning. Cloud-native comprises cloud technologies such as container
orchestrators, auto-scaling, and microservices.

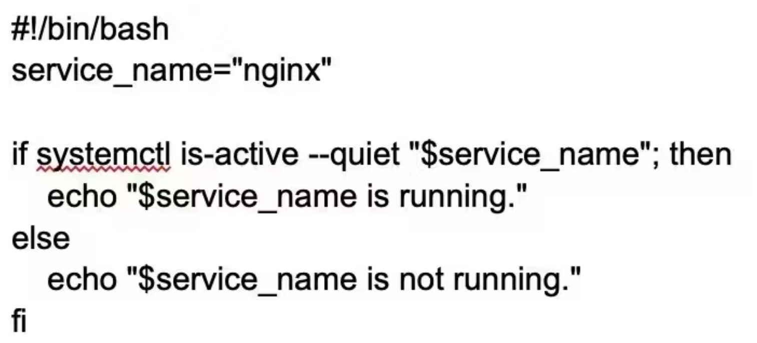
22. How can you get a list of every ansible_variable?

Hide Answer

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

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

Hide Answer



```
#!/bin/bash
service_name="nginx"

if systemctl is-active --quiet "$service_name"; then
   echo "$service_name is running."
else
   echo "$service_name is not running."
fi
```

---

24. Name the platforms Docker currently runs on.

Hide Answer

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

Linux:

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

Cloud:

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

Mac:
Docker runs on macOS 10.13 and newer versions.

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

25. What is the usage of a Dockerfile?

Hide Answer

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

26. List the top configuration management tools.

Hide Answer

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

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

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

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

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

27. How do you use Docker for containerization?

Hide Answer

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

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

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

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

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

---

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

Hide Answer

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

Some examples of these anti-patterns include:

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

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

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

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

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

29. Clarify the usage of Selenium for automated testing

Hide Answer

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

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

30. What is auto-scaling in Kubernetes?

Hide Answer

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

31. Explain EUCALYPTUS.

Hide Answer

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

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

Hide Answer

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

33. How to use Istio for service mesh?

Hide Answer

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

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

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

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

34. List the branching strategies you've used previously.

Hide Answer

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

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

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

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

35. Explain the usage of Grafana for data visualization.

Hide Answer

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

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

36. How do you use Elasticsearch for log analysis?

Hide Answer

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

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

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

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

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

37. How do you use Kubernetes for rolling updates?

Hide Answer

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

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

38. What are the benefits of A/B testing?

Hide Answer

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

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

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

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

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

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

39. Give a complete overview of Jenkins architecture.

Hide Answer

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

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

ADVANCED DEVOPS INTERVIEW QUESTIONS AND ANSWERS

1. Explain the Shift Left to Reduce Failure concept in DevOps.

Hide Answer

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

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

2. What are 'post mortem' meetings in DevOps?

Hide Answer

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

3. Explain the concept of 'pair programming'.

Hide Answer

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

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

---

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

Hide Answer

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

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

---

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

Hide Answer

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

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

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

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

6. Explain how the canary deployment works.

Hide Answer

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

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

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

Hide Answer

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

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

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

Hide Answer

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

Before deploying on a production environment

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

During deployment

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

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

Hide Answer

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

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

10. What is ‘session affinity’?

Hide Answer

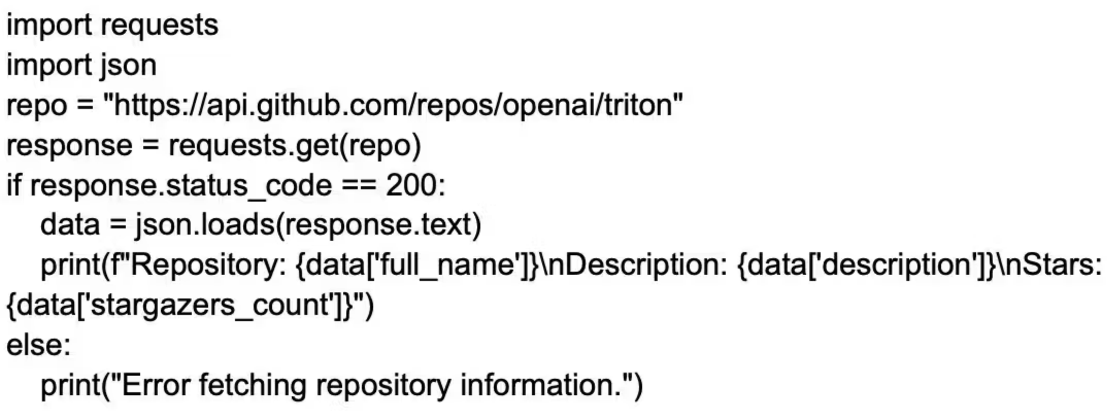
The session affinity technique, also known as a sticky session, is a popular load-balancing technique requiring an allocated machine always to serve a user session. When user information is stored in a load balancer server app during a session, the session data will be needed to be available to all machines.

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

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

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

Hide Answer



```
import requests
import json
repo = "https://api.github.com/repos/openai/triton"
response = requests.get(repo)
if response.status_code == 200:
   data = json.loads(response.text)
   print(f"Repository: {data['full_name']}\nDescription: {data['description']}\nStars:
{data['stargazers_count']}")
else:
   print("Error fetching repository information.")
```

12. How are SOA, monolithic, and microservices architecture different?

Hide Answer

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

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

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

Hide Answer

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

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

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

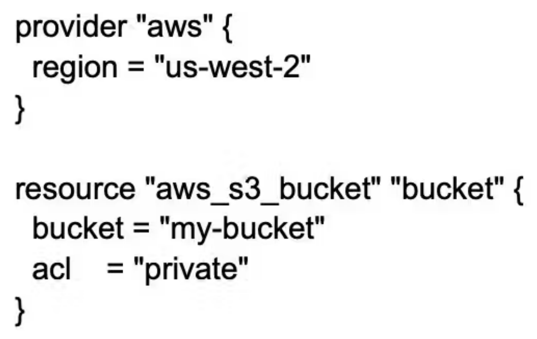
14. What are Puppet Modules, and how are they different from Puppet Manifests?

Hide Answer

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

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

Hide Answer



```
provider "aws" {
  region = "us-west-2"
}

resource "aws_s3_bucket" "bucket" {
  bucket = "my-bucket"
  acl    = "private"
}
```

16. Why should Nagios be used for HTTP monitoring?

Hide Answer

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

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

17. Explain the various components of Selenium.

Hide Answer

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

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

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

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

18. Is Docker better than virtual machines? Explain why.

Hide Answer

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

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

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

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

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

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

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

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

---

19. Explain the usage of SSL certificates in Chef.

Hide Answer

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

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

---

20. What is 'state stalking' in Nagios?

Hide Answer

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

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

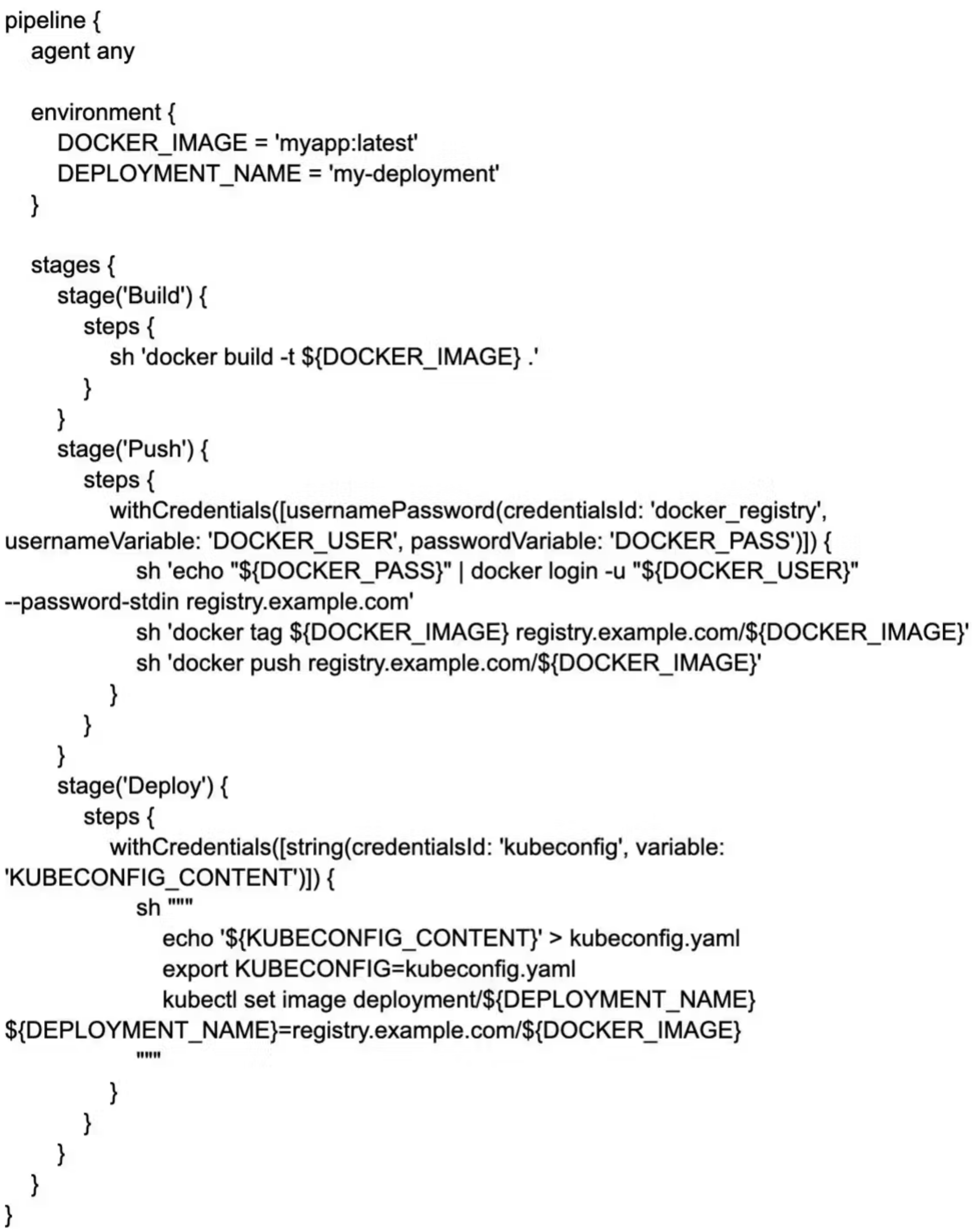
Hide Answer

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

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

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

Hide Answer



```
pipeline {
  agent any

  environment {
    DOCKER_IMAGE = 'myapp:latest'
    DEPLOYMENT_NAME = 'my-deployment'
  }

  stages {
    stage('Build') {
      steps {
        sh 'docker build -t ${DOCKER_IMAGE} .'
      }
    }
    stage('Push') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'docker_registry',
usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
          sh 'echo "${DOCKER_PASS}" | docker login -u "${DOCKER_USER}"
--password-stdin registry.example.com'
          sh 'docker tag ${DOCKER_IMAGE} registry.example.com/${DOCKER_IMAGE}'
          sh 'docker push registry.example.com/${DOCKER_IMAGE}'
        }
      }
    }
    stage('Deploy') {
      steps {
        withCredentials([string(credentialsId: 'kubeconfig', variable:
'KUBECONFIG_CONTENT')]) {
          sh """
            echo '${KUBECONFIG_CONTENT}' > kubeconfig.yaml
            export KUBECONFIG=kubeconfig.yaml
            kubectl set image deployment/${DEPLOYMENT_NAME}
${DEPLOYMENT_NAME}=registry.example.com/${DOCKER_IMAGE}
          """
        }
      }
    }
  }
}
```

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

Hide Answer

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

- Start by copying the .hpi file to the $JENKINS_HOME/plugins.
- Next, delete the plugin’s development directory.

- Create an empty file and name it .hpi.pinned.
- Restart Jenkins, and your custom build for the core plugin will be ready.

24. How can you use WebDriver to launch the Browser?

Hide Answer

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

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

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

25. How can you turn off auto-deployment?

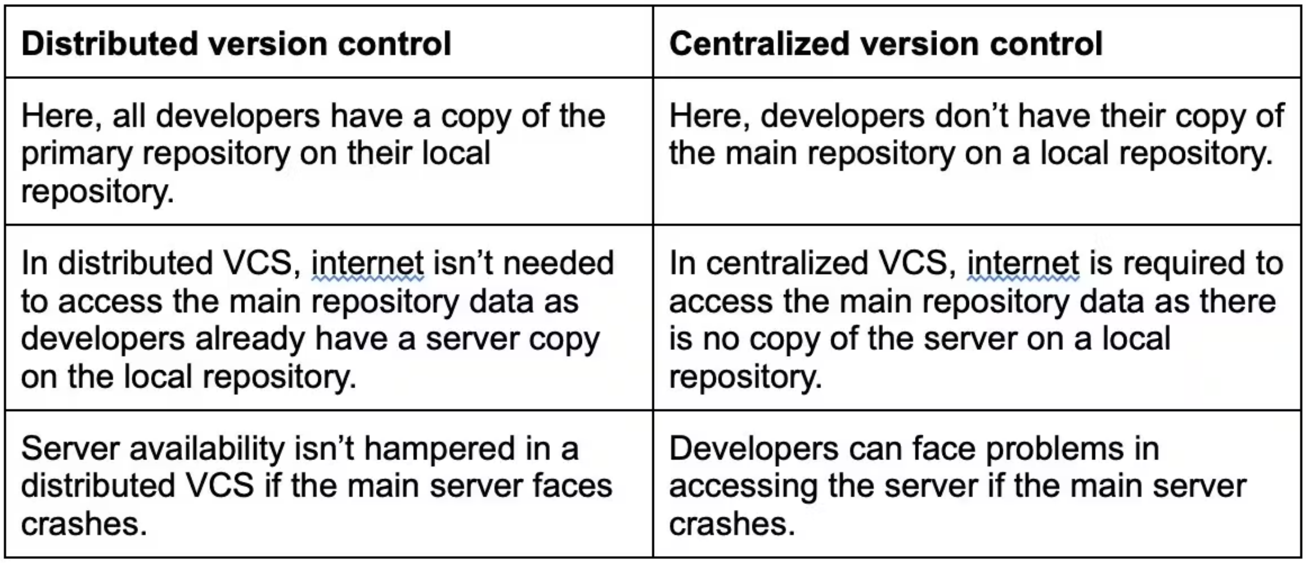
Hide Answer

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

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

26. How are distributed and centralized version control systems different?

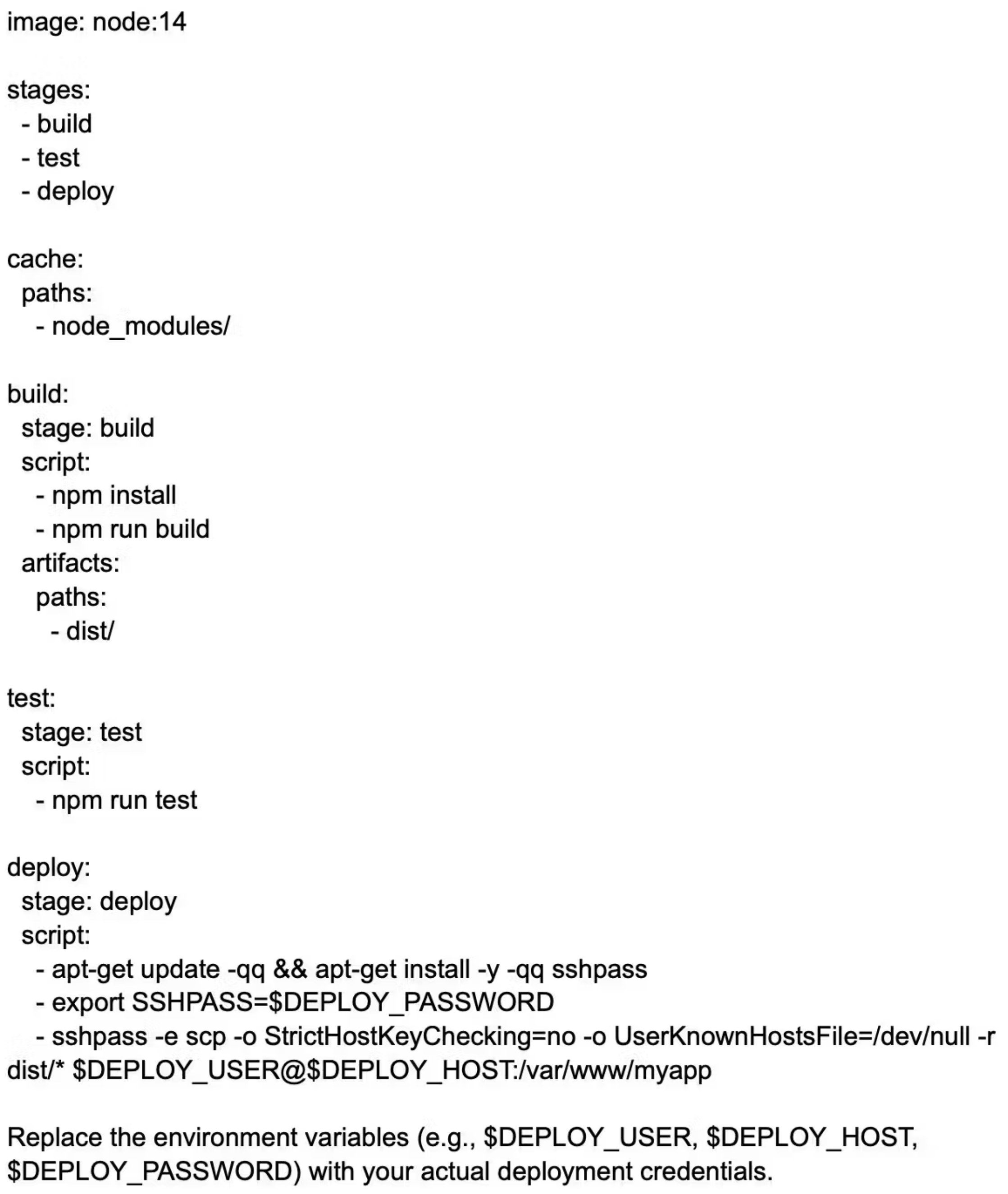
Hide Answer



Distributed version control	Centralized version control
Here, all developers have a copy of the primary repository on their local repository.	Here, developers don't have their copy of the main repository on a local repository.
In distributed VCS, internet isn't needed to access the main repository data as developers already have a server copy on the local repository.	In centralized VCS, internet is required to access the main repository data as there is no copy of the server on a local repository.
Server availability isn't hampered in a distributed VCS if the main server faces crashes.	Developers can face problems in accessing the server if the main server crashes.

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

Hide Answer



```
image: node:14

stages:
  - build
  - test
  - deploy

cache:
  paths:
    - node_modules/

build:
  stage: build
  script:
    - npm install
    - npm run build
  artifacts:
    paths:
      - dist/

test:
  stage: test
  script:
    - npm run test

deploy:
  stage: deploy
  script:
    - apt-get update -qq && apt-get install -y -qq sshpass
    - export SSHPASS=$DEPLOY_PASSWORD
    - sshpass -e scp -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -r
dist/* $DEPLOY_USER@$DEPLOY_HOST:/var/www/myapp
```

Replace the environment variables (e.g., $DEPLOY_USER, $DEPLOY_HOST, $DEPLOY_PASSWORD) with your actual deployment credentials.

28. Explain the process of setting up a Jenkins job

Hide Answer

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

- Optional triggers to control when Jenkins performs a build.
- Optional SCM, like Subversion or CVS, where the source code will be housed.

- A build script that will perform the build (maven, ant, batch file, shell script, etc.)
- Optional steps to inform other systems/people about the build result, like sending emails, updating issue trackers, IMs, etc.
- Optional steps to gather data out of the build, like recording javadoc and/or archiving artifacts and test results.

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

---

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

Hide Answer

The following steps can be taken to secure Jenkins:

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

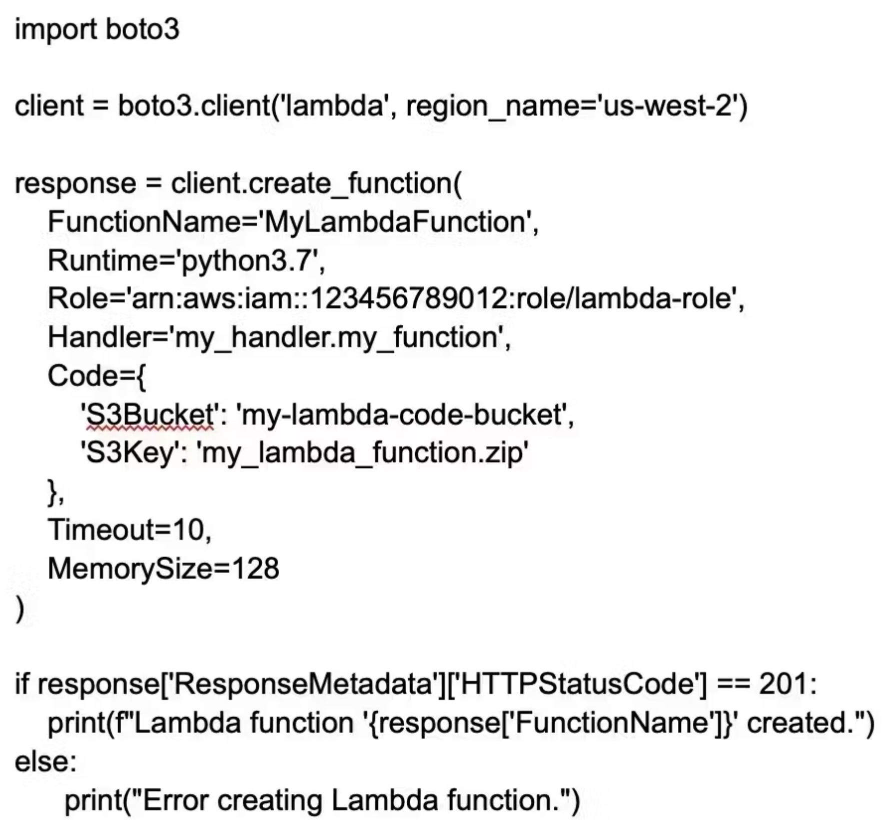
The three security mechanisms Jenkins can use to authenticate a user are:

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

---

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

Hide Answer



```
import boto3

client = boto3.client('lambda', region_name='us-west-2')

response = client.create_function(
   FunctionName='MyLambdaFunction',
   Runtime='python3.7',
   Role='arn:aws:iam::123456789012:role/lambda-role',
   Handler='my_handler.my_function',
   Code={
      'S3Bucket': 'my-lambda-code-bucket',
      'S3Key': 'my_lambda_function.zip'
   },
   Timeout=10,
   MemorySize=128
)

if response['ResponseMetadata']['HTTPStatusCode'] == 201:
   print(f"Lambda function '{response['FunctionName']}' created.")
else:
     print("Error creating Lambda function.")
```

31. How do the Verify and Assert commands differ in Selenium?

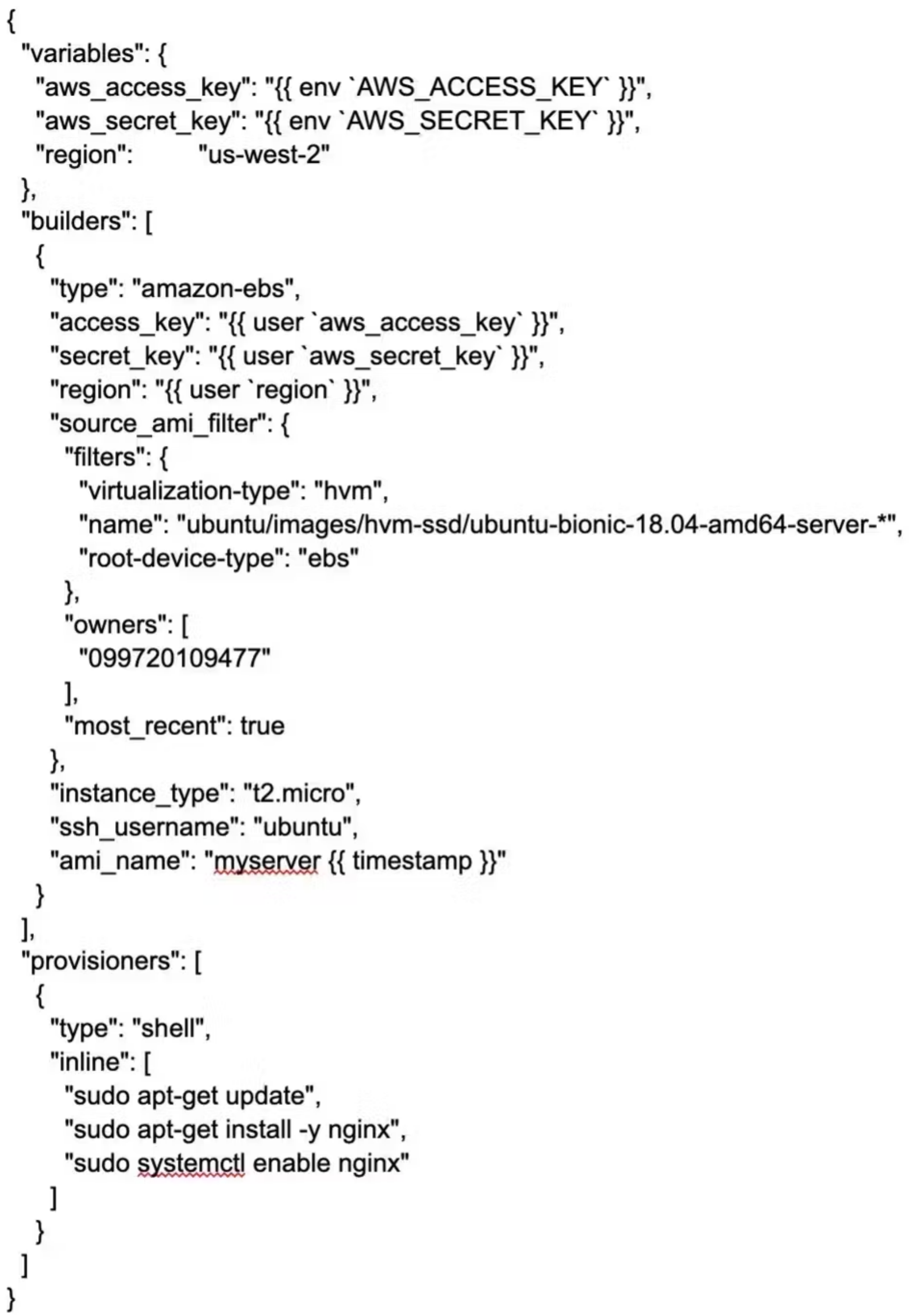
Hide Answer

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

verification failure won't halt the execution, and test steps will continue to be executed.

32. **Write a sample Packer configuration to build an AWS AMI.**

Hide Answer



```
{
 "variables": {
  "aws_access_key": "{{ env `AWS_ACCESS_KEY` }}",
  "aws_secret_key": "{{ env `AWS_SECRET_KEY` }}",
  "region":        "us-west-2"
 },
 "builders": [
  {
   "type": "amazon-ebs",
   "access_key": "{{ user `aws_access_key` }}",
   "secret_key": "{{ user `aws_secret_key` }}",
   "region": "{{ user `region` }}",
   "source_ami_filter": {
    "filters": {
     "virtualization-type": "hvm",
     "name": "ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-*",
     "root-device-type": "ebs"
    },
    "owners": [
     "099720109477"
    ],
    "most_recent": true
   },
   "instance_type": "t2.micro",
   "ssh_username": "ubuntu",
   "ami_name": "myserver {{ timestamp }}"
  }
 ],
 "provisioners": [
  {
   "type": "shell",
   "inline": [
    "sudo apt-get update",
    "sudo apt-get install -y nginx",
    "sudo systemctl enable nginx"
   ]
  }
 ]
}
```

---

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

Hide Answer

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

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

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

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

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

Hide Answer

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

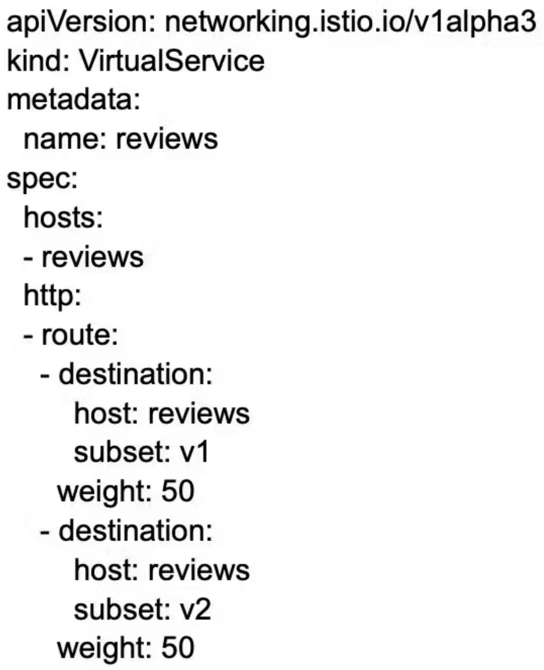
When you run a configure script, a sample main config file is created in the Nagios distribution's base directory. The file's default name is nagios.cfg and it's usually located in the etc/ subdirectory of the Nagios installation (/usr/local/bagios/etc/).

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

Hide Answer

A service mesh is a configurable infrastructure layer for the microservices application that makes communication between service instances flexible, reliable, and fast. The purpose is to handle the network communication between microservices, including load balancing, service discovery, encryption, and authentication.

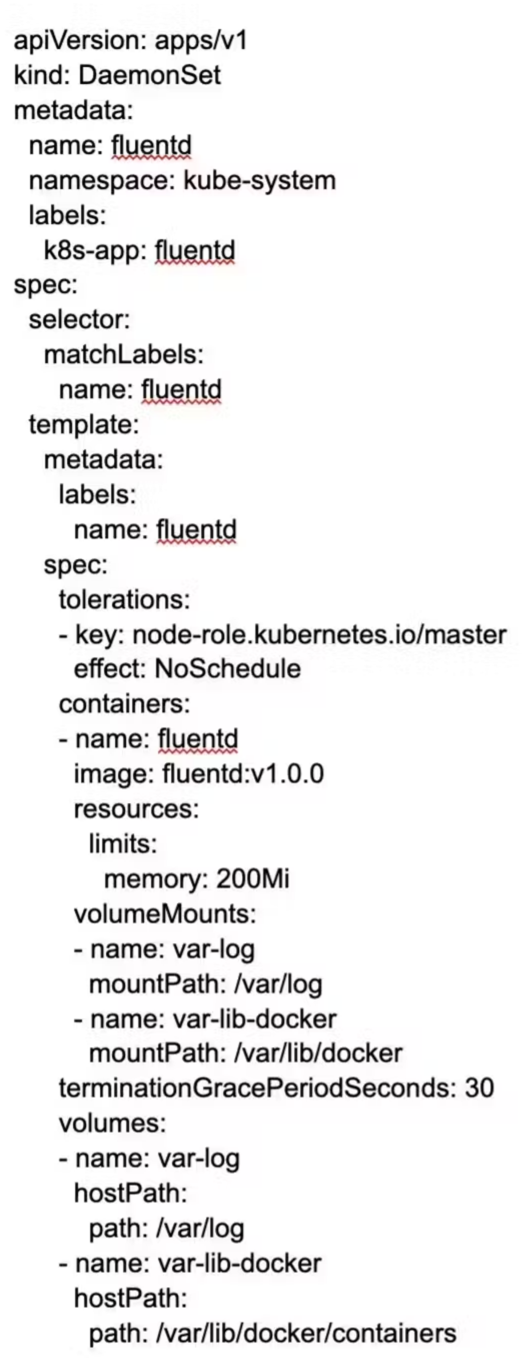
In DevOps, it simplifies network management, promotes security, and enables advanced deployment strategies. Istio is a popular service mesh that integrates with Kubernetes.



```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
      weight: 50
    - destination:
        host: reviews
        subset: v2
      weight: 50
```

36. Write a DaemonSet configuration manifest in Kubernetes.

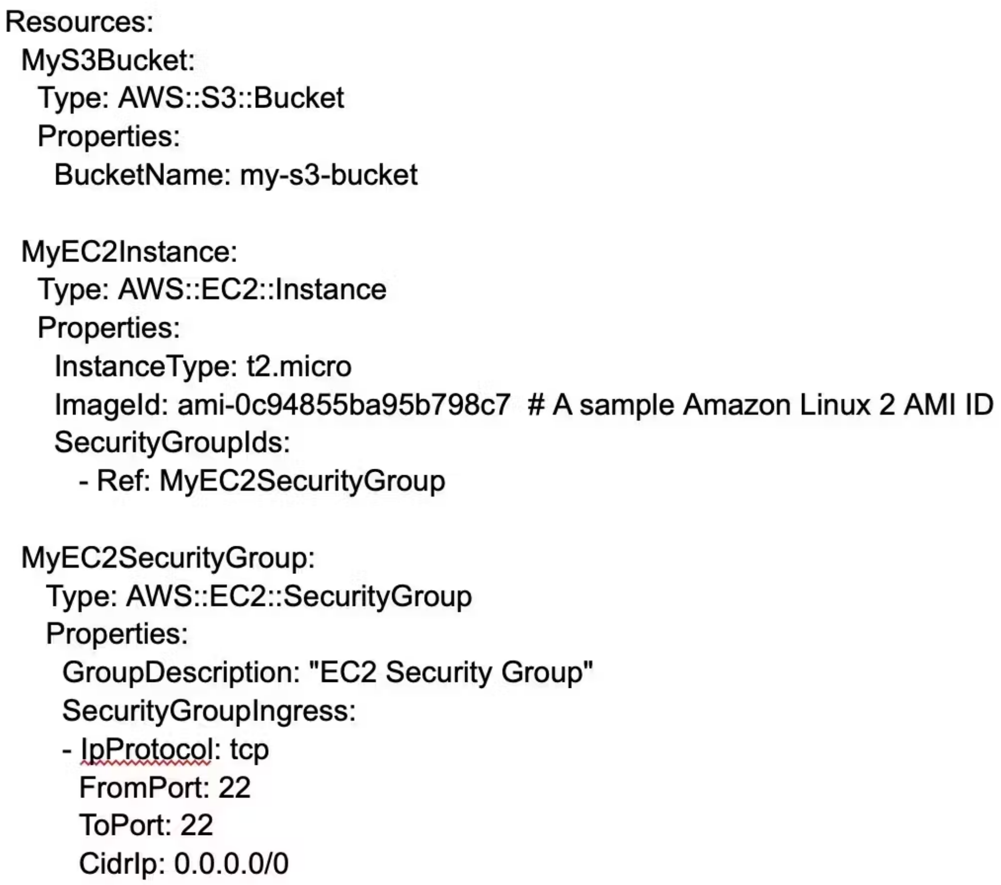
Hide Answer



```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd
  namespace: kube-system
  labels:
    k8s-app: fluentd
spec:
  selector:
    matchLabels:
      name: fluentd
  template:
    metadata:
      labels:
        name: fluentd
    spec:
      tolerations:
      - key: node-role.kubernetes.io/master
        effect: NoSchedule
      containers:
      - name: fluentd
        image: fluentd:v1.0.0
        resources:
          limits:
            memory: 200Mi
        volumeMounts:
        - name: var-log
          mountPath: /var/log
        - name: var-lib-docker
          mountPath: /var/lib/docker
      terminationGracePeriodSeconds: 30
      volumes:
      - name: var-log
        hostPath:
          path: /var/log
      - name: var-lib-docker
        hostPath:
          path: /var/lib/docker/containers
```

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

Hide Answer



```
Resources:
  MyS3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: my-s3-bucket

  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t2.micro
      ImageId: ami-0c94855ba95b798c7  # A sample Amazon Linux 2 AMI ID
      SecurityGroupIds:
        - Ref: MyEC2SecurityGroup

  MyEC2SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: "EC2 Security Group"
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
```

WRAPPING UP

The interview process for a DevOps engineer is not limited to DevOps technical interview questions. As a candidate looking for your dream DevOps job, you must also be ready to answer DevOps interview questions and questions related to your soft skills, such as communication, problem-solving, project management, crisis management, team management, etc.

As a recruiter, you are responsible for finding a DevOps engineer who complements your company's culture. Hence, in addition to the technical DevOps interview questions, you must ask questions to candidates about their team and social skills as well. If you want a DevOps engineer job with the best Silicon Valley companies, write the Turing test today to apply for these jobs. If you want to hire the best DevOps engineers, leave a message on Turing.com, and someone will contact you.