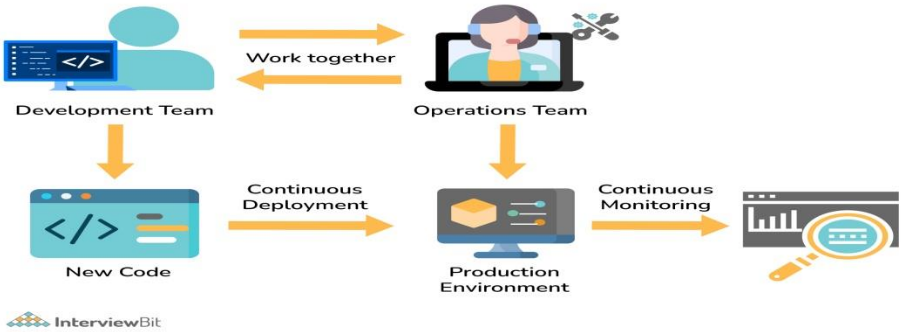
DevOps Interview Questions

Q1. What DevOps?

DevOps stands for Development and Operations. It is a software engineering practice that focuses on bringing together the development team and the operations team for the purpose of automating the project at every stage. This approach helps in easily automating the project service management in order to aid the objectives at the operational level and improve the understanding of the technological stack used in the production environment.

This way of practice is related to agile methodology and it mainly focuses on team communication, resource management, and teamwork. The main benefits of following this structure are the speed of development and resolving the issues at the production environment level, the stability of applications, and the innovation involved behind it.

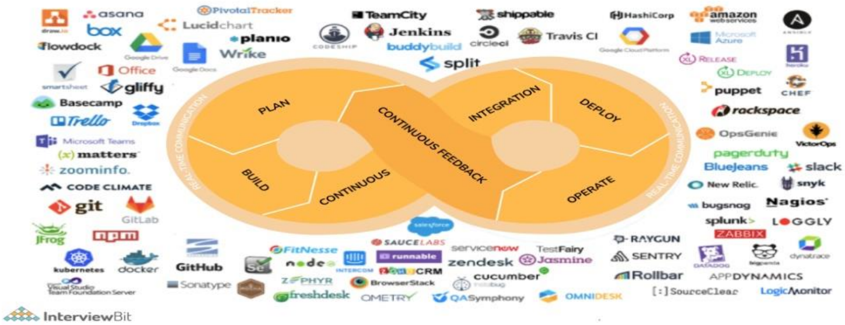


DevOps

Q2. Which are the top DevOps tools? Which tools have you worked on?

The most popular DevOps tools are mentioned below:

- Git : Version Control System tool
- Jenkins : Continuous Integration tool
- Selenium : Continuous Testing tool
- Puppet, Chef, Ansible : Configuration Management and Deployment tools
- Nagios : Continuous Monitoring tool
- Docker : Containerization tool



DevOps Tools

Q3. What is the need for DevOps?

According to me, this answer should start by explaining the general market trend. Instead of releasing big sets of features, companies are trying to see if small features can be transported to their customers through a series of release trains. This has many advantages like quick feedback from customers, better quality of software etc. which in turn leads to high customer satisfaction. To achieve this, companies are required to:

1. Increase deployment frequency
2. Lower failure rate of new releases
3. Shortened lead time between fixes
4. Faster mean time to recovery in the event of new release crashing

DevOps fulfills all these requirements and helps in achieving seamless software delivery. You can give examples of companies like Etsy, Google and Amazon which have adopted DevOps to achieve levels of performance that were unthinkable even five years ago. They are doing tens, hundreds or even thousands of code deployments per day while delivering world-class stability, reliability and security.

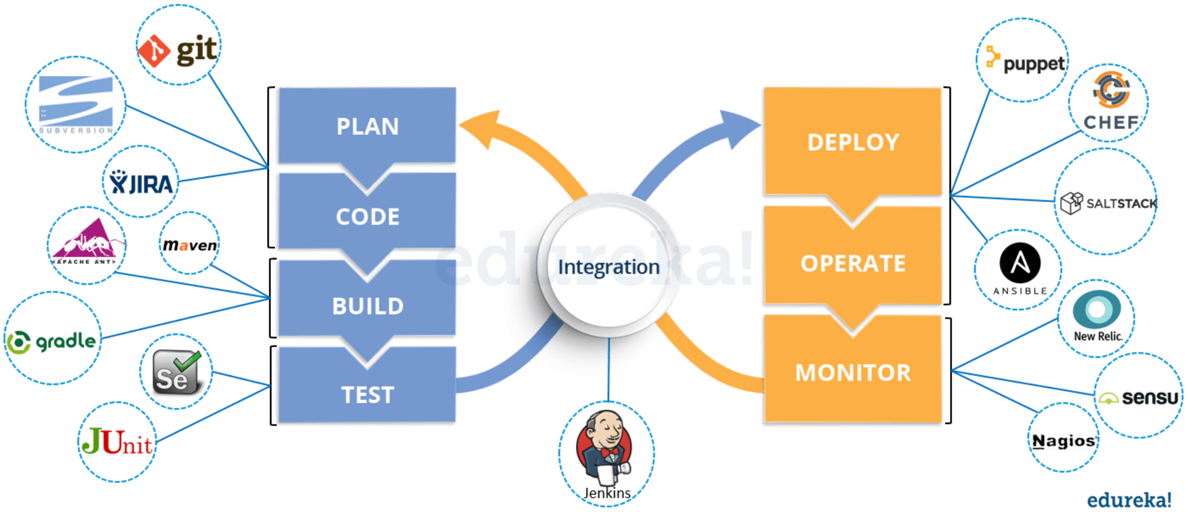
Q4. How is DevOps different from Agile / SDLC?

Agile is a set of values and principles about how to produce i.e. develop software. Example: if you have some ideas and you want to turn those ideas into working software, you can use the Agile values and principles as a way to do that. But, that software might only be working on a developer's laptop or in a test environment. You want a way to quickly, easily and repeatably move that software into production infrastructure, in a safe and simple way. To do that you need DevOps tools and techniques.

Q5. How do all these tools work together?

Given below is a generic logical flow where everything gets automated for seamless delivery. However, this flow may vary from organization to organization as per the requirement.

1. Developers develop the code and this source code is managed by Version Control System tools like Git etc.
2. Developers send this code to the Git repository and any changes made in the code is committed to this Repository.
3. Jenkins pulls this code from the repository using the Git plugin and build it using tools like Ant or Maven.
4. Configuration management tools like puppet deploys & provisions testing environment and then Jenkins releases this code on the test environment on which testing is done using tools like selenium.
5. Once the code is tested, Jenkins send it for deployment on the production server (even production server is provisioned & maintained by tools like puppet).
6. After deployment It is continuously monitored by tools like Nagios.
7. Docker containers provides testing environment to test the build features.



Q6. What are the advantages of DevOps?

Technical benefits:

- Continuous software delivery
- Less complex problems to fix
- Faster resolution of problems

Business benefits:

- Faster delivery of features
- More stable operating environments
- More time available to add value (rather than fix/maintain)

Q7. Mention some of the core benefits of DevOps?

- Faster development of software and quick deliveries.
- DevOps methodology is flexible and adaptable to changes easily.
- Compared to the previous software development models, confusion about the project is decreased due to increased product quality.
- The gap between the development team and operation team is bridged. i.e, the communication between the teams has been increased.
- Efficiency is increased by the addition of automation of continuous integration and continuous deployment.
- Customer satisfaction is enhanced.

Q8. What is the most important thing DevOps helps us achieve?

According to me, the most important thing that DevOps helps us achieve is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. This is the primary objective of DevOps. However, you can add many other positive effects of DevOps. For example, clearer communication and better working relationships between teams i.e. both the Ops team and Dev team collaborate together to deliver good quality software which in turn leads to higher customer satisfaction.

Q9. Explain with a use case where DevOps can be used in industry/ real-life.

There are many industries that are using DevOps so you can mention any of those use cases, you can also refer the below example:

Etsy is a peer-to-peer e-commerce website focused on handmade or vintage items and supplies, as well as unique factory-manufactured items. Etsy struggled with slow, painful site updates that frequently caused the site to go down. It affected sales for millions of Etsy's users who sold goods through online market place and risked driving them to the competitor.

With the help of a new technical management team, Etsy transitioned from its waterfall model, which produced four-hour full-site deployments twice weekly, to a more agile approach. Today, it has a fully automated deployment pipeline, and its continuous delivery practices have reportedly resulted in more than 50 deployments a day with fewer disruptions.

Q10. What are the core operations of DevOps in terms of development and Infrastructure?

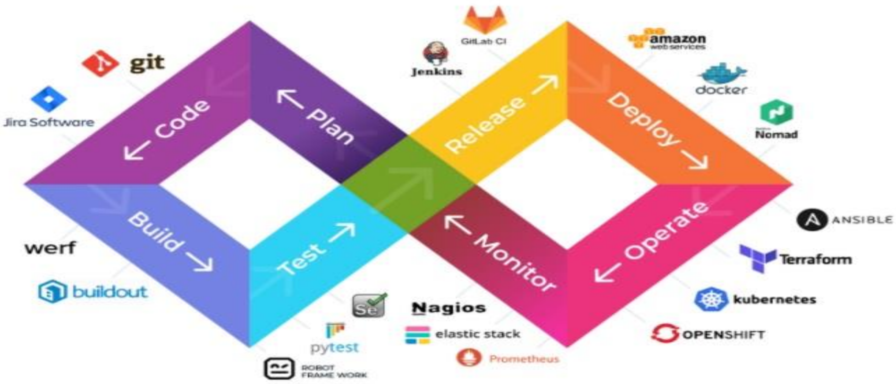
Ans. The core operations of DevOps are application development, code developing, code coverage, unit testing, packaging, deployment with infrastructure, provisioning, configuration, orchestration, and deployment.

Q11. What are the anti-patterns of DevOps?

A pattern is common usage usually followed. If a pattern commonly adopted by others does not work for your organization and you continue to blindly follow it, you are essentially adopting an anti-pattern. There are myths about DevOps. Some of them include:

- DevOps is a process
- Agile equals DevOps?
- We need a separate DevOps group
- Devops will solve all our problems
- DevOps means Developers Managing Production
- DevOps is Development-driven release management
    1. DevOps is not development driven.
    2. DevOps is not IT Operations driven.
- We can't do DevOps – We're Unique
- We can't do DevOps – We've got the wrong people

Q12. Explain the different phases in DevOps methodology?



The various phases of the DevOps lifecycle are as follows:

- **Plan –** In this stage, all the requirements of the project and everything regarding the project like time for each stage, cost, etc are discussed. This will help everyone in the team to get a brief idea about the project.
- **Code –** The code is written over here according to the client's requirements. Here codes are written in the form of small codes called units.
- **Build –** Building of the units is done in this step.
- **Test –** Testing is done in this stage and if there are mistakes found it is returned for re-build.
- **Integrate –** All the units of the codes are integrated into this step.
- **Deploy –** codeDevOpsNow is deployed in this step on the client's environment.
- **Operate –** Operations are performed on the code if required.
- **Monitor –** Monitoring of the application is done over here in the client's environment.

Q13. Explain your understanding and expertise on both the software development side and the technical operations side of an organization you have worked with in the past?

- **Deployment frequency:** This measures how frequently a new feature is deployed.
- **Change failure rate:** This is used to measure the number of failures in deployment.
- **Mean Time to Recovery (MTTR):** The time is taken to recover from a failed deployment.

Q14. What are the KPIs that are used for gauging the success of a DevOps team?

KPI Means Key Performance Indicators are used to measure the performance of a DevOps team, identify mistakes and rectify them. This helps the DevOps team to increase productivity and which directly impacts revenue.

There are many KPIs which one can track in a DevOps team. Following are some of them:

- **Change Failure rates:** This is used to measure the number of failures in deployments.
- **Meantime to recovery (MTTR):** The time is taken to recover from a failed deployment.
- **Lead time:** This helps to measure the time taken to deploy on the production environment.
- **Deployment frequency:** This measures how frequently a new feature is deployed.
- **Change volume:** This is used to measure how much code is changed from the existing code.
- **Cycle time:** This is used to measure total application development time.
- **Customer Ticket:** This helps us to measure the number of errors detected by the end-user.
- **Availability:** This is used to determine the downtime of the application.
- **Defect escape rate:** This helps us to measure the number of issues that are needed to be detected as early as possible.
- **Time of detection:** This helps you understand whether your response time and application monitoring processes are functioning correctly.

Q15. Why has DevOps become famous?

As we know before DevOps there are two other software development models:

- Waterfall model
- Agile model

In the waterfall model, we have limitations of one-way working and lack of communication with customers. This was overcome in Agile by including the communication between the customer and the company by taking feedback. But in this model, another issue is faced regarding communication between the Development team and operations team due to which there is a delay in the speed of production. This is where DevOps is introduced. It bridges the gap between the development team and the operation team by including the automation feature. Due to this, the speed of production is increased. By including automation, testing is integrated into the development stage. Which resulted in finding the bugs at the very initial stage which increased the speed and efficiency.

Q16. How does AWS contribute to DevOps?

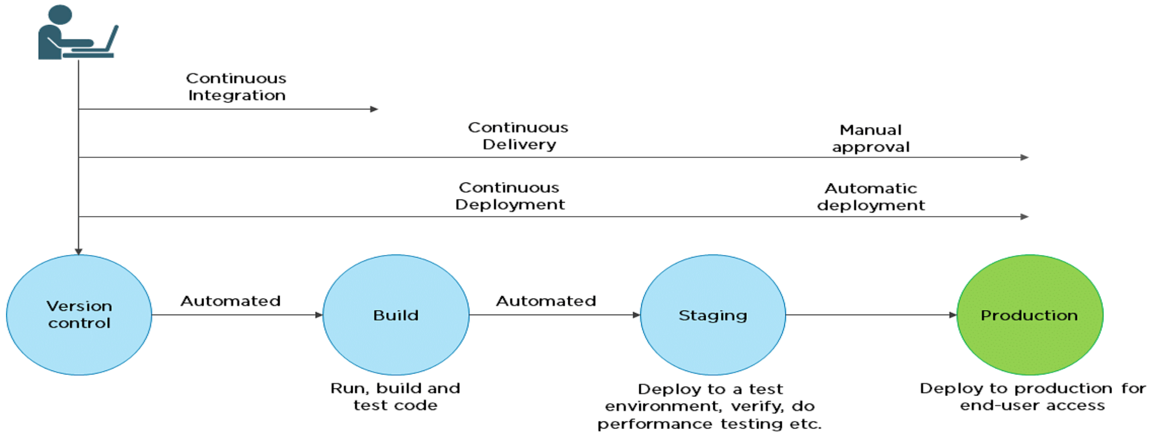
AWS [Amazon Web Services] is one of the famous cloud providers. In AWS DevOps is provided with some benefits:

- **Flexible Resources:** AWS provides all the DevOps resources which are flexible to use.
- **Scaling:** we can create several instances on AWS with a lot of storage and computation power.
- **Automation:** Automation is provided by AWS like CI/CD
- **Security:** AWS provides security when we create an instance like IAM

Q17. 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	There is no explicit approval from a developer and requires a

through rigorous automated testing	developed culture of monitoring



Q18. 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.

Q19. What is Multi-factor authentication?

In security implementation, we use Multi-factor authentication (MFA). In MFA, a user is authenticated by multiple means

before giving access to are source or service. It is different from simple user/password based authentication.

The most popular implementation of MFA is Two-factor authentication. In most of the organizations, we use

username/ password and an RSA token as two factors for authentication.

With MFA, the system be comes more secure and it cannot be easily hacked.

Q20. What are the main benefits of Nagios?

Nagios is open source software to monitor systems, networks and infrastructure. The main benefits ofNagiosareas follows:

I. Monitor : DevOps can configure Nagios to monitor IT infrastructure components, system metrics and

network protocols.

II. Alert :Nagios will send alerts when acritical component in infrastructure fails.

III. Response : DevOps acknowledges alerts and takes corrective actions.

IV. Report : Periodically Nagios can publish/send reports on outages, events and SLA setc.

V. Maintenance: During maintenance windows, we can also disable alerts.

VI. Planning : Based on past data, Nagios helps in infrastructure planning and upgrades.

Q 21. What is State Stalking in Nagios?

State Stalking is a very useful feature. Though all the users do not use it all the time, it is very helpful when we want to

Investigate an issue.

In State Stalking, we can enable stalking on a host. Nagios will monitor the state of the host very carefully and it will log any

changes in the state.

By this we can identify what changes might be causing an issue on the host

Q 22. What are the main features of Nagios?

Some of the main features of Nagios are as follows:

I. Visibility : Nagios provides a centralized view of the entire IT infrastructure.

II. Monitoring : We can monitor all the mission critical infrastructure components with Nagios.

III. Proactive Planning : With Capacity Planning and Trending we can proactively plan to scale up or scale down

The infrastructure.

IV. Extendable : Nagios is extendable to a third party tools in APIs.

V. Multi-tenant : Nagios supports multi-tenants architecture.

Q 23. What is Puppet?

Puppet Enterprise is a DevOps software platform that is used for automation of infrastructure operations. It runs on Unix as well as on Windows.

We can define system configuration by using Puppet’s language or Ruby DSL.

The system configuration described in Puppet’s language can be distributed to a target system by using RESTAPI calls.

Q 24. What is the architecture of Puppet?

Puppet is Open Source software. It is based on Client-server architecture. It is a Model Driven system. The client is also

called Agent. And server is called Master.

It has following architectural components:

I. Configuration Language : Puppet provides a language that is used to configure Resources. We have to Specify what Action has to be applied to which Resource.

The Action has three items for each Resource: type, title and list of attributes of aresource. Puppet code is written in Manifests files.

II. Resource Abstraction : We can create Resource Abstraction in Puppet so that we can configure resources on different platforms. Puppet agent uses a Facter for passing the information of an environment to Puppet server. In Facter   we have information about IP, hostname, OS etc of the environment.

III. Transaction : In Puppet, Agent sends Facter to Master server. Master sends back the catalog to Client. Agent applies any configuration changes to system. Once all changes are applied, there sult is sent to Server.

Q 25. What are the main use cases of Puppet Enterprise?

We can use Puppet Enterprise for following scenarios:

I. Node Management : We can manage a large number of nodes with Puppet.

II. Code Management : With Puppet we can define Infrastructure as code. We can review, deploy, and test the Environment configuration for Development, Testing and Production environments.

III. Reporting & Visualization : Puppet provides Graphical tools to visualize and see the exact status of Infrastructure configuration.

IV. Provisioning Automation : With Puppet we can automate deployment and creation of new servers and resources. So users and   business can get their infrastructure requirements completed very fast with Puppet.

V. Orchestration : For a large Cluster of nodes, we can orchestrate the complete process by using Puppet. It can follow the order in which we want to deploy the infrastructure environments.

VI. Automation of Configuration : With Configuration automation, the chances of manual errors are reduced. The process becomes more reliable with this.

Q 26. What is the use of Kubernetes?

We use Kubernetes for automation of large-scale deployment of Containerized applications.

It is an open source system based on concepts similar to Google’s deployment process of millions of containers.

It can be used on cloud, on-premise data center and hybrid infrastructure.

In Kubernetes we can create a cluster of servers that are connected to work as a single unit. We can deploy a containerized application to all the servers in a cluster without specifying the machine name.

We have to package applications in such a way that they do not depend on a specific host.

Q 27. What is the architecture of Kubernetes?

The architecture of Kubernetes consists of following components:

Master: The reisa master node that is responsible for managing the cluster. Master performs following functions in a cluster.

I. Scheduling Applications

II. Maintaining desired state of applications

III. Scaling applications

IV. Applying updates to applications

Nodes : A Node in Kubernetes is responsible for running an application. The Node can be a Virtual Machine or a Computer

in the cluster. There is software called Kubelet on each node. This software is used for managing the node and communicating

with the Master node in cluster.

There is a Kubernetes API that is used by Nodes to communicate with the Master. When we deploy an application on

Kubernetes, we request Master to start application containers on Nodes.

Q 28. How does Kubernetes provide high availability of applications in a Cluster?

In a Kubernetes cluster, there is a Deployment Controller. This controller monitors the instances created by Kubernetes in a

cluster. Once a node or the machine hosting the node goes down, Deployment Controller willreplacethe node.

It is a self-healing mechanism in Kubernetes to provide high availability ofapplications.

Therefore in Kubernetes cluster, Kubernetes Deployment Controller is responsible for starting the instances as well as

replacing theinstances in case ofafailure.

Q 29. Why Automated Testing is a must requirement for DevOps?

In DevOps approach we release software with high frequency to production. We have to run tests to gain confidence on the quality of software deliverables.

Running tests manually is a time taking process. Therefore, we first prepare automation tests and then deliver software. This

ensures that we catch any defects early in our process.

Q 30. What is Chaos Monkey in DevOps?

Chaos Monkey is a concept made popular by Netflix. In Chaos Monkey, we intentionally try to shut down the services or

Create failures. By failing one or more services, we test there liability and recovery mechanism of the Production architecture.

It checks whether our applications and deployment have survival strategy built into it or not.

Q 31. How do you perform Test Automation in DevOps?

We use Jenkins to create automated flows to run Automation tests. The first part of test automation is to develop test strategy

and test cases. Once automation test cases are ready for an application, we have to plug these into each Build run.

In each Build we run Unit tests, Integration tests and Functional tests.

With a Jenkins job, we can automate all these tasks. Once all the automated tests pass, we consider the build as green. This

helps in deployment and release processes to build confidence on theapplication software.

Q 32. What are the main services of AWS that you have used?

We use following main services of AWS in our environment:

I. EC2 : This is the Elastic Compute Cloud by Amazon. It is used to for providing computing capability to a

system. We can use it in places of our standalone servers. We can deploy different kinds ofapplications on

EC2.

II. S3 : We use S3 in Amazon for our storage needs.

III. Dynamo DB : We use Dynamo DB in AWS for storing data in NoSQL database form.

IV. Amazon Cloud Watch : We use Cloud Watch to monitor our application in Cloud.

V. Amazon SNS : We use Simple Notification Service to inform users about any issues in Production

environment.

33. Why GIT is considered better than CVS for version control system?

GIT is a distributed system. In GIT, any person can create its own branch and start checking in the code. Once the code is

tested, it is merged into main GIT repo. IN between, Dev, QA and product can validate the implementation ofthatcode.

In CVS, there is a centralized system that maintains all the commits and changes.

GIT is open source software and there are plenty of extensions in GIT for use by our teams.

Q34. What is the difference between a Container and a Virtual Machine?

We need to select an Operating System(OS) to get a specific Virtual Machine (VM). VM provides full OS to an application

for running in a virtualized environment.

A Container uses APIs of an Operating System(OS) to provide run time environment to an application.

A Container is very lightweight in comparison with a VM.

VM provides higher level of security compared to a Container.

A Container just provides the APIs that are required by the application.

Q35. What is Serverless architecture?

Serverless Architecture is a term that refers to following:

I. AnApplication that depends on athird-party service.

II. An Application in which Codeis run on ephemeral containers.

In AWS, Lambda is a popular service to implement Serverless architecture.

Another concept in Serverless Architecture is to treat code as a service or Function as a Service(FAAS). We just write code

thatcan berun on any environment or server without the need ofspecifying which server should be used to run thiscode.

Q 36. What are the main principles of DevOps?

DevOps is different from Technical Operations. It has following main principles:

I. Incremental : In DevOps we aim to incrementally release software to production. We do releases to

Production more often than Waterfall approach of one large release.

II. Automated :To enable use to make releases more often, we automate the operations from Code Check in to

Deployment in Production.

III. Collaborative :DevOps is not only responsibility of Operations team. It is a collaborative effort of Dev, QA,

Release and DevOps teams.

IV. Iterative : DevOps is based on Iterative principle of using a process that is repeatable. But with each iteration

We aim to make the process more efficient and better.

V. Self-Service : In DevOps, we automate things and give self-service options to other teams so that they are

Empowered to deliver the work in their domain.

Q 37. Are you more Dev or more Ops?

This is a tricky question. DevOps is a new concept and in any organization the maturity of DevOps varies from highly

Operations oriented to highly DevOps oriented. In some projects teams are very mature and practice DevOps in it true form.

In some projects, teams rely more on Operations team.

As a DevOps person I give first priority to the needs of an organization and project. At sometimes I may have to perform a lot

of operations work. But with each iteration, Iaimto bring DevOps changes incrementally to an organization.

Over time, organization/project starts seeing results ofDevOps practices and embraces it fully.

Q 38. What is a REST service?

REST is also known as Representational State Transfer. A REST service is a simple software functionality that is available

over HTTP protocol. It is a light weight service that is widely available due to the popularity of HTTP protocol.

Sine REST is lightweight; it has very good performance in a software system. It is also one of the foundations for creating

highly scalable systems that provide a service to large number of clients.

Another key feature of a REST service is that as long as the interface is kept same, we can change the underlying

implementation. E.g. Clients of REST service can keep calling the same service while we change the implementation from php

to Java.

Q39. What are the Three Ways of DevOps?

Three Ways of DevOps refers to three basic principles of DevOps culture. These areas follows:

I. The First Way: Systems Thinking : In this principle we see the DevOps as a flow of work from left to right.

This is the time taken from Code check in to the feature being released to End customer. In DevOps culture

We try to identify the bottlenecks in this.

II. The Second Way: Feedback Loops : Whenever there is an issue in production it is a feedback about the

whole development and deployment process. We try to make the feedback loop more efficient so that teams

can get the feedback much faster. It is a way of catching defect much earlier in process than it being reported

by customer.

III. The Third Way: Continuous Learning : We make use of first and second way principles to keep on making

improvements in the overall process. This is the third principle in which over the time we make the process and

our operations highly efficient ,automated and error free by continuously improving them.

Q40. How do you apply DevOps principles to make system Secure?

Security of a system is one of the most important goals for an organization. We use following ways to apply DevOps to

security.

I. Automated Security Testing : We automate and integrate Security testing techniques for Software

Penetration testing and Fuzz testing in software development process.

II. Early Security Checks : We ensure that teams know about the security concerns at the beginning of a

project, rather than at the end of delivery. It is achieved by conducting Security trainings and knowledge

sharing sessions.

III. Standard Process : At DevOps we try to follow standard deployment and development process that has

Already gone through security audits. This helps in minimizing the introduction of any new security loopholes

Due to change in the standard process.

Q41. How does Ansible work?

Ansible has two types of servers categorized as:

- Controlling machines
- Nodes

For this to work, Ansible is installed on controlling machine using which the nodes are managed by means of using SSH. The location of the nodes would be specified and configured in the inventories of the controlling machine.

Ansible does not require any installations on the remote node servers due its nature of being agentless. Hence, no background process needs to be executed while managing any remote nodes.

Ansible can manage lots of nodes from a single controlling system my making use of Ansible Playbooks through SSH connection. Playbooks are of the YAML format and are capable to perform multiple tasks.

Q42. What can be a preparatory approach for developing a project using the DevOps methodology?

The project can be developed by following the below stages by making use of DevOps:

- **Stage 1: Plan:** Plan and come up with a roadmap for implementation by performing a thorough assessment of the already existing processes to identify the areas of improvement and the blindspots.
- **Stage 2: PoC:** Come up with a proof of concept (PoC) just to get an idea regarding the complexities involved. Once the PoC is approved, the actual implementation work of the project would start.
- **Stage 3: Follow DevOps:** Once the project is ready for implementation, actual DevOps culture could be followed by making use of its phases like version control, continuous integration, continuous testing, continuous deployment, continuous delivery, and continuous monitoring.

Q 43. Can you explain the "Shift left to reduce failure" concept in DevOps?

In order to understand what this means, we first need to know how the traditional SDLC cycle works. In the traditional cycle, there are 2 main sides -

- The left side of the cycle consists of the planning, design, and development phase
- The right side of the cycle includes stress testing, production staging, and user acceptance.

In DevOps, shifting left simply means taking up as many tasks that usually take place at the end of the application development process as possible into the earlier stages of application development. From the below graph, we can see that if the shift left operations are followed, the chances of errors faced during the later stages of application development would greatly reduce as it would have been identified and solved in the earlier stages itself.

The most popular ways of accomplishing shift left in DevOps is to:

- Work side by side with the development team while creating the deployment and test case automation. This is the first and the obvious step in achieving shift left. This is done because of the well-known fact that the failures that get notices in the production environment are not seen earlier quite often. These failures can be linked directly to:
  - Different deployment procedures used by the development team while developing their features.
  - Production deployment procedures sometimes tend to be way different than the development procedure. There can be differences in tooling and sometimes the process might also be manual.
- Both the dev team and the operations teams are expected to take ownership to develop and maintain standard procedures for deployment by making use of the cloud and the pattern capabilities. This aids in giving the confidence that the production deployments would be successful.
- Usage of pattern capabilities to avoid configurational level inconsistencies in the different environments being used. This would require the dev team and the operation team to come together and work in developing a standard process that guides developers to test their application in the development environment in the same way as they test in the production environment.

Q 44. Do you know about post mortem meetings in DevOps?

Post Mortem meetings are those that are arranged to discuss if certain things go wrong while implementing the DevOps methodology. When this meeting is conducted, it is expected that the team has to arrive at steps that need to be taken in order to avoid the failure(s) in the future.

Q45. What is the concept behind sudo in Linux OS?

Sudo stands for 'super user do' where the super user is the root user of Linux. It is a program for Linux/Unix-based systems that gives provision to allow the users with super user roles to use certain system commands at their root level.

Q46. Can you explain the architecture of Jenkins?

Jenkins follows the master-slave architecture. The master pulls the latest code from the GitHub repository whenever there is a commitment made to the code. The master requests slaves to perform operations like build, test and run and produce test case reports. This workload is distributed to all the slaves in a uniform manner.

Jenkins also uses multiple slaves because there might be chances that require different test case suites to be run for different environments once the code commits are done.

Q47. Can you explain the "infrastructure as code" (IaC) concept?

As the name indicates, IaC mainly relies on perceiving infrastructure in the same way as any code which is why it is commonly referred to as "programmable infrastructure". It simply provides means to define and manage the IT infrastructure by using configuration files.

This concept came into prominence because of the limitations associated with the traditional way of managing the infrastructure. Traditionally, the infrastructure was managed manually and the dedicated people had to set up the servers physically. Only after this step was done, the application would have been deployed. Manual configuration and setup were constantly prone to human errors and inconsistencies.

This also involved increased cost in hiring and managing multiple people ranging from network engineers to hardware technicians to manage the infrastructural tasks. The major problem with the traditional approach was decreased scalability and application availability which impacted the speed of request processing. Manual configurations were also time-consuming and in case the application had a sudden spike in user usage, the administrators would desperately work on keeping the system available for a large load. This would impact the application availability.

IaC solved all the above problems. IaC can be implemented in 2 approaches:

- Imperative approach: This approach "gives orders" and defines a sequence of instructions that can help the system in reaching the final output.
- Declarative approach: This approach "declares" the desired outcome first based on which the infrastructure is built to reach the final result.

Q48. What is 'Pair Programming'?

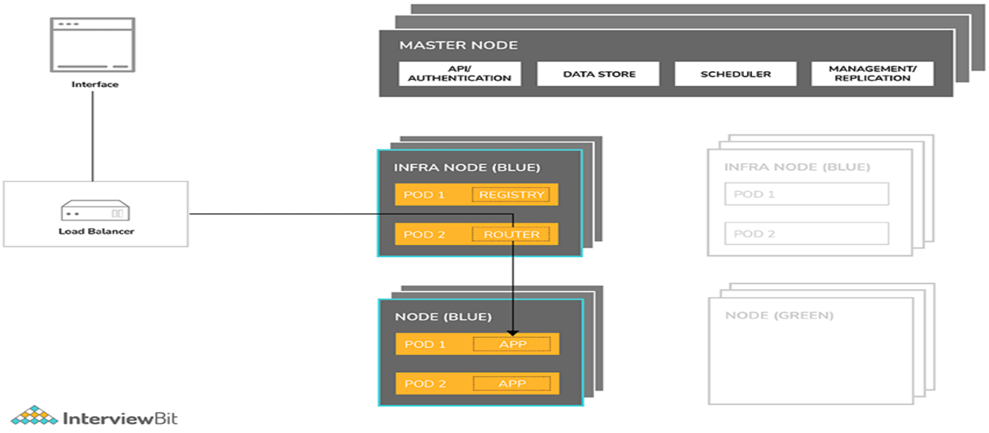
Pair programming is an engineering practice where two programmers work on the same system, same design, and same code. They follow the rules of "Extreme Programming". Here, one programmer is termed as "driver" while the other acts as "observer" which continuously monitors the project progress to identify any further problems.

Q49. What is Blue/Green Deployment Pattern?

A blue-green pattern is a type of continuous deployment, application release pattern which focuses on gradually transferring the user traffic from a previously working version of the software or service to an almost identical new release - both versions running on production.

The blue environment would indicate the old version of the application whereas the green environment would be the new version.

The production traffic would be moved gradually from blue to green environment and once it is fully transferred, the blue environment is kept on hold just in case of rollback necessity.



In this pattern, the team has to ensure two identical prod environments but only one of them would be LIVE at a given point of time. Since the blue environment is more steady, the LIVE one is usually the blue environment.

Q50. What is Dogpile effect? How can it be prevented?

It is also referred to as cache stampede which can occur when huge parallel computing systems employing caching strategies are subjected to very high load. It is referred to as that event that occurs when the cache expires (or invalidated) and multiple requests are hit to the website at the same time. The most common way of preventing dogpiling is by implementing semaphore locks in the cache. When the cache expires in this system, the first process to acquire the lock would generate the new value to the cache.

Q51. What are the steps to be undertaken to configure git repository so that it runs the code sanity checking tooks before any commits? How do you prevent it from happening again if the sanity testing fails?

Sanity testing, also known as smoke testing, is a process used to determine if it's reasonable to proceed to test. Git repository provides a hook called pre-commit which gets triggered right before a commit happens. A simple script by making use of this hook can be written to achieve the smoke test.

The script can be used to run other tools like linters and perform sanity checks on the changes that would be committed into the repository.

The following snippet is an example of one such script:

```
#!/bin/sh
files=$(git diff –cached –name-only –diff-filter=ACM | grep ‘.py$’)
if [ -z files ]; then
exit 0
fi
unfmtd=$(pyfmt -l $files)
if [ -z unfmtd ]; then
exit 0
fi
echo “Some .py files are not properly fmt’d”
exit 1
```

The above script checks if any .py files which are to be committed are properly formatted by making use of the python formatting tool pyfmt. If the files are not properly formatted, then the script prevents the changes to be committed to the repository by exiting with status 1.

Q52. How can you ensure a script runs every time repository gets new commits through git push?

There are three means of setting up a script on the destination repository to get executed depending on when the script has to be triggered exactly. These means are called hooks and they are of three types:

- **Pre-receive hook:** This hook is invoked before the references are updated when commits are being pushed. This hook is useful in ensuring the scripts related to enforcing development policies are run.
- **Update hook:** This hook triggers the script to run before any updates are actually made. This hook is called once for every commit which has been pushed to the repository.
- **Post-receive hook:** This hook helps trigger the script after the updates or changes have been accepted by the destination repository. This hook is ideal for configuring deployment scripts, any continuous integration-based scripts or email notifications process to the team, etc.

Q53. What are the benefits of using version control?

- **It helps improve the collaborative work culture**: Here, team members are allowed to work freely on any file at any time. The version control system allows us to merge all changes into a common version.
- **It keeps different versions of code files securely:** All the previous versions and variants of code files are neatly packed up inside the version control system.
- **It understands what happened**: Every time we save a new version of our project, the version control system asks us to provide a short description of what was changed. More than that it allows us to see what changes were made in the file’s content, as well as who has made those changes.
- **It keeps backup**: A distributed version control system like Git allows all team members to have the complete history of the project file so that in case there is a breakdown in the central server, they can use any of their teammate’s local Git repository.

Q54. What are the types of Http requests?

The types of Http requests are

- GET

- HEAD
- PUT
- POST
- PATCH
- DELETE
- TRACE
- CONNECT
- OPTIONS

Q55. Give some benefits of using Version Control system

- The version Control system allows team members to work freely on any file at any time.
- All the past versions and variants are closely packed up inside the VCS.
- A distributed VCS like helps you to store the complete history of the project so in case of a breakdown in the central server you can use your team member's local Git repository.
- Allows you to see what exact changes are made in the file's content

56. Explain Git Bisect

Git bisect helps you to find the commit which introduced a bug using binary search.

Q57. What is the build?

A build is a method in which the source code is put together to check whether it works as a single unit. In the build creation process, the source code will undergo compilation, inspection, testing, and deployment.

Q58. What is the Docker hub?

Docker Hub is a cloud-based repository of Docker. It allows users to create, test, store, and distribute container images. It helps to –

- Access public, open-source image repositories,
- Use space to create their own private repositories
- Build automated build functions, webhooks, and workgroups
- Store manually pushed images and links to Docker cloud

Q59. What is Docker Swarm?

Docker Swarm is a tool for Docker containers, responsible for clustering and scheduling. It helps users establish and manage a cluster of Docker nodes as a single virtual system.

Q60. Do you have ever used DevOps implementation in the cloud? Which are the most popular Cloud Computing Platforms?

Some popular Cloud Computing Platforms are –

- AWS (Amazon Web Services)

- Microsoft Azure
- Google Cloud