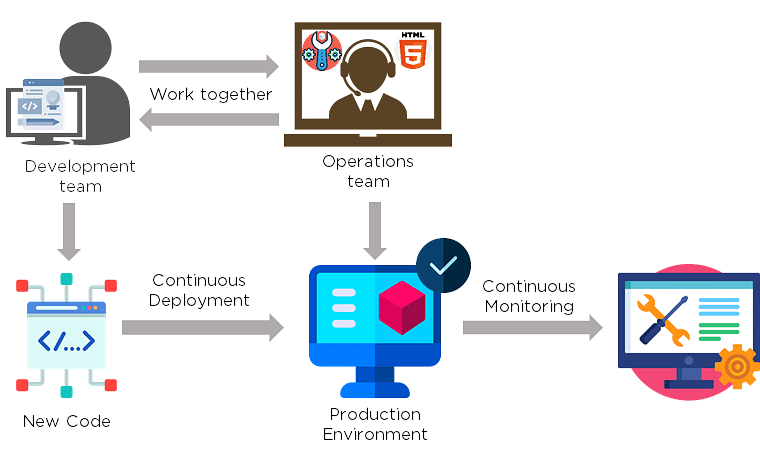
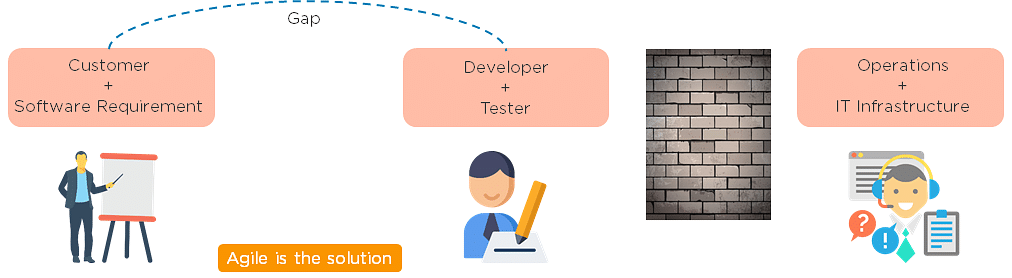
**DEVOPS INTERVIEW QUESTIONS**

How is DevOps different from agile methodology?

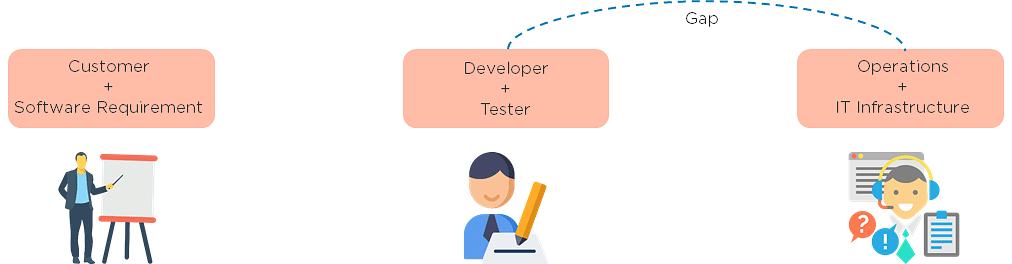
Devops ia a culture that allows the development and the operations team to work together. This results in ci, testing, integration, deployment, and monitoring of the software throughout the lifecycle.



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



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



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

The most popular Devops tools include:

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

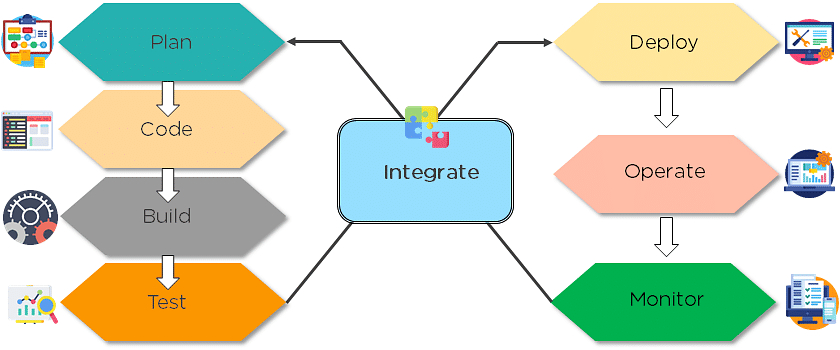
docker

Get Certified in DevOps with Caltech CTME

4. What are the different phases in DevOps?

The various phases of the DevOps lifecycle are as follows:

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



The above figure indicates the DevOps lifecycle.

5. Mention some of the core benefits of DevOps.

The core benefits of DevOps are as follows:

Technical benefits

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

Business benefits

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

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

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

Stage 1

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

Stage 2

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

Stage 3

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

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

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

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

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.

What is the role of AWS in DevOps?

AWS has the following role in DevOps:

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

11. Name three important DevOps KPIs.

The three important KPIs are as follows:

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

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

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

13. How is IaC implemented using AWS?

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

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

Before talking about the growing popularity of DevOps, discuss the current industry scenario. Begin with some examples of how big players such as facebook are investing in DevOps to automate and accelerate application deployment and how this has helped them grow their business. Using Facebook as an example, you would point to Facebook’s continuous deployment and code ownership models and how these have helped it scale up but ensure the quality of experience at the same time. Hundreds of lines of code are implemented without affecting quality, stability, and security.

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

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

What are the benefits of using version control?

Here are the benefits of using Version Control:

* All team members are free to work on any file at any time with the Version Control System (VCS). Later on, VCS will allow the team to integrate all of the modifications into a single version.
* The VCS asks to provide a brief summary of what was changed every time we save a new version of the project. We also get to examine exactly what was modified in the content of the file. As a result, we will be able to see who made what changes to the project.
* Inside the VCS, all the previous variants and versions are properly stored. We will be able to request any version at any moment, and we will be able to retrieve a snapshot of the entire project at our fingertips.
* A VCS that is distributed, such as Git, lets all the team members retrieve a complete history of the project. This allows developers or other stakeholders to use the local Git repositories of any of the teammates even if the main server goes down at any point in time.

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

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

Explain the concept of branching in Git.

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

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

