**DEVOPS (Development + Operation)**

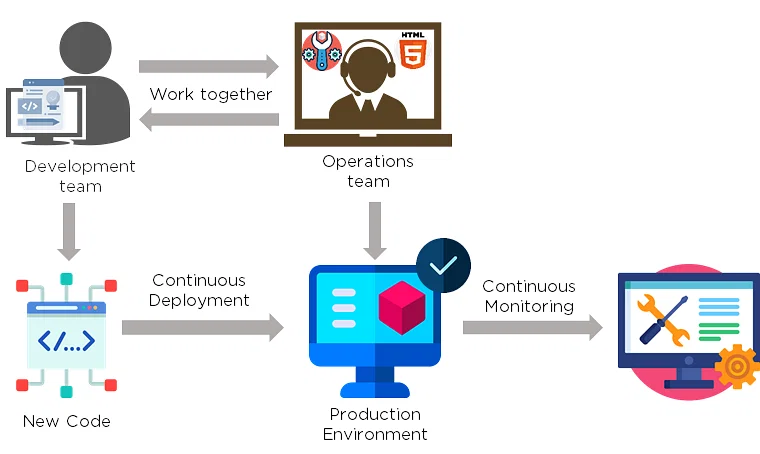
**What is DEVOPS engineer?** To improve the efficiency and quality of software development, delivery, and deployment, a group of activities and approaches called DevOps combines software development (Dev) with information technology operations (Ops).DevOps' primary objective is to foster teamwork between the development and operations teams so that they may collaborate easily across the whole software development life cycle. In addition, automation, continuous integration, delivery, and deployment are used to speed up and reduce mistakes in the software development process. Monitoring and feedback are also emphasized in DevOps, which enables the development and operations teams to see problems early and proactively handle them. Using DevOps methods, businesses may improve their agility, competitiveness, and overall productivity by achieving quicker release cycles, higher-quality software, and enhanced team cooperation.

1. What do you know about DevOps?

Your answer must be simple and straightforward. Begin by explaining the growing importance of DevOps in the [IT industry.](https://www.simplilearn.com/6-trends-shaking-up-the-it-industry-article) Discuss how such an approach aims to synergize the efforts of the development and operations teams to accelerate the delivery of software products, with a minimal failure rate. Include how DevOps is a value-added practice, where development and operations engineers join hands throughout the product or service lifecycle, right from the design stage to the point of deployment.

### 2. How is DevOps different from agile methodology?

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

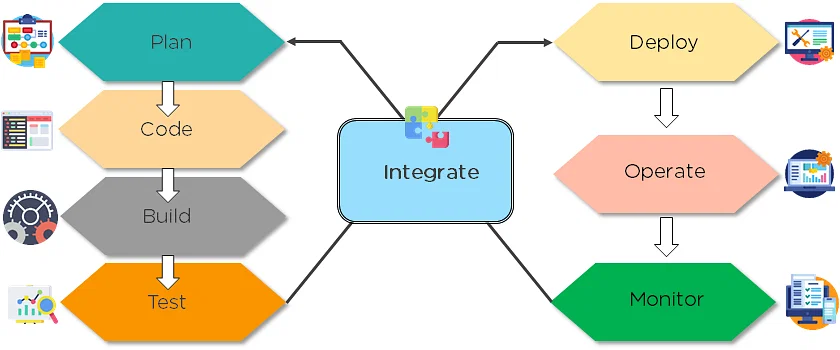
The most popular [DevOps tools](https://www.simplilearn.com/tutorials/devops-tutorial/devops-tools) include:

1. [Selenium](https://www.simplilearn.com/tutorials/selenium-tutorial/what-is-selenium)
2. [Puppet](https://www.simplilearn.com/puppet-tutorial-article)
3. [Chef](https://www.simplilearn.com/chef-tutorial-article)
4. [Git](https://www.simplilearn.com/tutorials/git-tutorial)
5. [Jenkins](https://www.simplilearn.com/tutorials/jenkins-tutorial/what-is-jenkins)
6. [Ansible](https://www.simplilearn.com/tutorials/ansible-tutorial)
7. [Docker](https://www.simplilearn.com/tutorials/docker-tutorial)

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

Also Read: [How to Become a DevOps Engineer](https://www.simplilearn.com/tutorials/devops-tutorial/how-to-become-devops-engineer)

### 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.

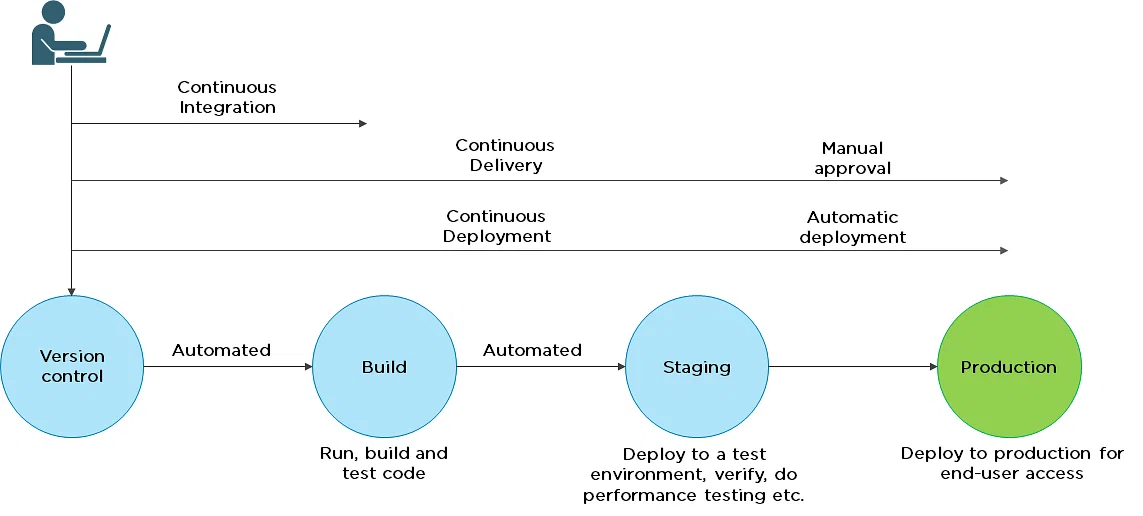
#### **Stage 3**

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](https://www.simplilearn.com/tutorials/devops-tutorial/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 |

****