



**ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY**

# **CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY USING DevOps**

**Exp-3**

**By**

**B Manikyala Rao M.Tech(Ph.d)**

**Assistant Professor**

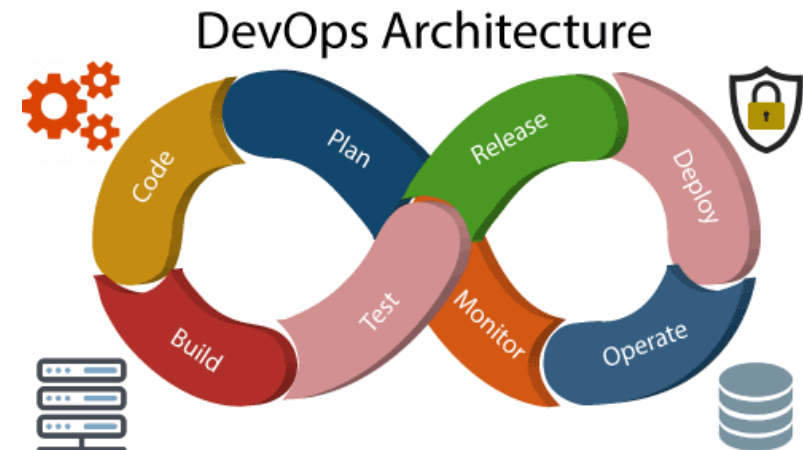
**Dept of Computer Science & Engineering**

**Aditya College of Engineering & Technology**

**Surampalem**

# DevOps

- The word DevOps is a combination of two words that is development and operations. This is a culture that promotes the collaboration between development team and the operations team. With the help of DevOps, The applications can be delivered faster and serve their customer needs nicely.
- DevOps is the effective collaboration between Development and IT operations. It is not possible to understand DevOps without knowing DevOps lifecycle. The important phase of DevOps life cycle are shown below that promotes continuous integration and continuous monitoring.





# Devops Life Cycle

- 1) Build: With DevOps, the usage of cloud, sharing of resources comes into the picture, and the build is dependent upon the user's need, which is a mechanism to control the usage of resources or capacity.
- 2) Code: Many good practices such as Git enables the code to be used, which ensures writing the code for business, helps to track changes, getting notified about the reason behind the difference in the actual and the expected output, and if necessary reverting to the original code developed. The code can be appropriately arranged in **files, folders**, etc. And they can be reused.
- 3) Test: In the case of manual testing, it consumes more time in testing and moving the code to the output. The testing can be automated, which decreases the time for testing so that the time to deploy the code to production can be reduced as automating the running of the scripts will remove many manual steps.
- 4) Plan: DevOps use Agile methodology to plan the development. With the operations and development team in sync, it helps in organizing the work to plan accordingly to increase productivity.

- 4) Monitor: Continuous monitoring is used to identify any risk of failure. Also, it helps in tracking the system accurately so that the health of the application can be checked. The monitoring becomes more comfortable with services where the log data may get monitored through many third-party tools such as **Splunk**.
- 6) Deploy: Many systems can support the scheduler for automated deployment. The cloud management platform enables users to capture accurate insights and view the optimization scenario, analytics on trends by the deployment of dashboards.
- 7) Operate: DevOps changes the way traditional approach of developing and testing separately. The teams operate in a collaborative way where both the teams actively participate throughout the service lifecycle. The operation team interacts with developers, and they come up with a monitoring plan which serves the IT and business requirements.
- 8) Release: Deployment to an environment can be done by automation. But when the deployment is made to the production environment, it is done by manual triggering. Many processes involved in release management commonly used to do the deployment in the production environment manually to lessen the impact on the customers.



# DevOps Lifecycle

- 1) Continuous Development: This phase involves the planning and coding of the software. The vision of the project is decided during the planning phase. And the developers begin developing the code for the application. There are no DevOps tools that are required for planning, but there are several tools for maintaining the code.
- 2) Continuous Integration: This stage is the heart of the entire DevOps lifecycle. It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or weekly basis. Then every commit is built, and this allows early detection of problems if they are present. Building code is not only involved compilation, but it also includes **unit testing, integration testing, code review, and packaging**.
- 3) Continuous Testing: This phase, where the developed software is continuously testing for bugs. For constant testing, automation testing tools such as **TestNG, JUnit, Selenium**, etc are used. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there is no flaw in the functionality. In this phase, **Docker** Containers can be used for simulating the test environment.

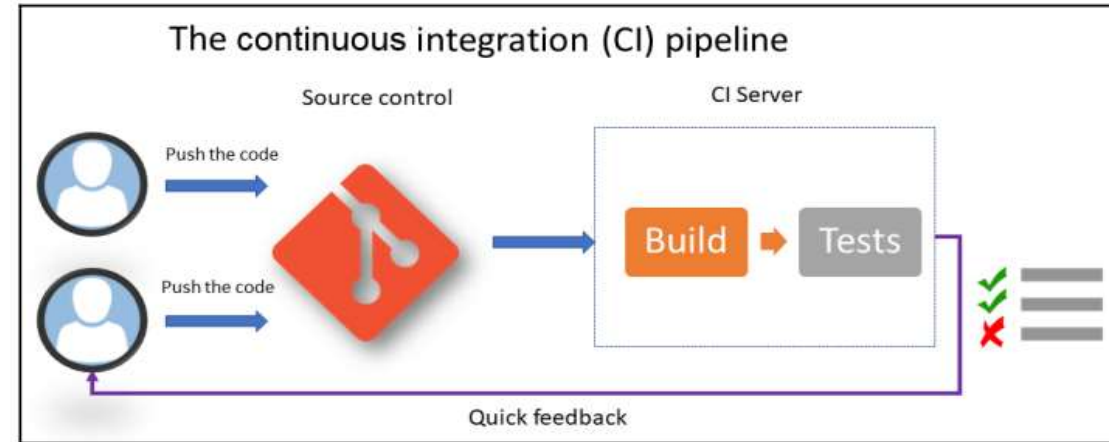
**Selenium** does the automation testing, and TestNG generates the reports. This entire testing phase can automate with the help of a Continuous Integration tool called **Jenkins**.

- 4) Continuous Monitoring: Monitoring is a phase that involves all the operational factors of the entire DevOps process, where important information about the use of the software is recorded and carefully processed to find out trends and identify problem areas. Usually, the monitoring is integrated within the operational capabilities of the software application.
- It may occur in the form of documentation files or maybe produce large-scale data about the application parameters when it is in a continuous use position. The system errors such as server not reachable, low memory, etc are resolved in this phase. It maintains the security and availability of the service.

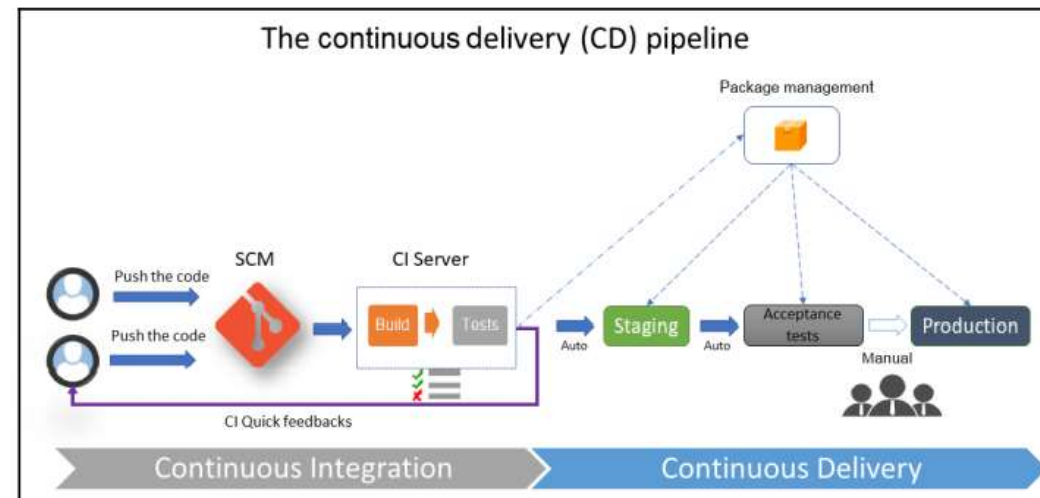
- 5) Continuous Feedback: The application development is consistently improved by analyzing the results from the operations of the software. This is carried out by placing the critical phase of constant feedback between the operations and the development of the next version of the current software application.
- 6) Continuous Deployment: In this phase, the code is deployed to the production servers. Also, it is essential to ensure that the code is correctly used on all the servers.
- 7) Continuous Operations: All DevOps operations are based on the continuity with complete automation of the release process and allow the organization to accelerate the overall time to market continuingly.
- It is clear from the discussion that continuity is the critical factor in the DevOps in removing steps that often distract the development, take it longer to detect issues and produce a better version of the product after several months. With DevOps, we can make any software product more efficient and increase the overall count of interested customers in your product.

# DevOps

Continuous integration (CI):



Continuous delivery (CD):





# DevOps Tools

Here are some most popular DevOps tools

## 1) Puppet

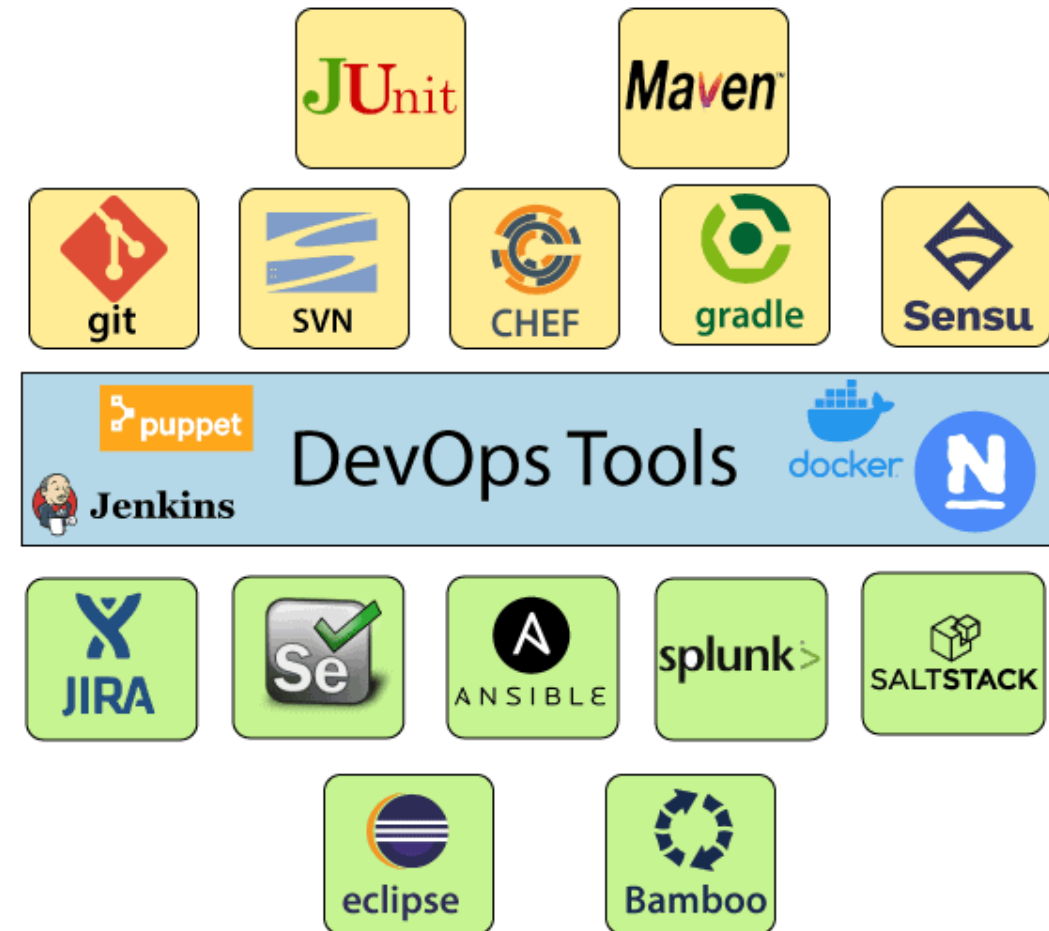
Puppet is the most widely used DevOps tool.

It allows the delivery and release of the technology changes quickly and frequently.

It has features of versioning, automated testing, and continuous delivery.

## 2) Ansible

Ansible is a leading DevOps tool. Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools. It makes it easier for DevOps teams to scale automation and speed up productivity.



# Devops Tools

## 3) Docker

Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

## 4) Nagios

Nagios is one of the more useful tools for DevOps. It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.

It provides complete monitoring of desktop and server operating systems.

In the event of a failure, Nagios can alert technical staff of the problem, allowing them to begin remediation processes before outages affect business processes, end-users, or customers.

## 5) CHEF

A chef is a useful tool for achieving scale, speed, and consistency. The chef is a cloud-based system and open source technology. This technology uses Ruby encoding to develop essential building blocks such as recipes and cookbooks. The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.

# Devops Tools

## 6) Jenkins

Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly.

### Features

- Jenkins increases the scale of automation.
- It can easily set up and configure via a web interface.
- It can distribute the tasks across multiple machines, thereby increasing concurrency.
- It supports continuous integration and continuous delivery.
- It offers 400 plugins to support the building and testing any project virtually.
- It requires little maintenance and has a built-in GUI tool for easy updates.



# Devops Tools

## 7) Git

- Git is an open-source distributed version control system that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency. It is developed to co-ordinate the work among programmers. The version control allows you to track and work together with your team members at the same workspace. It is used as a critical distributed version-control for the DevOps tool.

### Features:

- It is a free open source tool.
- It allows distributed development.
- It supports the pull request.
- It enables a faster release cycle.
- Git is very scalable.
- It is very secure and completes the tasks very fast.



# Devops Tools

## 8) Selenium

Selenium is a portable software testing framework for web applications. It provides an easy interface for developing automated tests.

### Features

- It is a free open source tool.
- It supports multiplatform for testing, such as Android and ios.
- It is easy to build a keyword-driven framework for a WebDriver.
- It creates robust browser-based regression automation suites and tests.



# Devops Tools

## 9) SonarQube :

It is an open-source platform developed by SonarSource for **continuous inspection of code quality**. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

### Features:

- **Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
- **Increase productivity** - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- **Quality code** - Code quality control is an inseparable part of the process of software development.
- **Detect Errors** - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- **Increase consistency** - Determines where the code criteria are breached and enhances the quality
- **Business scaling** - No restriction on the number of projects to be evaluated
- **Enhance developer skills** - Regular feedback on quality problems helps developers to improve their coding skills



# Devops Tools

## Maven

- [Maven](#) is one of the important DevOps tools for building projects. Unlike the ANT build system, Apache Maven is more than just an automation build framework. It is also designed to manage reporting, documentation, distribution, releases, and dependencies processes. Written in Java language, Maven can build and manage projects written in [Java](#) or [C#](#), [Ruby](#), [Scala](#), and other languages using project object model (POM) plugins.



# ANY QUERIES

