

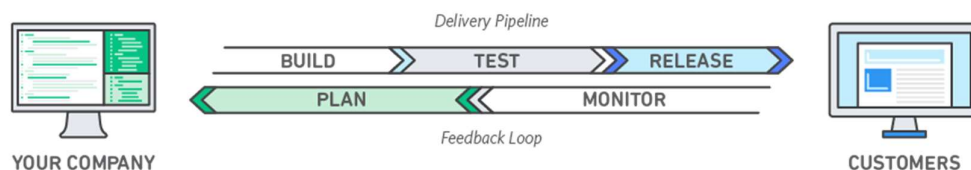
**Aim:** To study DevOps: principles, practices, role and responsibilities of DevOps Engineers.

## **What is DevOps?**

DevOps is a methodology in the software development and IT industry. Used as a set of practices and tools, DevOps integrates and automates the work of software development and IT operations as a means for improving and shortening the systems development life cycle.

DevOps is a combination of software development (dev) and operations (ops). It is defined as a software engineering methodology which aims to integrate the work of development teams and operations teams by facilitating a culture of collaboration and shared responsibility.

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.



The DevOps methodology comprises four key principles that guide the effectiveness and efficiency of application development and deployment. These principles, listed below, centre on the best aspects of modern software development.

1. **Automation of the software development lifecycle.** This includes automating testing, builds, releases, the provisioning of development environments, and other manual tasks that can slow down or introduce human error into the software delivery process.
2. **Collaboration and communication.** A good DevOps team has automation, but a great DevOps team also has effective collaboration and communication.
3. **Continuous improvement and minimization of waste.** From automating repetitive tasks to watching performance metrics for ways to reduce release times or mean-time-to-recovery, high performing DevOps teams are regularly looking for areas that could be improved.
4. **Hyperfocus on user needs with short feedback loops.** Through automation, improved communication and collaboration, and continuous improvement, DevOps teams can take a moment and focus on what real users really want, and how to give it to them.

## **TOOLS OF DEVOPS:**

### **1. Version Control Tool: Git (GitLab, GitHub, Bitbucket):**

Git is perhaps the best and most widely used version control tool in a development era characterized by dynamism and collaboration.

Version control provides developers with a means by which they can keep track of all the changes and updates in their codes such that in the event of a mishap, it is quite easy to return to and use the previous versions of the code and Git happens to be the best for many reasons.

### **2. AWS Cloud Computing and Storage in DevOps:**

AWS features the widest range of service offerings under PaaS, SaaS, and IaaS categories including compute, identity and access management (IAM), networking, and storage. While AWS offers public, private, and hybrid clouds, its focus is more on the public cloud.

### **3. Continuous Integration Tool: Jenkins**

Jenkins is an integration DevOps tool. For continuous integration (CI), Jenkins stands out as it is designed for both internal and plugin extensions. Jenkins is an open-source Java-based automation CI server that is supported by multiple operating systems including Windows, macOS, and other Unix OSs. Jenkins can also be deployed on cloud-based platforms.

#### **4. Container Platforms: Docker**

Container platforms are application solutions that allow developers to build, test, and ship applications in resource-independent environments. Each container comprises a complete runtime environment including the specific application, its libraries, source code, configurations, and all its dependencies. Container platforms offer orchestration, automation, security, governance, and other capabilities.

#### **5. Google Cloud Platform Cloud Computing and Storage in DevOps:**

GCP supports DevOps by providing the services required to develop, store, and deploy high-quality software in shorter cycles. The Google Cloud platform features instances of up to 96 vCPUs and 624 GB RAM alongside services like the cloud console, Google compute engine, and the GCP deployment manager that supports the implementation of DevOps on the Google Cloud Platform.

#### **6. Testing Tool: Selenium**

Selenium is a top open-source testing framework for web applications that supports all major browsers and platforms like Linux, Windows, and macOS. The beauty of Selenium is that it integrates with a wide range of programming languages including Python, C#, Ruby, Java, JavaScript, PHP, and PERL, and several other automation test frameworks.

## **7. CHEF:**

A chef is a useful tool for achieving scale, speed, and consistency. The chef is a cloud-based system and open source technology. The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.

Chef has got its convention for different building blocks, which are required to manage and automate infrastructure.

## **8. 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. It enables to manage entire infrastructure as code without expanding the size of the team.

## **9. 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.

## **10. Nagios:**

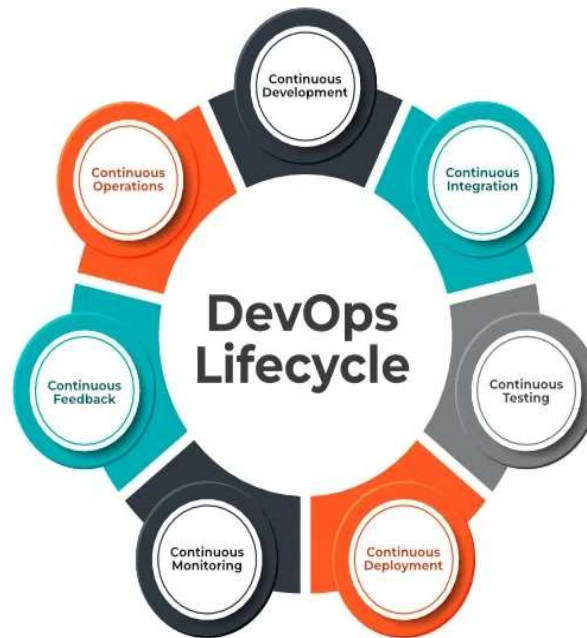
Nagios is one of the more useful tools for DevOps. It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.

## **LIFECYCLE OF DEVOPS**

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from beginning to the final stage of the product. DevOps lifecycle is effortless to manipulate and it helps satisfactory delivery.

### **7 Cs of DevOps:**

1. Continuous Development
2. Continuous Integration
3. Continuous Testing
4. Continuous Deployment/Continuous Delivery
5. Continuous Monitoring
6. Continuous Feedback
7. Continuous Operations



### **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.

### **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.

### **5) Continuous Feedback:**

The application development is consistently improved by analysing 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.



The new code is deployed continuously, and configuration management tools play an essential role in executing tasks frequently and quickly. Here are some popular tools which are used in this phase, such as Chef, Puppet, Ansible, and SaltStack.

### **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.

## **ROLES AND RESPONSIBILITIES OF** **DEVOPS ENGINEERS**

DevOps engineers work full time. They are responsible for the production and continuing maintenance of a software application platform. Below are some roles, responsibilities, and skills which are expected from DevOps engineers, such as:

- Manage projects effectively through an open standard based platform.
- Increases project visibility through traceability.
- Improve quality and reduce the development cost with collaboration.
- DevOps should have the soft skill of problem solver and a quick learner.
- Analyse, design, and evaluate automation scripts and systems.
- Able to perform system troubleshooting and problem-solving across the platform and application domains.
- Ensuring the critical resolution of system issues by using the best cloud security solution services.

**Lab Outcome:** **LO1**-To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements