DEVOPS

➢ What is DevOps?

- DevOps is a combination of Development (Dev) and Operations (Ops) that focuses on collaboration, automation, and continuous improvement in software development and IT operations.
- It aims to deliver software faster, with better quality and reliability.

✓ **Real time example:**

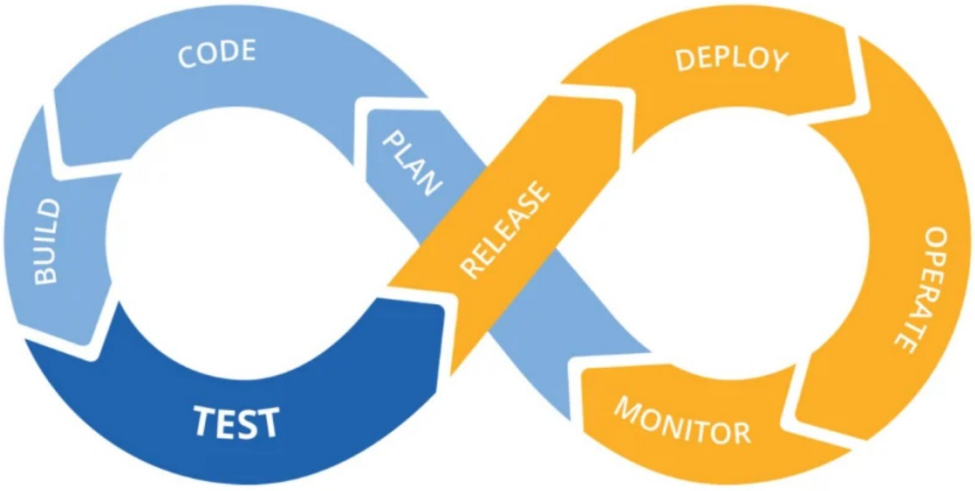
- Imagine a restaurant where the chefs (developers) prepare food and the waiters (operations team) serve customers. Without proper coordination, orders get delayed, food quality suffers, and customers complaint.
- Now, imagine a system where chefs and waiters work together efficiently with automation, quick feedback, and continuous improvements—this is DevOps in action for software development!

➢ Why is DevOps Important?

- Before DevOps, software development teams (developers) and IT operations teams worked separately. This caused delays, inefficiencies, and conflicts. DevOps bridges the gap between these teams, leading to:
- Faster Delivery: Automates software development, reducing time-to-market.
- Higher Quality: Detects bugs early with continuous testing.
- Increased Collaboration: Developers and IT operations work as a team.
- Reliability: Ensures stable applications with automation and monitoring.
- Scalability: Helps businesses handle increasing users and traffic.

➢ DevOps Lifecycle – How It Works

✓ Plan → Develop → Build → Test → Release → Deploy → Operate → Monitor (Repeat)



- Plan:

What Happens in This Stage?

Define project requirements and objectives.

Collaborate between development, operations, and business teams.

Use project management tools to track progress.

Example:

A company decides to develop an online shopping app. They create a roadmap for features like payment integration, user authentication, and product listings.

Tools Used:

Jira, Trello, Confluence, Slack

- Develop (Coding & Version Control)

What Happens in This Stage?

Developers write and modify source code.

Code is stored in a version control system (VCS).

Team members collaborate on features and bug fixes.

Example:

A developer writes code for a login feature and commits changes to GitHub. Other team members review and merge the code into the main project.

Tools Used:

Git, GitHub, GitLab, Bitbucket

- Build (Compiling & Packaging Code)

What Happens in This Stage?

Convert source code into executable software.

Combine different components to create a deployable application.

Automate this process to ensure efficiency.

Example:

Once developers finish coding, Jenkins automatically compiles the code into a working application and prepares it for testing.

Tools Used:

Maven, Gradle, Jenkins, GitHub Actions

- Test (Quality Assurance & Automation Testing)

What Happens in This Stage?

Run automated and manual tests to find bugs.

Perform security, performance, and functional testing.

Ensure application stability before release.

Example:

A test script runs automatically to check if the login feature is working as expected. If it fails, the developer is notified to fix the issue.

Tools Used:

Selenium, JUnit, TestNG, Postman

- Release (Preparing for Deployment)

What Happens in This Stage?

Approve tested code for production.

Ensure all changes are documented and reviewed.

Deploy in a staged manner to reduce risks.

Example:

A company schedules a feature release at midnight to minimize user impact.

Tools Used:

Jenkins, GitHub Actions, AWS CodePipeline

- Deploy (Pushing to Production)

What Happens in This Stage?

The tested software is deployed to a production environment.

Deployment can be manual or automated (Continuous Deployment).

Uses containers and cloud platforms for scalability.

Example:

Using AWS Fargate, an application update is deployed automatically without downtime.

Tools Used:

Docker, Kubernetes, AWS CodeDeploy, Azure DevOps

- Operate (Infrastructure Management & Performance Optimization)

What Happens in This Stage?

Maintain servers, databases, and network resources.

Optimize cloud infrastructure for cost and performance.

Ensure high availability and security.

Example:

A company configures Auto Scaling to handle increased traffic during a sale event.

Tools Used:

Terraform, Ansible, Chef, Puppet

- Monitor (Logging & Continuous Feedback)

What Happens in This Stage?

Monitor application performance and detect issues.

Track logs, errors, and system metrics.

Provide real-time feedback for future improvements.

Example:

A CloudWatch alert is triggered when the application CPU usage exceeds 80%, notifying the DevOps team.

Tools Used:

Prometheus, Grafana, Datadog, Amazon CloudWatch

Devops Tools:

Category	Tools
Version Control	Git, GitHub, GitLab
Continuous Integration (CI)	Jenkins, GitHub Actions
Configuration Management	Ansible, Puppet
Containerization	Docker, Kubernetes
Cloud Providers	AWS, Azure, Google Cloud
Monitoring & Logging	Prometheus, CloudWatch