UNIT III – DEVOPS AND CI/CD CONCEPTS [9 hours]

What is DevOps? Why is it used in industry?,CI/CD – Continuous Integration and Deployment, Introduction to GitHub Actions / Jenkins, Introduction to Docker and Dockerfile, Automating build and test for an app

---

INTRODUCTION

DevOps is a collection of two words, "Development" and "Operations," representing a cultural approach that emphasizes collaboration between development and operations teams to streamline the entire software delivery lifecycle.

What is DevOps?

DevOps is a software development approach emphasizing collaboration, automation, and continuous delivery to provide high-quality products to customers quickly and efficiently. DevOps breaks down silos between development and operations teams to enable seamless communication, faster time-to-market, and improved customer satisfaction.

It allows a team to handle the complete application lifecycle, from development to testing, operations, and deployment. It shows cooperation between Development and Operations groups to deploy code to production quickly in an automated and repeatable manner.

Every phase of the software development lifecycle, including planning, coding, testing, deployment, and monitoring, is heavily automated in DevOps. This improves productivity, ensures consistency, and lowers error rates in the development process.

A culture of continuous improvement is also promoted by DevOps, where feedback loops are incorporated into the procedure to facilitate quicker iteration and better decision-making. Organizations can increase their agility, lower costs, and speed up innovation by adopting DevOps.

Why is DevOps Needed?

- Before DevOps, the development and operation team worked in complete isolation.
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members spend a large amount of their time testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production.
- Coding & operation teams have separate timelines and are not sync, causing further delays.

How is DevOps different from traditional IT

In this DevOps training, let's compare the traditional software waterfall model with DevOps to understand the changes DevOps brings. We assume the application is scheduled to go live in 2 weeks, and coding is 80% done. We assume the application is a fresh launch, and the process of buying servers to ship the code has just begun-

Sl.No	**Old Process**	**DevOps**
1.	After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.	After placing an order for new servers, the Development and Operations team work together on the paperwork to set up the new servers. This results in better visibility of infrastructure requirements.
2.	Projections about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers who have deep knowledge of the application.	Projections about failover, redundancy, disaster recovery, data center locations, and storage requirements are pretty accurate due to the inputs from the developers.
3.	The operations team has no clue about the progress of the Development team. The operations team develops a monitoring plan as per their understanding.	In DevOps, the Operations team is completely aware of the developers' progress. Operations teams interact with developers and jointly develop a monitoring plan that caters to IT and business needs. They also use advanced Application Performance Monitoring (APM) Tools.
4.	Before going go-live, the load testing crashes the application, and the release is delayed.	Before going go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks, and the application is released on time.

Why is DevOps used?

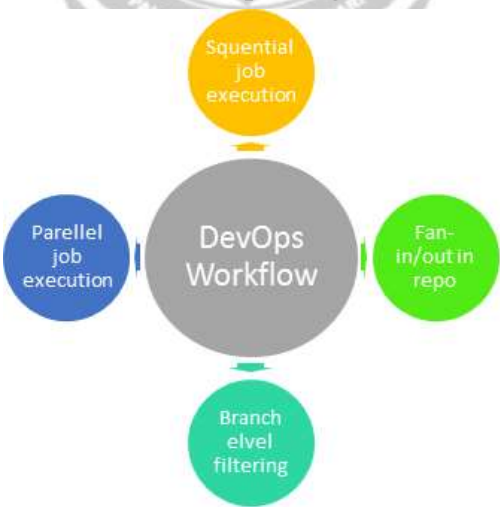
DevOps allows Agile Development Teams to implement Continuous Integration and Continuous Delivery, which helps them launch products faster into the market.

Other Important reasons are:

1. **Predictability**: DevOps offers a significantly lower failure rate of new releases.
2. **Reproducibility**: Version everything so that earlier versions can be restored anytime.
3. **Maintainability**: Effortless recovery process in the event of a new release crashing or disabling the current system.
4. **Time to market:** DevOps reduces the time to market up to 50% through streamlined software delivery. It is particularly the case for digital and mobile applications.
5. **Greater Quality:** DevOps helps the team improve application development quality by incorporating infrastructure issues.
6. **Reduced Risk:** DevOps incorporates security aspects in the software delivery lifecycle, and it helps reduce defects across the lifecycle.
7. **Resiliency:** The Operational state of the software system is more stable, secure, and changes are auditable.
8. **Cost Efficiency:** DevOps offers cost efficiency in the software development process, which is always an aspiration of IT management.
9. **Breaks larger code base into small pieces:** DevOps is based on the agile programming method. Therefore, it allows breaking larger codebases into smaller and manageable chunks.

DevOps Workflow

Workflows provide a visual overview of the sequence in which input is provided. It also tells about performed actions, and output is generated for an operations process.



Workflow allows the ability to separate and arrange jobs that the users top request. It also can mirror their ideal process in the configuration jobs.

How is DevOps different from Agile? DevOps Vs Agile

Stakeholders and communication chain a typical IT process.

Agile addresses gaps in Customer and Developer communications

DevOps addresses gaps in Developer and IT Operations communications

Difference between DevOps and Agile

Agile	Devops
Emphasize breaking down barriers between developers and management	DevOps is about software deployment and operation teams.
Addresses gaps between customer requirements and development teams.	Addresses the gap between the development and Operation team
Focuses more on functional and non-functional readiness	It focuses on operational and business readiness.
Agile development pertains mainly to the company's way development is thought out.	DevOps emphasises deploying software in the most reliable and safest ways that aren't always the fastest.
Agile development emphasises training all team members to have varieties of similar and equal skills. So that, when something goes wrong, any team member can get assistance from any member in the	DevOps likes to divide and conquer, spreading the skill set between the development and operation teams. It also maintains consistent communication

absence of the team leader	
Agile development manages “sprints”. It means that the timetable is much shorter (less than a month), and several features are to be produced and released in that period.	DevOps strives for consolidated deadlines and benchmarks with significant releases rather than smaller and more frequent ones.

DevOps Automation Tools

Automating all the testing processes and configuring them to achieve speed and agility is vital. This process is known as DevOps automation.

Classified briefly into six different categories.

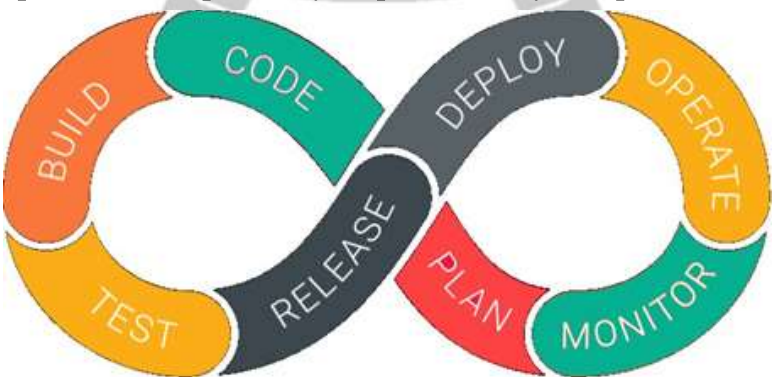
1. Infrastructure Automation- ex: AWS
2. Configuration Management- ex: Chef
3. Deployment Automation- ex: Jenkins
4. Performance Management- ex: App dynamic
5. Log Management- ex: Splunk
6. Monitoring- ex: Nagios

What is DevOps Lifecycle?

The DevOps Lifecycle is a series of development stages that guide everyone as efficiently as possible through the end-to-end process of product development. All of these components of the DevOps lifecycle is necessary to take the maximum leverage of the DevOps methodology.

DevOps Lifecycle: Key Components

Here are some important DevOps Lifecycle phases / Key components of DevOps:



Stage 1) Continuous Development:

This practice spans the planning and coding phases of the DevOps lifecycle. Version-control mechanisms might be involved.

Stage 2) Continuous Integration:

This software engineering practice develops software by frequently integrating its components. It helps to ensure that changes in the source code do not break the build or cause other problems.

Stage 3) Continuous Testing:

This DevOps lifecycle stage incorporates automated, prescheduled, continued code tests as application code is written or updated. Such tests can be written manually or in conjunction with continuous integration tools.

Stage 4) Continuous Deployment:

The deployment process takes place continuously in this DevOps lifecycle phase. It is performed so that any changes made in the code should not affect the functioning of a high traffic website.

Stage 5) Continuous Monitoring:

During this phase, developers collect data, monitor each function, and spot errors like low memory or server connection are broken. For example, when users log in, they should access their account, and a failure to do so means there's a problem with your application.

Stage 6) Continuous Feedback:

Continuous feedback is like a progress report. In this DevOps stage, the software automatically sends out information about performance and issues experienced by the end-user. It's also an opportunity for customers to share their experiences and provide feedback.

Stage 7) Continuous Operations:

It is the last, shortest, and most straightforward phase of DevOps. It also involves automating the application's release and all these updates that help you keep cycles short and give developers more time to focus on developing.
