02/9/202

DevOps Curriculum Using with Tools.

Overview of Devops Architecture Design.

UNIT - I : DevOps Workflow

Introduction to DevOps.

1.1.1 Definition and goals of devops

1.1.2 Devops Architecture

1.1.3 Devops Architecture Workflow

Definition and goals of DevOps:

The main goals of DevOps are to improve the speed, efficiency and quality of software development and delivery.

Increase Deployment Frequency

Improve Deployment Quality

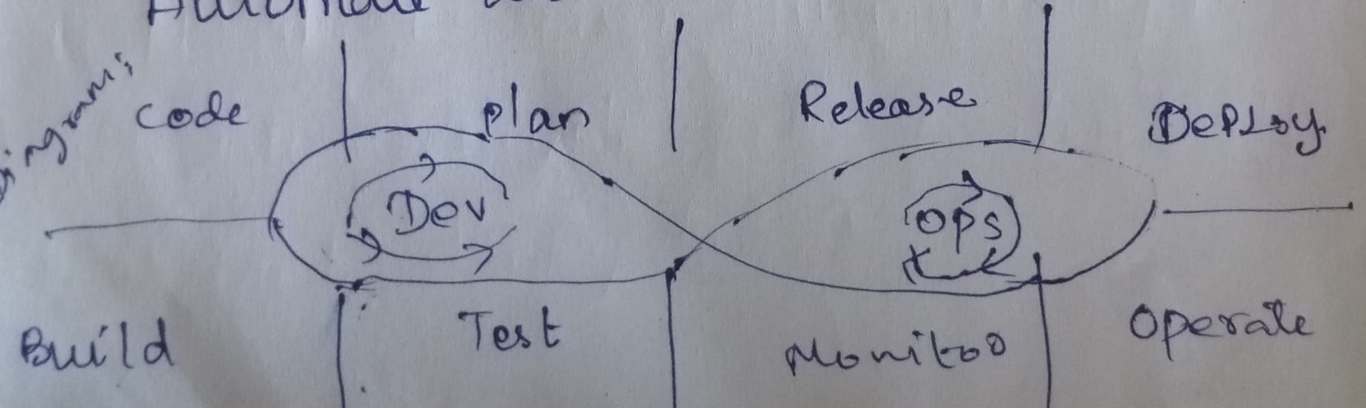
Reduce Lead time for changes

Enhance Collaboration and Communication

Improve Recovery time

Automate and streamline Processes.

Diagrams:



1.2 Devops architecture

key components of Devops architecture

* version control system (vcs)

purpose: Manages code versions, tracks, changes and facilitates collaboration among develops

* Continous Integration (CI):

Autmes the process of integrating code changes from multiple contributors into a single software project.

* Continous Delivery / Continous Deployment (CD)

Automates the deployment of code changes to various environments, ensuring that software can be released reliably at any time.

Configuration Management

Manages and maintain consistency in software environments (development, testing, production)

Infrastructure as code (Iac):

Manages and provisions Computing infrastructure through machine-readable defenition files, rather than physical hard or interactive configuration tools.

Containerization and orchestration:

Packages applications and their dependencies into Containers to ensure Consistency across environments and Simplifies deployment.

Continous Monitoring and Logging:-

Monitors applications and infrastructure to detect the performance issues, errors, and Security threats.

Colloboration and Communication tools:-

Facilitates Communication and collaboration among team numbers, enabling faster decision-making and issue ~~&~~ resolution.

1.1.3 DevOps Workflow

Code: Developers write and Commit Code to version Control system (eg: Git).

Build: The CI Server automatically builds the Code into executable files, Creating artifacts that Can be deployed.

Test: Automated tests are run to ensure the quality of the code. This includes unit test, integration tests and sometimes security checks.

Releases: If all test pass, the code is packages and prepared for deployment.

Deploy: The code is automatically deploy to the target environment. (eg: staging, production). Continuous deployment involves deploying to production automatically, whereas continous delivery, might require manual approval.

Operate: The deployed applications are monitored for performance, reliability and security. Continuous monitoring tools collect metrics and logs, providing insights into the application's behaviour.

Monitor: Feedback is collected from monitoring and users, providing data for continuous improvement. Any issues detected are fed back into the development process for resolution.

1.2 Devops vs. Traditional IT Operations

1.2.1 Differences between Devops and traditional software Developmend and IT operation

1.2.2. Benefits of adopting DevOps practices

1.2.3 Building a culture of Colloboration and Communication between development and operations teams.

1.2.4 The role of automation and monitoring in enhancing team efficiency.

1.2.1:

Colloboration and Communication,

Traditional Approach: Development and IT Operations team work in Silos. Developers focus on writing Code, and operations are responsible for deploying and maintaining the Application. This often leads to misCommunication

Develops Approach: devops encourages Continous colloboration and communication between development and operations teams.

Process and workflow:

* Traditional Approach: Uses a Sequential development process (eg: waterfall model).

Develops Approach: Follows as agile and iterative Approach where development, testing, and deployment are done Continously and Concurrently.

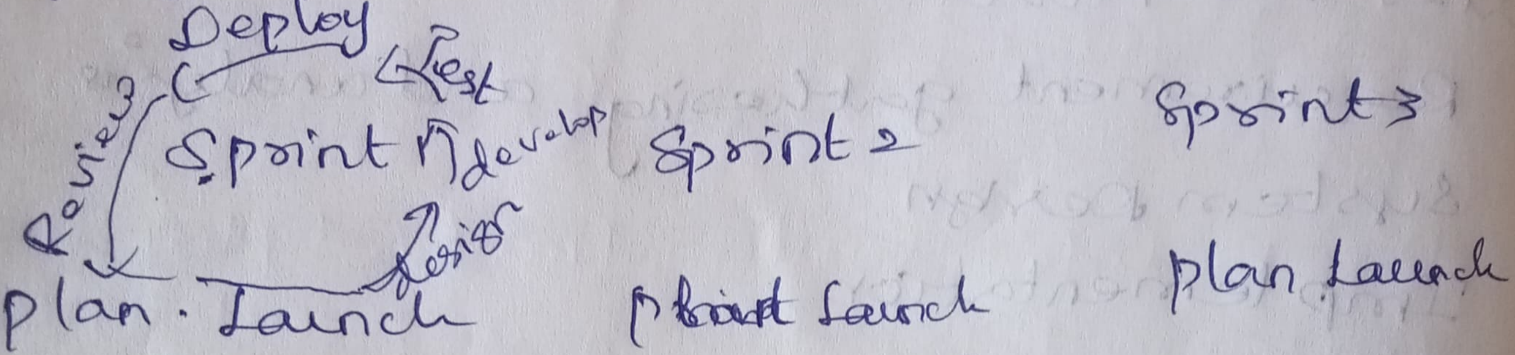
Water Fall Model:

Requirement gathering and analysis
System Design
Implementation
Testing
Development
Maintenance.

It can make your projects flow smoothly, avoid bottle necks, help you hit deadlines, ensures deliverables are met before the next phase begins, and allow the team overall to shine with

Agile:
Agile development is important because it helps to ensure that development teams Complete projects on time and within budget. It also helps to improve Communication between the development team and the product Owner. Additionally,

Agile development methodology can help reduce the risks associated with Complex projects.


Deploy
Test
Review
Sprint
develop
Design
plan
Launch
Sprint 2
plan Launch
Sprint 3
plan Launch