02/01/24

Devops curriculum using with tools.

overview of Devops Architecture design

UNIT - 1 Devops workflow:

Visual studio → collection of Languages: MS word collection of document.
ex: Excel, - - -

net using in intesmediator Language.

Introduction to devops:

i) Definition and goals of Devops.

ii) Devops Architecture

iii) Devops Architecture window.

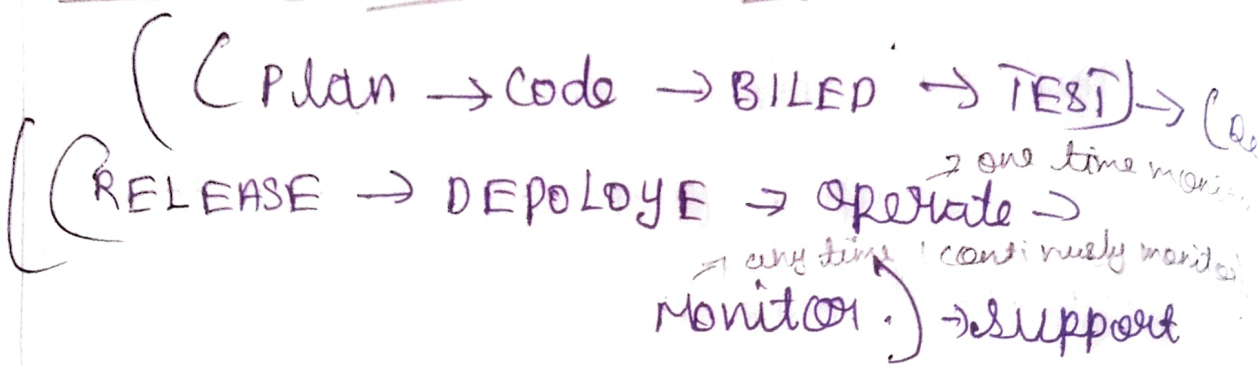
Definition and goals of Devops:

The main goals of Devops are to improve the speed, efficiency, and quality of software development and delivery. Here are the primary objectives.

* increase deployment Frequency.
* improve Deployment quality.
* Reduce Lead Time for change.
* Enhance collabration and communicat
* improve Recovery Time.

* Automation and streamline Process.

Devops Architecture Diagram:



Dev – Developers (Technical Teams)
Ops – operations. (End users).

Two Type of case;

* Positive case, ex: Palanibo@gmail.
* negative case. / ex: @palan gmail bo

Devops Architecture:

Key components of Devops Architecture.

* Version control system (VCS):

⇒ Purpose: manges code version. tracks changes, and facilitates collaboration among developers.

* continous integration (CI):

⇒ Purpose: Automates the process

of integrating code changes from multiple contibutors into a single software project.

* continous delivery / continous deployment (CCD):
⇒ Purpose: Automates the deployements of code changes to various environments, ensuring that software can be released reliably at any time.

* configuration as code (Cac) management:
Purpose: manages and maintains consistency in software environment (development, testing, production).

Infrastructure as code (Iac):
Purpos: manages and provisions computing infrastructure through machine-readable definition files, rathes than Physical hardwas or interactive configuration tools.

Containerization and orchestration.

Purpose: Packages applications and their dependencies into containers to ensure consistency across environments and simplifies deployment.

continuous monitoring and Logging:

Purpose: Monitors applications and infrastructure to detect performance issues, errors, and security threats.

Collaboration and communication Tools:.

Purpose: Facilitates communication and collaboration among team members, enabling faster decision-making and issue resolution.

Devops workflow:.

Code: Developers write and commit code to a version control system (e.g, Git).

Build: The CI server automatically builds the code into executables files,

creating artifacts that can be deployed.
Test: Automated tests are run to ensure the quality of the code. This includes unit tests, integration tests, and sometimes security checks.
Release: If all tests pass, the code is packaged and prepared for deployment.
Deploy: The code is automatically deployed to the target environment (eg, staging, production). Continous deployedment involves deploying to production automatically, whereas continous delivery might require manual approval.
Operate: The deployed application are monitored for performance, reliability, and security. Continous 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.

Devops vs Traditional IT operations.

* Differences between Devops and traditional software development and IT operations.

* Benefits of adopting Devops practices.

* Building a culture of collaboration and communication between development and operations teams.

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

Differences between Devops and traditional software development and IT operations:

Collaboration and communication:

* Traditional approach:

=) Development and IT operations teams work in silos. Developers focus on writing code, and operations teams are responsible for deploying and maintaining the application. This

often leads to miscommunication, delays, and a lack of shared understanding.

DevOps Approach:

=> DevOps encourages continous collaboration and communication between development and operations teams, both teams work togethes throughout the software development lifecycle, fostering a culture of shred responsibility.

Process and Workflow:

* Traditional Approach: uses a sequential development process (e.g. waterfall model) where each phase must be completed before the next begins. This can create bottlenecks and slow down the process.

DevOps Approach: Follows an agile and iterative approach where development, testing, and deployment are done continously and concurrently. This helps identify and fix issues earlier in the development process.

water fall model:,

* it can make your projects flow smoothly, avoid bottlenecks, help you hit deadlines, ensure deliverables are met before the next phase begins, and allow the team overall to shine with perfection. This in-depth guid analyzes the advantages of the water fall methodology.

Requesment gathering

Agile:,

Agile development is important because it helps to ensure that development teams complete projects on time and within budget. it als helps to improve communication between the development team and the product owner. Additionaly, Agile development methodology can help reduce the

Risks associated with complex projects.