2/09/24

Devops curriculum using with tools.

S.Mathanki

overview of Devops Architecture Design.

Unit - 1

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. Here are the primary objectives:

* Increase Development Frequency

* Improve Development Quality

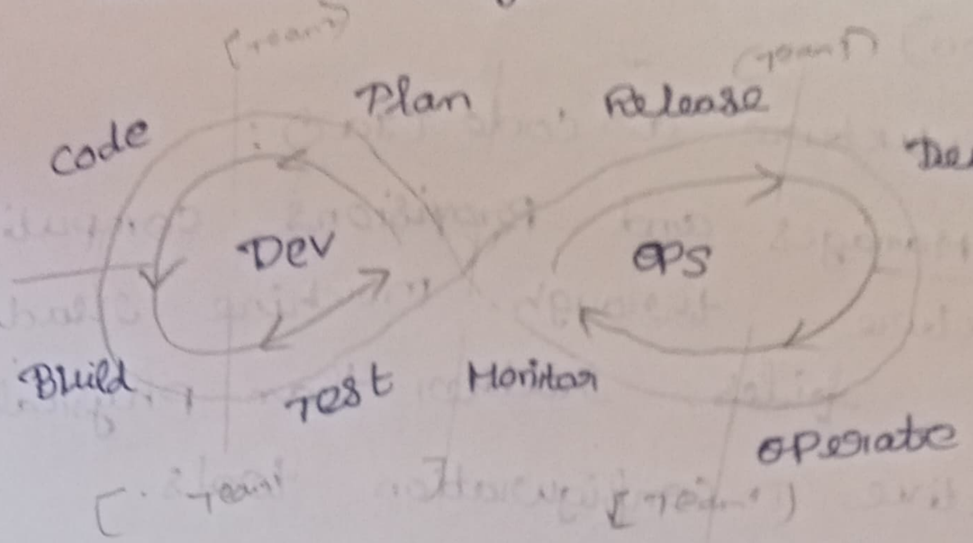
* Reduce Lead Time for changes

* Enhance collabration and communication

* Improve Recovery Time

* Automate and Streamline Processes

DevOPS Architecture Diagram



1.1.2 DevOPS Architecture.

Key components of DevOPS Architecture.

* version control system (vcs):

purpose: Manage code versions, tracks changes, and facilitage collaboration among developers.

* continuous integration (CI):

purpose: Automates the process of integrating code changes from multiple contributors into a single software project.

* continuous delivery/continuous deployment (CD):

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

* configuration Management:

purpose: Manages and maintains consistency in software environments (development, testing, Production)

* Infrastructure as code (IaC):

Purpose: Manages and provisions computing infrastructure through machine readable definition files, rather than physical hardware 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 application and infrastructure to detect Performance issues, errors and Security threats.

* collabration and communication Tools:

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

1.1.3 Devops Workflow

Code: Developers write and commit code to a Version Control system (eg. Git)

Build: The CI Server automatically builds the code into executable files, creating artifacts that can be deployed.

Test: Automated test 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 develoyed to the target environment (e.g, Staging, Production) Continuous Develoyment involves deploying to Production automatically, whereas continuous Monitoring 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 behavior.

Monitor:

Feedback is collected from monitorin

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 development and IT operations:

1.2.2 Benefits of adopting DevOPS Practices

1.2.3 Building a culture of Collabration and communication between development and operations teams.

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

Difference between DevOPS and traditional Software development and IT operations.

* collaboration and communication:

* Traditional APProach: Development and IT operations team work in silos. Developers focus on writing code and opriations.

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 continuous collaboration, & communication b/w development

Waterfall model:

*It can make your Project 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 guide analyses the advantages of the waterfall

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 risk associated with complex project.