

Overview of Devops Architecture Design

Unit 1: Devops workflow

Introduction to Devops

- i) Definition and goals of Devops
- ii) Devops architecture
- iii) Devops architecture workflow.

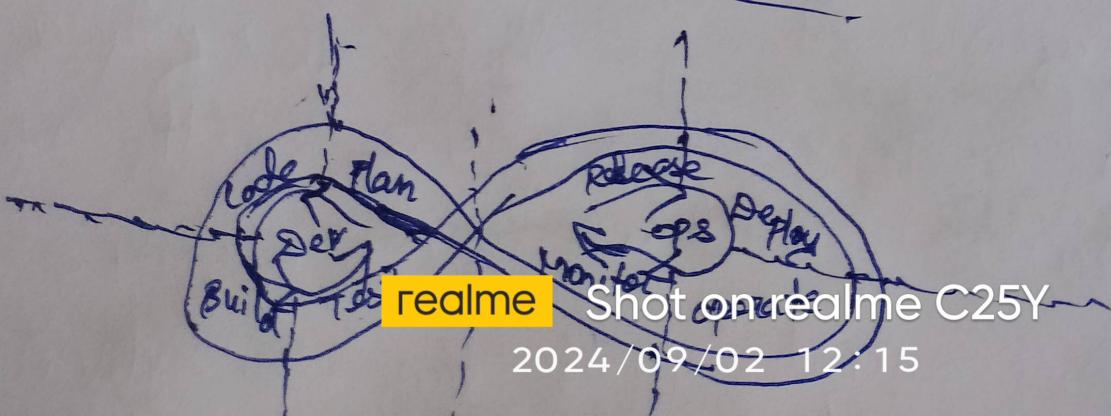
Definition and goals of Devops.

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

These are the primary objectives.

- * Increase deployment frequency
- * Improve deployment quality
- * Reduce lead time for changes.
- * Enhance collaboration and communication
- * Improve recovery time.
- * Automate and streamline processes.

Devops architecture Diagram.



Devops Architecture

Key Components of Devops Architecture:

- * Version Control System (VCS):

Purpose: Manages code versions, tracks changes, and facilitates 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.

purpose - maintains applications and infrastructure to detect performance issues, errors, and security threats.

The Collaboration and Communication tools

PurPOSE : facilitates communication and collaboration among team members, enabling fast decision-making and issue resolution.

DevOps Workflow:

Code is developed, written and committed. Commit code to a version control system (e.g. Git).
Build: The CI server automatically builds the code into executable files, creating artifacts, and creating it can be deployed.

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

release: If all tests pass, the code is packaged and prepared for deployment. These are automated.

Deploy: The code is automatically deployed to the target environment (e.g., containers).

Deployment: Continuous deployment involves deploying to production automatically. Whereas continuous delivery might require manual approval.

and continuous monitoring tools collect metrics, logs, providing insights into the applications behaviour.

DevOps vs. traditional IT operations:

- 1) Differences between DevOps and traditional software development and IT operations.
 - 2) Benefits of adopting DevOps practices.
 - 3) Building culture of collaboration and communication between development and operations teams.
- a) The role of communication and monitoring in enhancing team efficiency.
- Difference between DevOps and traditional software development and IT operations.
- b) Collaboration and Communication:
- # Collaboration approach: Development and traditional operations teams work in silos. Development focuses on writing code, and operations focus on responsible deployment and teams are the application. This often main failing leads to miscommunications, delays, and

a lack of shared understanding.

~~DevOps Approach~~: DevOps encourages collaboration and communication between development and operations teams. Both teams work together throughout the software development lifecycle, fostering a culture of shared responsibility.

Waterfall 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 analysis of the waterfall methodology

guides us through the waterfall methodology.

Requirement gathering & Analysis

System Design

Implementation

Testing

Development

Maintenance

Agile

Agile development is important because it helps to ensure that development teams complete projects on time and within budget.

realme gShot on realme C25Y for image.