ROUNAK KATIYAR
Batch: T-13   Rollno. 47
Sub: SEPM

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Expt. 1: To understand DevOps: Principles, Practices, and DevOps Engineer Role & Responsibilities

DevOps:

DevOps is a combination of 'Development' (Dev) and 'Operations' (Ops) practices aimed at automating and integrating the processes of software development and IT operations. It emphasizes collaboration, communication, and integration between development and operations team to deliver Software quickly, efficiently, and reliably.

Key Principles of DevOps:

i. Collaboration and Communication:
Breaking down Silos between teams to work together Seamlessly.

ii. Automation:
Automative repetitive tasks Such as code deployment, testing and monitoring.

iii. Continous Integration & Continous Delivery : (CI/CD):
CI: Integrating code changes into a Shared respository frequently
CD: Automating the deployment of applications to production environments.

iv. Infrastructure as Code (IaC):
Managing and Provisioning infrastructure through code rather than manual processes.

v. Monitoring and Feedback:
Continous monitoring of applications and infrastructure to improve performance and reliability.

Need of DevOps:
DevOps was needed to address challenges in traditional software development and IT operations, where ineffiencies and conflicts often arose due to lack of collaboration and automation.

i. Silos between Teams
→ Development Teams focused on building features, while Operations Teams were responsible for stability and reliability
→ DevOps promotes collaboration and shared responsibility, ensuring both teams work together seamlessly.

ii. Slow Development & Deployment Cycles:
→ Traditional methods like Waterfall resulted in long development cycles, making it difficult to adapt to changing market needs.
→ DevOps encourages iterative development and Continous
~~iii.~~ delivery (CI/CD), enabling faster and more frequent errors.

iii. Manual and Error-Prone Processes:
→ Manual tasks like deployment, testing, and Configuration were time-consuming and prone to errors.
→ Automation in DevOps reduces human errors and accelerates processes.

iv. Inconsistent Environments :

→ Applications worked in development but often failed in production due to environment differences.

→ Tools like Docker and Kubernetes ensure Consistency across enviroments.

v. Lack of Scalibility :

→ Scaling infrastructure manually was slow and inefficient.

→ DevOps practices like cloud adoption and automated provisioning make Scaling Seamless and cost-effective.

DevOps Practices :

i. Continous Development :

This phase is the phase that involves planning and coding, versioning and managing builds of the Software's functionality.

Tools : Git, Github, Maven, Apache Ant.

ii. Continous Testing :

Continous testing is, executing automated tests, Continously and repeatedly against the Code base and various environments.

Tools : Bamboo, Appium.

iii. Continous Integration :

It refers to the build and unit testing stages of the Software release process.

Tools : Jenkins, Travis CI, Circle CI.

iv. Continous Delivery & Deployment:
These originate from Continous integration, a method to develop, build and test new code rapidly with automation so that only code that is known to be good becomes part of a software.

v. Infrastructure Management:
It is an automated method for maintaining computer systems and software in a known, consistent state.

vi. Configuration Management:
It is the practice of describing all software runtime environment and network settings and parameters in simple textual format, that can be stored in VCS and versioned on request.
Tools: Chef, SaltStack, Terraform

vii. Microservices Architecture:
It is used to easy the process to create, deploy, and run applications by using containers, which allow packing up an application with all parts it needs.
Tools: Docker.

~~viii.~~

DevOps Engineer :

Roles and Responsibilites :

A ~~Dev~~ DevOps Engineer plays a critical role in bridging the gap between development and operations teams to ensure smooth, efficient, and reliable software delivery.

i. Collaboration & Communication:
Acts as a bridge between development, QA, and operations teams to foster a culture of collaboration.

ii. CI/ CD Pipeline Management:
Design, build, and maintain ~~in~~ using CI/CD pipelines. Automate build, test, and deployment processes to ensure faster and error-free releases.

iii. Infrastructure as Code (IaC):
Implement and manage IaC using DevOps tools, ensuring consistency, scalability, and reliability in infrastructure provisioning and configuration.

iv. Monitoring & Logging:
~~Setting~~ Set up monitoring and alerting systems to track application and infrastructure performance

v. Automation:
Automate repetitive tasks, including deployment, scaling and testing.