THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

# SEPM

Name:- Ajinkya Dahiwal
Roll no:- 21 [TY-AI&DS]

To understand DevOps : Principles, Practices & DevOps Egineer Role & Responsibilities.

What is DevOps?

→ DevOps is a collabarative approach where teams work together to build & deliver secure software efficiently. It combines Software development (dev) & operations (ops) to accelerate delivery through automation, collabration, fast feedback & iterative improvement. Built on Agile methadology. DevOps creates a culture of accountability, collabration & shared responsibility for business outcomes.

Core Principles of DevOps :

- Develop & test in production - like enviromments.
- Develop builds frequently.
- Continuously validate operational - quality.

Key Practices of DevOps :

1. Continuous Deployment :
   Continuous delivery & deployment originate from continious integration, a method to rapidly develop, build & test new code with automation to that only code that is known to be good becomes part of a software product.

2. Continuous Development :
   This is the phase that involves planning & coding, versioning & managing builds of the software applications functionality.

Eg: Git, Github, Maven.

3. Continuous Testing:
Continuous testing is executed automated tasks, continuously & repeatedly against the code base & the various deployment environments. It is a software testing methodology which focuses on achieving continuous quality & improvement.
Eg: Appium, Bamboo.

4. Continuous Integration:
Continuous Integration refers to the build & unit testing stages of the software release process. Every revision that is committed triggers an automated build & test.
Eg: Tenkins, Travis, CI.

5. Infrastructure Management:
Without automation, building & maintaing large-scale modern without automation IT systems can be a resource intensive undertaking & can lead to increased risk due to manual error. Configuration & resource management is an automated method for maintaining computer systems & software in a known, consistent attack.

- Configuration Management:
Infrastructure as code is the practice of describing all software runtime environment & networking settings & parameters in simple textual format, that can be stored in your version control systems [VCS] & versioned on request. These test files are called manifests & are used by DevOps tools to

automatically provision & configure build servers, testing, staging & production environments.
Eg: chef, saltstack

DevOps Engineer Role:
A DevOps engineer manages a company i.e. It infrastructures, bridging development & operation, the primary goal is to improve the process and efficiency throughout the software development lifecycle.

Key Role:
1. Facilitator of Collabration:
Bridging the gap between development, operation & QA teams to streamline communication.
2. Automation specialist:
Automate repetitive tasks like testing, deployment & monitoring.
3. Continuous Integration & Continuous Delivery [CI/CD]:
Design, implement & maintain CI/CD pipelines to enable faster, reliable & repetable software releases.
4. Infrastructure as Code:
Use tools like Terraform, Ansible or cloud formation to define & provision infrastructure through code.
5. Monitoring & Incident Management:
Set up monitoring system to track application performance and troubleshoot issue in real time. It also ensures that systems are resistent and downtime is minimized.
6. Cloud & Infrastructure Management:
Deploy, manage & optimize application on cloud platform

like AWS, Azure or Google Cloud, also handles container orchestration.

Key Responsibilities :

1. Collabration & Planning :
   Work with development & operations teams to plan & design scalable solution.
2. Configuration Management :
   Uses tools like Puppet, chef or Aricble to manage server configuration & ensure consistency.
3. Pipeline Management :
   Maintain CI/CD pipelines to ensure seamless build, test & deployment workflows.
4. Monitoring & logging :
   Implement monitoring tools like Promethem, curafana or splunk to track system health & measurement performance.
5. Support & Troubleshooting :
   Respond to incidents & resolve production issues promptly & identify root causes of failure & implement fines.
6. Documentation & Reporting :
   Document system configurations, deployment processes & troubleshooting guides.