Tanush Bidkar
TII-11

SEPM

To understand DevOps Principles, Practices & DevOps Eginer Role & Responsibilities

What is DevOps?
DevOps is a collaborative approach where teams work together to build & deliver secure software efficiently. It combines software development (dev) & operations (ops) to accelerate delivery through automation, collaboration, fast feedback & iterative improvement. Build on Agile methodology, DevOps creates a culture of accountability, collaboration and shared responsibility for business outcomes.

Core Principles of DevOps:
- Develop and test in production - like environments
- Develop builds frequently.
- Continuosly validate operational quality.

Key Practices of DevOps:
(1) Continuous Deployment
Continuous delivery and deployment originate from continuous integration, a method to rapidly develop, build and test new code with automation to that only code that is known to be good becomes part of a software product.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

(2) Continuous Development
This is the phase that involves planning and coding, versioning and managing builds of the software applications functionality. Eg. Git, Github, Maven.

(3) Continuous Testing:
Continuous testing is executing automated tests, continuously and repeated against the code base and the various deployment environments. It is a software testing methodology which focuses on achieving continuous quality and improvement.
Eg. Appium, Bamboo.

(4) Continuous Integration:
Continuous Integration refers to the build and 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 and maintaining large-scale modern without automation. IT systems can be a resource intensive under-taking and can lead to increased risk due to manual errors. Configuration and resource management is an automated method for maintaining computer systems and software in a known, consistent attact.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

(6) Configuration Management

Infrastructure as Code is the practice of describing all software runtime environment and networking settings and parameters in simple textual format, that can be stored in your version control system (VCS) and versioned on request. Those text files are called manifests and are used by DevOps tools to automatically provision and configure build servers, testing, Staging and production environments.

Eg. Chef, saltstack

DevOps Engineer Role:

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

Key Role:

(1) ~~Facilitat~~ Facilitator of collaboration

Bridging the gap between development, operation and QA teams to streamline communication

(2) Automation Specialist

Automate repetitive task like testing, deployment monitoring

(3) Continuous Integration and Continuous Delivery (CI/CD)

design, implement and maintain CI/CD pipelines to enable faster, reliable, and repeatable software releases.

4) Infrastructure as code:

Use tools like Terraform, Ansible or cloud formation to define and provision infrastructure through code.

5) Monitoring and 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 and Infrastructure Management.
Deploy, manage and optimize applications on cloud platform like Aws, Azure or Google Cloud, also handles container orchestration.

Key Responsibilites

1) Collaboration and planning:
Work with development and operations teams to plan and design scalable solution.
2) Configuration Management:
Use tools like Puppet, chef or Ancible to manage server configuration and ensure consistency.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

(3) Pipeline Management
Maintain CI/CD pipelines to ensure seamless build, test and deployment workflows.

(4) Monitoring and logging
Implement monitoring tools like Prometheses, curafana as splunk to track system health and measurement performance.

(5) Support and trouble Shooting
Respond to incident and resolve production issues promptly and identify root causes of failure and implement fines

(6) Documentation and Reposting
Pocument system configuration, deployment processes and troubleshooting guides.