

SEPM

Atharva Kheule
T-13 - 48

To understand DevOps : Principles, Practices & DevOps Engineer Role & Responsibilities

What is DevOps?

It 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. Built on Agile methodology, DevOps creates a culture of accountability, collaboration & shared responsibility for business outcomes.

Core Principles of DevOps:

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

Key practices of DevOps:

1. Continuous Deployment:

Continuous delivery & deployment originate from continuous integration, a method to rapidly develop, build & test new code with automation so 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 and coding, versioning & managing builds of the software application functionality.

eg: Git, GitHub, Maven.

3. Continuous Testing:

It is executed automated tests, continuously and repeatedly against the code base & the various deployment environments. It is a software testing methodology which focuses on achieving continuously quality & improvement.
Gg: 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 and test.

eg: Jenkins, Travis, CI.

5. Infrastructure management.

Without automation, building & maintaining large-scale modern with automation IT system can be lead to increased risk due to manual error. Configuration & resource management is an automated method for maintaining computer systems & software in a known consistent state to avoid attack.

6. 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 text files are called manifests & are used by DevOps tools to automatically provision & configure build servers, testing, staging & production environments.

eg: Chef, Salt stack.

DevOps Engineer Role:

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

Key Role:

1. Facilitator of Collaboration:

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

2. Automation Specialist.

Automate integration repetitive tasks like testing, deployment and monitoring.

3. Continuous Integration & Continuous Delivery (CI/CD):
Design, implement and maintain CI/CD pipelines to enable faster, reliable & repeatable software releases.

4. Infrastructure as Code:
Use tools like Terraform, Ansible or CloudFormation to define & provision infrastructure through code.

5. Monitoring & Incident Management:
Setup monitoring system to track application performance & troubleshoot issues in real time. It also ensures that system are resilient & downtime is minimized.

6. Cloud & Infrastructure Management:
Deploy, manage & optimize applications on cloud platform like AWS, Azure or Google Cloud, also handles container orchestration.

Key Responsibilities:

1) Collaboration & planning:
Work with development & operations teams to plan & design scalable solutions.

2) Configuration Management:
Uses tools like puppet, chef or Ansible to manage server configurations & ensure consistency.

3. Pipeline Management:
maintain CI/CD pipelines to ensure seamless build, test & deployment workflows.
4. Monitoring & Logging:
implement monitoring tools like Prometheus, Grafana or Splunk to track system health & measurement performance.
5. Support & Troubleshooting:
respond to incident & resolve production issues promptly & identify root cause of failure & implement fixes.
6. Documentation & Reporting:
Document system configurations, deployment process & troubleshooting guides.