

To understand Devops: Principles, Practices and Devops Engineer Role & Responsibilities

What is Devops?

Devops is a collaborative approach where teams work together to build and deliver secure software efficiently. It combines software development and operations (Ops) to accelerate, deliver 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
- Continuously 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 so that only code that is known to be good becomes part of Software product

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 repeatedly against the code base and the various deployment environments. Eg. Appium, Bamboo

④ Continuous Integration:

Continuous Integration refers to the build and unit testing stages of the software release process
eg. Jenkins, CI, travis

⑤ Infrastructure Management.

Without automation, building and maintaining large-scale modern without automation. IT systems can be a resource intensive undertaking and can lead to increase risk due to manual errors.

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)

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

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

Infrastructure as code.

Use tools like Terraform, Ansible or Cloud Formation to define and provision infrastructure through code.

④

Monitoring and Incident Management.

Set up monitoring system to track application performance and ~~track~~ trouble shoot issue in real time. It also ensures that systems are resistant and downtime is minimized.

Key Responsibilities

- 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 Ansible to manage server configuration and ensure consistency.
- 3) Pipeline Management
Maintain CI/CD pipelines to ensure seamless build, test and deployment workflows.
- 4) Monitoring and logging.
Implement monitoring tools like Prometheus, Grafana and Splunk to track system health and measurement performance.
- 5) Support and Troubleshooting
Respond to incident and resolve production issues promptly and identify root causes of failure and implement fixes.
- 6) Documentation and Reposting.
Document system configuration, deployment processes and troubleshooting guide.