

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: Jenkins, Travis CI

~~5. An~~

5. Infrastructure Management:

Without automation, building & maintaining large-scale modern 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 state.

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 system (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, Saltstack

DevOps Engineer Role:

A DevOps engineer manages a company's IT infrastructure, bridging development & operations. The primary goal is to automate processes & improve efficiency through the software development lifecycle.

Key Roles:

1. **Facilitator of Collaboration:**
Bridging the gap between development, operations & 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 & repeatable software releases.
4. **Infrastructure as Code:**
Use tools like Terraform, Ansible or CloudFormation to define & provision infrastructure through code.
5. **Monitoring & Incident Management:**
Set up monitoring systems to track application performance and troubleshoot issues in real time. It also ensures that systems are resilient & downtime is minimized.
6. **Cloud & Infrastructure Management:**
Deploy, manage & optimize applications on cloud platform like AWS, Azure or Google Cloud. It 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 configuration & 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 of performance.
5. Support & Troubleshooting:
Respond to incidents & resolve production issues promptly & identify root causes of failure & implement fixes.
6. Documentation & Reporting:
Document system configurations, deployment processes & troubleshooting guides.