

Vanshel S. Chavan

T31 - 19

23/01/2025



## SEPM

To understand Dev Ops, principles, practice & Dev Ops Engineer Role & Responsibilities.

What is Dev Ops?

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, to accelerate fast feedback & iterative improvement. Built on agile methodology, DevOps creates a culture of accountability, collaboration & shared responsibility for business outcomes.

Core Principles of Dev Ops.

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

Key Practices of Dev Ops.

### 1. Continuous Develop. 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.

### 2. Continuous Development:

This is the phase that involves planning & coding, versioning & managing builds of the software application functionality. Eg - Git, GitHub, Maven..



3. Continuous Testing  
Continuous testing is executing automated test, continuously & repeated against, the code base & the various deployment environment. 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. Infrastructure Management  
Without automation, building & maintaining large-scale modern without automation. IT system can be a resource intensive undertaking & can lead to increased risk due to manual errors. Configuration & resource management is an automated method for maintaining computer system & 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 the simple textual format that can be stored in your version control system (VCS)



versioned on request. These text files are called manifest are used by DevOps tools to automatically provision & configure build servers, testing, staging & production environment.  
Eg:- Chef, saltStack

### Dev Ops Engineer Role.

A DevOps engineer manages a company i.e IT infrastructure bringing 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 repetitive task like testing, deployment & monitoring.

##### 3] Continuous Integration & Continuous Delivery (CI/CD).

Design, implement & maintain CI/CD pipeline to enable faster, reliable & repeatable 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 & troubleshoot issue in real time. It also ensures that system are resilient & down time is minized.



## 6. Cloud & Infrastructure Management

Deploy, manage & optimize application on cloud platform like A.W.S, Azure or Google Colab, also handles container orchestration.

### Key Responsibilities.

#### 1] Collaboration & Planning

Work with development & operation teams to plan & design scalable solution.

#### 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 & measure performance.

#### 5] Support & Troubleshooting

Respond to incident & resolve production issues promptly & identify root causes of failure & implement fixes.

#### 6] Documentation & Reporting

Document system configuration, deployment, processes & troubleshooting guides.