

Experiment No-1

Date of Performance:

Date of Submission:

Aim: To understand DevOps: Principles, Practices, and DevOps Engineer Role and Responsibilities

Theory: DevOps stands for development and operations. It's a practice that aims at merging development, quality assurance, and operations (deployment and integration) into a single, continuous set of processes. This methodology is a natural extension of Agile and continuous delivery approaches.

DevOps principles

1. Collaboration

The key premise behind DevOps is collaboration. Development and operations teams coalesce into a functional team that communicates, shares feedback, and collaborates throughout the entire development and deployment cycle. Often, this means development and operations teams merge into a single team that works across the entire application lifecycle.

2. Automation

An essential practice of DevOps is to automate as much of the software development lifecycle as possible. This gives developers more time to write code and develop new features. Automation is a key element of a CI/CD pipeline and helps to reduce human errors and increase team productivity. With automated processes, teams achieve continuous improvement with short iteration times, which allows them to quickly respond to customer feedback.

3. Continuous Improvement

Continuous improvement was established as a staple of agile practices, as well as lean manufacturing and Improvement Kata. It's the practice of focusing on experimentation, minimizing waste, and optimizing for speed, cost, and ease of delivery. Continuous improvement is also tied to continuous delivery, allowing DevOps teams to continuously push updates that improve the efficiency of software systems.

3. Customer-centric action

DevOps teams use short feedback loops with customers and end users to develop products and services centered around user needs. DevOps practices enable rapid

collection and response to user feedback through use of real-time live monitoring and rapid deployment.

4. Create with the end in mind

This principle involves understanding the needs of customers and creating products or services that solve real problems. Teams shouldn't 'build in a bubble', or create software based on assumptions about how consumers will use the software. Rather, DevOps teams should have a holistic understanding of the product, from creation to implementation.

DevOps Practices:

DevOps practices reflect the idea of continuous improvement and automation. Many practices focus on one or more development cycle phases. These practices include:

1.Continuous development:-This practice spans the planning and coding phases of the DevOps lifecycle. Version-control mechanisms might be involved.

2.Continuous testing:-__This practice incorporates automated, prescheduled, continued code tests as application code is being written or updated. Such tests can speed the delivery of code to production.

3.Continuous integration (CI):-This practice brings configuration management (CM) tools together with other test and development tools to track how much of the code being developed is ready for production. It involves rapid feedback between testing and development to quickly identify and resolve code issues.

4.Continuous delivery:- This practice automates the delivery of code changes, after testing, to a preproduction or staging environment. An staff member might then decide to promote such code changes into production.

5.Continuous deployment (CD):- Similar to continuous delivery, this practice automates the release of new or changed code into production. A company doing continuous deployment might release code or feature changes several times per day. The use of container technologies, such as Docker and Kubernetes, can enable continuous deployment by helping to maintain consistency of the code across different deployment platforms and environments.

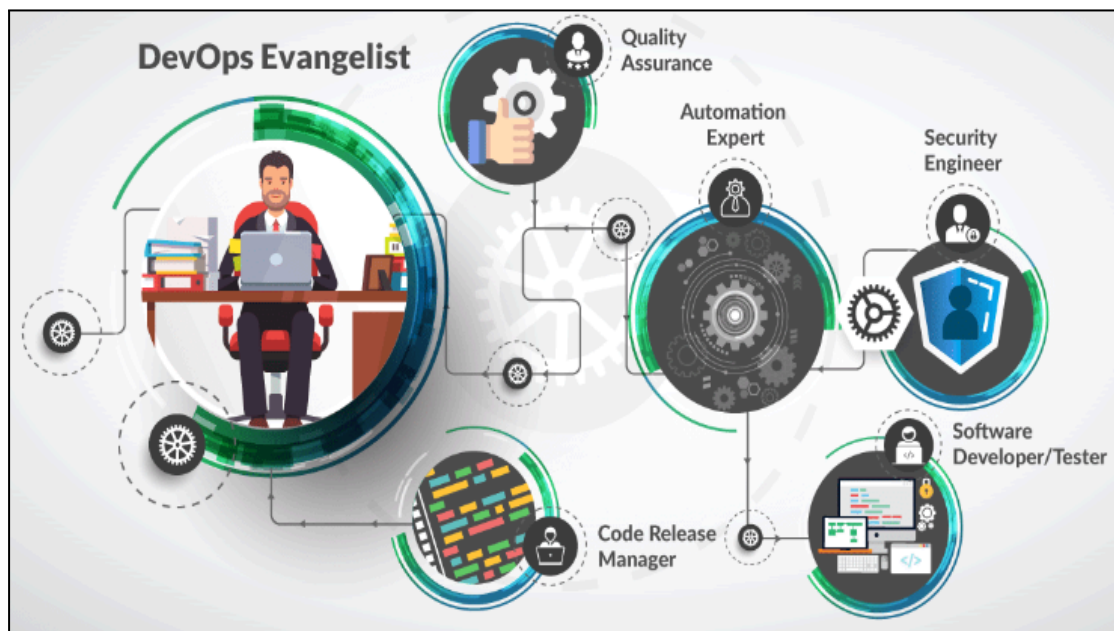
6.Continuous monitoring:- This practice involves ongoing monitoring of both the code in operation and the underlying infrastructure that supports it. A feedback loop that reports on bugs or issues then makes its way back to development.

7.Infrastructure as code:- This practice can be used during various DevOps phases to automate the provisioning of infrastructure required for a software release. Developers add infrastructure "code" from within their existing development tools. For example, developers might create a storage volume on demand from Docker, Kubernetes, or OpenShift. This practice also allows operations teams to monitor environment configurations, track changes, and simplify the rollback of configurations.

DevOps Engineer Roles and Responsibilities:

1)DevOps Engineer Roles:

- 1.DevOps Evangelist – The principal officer (leader) responsible for implementing DevOps
- 2.Release Manager – The one releasing new features & ensuring post-release product stability
- 3.Automation Expert – The guy responsible for achieving automation & orchestration of tools
- 4.Software Developer/ Tester – The one who actually develops the code and tests it
- 5.Quality Assurance – The one who ensures quality of the product confirms to its requirement
- 6.Security Engineer – The one always monitoring the product's security & health



2)DevOps Engineer Responsibilities:

A DevOps engineer works with diverse teams and departments to create and implement software systems. People who work in DevOps are experienced IT professionals who collaborate with software developers, quality assurance professionals, and IT staff to manage code releases.

DevOps engineers need to be able to multitask, demonstrate flexibility, and deal with many different situations at a time. Specifically, a DevOps engineer's responsibilities include:

1.Documentation:- Writes specifications and documentation for the server-side features.

2.Systems analysis:- Analyzes the technology currently being used and develops plans and processes for improvement and expansion. The DevOps engineer provides support for urgent analytic needs.

3.Development:- Develops, codes, builds, installs, configures, and maintains IT

SIGN AND REMARK:

DATE:

R1 (3 Marks)	R2 (5 Marks)	R3 (4 Marks)	R4 (3 Marks)	Total (15 Marks)