Name: Mrunal Vaidya XIE ID: 202003060
Roll : 68 Batch: C

Experiment 1

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

Theory:

What is DevOps?

DevOps is a combination of software developers (dev) and operations (ops). It is defined as a software engineering methodology which aims to integrate the work of software development and software operations teams by facilitating a culture of collaboration and shared responsibility.

DevOps can be best explained as people working together to conceive, build and deliver secure software at top speed. DevOps practices enable software developers (devs) and operations (ops) teams to accelerate delivery through automation, collaboration, fast feedback, and iterative improvement.

DevOps Principles

The DevOps methodology comprises four key principles that guide the effectiveness and efficiency of application development and deployment. These principles, listed below, center on the best aspects of modern software development.

1. Automation of the software development lifecycle

   The DevOps team should automate as much of the process as possible. Automation can benefit operations like Continuous Integration/Continuous Delivery (CI/CD), infrastructure provisioning, security compliance verification tests, functionality tests, and software deployment. The more you can automate processes, the safer, faster, and more reliable the product release.

2. Collaboration and communication

   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.

3. Continuous improvement and minimization of waste

   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. The constant pipeline of new releases means teams consistently push code changes that eliminate waste, improve development efficiency, and bring more customer value.

4. Hyperfocus on user needs with short feedback loops

   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. Teams get immediate visibility into how live users interact with a software system and use that insight to develop further improvements.

By adopting these principles, organisations can improve code quality, achieve a faster time to market, and engage in better application planning.

DevOps Practices

1. Gain the stakeholders' active participation as soon as possible
2. Developers and testers should test code often and early, using automated testing
3. Bring change management to every stage of the project, exposing a larger audience to enterprise-level issues
4. Make sure users have development support after you release new builds
5. Define your integrated deployment best practices and implement them across the internal and external communities
6. Keep code repositories regularly updated and ensure that updates are continuously integrated into the workflow
7. Build, test, and release code faster via continuous delivery
8. Build system-wide structures to facilitate configuration management, consolidating operations, and adding visibility to Information Technology leaders
9. Quickly bring in new features by taking advantage of continuous deployment tools
10. Automate dashboards to permit team members and leaders to spot bottlenecks and further investigate roadblocks quickly
11. Ensure applications have sufficient monitoring, usually automated so that development teams can quickly identify production code difficulties

DevOps Engineer Role and Responsibilities

DevOps Engineer is somebody who understands the Software Development Lifecycle and has the outright understanding of various automation tools for developing digital pipelines (CI/ CD pipelines).

Roles of DevOps Engineer:

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

Responsibilities of DevOps Engineer:

- building and setting up new development tools and infrastructure
- understanding the needs of stakeholders and conveying this to developers
- working on ways to automate and improve development and release processes
- testing and examining code written by others and analysing results
- ensuring that systems are safe and secure against cybersecurity threats
- identifying technical problems and developing software updates and 'fixes'
- working with software developers and software engineers to ensure that development follows established processes and works as intended
- planning out projects and being involved in project management decisions.

Tools

- Version Control Tool: Git (GitHub)
- Build Tool: Maven
- Continuous Integration Tool: Jenkins
- Configuration Management Tool: Ansible
- Container Platforms: Docker
- Communication and Collaboration: Slack
- Testing Tool: Selenium
- Continuous Monitoring: Nagios
- Cloud Computing and Storage: AWS

Conclusion:

From the above Experiment, the concept of DevOps and DevOps Engineers along with their roles and responsibilities understood clearly.And hence, with this experiment we have achieved the Lab Outcome One (LO1).

POs Achieved: PO1, PO4