MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI

Experiment 1

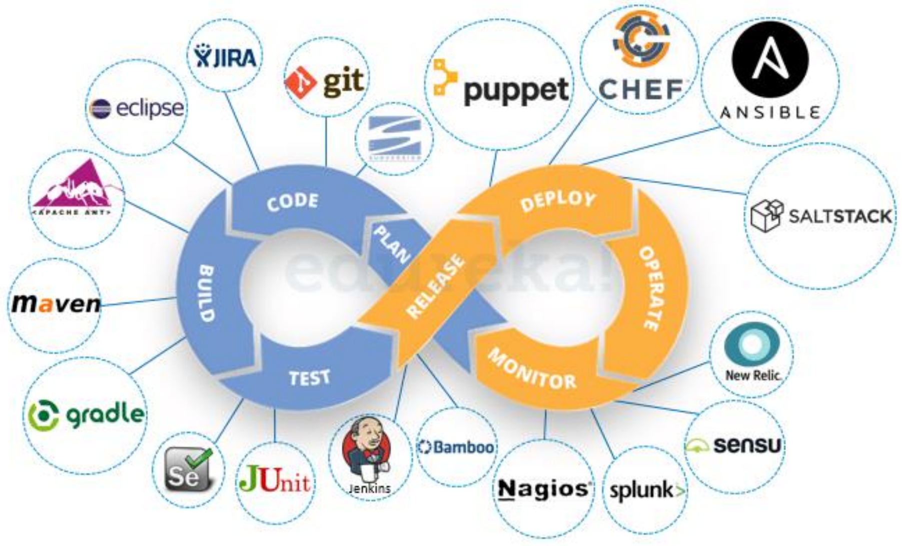
Aim: To understand DevOps, principles, practices, and DevOps engineer roles and responsibilities.

Theory:

DevOps:

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams. It emphasizes team empowerment, cross-team communication and collaboration, and technology automation.

The DevOps movement began around 2007 when the software development and IT operations communities raised concerns about the traditional software development model, where developers who wrote code worked apart from operations who deployed and supported the code. The term DevOps, a combination of the words development and operations, reflects the process of integrating these disciplines into one, continuous process.



1.1 DevOps lifecycle

Phases:

- **Plan:** Professionals determine the commercial need and gather end-user opinions throughout this level. In this step, they design a project plan to optimize business impact and produce the intended result.

- **Code:** During this point, the code is being developed. To simplify the design process, the developer team employs lifecycle DevOps tools and extensions like Git that assist them in preventing safety problems and bad coding standards.
- **Build:** After programmers have completed their tasks, they use tools such as Maven and Gradle to submit the code to the common code source.
- **Test:** To assure software integrity, the product is first delivered to the test platform to execute various sorts of screening such as user acceptability testing, safety testing, integration checking, speed testing, and so on, utilizing tools such as JUnit, Selenium, etc.
- **Release:** At this point, the build is prepared to be deployed in the operational environment. The DevOps department prepares updates or sends several versions to production when the build satisfies all checks based on the organizational demands.
- **Deploy:** At this point, Infrastructure-as-Code assists in creating the operational infrastructure and subsequently publishes the build using various DevOps lifecycle tools.
- **Operate:** This version is now convenient for users to utilize. With tools including Chef, the management department take care of server configuration and deployment at this point.
- **Monitor:** The DevOps workflow is observed at this level depending on data gathered from consumer behavior, application efficiency, and other sources. The ability to observe the complete surroundings aids teams in identifying bottlenecks affecting the production and operations teams’ performance.

DevOps principle:

1. **Collaboration**

DevOps refers to the integration of the operation and the development of teams which means that collaboration is the major part of DevOps. Therefore by working together, the development team can better configure the software for the operating phase and the operations to test the software to make sure that it fulfills the needs.

2. **Customer-Centric Decision Making**

Customer-centric decision-making is an important DevOps principle that is mainly focused on the DevOps lifecycle and it is equally important as data and decisions should be made in such a way that the question should come to mind “Will this benefit the customer or their needs”. Therefore gathering feedback from the customers on the already existing products will further guide the future optimization.

3. **Making Data Based Decision**

Data-based decision-making is another DevOps principle which is used for informing decisions with data. Therefore whether selecting the right tech stack or selecting the various types of tools to streamline the pipeline, the developers should always collect the data all around each decision to make sure that the choices agree with the historical data and team metrics

4. **Automation**

The major benefit of the DevOps approach is the speed of the software delivery and the speed of the patches. This momentum is mainly achieved with automation. The DevOps team has an objective to automate every single phase of the process from the code reviews to handoff to provisioning and deployment.

5. **Regular Improvement**

The process of DevOps mainly focuses on a regular improvement and the development team should regularly focus on the new upgrades and features. The idea of Agile methodology mainly focuses on incremental releases.

6. **Failure as a Learning Opportunity**

The process of development is getting better day by day as the software itself is continuously improving. Thus the part of maintaining flexibility is to view the failure as an opportunity to learn and improve instead of trying to avoid the failure at any cost by encouraging risk-taking in the right context.

7. **Responsibility Throughout the Lifecycle**

Earlier in the past the software development models, the development team coded and developed the applications. Then they hand it to the operations teams to test, deploy, and deliver to the customer.

DevOps practices:

1. **Developers Should Be Involved in Operations**

When developers are involved in Operations as part of a holistic and integrated team, they step out of their enclosed development environments and into the real world of production systems where their code is designated to run.

2. **Operations Should Be Involved in Development**

When Operations are aware of code changes that are planned for development, they can anticipate how they might affect production infrastructure and ensure that developers consider factors such as hardware constraints, monitoring, deployment, troubleshooting, security and tooling needed for production systems.

3. **Version Control Everything**

Using version control systems for source code is widely accepted as a standard practice. In a DevOps environment the concept of “code” is extended to everything that is involved in production systems. So in addition to source code, you should version control software and hardware configuration files, settings, parameters and anything else that take part in your runtime systems.

4. **Be Agile With Changes in Infrastructure**

Using Agile methodologies for software development is commonplace, but they should also be used for infrastructure. Infrastructure should be treated and managed as code. Which means, changes should be versioned and applied in small, discrete steps, and at each step, your systems should be tested to ensure nothing has broken.

5. **Automate Everything**

Manual tasks and processes are error-prone and not scalable. In a DevOps environment, every change should undergo automation as part of a CI/CD pipeline. Everything that can be automated, should be automated. This includes automated deployment processes, automated testing procedures, and more.

6. **Continuous Integration/Delivery/Deployment**

A change at any stage of the software delivery pipeline, from development to production systems, should go through a CI/CD system. This ensures that if anything fails, there is a fast feedback loop leading to rapid recovery.

7. **Unified Tools and Platforms**

In DevOps, development, staging and production environments should use the same tools, configuration and hardware resources to the greatest extent possible. This is to ensure that anything that works in development will transition successfully to the staging and production systems.

DevOps roles and responsibilities:

- **Collaboration**: DevOps Engineers foster collaboration between development and operations teams, breaking down silos and enabling effective communication and cooperation.
- **Automation**: They automate processes and workflows, utilizing tools and frameworks to streamline build, test, deployment, and monitoring processes. This includes configuration management, infrastructure provisioning, and code deployment automation.
- **Continuous Integration and Delivery**: DevOps Engineers implement and maintain continuous integration and delivery pipelines, enabling frequent software releases while ensuring quality, stability, and scalability.
- **Infrastructure Management**: They optimize infrastructure, including cloud platforms, servers, networking, and storage. DevOps Engineers ensure the infrastructure's scalability, availability, and security through infrastructure-as-code practices.
- **Monitoring and Troubleshooting**: They implement monitoring solutions and establish proactive alerts to identify and address issues promptly. DevOps Engineers possess strong troubleshooting skills to resolve incidents and minimize downtime.

Conclusioin: We have successfully understood DevOps, principles, practices, and DevOps engineer roles and responsibilities.