Assignment 1

Aim: To understand DevOps: principles, practices & DevOps engineer role

Lo Mapped: LO1

Theory:

1. Principles of DevOps

DevOps workflow provides a visual overview of the sequence in which input is provided. Also, it tells about which one action is performed, and output is generated for an operations process.

- **End to End Responsibility:** DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.
- **Continuous Improvement:** DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.
- **Automate Everything:** Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.
- **Custom Centric Action:** DevOps team must take customer-centric for that they should continuously invest in products and services.
- **Monitor and test everything:** The DevOps team needs to have robust monitoring and testing procedures.
- **Work as one team:** In the DevOps culture role of the designers, developers, and testers are already defined. All they needed to do is work as one team with complete collaboration.

2. DevOps Practices

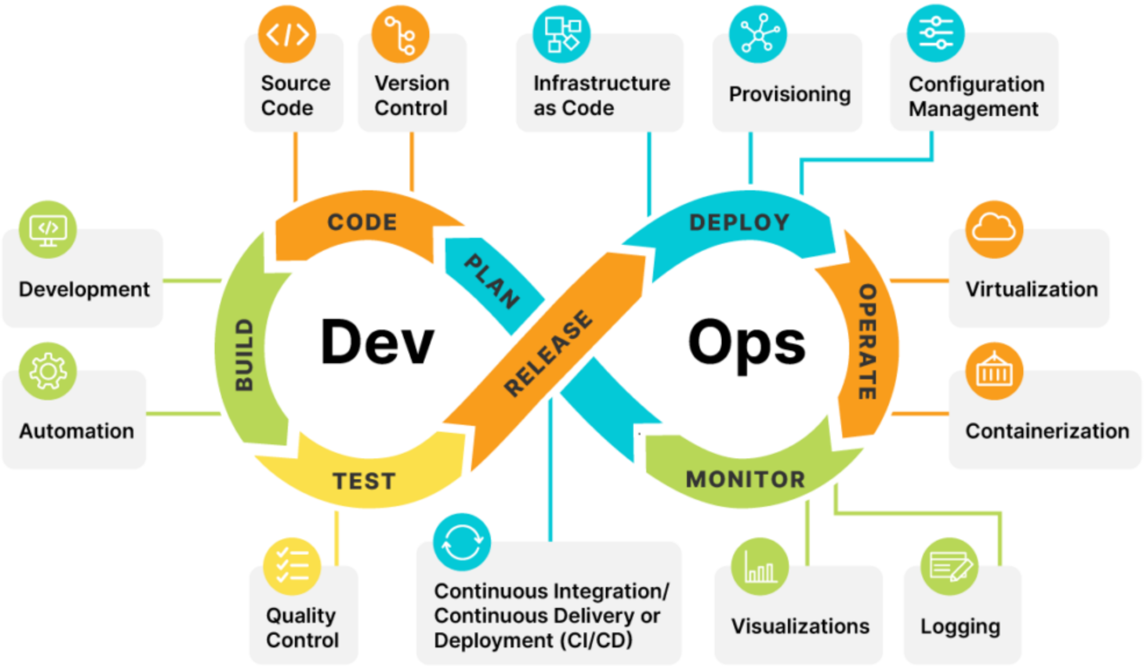
DevOps practices reflect the idea of continuous improvement and automation. Many practices focus on one or more development cycle phases. These practices include:

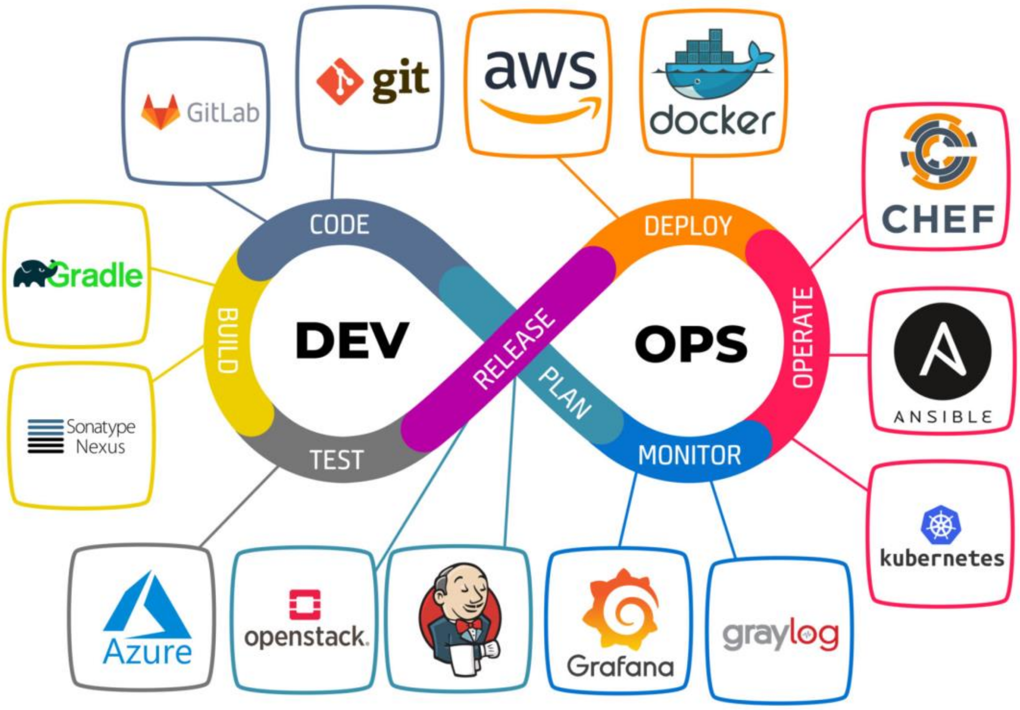
- **Continuous development:** This practice spans the planning and coding phases of the DevOps lifecycle. Version-control mechanisms might be involved.
- **Continuous testing:** This practice incorporates automated, prescheduled, continued code tests as application code is being written or updated. Such tests can speed the delivery of code to production.
- **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.

- **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.
- **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.
- **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.
- **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.

3. Role of DevOps Engineer

- DevOps engineers reduce that complexity, closing the gap between actions needed to quickly change an application, and the tasks that maintain its reliability.
- Development teams and IT operations teams can have different skills and different goals. Developers want to introduce new features to an application, while operations teams want to preserve the stability of an application once it is released.
- DevOps is all about the unification and automation of processes, and DevOps engineers are instrumental in combining code, application maintenance, and application management. All of these tasks rely on understanding not only development life cycles, but DevOps culture, and its philosophy, practices, and tools.
- Within an agile environment, developers, system administrators, and programmers can be siloed, working on the same product but not sharing information necessary to ensure value to the user.





Conclusion:

In this assignment, we understood the principles of DevOps as well as the various practices and the role and responsibilities of a DevOps engineer.