**What is DevOps?**

DevOps is a culture, process, methodology to software development that prioritizes collaboration, communication, automation, and continuous improvement with the help of faster feedback loops for more frequent delivery of updates and features. The DevOps model helps organizations to overcome the gap between development and operations teams, streamline the software development process, and deliver software faster and with higher quality.

DevOps practices include continuous integration, continuous delivery, and continuous deployment, which are automated processes that help teams deliver code changes more frequently and reliably. DevOps methodologies generally emphasize the use of infrastructure as code where managing infrastructure through code instead of manual processes, monitoring and logging are essential for identifying and resolving issues quickly.

**Why DevOps is important**

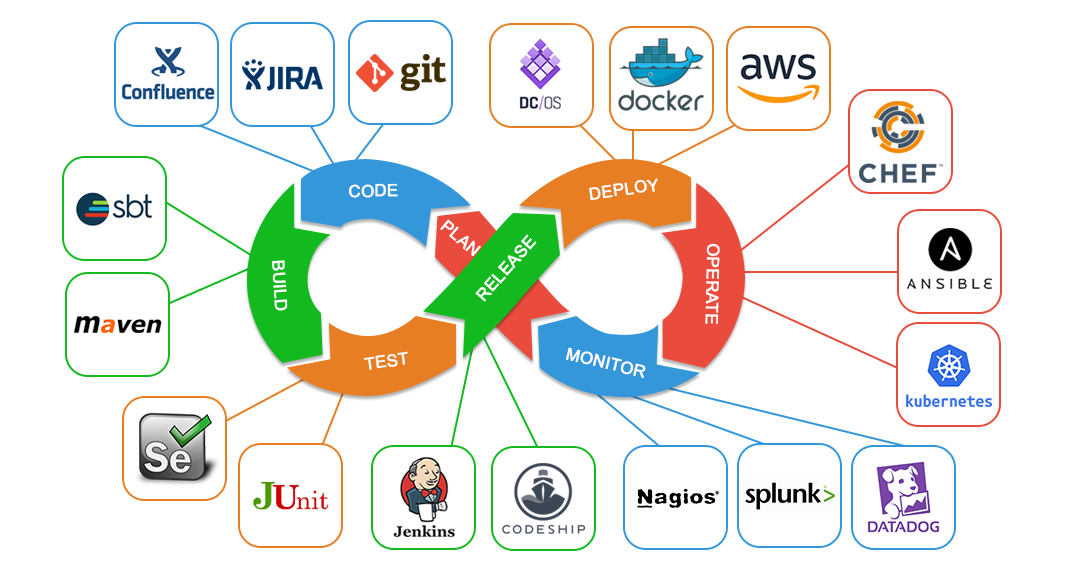
The goal of DevOps is to provide a culture of collaboration and communication between software development teams and IT operations teas, enabling organizations to deliver software faster, with greater efficiency and quality, and with reduced time to market. It encourages automation and continuous improvement to ensure that code changes can be tested and deployed quickly and reliably.

**Here are the benefits of the DevOps approach:**

* **Faster delivery:** DevOps can help organizations to deliver new software and updates faster
* **Improved Quality:** DevOps can help to improve the quality of software by catching bugs early in the development process. This can lead to fewer customer support issues and increased customer satisfaction.
* **Reduced Cost:** DevOps can help to reduce costs by streamlining the software delivery process and eliminating waste.
* **Better collaboration and communication:** DevOps focuses on breaking down silos between development and operations teams. By promoting a culture of collaboration and communication, DevOps helps to align teams towards common goals, improving morale and reducing frustration.

**High Level overview of the DevOps infinity loop with tools**

A DevOps pipeline for continuous integration and continuous delivery consists of several stages that help automate the process of building, testing, and deploying software.



**Development or Integration Stages**

**Plan**: Planning your project, technology, environment, structure, and architecture creates a roadmap to successfully reaching your project goals. Planning is also the stage where you want to decide what software and tools you will use — more on this later.

**Code**: In this stage, we start writing code for the project. Effectively, we're getting ready to build a testable product. However, since writing code can take a lot of time, this is a prime opportunity to maximize automation tools.

**Build**: In the build stage, we take the provided code and build it for testing purposes. The code is built in a development environment to allow testing and bug fixes.

**Test:** Automated testing ensures the project is functioning as expected and finds any bugs or issues in behaviour. Depending on your team workflow, UI/UX or performance testing will also happen in this stage.

**Operations or Deployment Stages**

**Release:** The release stage is when the Ops team confirms that the project is ready to be released and builds it into the production environment. This stage is critical as it is the last stop after multiple stages for checks — like vulnerabilities and bugs — just before deployment.

**Deploy:** Deployment is the stage where we move the project — in its current state — to the production environment for the end-users to access. This stage is where approved changes get deployed to the user.

**Operate:** In the Operate stage, the operations team will configure and manage the project in the production environment. Typically, the team will rely on automation to help maintain the DevOps project in this stage.

**Monitor:** In the Monitor stage, the project is being used. From the recorded results, teams will gain more insight into behaviour, user response, and the general success of the product.