# CI/CD

# CI/CD

## Continuous Integration (CI)

### Definition

Software development practice where members of a team integrate their work frequently by committing code changes back into the main/master branch.

### Frequency

Each member integrates at least daily, leading to multiple integrations per day.

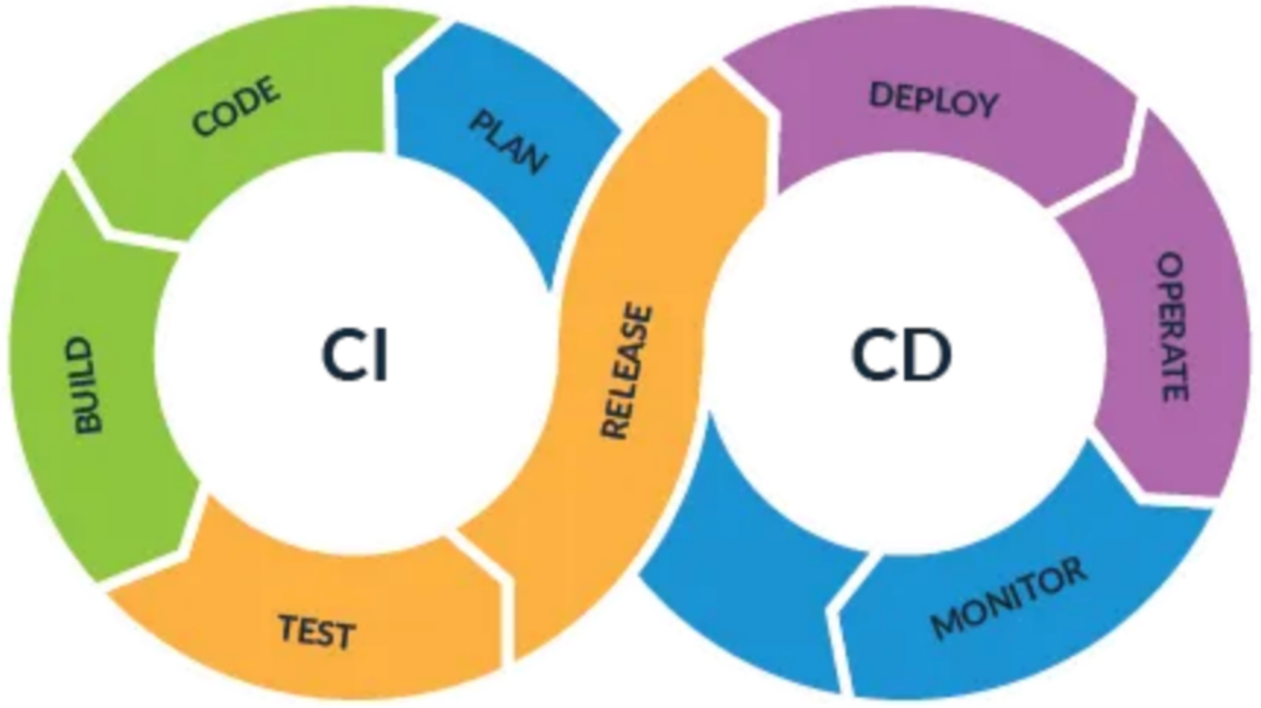
### Automation

Each integration is verified by an automatic build and tests to detect errors or bugs as quickly as possible.

# CI Main Goals

- Improve software quality.
- Reduce time to deliver software.
- Ensure code in the shared repository is always in a releasable state.

# Plan - Code - Build - Test



**Plan:** The process begins with planning the features and improvements that need to be developed. This stage involves understanding the requirements and setting the necessary tasks.

**Code:** Developers write code to add new features or fix bugs based on planned tasks. The code should be self-explaining, well-documented, and follow the best practices and coding standards.

**Build:** The code is compiled, dependencies are managed, and it is transformed into executable code. In interpreted languages like Python or JavaScript, this may equate to a syntax check or running automated style checkers, etc.

**Test:** Automated tests are then run against the build to ensure that any new changes have not broken existing functionality and that new functionality works as expected.

**Summary:** The process involves planning tasks, coding features or fixes, transforming this code into an executable build, and running automated tests to ensure functionality.

# CI Demonstration

https://github.com/dailydoseofdevops/ci-calculator