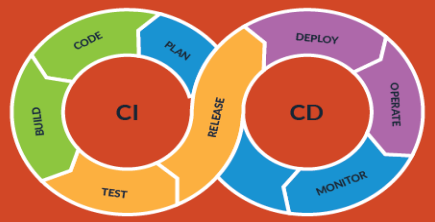
Continuous Integration and Continuous Deployment

Fundamentals and benefits of CI/CD to achieve, build, and deploy automation for cloud-based software products.



Agenda

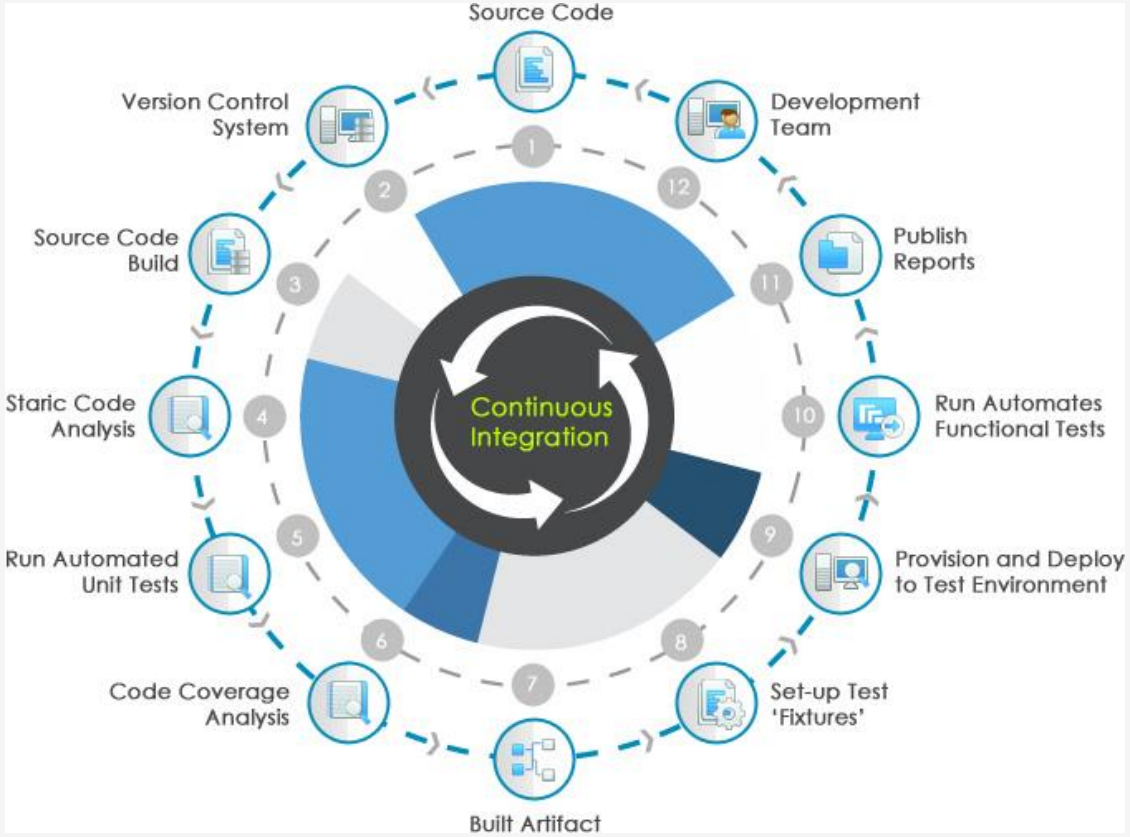
- CI/CD Overview
  - Continuous Integration (CI)
  - Continuous Deployment (CD)
- Benefits of CI/CD

CI/CD Overview

Continuous Integration (CI)

The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "**Making**". Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact! Some common CI-related phases might include:

- Compile
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact

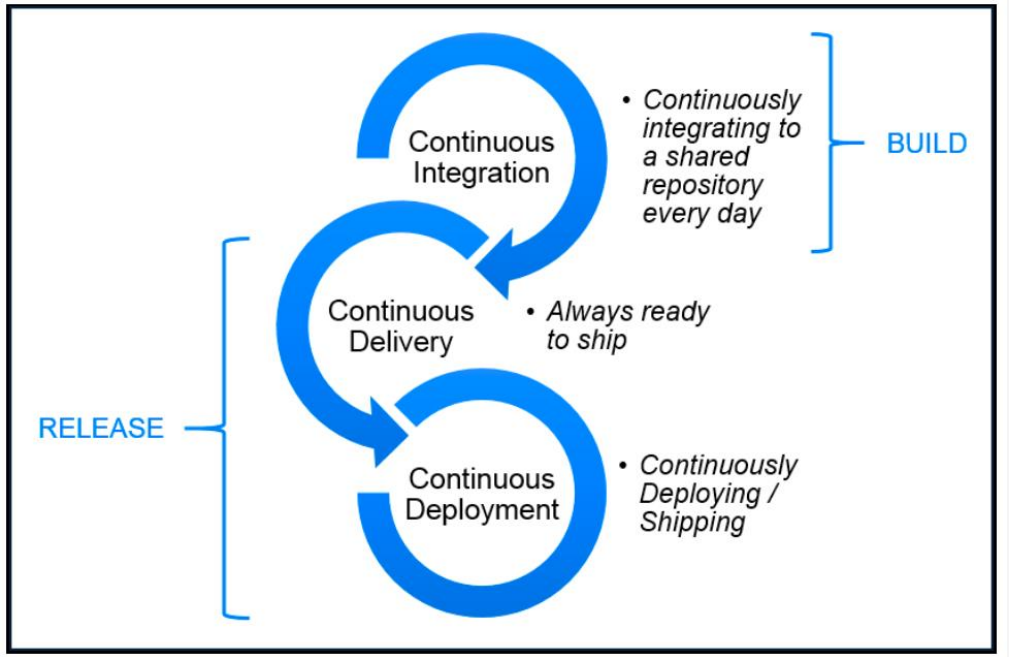


CI/CD Overview

Continuous Deployment (CD)

A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. It's the process of "**Moving**" the artifact from the shelf to the spotlight. Some common CD-related phases might include:

- Creating infrastructure
- Provisioning servers
- Copying files
- Promoting to production
- Smoke Testing (aka Verify)
- Rollbacks



Benefits of CI/CD

Benefits of CI/CD

1

Automate Infrastructure Creation and clean up:

Eliminating human errors and avoid **unnecessary cost of unused or invalid infrastructure**.

2

Faster to production:

By automating the pipeline to production this way we can deploy features as soon as created which will help **increase revenue**.

3

Automated Rollback Triggered by Job Failure:

Automate the process of rolling back and cleaning any infrastructure left which would help in **reducing cost and lower down time**.

4

Catch Compile Errors After Merge:

Discover errors as soon as the developer make his commit which will help **reduce the time** of developers and **reduce cost**.

5

Catch Unit Test Failures:

Unit tests are not neglected with CICD which will **increase code quality** and catch errors early before production which would **decrease cost**.

6

Automated Smoke Tests:

Automate smoke test after deployment and automatic rollback in case of failure which will **decrease downtime** and **reduce cost**.