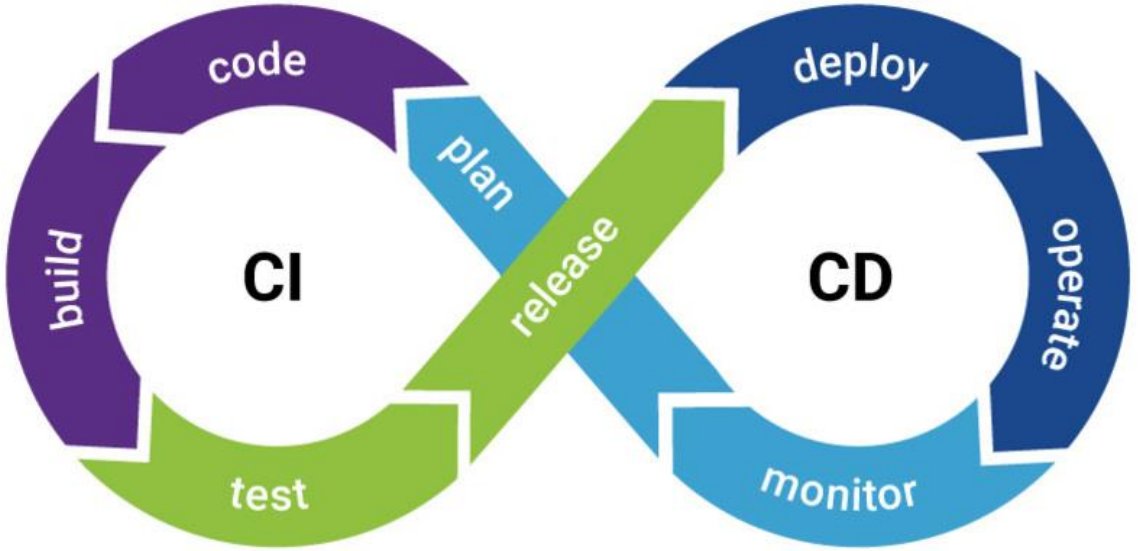
Continuous Integration Continuous Deployment

Fundamentals and benefits of CI/CD to build, deploy application automatedly

Agenda

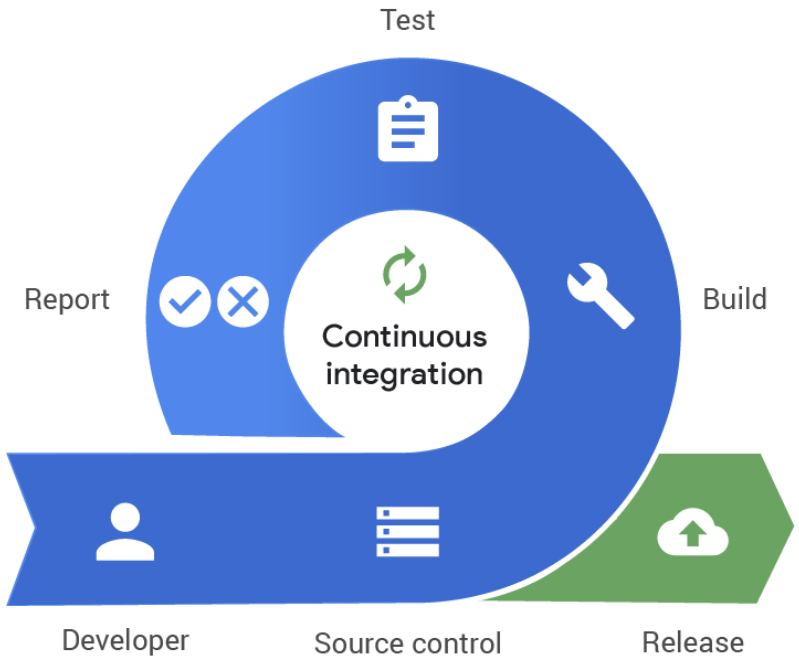
- **CI/CD Fundamentals**
- **Benefits of CI/CD**



Continuous Integration

*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



Continuous Deployment

*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

Increase Revenue

Avoid costs

Reduce costs

Benefits of CI/CD

- **Faster time and more frequent to market**
  - Increase valuable features, **Increase revenue**
- **Automate Infrastructure cleanup**
  - Less cost from unused infrastructure, **Reduce costs**
- **Automate Infrastructure creation**
  - Less human error, faster deployments**, Avoid cost**
- **Catch Unit Test Failures**
  - Less bugs, **Avoid cost**
- **Detect security vulnerabilities**
  - Prevent security holes, **Avoid cost if issues**
- **Automate Smoke Test**
  - Reduce downtime from failure deployment, **Avoid cost**
- **Automate Rollback Failure**
  - Quick undo Production to working state**, Avoid cost**
- **Catch Compile Errors From Merges**
  - Less time spent on issues, **Reduce cost**