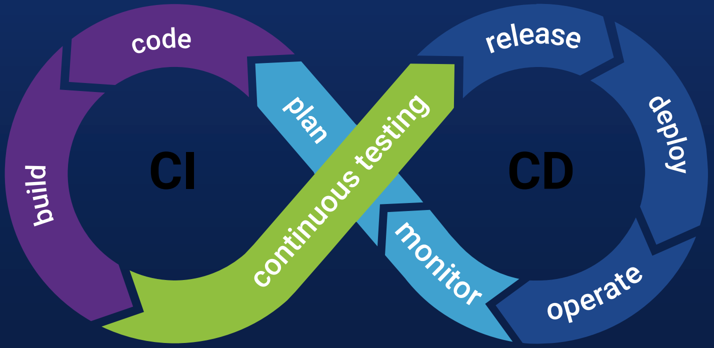
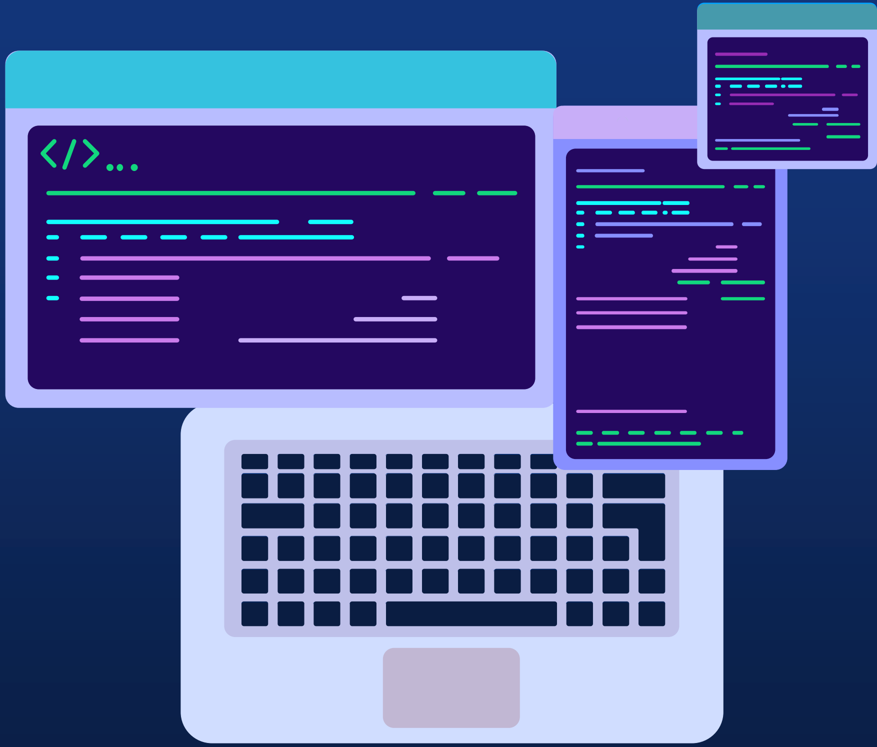
Why We Should Switch To CI/CD

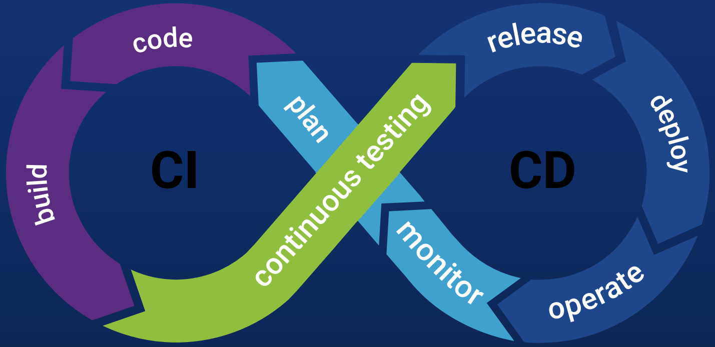




CI (Continuous Integration)

The practice of merging all developers' working copies to a shared mainline several times a day

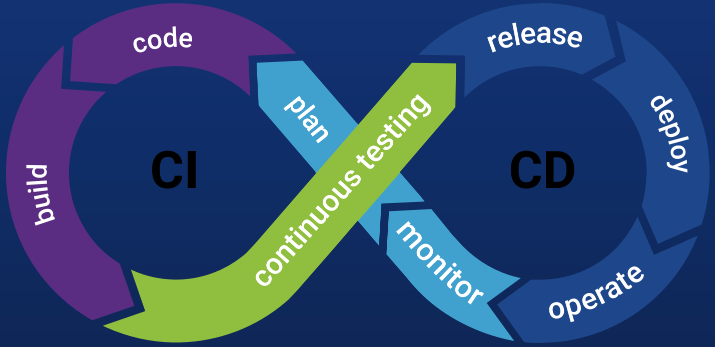
- Compile
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact



CD (Continuous Delivery/Deployment)

A software engineering approach in which the value is delivered frequently through automated deployments

- Creating infrastructure
- Provisioning servers
- Copying files
- Promoting to production
- Smoke Testing
- Rollbacks



Main Benefits Of CI/CD

01
Catch Compile Errors After Merge

Less developer time on issues from new developer code

02
Catch Unit Test Failures

Less bugs in production and less time in testing

03
Detect Security Vulnerabilities

Prevent embarrassing or costly security holes

04
Automate Infrastructure Creation

Less human error, Faster deployments

05
Automate Infrastructure Cleanup

Less infrastructure costs from unused resources

06
Faster and More Frequent Production Deployments

New value-generating features released more quickly

Main Benefits Of CI/CD

07

Deploy to Production Without Manual Checks

Less time to market

08

Automated Smoke Tests

Reduced downtime from a deploy-related crash or major bug

09

Automated Rollback Triggered by Job Failure

Quick undo to return production to working state