**Fundamentals and Benefits of CI/CD**

**Continuous Integration (CI)** - it’s the practice of merging all developers' working copies to a shared mainline several times a day. Everything related to the code fits here.

Some common CI-related phases might include:

• Compile

• Unit Test

• Static Analysis

• Dependency vulnerability testing

• Store artifact

**Continuous Deployment (CD) –** it’s a software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here.

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**

* Help catch compile errors after merge thus less developer time on issues from new developer code, this helps reduce cost.
* Helps catch unit test failures thus less bugs in production and less time in testing, this will help reduce cost.
* Helps detect security vulnerabilities thus prevent embarrassing or costly security holes, this will help avoid cost.
* Helps automate infrastructure creation thus less human error and faster deployments, this will help avoid cost.
* Automate infrastructure cleanup thus less infrastructure costs from unused resources, this will help reduce cost.
* Faster and more frequent production deployments thus new value-generating features released more quickly, this will help increase revenue.
* Deploy to production without manual checks thus less time to market, this will help increase revenue.
* Automated smoke tests thus reduced downtime from a deploy-related crash or major bug, this will help protect revenue.
* Automated rollback triggered by job failure thus quick undo to return production to working state, this will help protect revenue.