CI/CD is short for Continuous Integration and/ Continuous Deployment.
Before the advent of CI/CD, everything required human intervention, which was error prone, and unreliable.
With CI/CD however, we can drastically reduce the frequency of things going wrong, and focus more on efficiency and delivering quality software, products and services.

Continuous Integration is the practice of merging all developers' working copies to a shared mainline several times a day.
Some common CI-related phases might include:

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

Benefits of CI:

- Reduces the cost of catching compile errors after Merge
- Less bugs in production and less time in testing help avoid/reduce cost
- Detecting security vulnerabilities on time help reduce cost

Whereas, **Continuous Deployment** refers to a software engineering approach in which value is delivered frequently through automated deployments.
Some common CD-related phases might include:

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

Benefits of CD:

- Less developer time on issues from new developer code by automating Infrastructure creation can help avoid unnecessary
- Automation of infrastructure cleanups can also help reduce cost as unused resources get cleaned up automatically
- Faster and more frequent production deployments can boost revenue
- Deploying to production without manual checks can facilitate increase in revenue

Overall, CI/CD can help create and protect revenue, and also control and reduce costs.