Fundamentals of CI/CD

- **Continuous integration** means that developers frequently merge their code changes to a shared repository. It's an automated process that allows multiple developers to contribute software com ponents to the same project without integration conflicts. CI involves automated testing whenev er a software change is integrated into the repository. *Some common CI-related phases might inc lude:* compile, unit Test, static Analysis, dependency vulnerability testing, store artifact.

- **Continuous Deployment** means that *a software engineering approach in which the value is deliv ered 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-r elated phases might include: c*reating infrastructure, provisioning servers, promoting to producti on, smoke Testing (aka Verify), rollbacks.

- **A CI/CD pipeline** is a set of data processing elements connected in series, where the output of o ne element is the input of the next one. In the context of software development, a CI/CD pipelin e is a series of automated steps that allow developers to quickly and safely release new code cha nges to production.

Benefits of CI/CD

Technical terms	Translation in business
- Catch Compile Errors After Merge	- Reduce Cost. Since the error will almost immediately catch after the developer finishes their job of writing the code and storing it in the code repository.
- Smoke Tests	- Protecting revenue. Think about a product, which has a lot of potential errors that never be tested. What do we need to do when releasing that product? The answer is making sure it will be tested before the release happens, by a full-time job QA or automated test system. And what happens if the tester misses some important test due to human errors, which cannot be pre vented every single time, especially when the task is complex a nd must be done repeatedly? I can see a severe bug occurring on the production site because of human mistake during the te st, causing the server to go offline for at least a few minutes to an hour or possibly a day to fix the issue. The solution to this pr oblem will be an automated test system.

Benefits of CI/CD

Technical terms	Translation in business
- Automate Infrastructure Cleanup	- Reducing cost. By manually removing the infrastructure, we c an easily forget one or more things that need to be destroyed to avoid the extra cost. Within CI/CD, automation clean-up will take the best efficiency. Every single piece will be cleaned immediately after the test is passed. No more extra time for un used infrastructure, no more wasting money
- Faster and More Frequent Production Deployments	- Increasing revenue. The faster deployment of the product, the more quickly feedback from customers. It is very meaningful. Within the CI/CD, we can quickly catch up with the customer, solve their problem, give them more ideas for shopping, or serve them better. I can't see any of these benefits that don't lead to an increase in revenue.

Why is CI/CD Important?

- When large pieces of a code base change at a time it puts an application's quality at higher risk. This is bec ause there is more likely a chance that something will break the larger the change - and troubleshooting is harder the larger the change. Agile organizations frequently integrate their code and perform automated tests to reduce the cost of introduction, identifying root causes, and fixing bugs.

- Automation is key to CI. There is no way someone could keep up manually at the speed needed for continuous integration to be successful. Developers need to integrate frequently and need feedback as so on as possible.

- With continuous deployment, in which the release to production is fully automated, you relinquish some control. At the same time, you gain additional advantages. You can develop at an even higher velocity than the already-fast continuous delivery, since you don’t need to pause development for releases, and yo ur customers will appreciate the steady stream of improvements.