Fundamentals and Benefits of CI/CD to Achieve, Build, and Deploy Automation for Cloud-Based Software Products

What is CI/CD

CI/CD is a set of development practices aimed at automating the building, testing, and deployment of applications. These practices enable developers to deliver software products to the market very rapidly.CI/CD comprises two phrases, **CI** which is an abbreviation for Continous Integration. And **CD** which in full means continuous deployment.

Fundamentals of CI/CD

What is Continuous Integration

Continuous Integration is the practice of merging all developers' working copies to 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**-This is the phase in which the code is compiled and executed.
- **Unit Test**-This is the phase in which the unit test is run against the codebase to ascertain that it doesn't contain any errors and that if any errors occur during this phase it is caught before the code is deployed to production
- **Static Analysis**-This is the phase in which all static analysis is done, this is where code quality and code coverage are determined.
- **Dependency vulnerability testing**-in this phase the code is scanned and security vulnerabilities are checked.
- **Store artifact-** This is the phase where code that has passed all the builds and tests is stored awaiting to be executed in the production environment.

What is 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** -This is the phase in which infrastructures are being created.
- **Provisioning servers** -In this phase servers are being provisioned.
- **Copying files** -This is the phase where files are being copied to the server
- **Promoting to production** -In this phase, the code is ready for release.
- **Smoke Testing (aka Verify)** - The code is being verified in this phase.
- **Rollbacks** -This is where the code which does not pass the smoke test is being rolled back to the initial version that was working.

Continuous Delivery: An engineering practice in which teams produce and release value in short cycles.

Benefits of CI/CD

- Fewer bugs get shipped to production as regressions are captured early by the automated tests hence reducing production cost.
- Building the release is easy as all integration issues have been solved early.
- Less context switching as developers are alerted as soon as they break the build and can work on fixing it before they move to another task.
- Testing costs are reduced drastically – your CI server can run hundreds of tests in a matter of seconds.

- Your QA team spends less time testing and can focus on significant improvements to the quality culture.
- The complexity of deploying software has been taken away. Your team doesn't have to spend days preparing for a release anymore.
- You can release more often, thus accelerating the feedback loop with your customers.
- There is much less pressure on decisions for small changes, hence encouraging iterating faster.
- You can develop faster as there's no need to pause development for releases. Deployments pipelines are triggered automatically for every change.
- Releases are less risky and easier to fix in case of problem as you deploy small batches of changes.
- Customers see a continuous stream of improvements, and quality increases every day, instead of every month, quarter or year.
-