

UdaPeople Project

WHY CI / CD

WHAT IS CI/CD

❖ Continuous Integration

- ❖ The practice of merging all developers' working copies to a 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 ,Unit Test, Static Analysis, Dependency vulnerability testing, Store artifact)

❖ 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 , Provisioning servers , Copying files , Promoting to production , Smoke Testing (aka Verify) , Rollbacks)

BENEFITS OF CI/CD

❖ Faster time to market

- ❖ The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently. Tech giants may have led the way, adopting Agile and DevOps techniques to transform their development processes and deliver constant improvements to their users, but with many smaller organizations following suit the landscape is becoming increasingly competitive

❖ Reduced risk

- ❖ Having a shorter time to market doesn't just help you keep up with the competition. Rapid releases provide an opportunity for product managers and marketing professionals to engage more closely with the development process.

CONT..

❖ Shorter review time

- ❖ With continuous integration, developers are encouraged to commit their code changes more frequently – at least once a day as a rule of thumb. Sharing code with the rest of the team regularly not only ensures everyone is building on the same foundation, but also results in faster code reviews and makes it easier to integrate changes.

❖ Reduce Cost

- ❖ Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD. Doing so also frees up developer time that could be spent on product development as there aren't as many code changes to fix down the road if the error is caught quickly. Another thing to keep in mind: increasing code quality with automation also increases your ROI.

❖ Better code quality

❖ Smoother path to production

❖ Faster bug fixes

❖ Smoother path to production

Why do we need to use CI/CD ?

- ❖ When we are releasing a new feature for our application we go through different stages [code compile/build - unit/static test - vulnerability detection - smoke testing - infrastructure create/clean - roll backs]
- ❖ These steps are repeatable for every release, humans get bored when they keep repeating tasks which will lead to manual mistakes and a lot of paid time for the repetitive tasks and roll backs.
- ❖ One or more team member will only know how to deploy the new release.
- ❖ The manual deployment limits your team

Is CI/CD the solution for our project ?

- ❖ CI/CD requires automation knowledge and discipline from team members but in return it will :
- ❖ Reduce costs by catch compiling error after merge and automate the configuration and clean up of the environment.
- ❖ Avoid unnecessary costs coming from security vulnerability and bugs in production Increase revenue by deploying to production more frequent and reduce time to market.
- ❖ Focus on business by automate the deployment phases and free the employees to focus on the competitive features