THE FUNDAMENTALS AND BENEFITS OF CI/CD TO ACHIEVE, BUILD, AND DEPLOY AUTOMATION FOR CLOUD-BASED SOFTWARE PRODUCTS.

Continuous Deployment

Continuous Deployment can be an incredible tool in your arsenal. Not only does CD save time, but it opens some unexpected doors that have a ripple effect over the entire organization. Let's take a look at where CD fits into the overall **S**oftware **D**evelopment **L**ife**c**ycle.

Continuous Deployment

Continuous Deployment can be an incredible tool in your arsenal. Not only does CD save time, but it opens some unexpected doors that have a ripple effect over the entire organization. Let's take a look at where CD fits into the overall **S**oftware **D**evelopment **L**ife**c**ycle.

BENEFITS OF IMPLEMENTING A CI/CD PIPELINE

1. REDUCE RISK

Finding and fixing bugs late in the development process is expensive and time-consuming. This is especially true when there are issues with features that have already been released to production.
With a CI/CD pipeline, you can test and deploy code more frequently, giving testers the ability to detect issues as soon as they occur and to fix them immediately. You are essentially mitigating risks in real time.

2. DELIVER FASTER

Organizations are moving toward releasing features multiple times a day. This is not an easy task; only a handful of companies like Netflix, Amazon, and Facebook have been able to achieve this goal. But, with a seamless CI/CD pipeline, multiple daily releases can be made a reality.
Teams can build, test and deploy features automatically with almost no manual intervention. This is accomplished using various tools, frameworks, and systems like Travis CI, Docker, Kubernetes, LaunchDarkly ,and CircleCI

3. EXPEND LESS MANUAL EFFORT

To align with the shift-left paradigm, we need automation right from the start. This is also a vital component of having a successful CI/CD implementation. Once you build features and check in code, tests should be automatically triggered to make sure that the new code does not break existing features and that the new features are working correctly.
After the tests run, the code gets deployed to different environments, including QA, staging and production. Throughout this process, you will be getting constant notifications through different channels, giving you plenty of information about the build, test and deploy cycles.

4. GENERATE EXTENSIVE LOGS

Observability is one of the biggest aspects of DevOps and CI/CD integration. If something is wrong, you need to understand why. You need a mechanism to study the system in production over time and identify key performance metrics. Observability is a technical solution that helps in this effort.
One key aspect of observability is logging information. Logs are a rich source of information to understand what is happening beneath the UI and study application behavior.
With a CI/CD pipeline, extensive logging information is generated in each stage of the development process. There are various tools available to analyze these logs effectively and get immediate feedback about the system.

5. MAKE EASIER ROLLBACKS

One of the biggest advantages of a CI/CD pipeline is you can roll back changes quickly. If any new code changes break the production application, you can immediately return the application to its previous state. Usually, the last successful build gets immediately deployed to prevent production outages.
The world is moving toward rapid release cycles, and CI/CD pipelines have accelerated the release rate. With careful planning and implementation, such a pipeline can help you find defects faster, implement fixes immediately, and increase overall customer satisfaction.