CI/CD PROPOSAL

What is Continuous Integration

Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. Continuous integration covers the process of multiple developers attempting to merge their code changes with the main code repository of a project.

What is Continuous Delivery

Continuous delivery is the next extension of continuous integration. The delivery phase is responsible for packaging an artifact together to be delivered to end-users. This phase runs automated building tools to generate this artifact. This build phase is kept 'green,' which means that the artifact should be ready to deploy to users at any given time.

What is Continuous Deployment

Continuous deployment is the final phase of the pipeline. The deployment phase is responsible for automatically launching and distributing the software artifact to end-users. At deployment time, the artifact has successfully passed the integration and delivery phases. Now it is time to automatically deploy or distribute the artifact. This will happen through scripts or tools that automatically move the artifact to public servers or to another mechanism of distribution, like an app store.

Why is Continuous Integration Needed?

In the past, developers on a team might work in isolation for an extended period of time and only merge their changes to the master branch once their work was completed. This made merging code changes difficult and time-consuming, and also resulted in bugs accumulating for a long time without correction. These factors made it harder to deliver updates to customers quickly.

Benefits of CI/CD Adoption

1. **Enable scaling:** CI enables organizations to scale in engineering team size, codebase size, and infrastructure. By minimizing code integration bureaucracy and communication overhead, CI helps build DevOps and agile workflows. It allows each team member to own a new code change through to release. CI enables scaling by removing any organizational dependencies between

development of individual features. Developers can now work on features in an isolated silo and have assurances that their code will seamlessly integrate with the rest of the codebase, which is a core DevOps process.

2. **Enables quick time to market:** If we are able to deploy changes on time, we will be able to increase the company revenue bacause new features will attract more customers
3. **Improved feedback loop:** Faster feedback on business decisions is another powerful side effect of CI. Product teams can test ideas and iterate product designs faster with an optimized CI platform. Changes can be rapidly pushed and measured for success. Bugs or other issues can be quickly addressed and repaired. Ensuring increased and protected revenue.
4. **Reduced downtime**: This helps to protect the company revenue by ensuring we don't lose customers because of product downtime encountered by the user.
5. **Improved developer productivity:** Continuous integration helps your team be more productive by freeing developers from manual tasks and encouraging behaviors that help reduce the number of errors and bugs released to customers
6. **Less bug gets to production:** We will be able to reduce cost and also avoid cost here, since the automated test will catch those bugs on time and prevent it from getting to production.