CI/CD

Leveraging modern concepts

to improve project delivery.

What is Continuous Integration/Delivery (CI/CD)?

Continuous Integration is the process of checking in codes into a shared repository as often as (often several times a day) possible while making sure the changes are automatically tested and built.

Continuous Delivery ensure that a team produces and releases value as often as possible, most times several times a day.

Continuous Deployment is an approach in which the value is promoted to production frequently through automated deployments.

Why Continuous Integration/Delivery (CI/CD)?

- While it is possible to have an entire team working without CI/CD, CI/CD helps alleviate a lot of pain points that has to do with managing software as a team.

- Adopting CI/CD into any team automatically improves team productivity by taking over processes that normally takes time there by making that available for team members to channel into building things that matter such as, more features, etc.

- CI/CD also helps to improve quality of outputs from team members by automatically applying checks to developers outputs while providing feedbacks in real time while everything is still fresh to the team member.

Why Continuous Integration/Delivery (CI/CD) ctnd?

The following are the benefits of CI/CD:

- **Catch Compile Errors After Merge** which **reduces costs** by requiring less developer time on issues of new developer code
- **Catch Unit Test Failures** which **avoids costs** by introducing less bugs in production and less time in testing
- **Detect Security Vulnerabilities** which **avoids costs** by preventing embarrassing or costly security holes
- **Automate Infrastructure Creation** which **avoids costs** by introducing Less human error which translates to faster deployments

Why Continuous Integration/Delivery (CI/CD) ctnd?

The following are the benefits of CI/CD (ctnd):

- **Automate Infrastructure Cleanup** which **reduces costs** by introducing less infrastructure costs from unused resources
- **Faster and More Frequent Production Deployments** which **increases revenue** by making it possible for new value-generating features to be released more quickly
- **Deploy to Production Without Manual Checks** which **increases revenue** by reducing time to market
- **Automated Smoke Tests** which **protects revenue** by reducing downtime from a deploy-related crash or major bugs
- **Automated Rollback Triggered by Job Failure** which **protects revenue** by quickly rolling back to return production to working state

Conclusion

The benefits of adopting CI/CD in your development process cannot be overemphasized.

In simple terms, if your development process involves anything that can be automated, then CI/CD is for you.