ADOPTING CI/CD: A BENEFICIAL TOOL FOR UDAPEOPLE APP

Presented By:

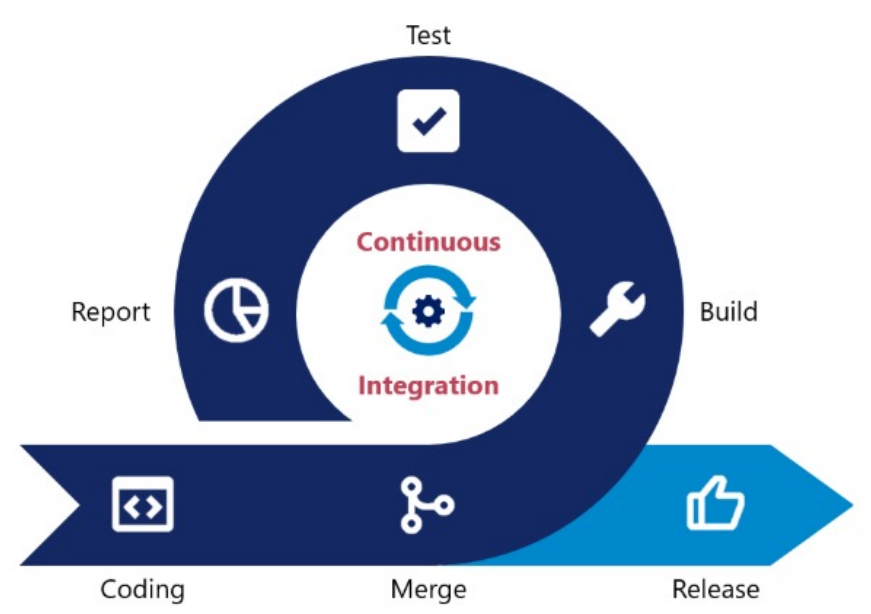
HABIB AJAO
Cloud Devops Professional, Technical support specialist

8th August ,2022

WHAT IS CI/CD

- CI/CD consist of three major concepts
- **Continuous Integration**
- Continuous Integration describes the process of merging developer branches to the main branch several times a day. CI puts an emphasis on test automation and finally generates a high quality, deployable artifact.
- **Continuous Delivery**
- In addition to Continuous Integration, Continuous Delivery makes sure that changes of a software product can be released quickly to customers in an automated way and at any point in time. Continous Deloyment is a software engineering approach in which value is delivered frequently through automated deployments.
- **Continuous Deployment**
- Continuous Deployment is a software engineering approach which extends Continuous Delivery in such a way that it allows frequent automated deployments without any human interaction. Typical phases in Continuous Deployment are Infrastructure Provisioning, Smoke Testing, Production Deployments and automated Rollbacks.

CONTINOUS INTEGRATION



Continous integration is the practice of merging all developer working copies of a project into a shared main copy several times a day.

A CI pipeline typically involves the following tasks:

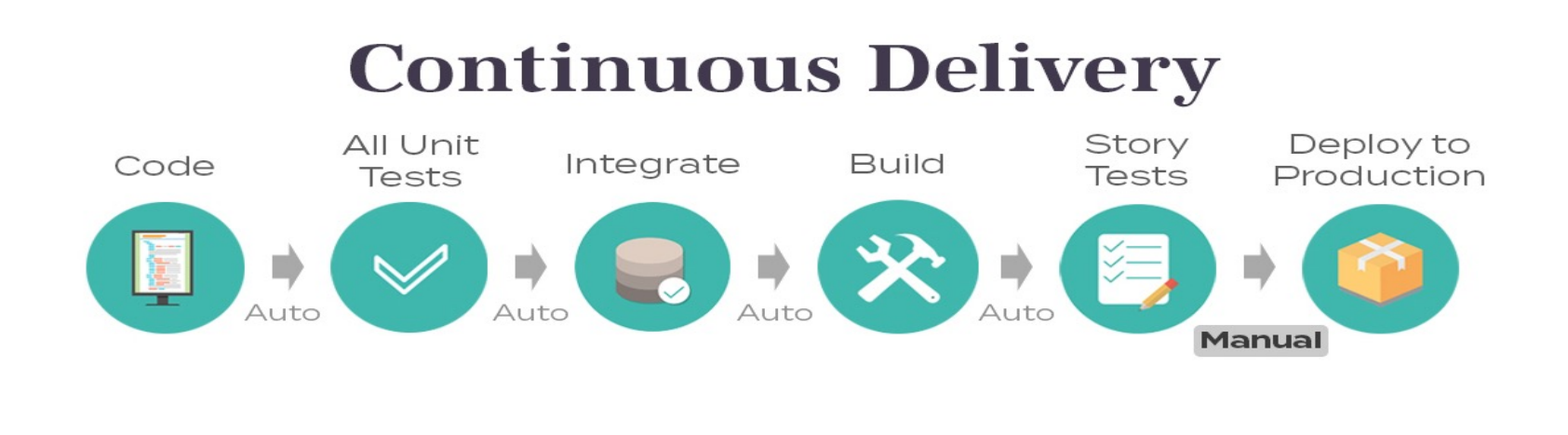
- Detect changes in the source code repository (new commits appear)
- Source code quality analysis
- Build
- Execute all unit tests
- Execute all integration tests
- Generate deployable artifacts
- Report status

If one of the steps above fails:

- Integration may stop or continue depending on defect severity and configuration
- Results are notified to the team via email or chat system
- The team fixes defects and commits again
- Tasks are performed again

CONTINOUS DEPLOYMENT

- **Continous delivery** makes it possible to release builds to the production environment when needed. Allowing the team to deploy at will, CD effectively reduces time to market.
- **Continuous deployment** is an extension of continuous delivery that automatically deploys each build that passes the full test cycle. Instead of waiting for a human gatekeeper to decide what and when to deploy to production, a continuous deployment system deploys everything that has successfully traversed the deployment pipeline. Keep in mind that when new code is automatically *deployed*, new features can still be activated conditionally at a later time or for a subset of users. Deploying automatically pushes features and fixes to customers quickly, encourages smaller changes with limited scope, and helps avoid confusion over what is currently deployed to production.NG all deployments to reduce manual intervention
-



BENEFIT OF CI/CD

1. **Smaller Code Changes**

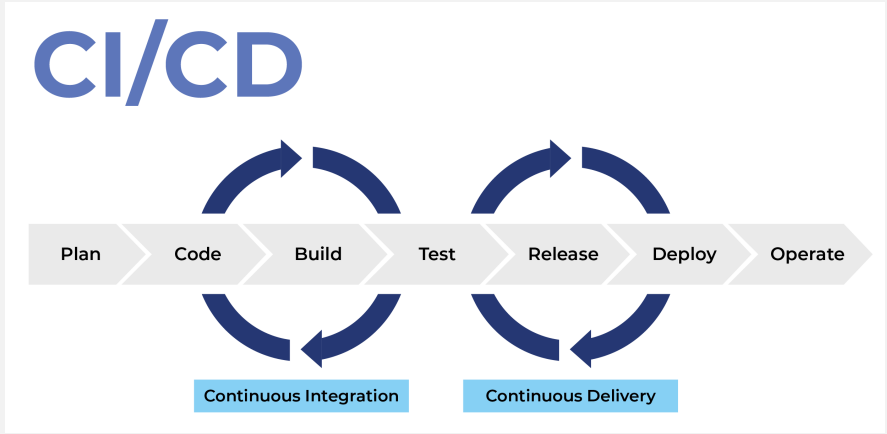
One technical advantage of continuous integration and continuous delivery is that it allows you to integrate small pieces of code at one time. These code changes are simpler and easier to handle than huge chunks of code and as such, have fewer issues that may need to be repaired at a later date.

2. **Fault Isolations**

Designing your system with CI/CD ensures that fault isolations are faster to detect and easier to implement. Fault isolations combine monitoring the system, identifying when the fault occurred, and triggering its location.

3. **Faster Mean Time To Resolution (MTTR)**

CI/CD reduces the MTTR because the code changes are smaller and fault isolations are easier to detect. One of the most important business risk assurances is to keep failures to a minimum and quickly recover from any failures that do happen. Application monitoring tools such as Prometeus are a great way to find and fix failures while also logging the problems to notice trends faster.



BENEFIT OF CI/CD

4. Faster Release Rate

Failures are detected faster and as such, can be repaired faster, leading to increasing release rates. CI/CD continuously merges codes and continuously deploys them to production after thorough testing, keeping the code in a release-ready state.

5. Smaller Backlog

Incorporating CI/CD into your organization's development process reduces the number of non-critical defects in your backlog. These small defects are detected prior to production and fixed before being released to end-users.

6. Customer Satisfaction

Utilizing a CI/CD approach also keeps your product up-to-date with the latest technology and allows you to gain new customers who will select you over the competition through word-of-mouth and positive reviews.

7. Reduce Costs

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.

THANK YOU!

- For this task of Automatic Deployment (Continuous Delivery) we are going to be using the following tools and we want you to confirm with your technical team of the safety, security and trustworthiness of these tools.
- 1. AWS – Cloud Provider
  2. Circle CI– Cloud Integration Pipeline 3. Ansible – IT Automation
  4. Prometheus – Analytics and Metrics