



Continuous Integration
Continuous Delivery
Continuous Deployment

CICD

GODWIN AGBENYENU, UDAPEOPLE PROJECT



What is CICD ?



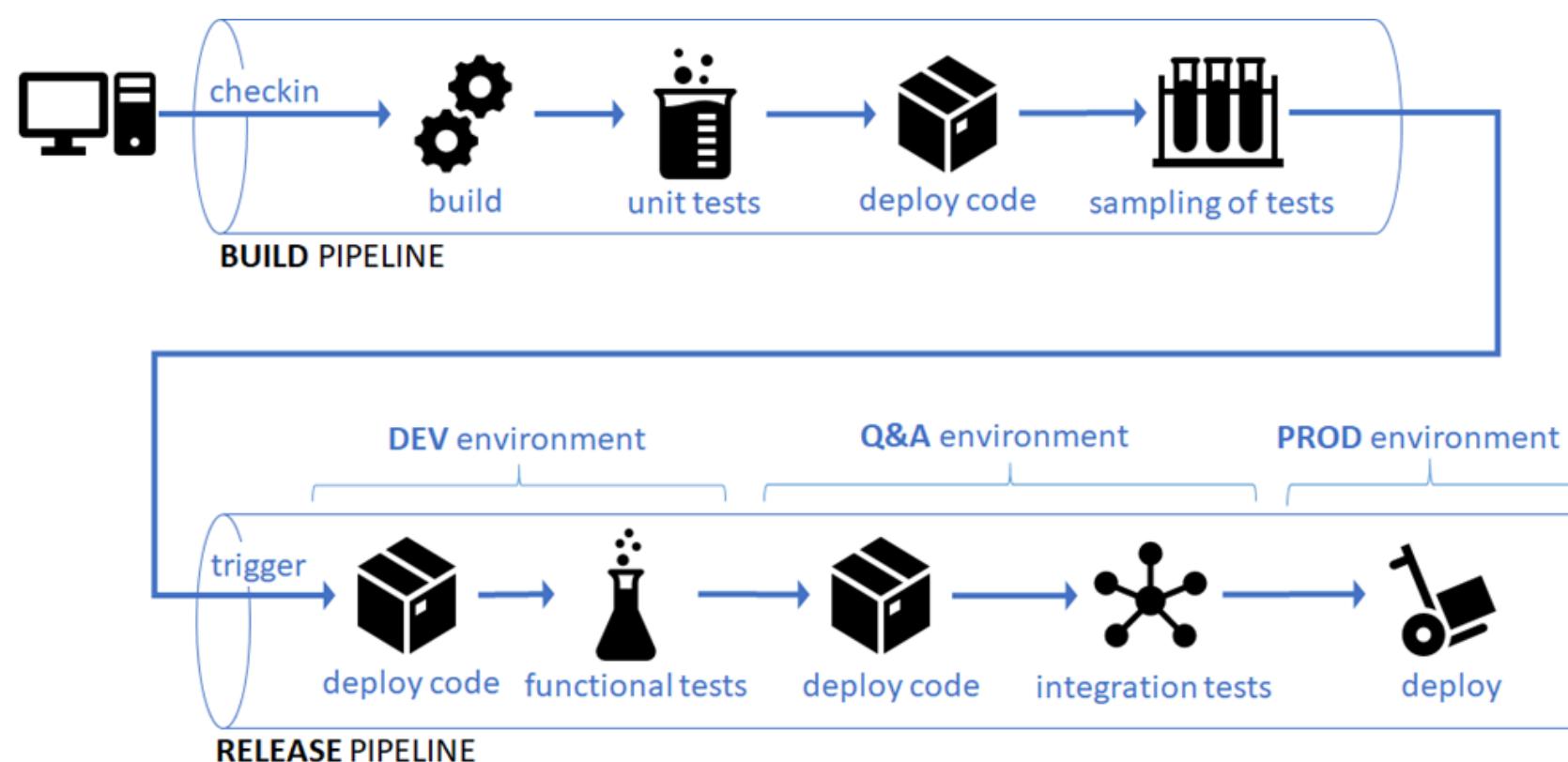
Continuous Integration

Continuous integration is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.



Continuous Deployment

Continuous deployment is a strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software's users.





Benefits Of CICD

1. SPEED

CI/CD speeds up the time it takes to resolve an issue since the code changes are smaller and blame isolations are simpler to distinguish

2. ROLLBACKS

By using CICD rollbacks are automated in getting the system back into a working state

3. CHANGE

By using CICD it is easier to continuously merge changes in a shared branch and this reduces the chances of merge conflicts

4. INTEGRATION

One specialized advantage of CICD is the ceaseless integration and nonstop conveyance in that it permits you to coordinate little pieces of code at one time.





Why you should use CICD



Detection & Isolation- In CI/CD, you ceaselessly coordinate little clusters of code rather than the whole application, which makes it easier for developers to spot issues and eliminate them. You'll be able limit potential breakdowns and other basic issues by confining bugs and vulnerabilities before they put the whole app at risk.

Automated builds - Incorporating CI/CD into your firm helps with automation will also execute unit tests and other types of tests. Testing gives imperative input to the coders that their code changes didn't break anything.

Reducing Cost - Incorporating CI/CD helps reduce the time to finish a project and ship new features and products to the market.



How the Firm Benefits from CICD

Deed

1. It takes less time to take to market new features
2. Automation reduces the cost
3. Fault isolation
4. Improved mean time to resolution (MTTR)
5. Catch Unit Test Failures Faster

Value

1. Increase Revenue
2. Avoid Cost
3. Avoid Cost and Improve Productivity
4. Increase Revenue
5. Avoid Cost