UDAPeople

CI/CD IMPLEMENTATION PROPOSAL

Mubarak Imam

DevOps Engineer

Overview

01 **What is CI/CD**
The concept, the tools and the practice

02 **Why CI/CD**
Do we really need this?

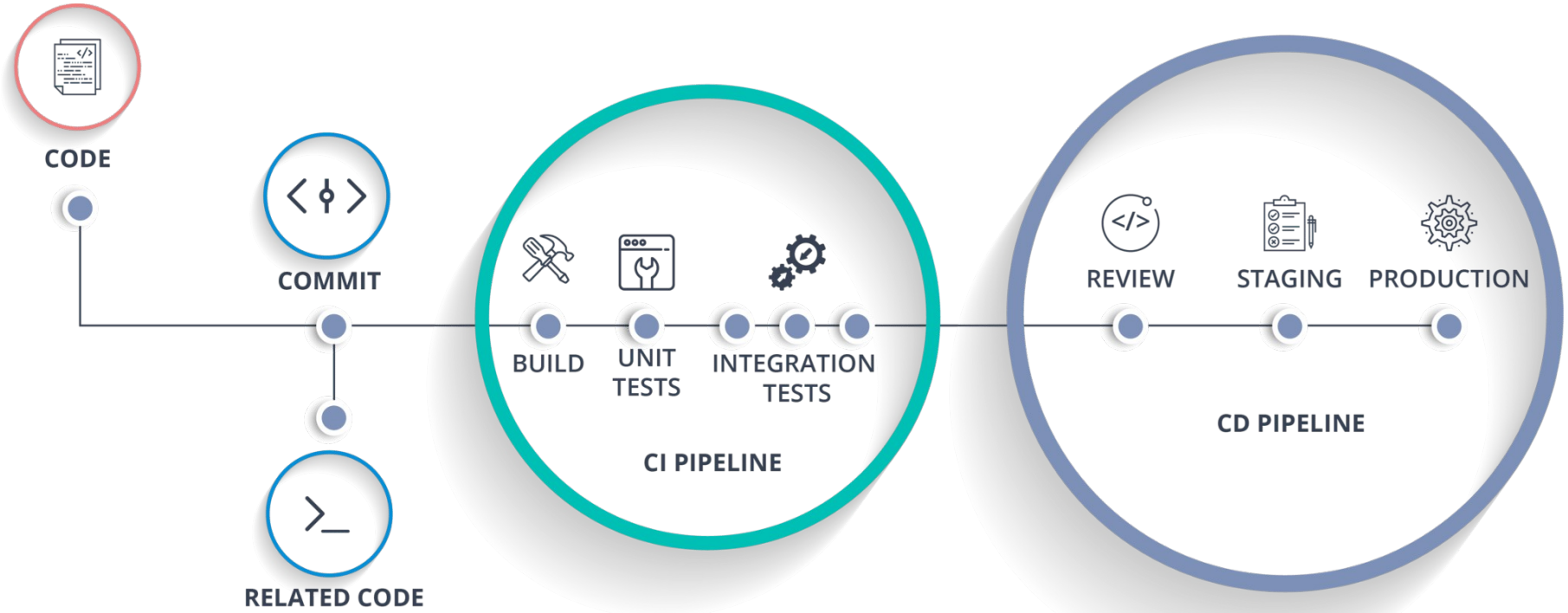
03 **A future with CI/CD**
Benefits of CI/CD and a great overall DevOps culture

04 **Challenges ahead**
Implementation challenges and our plan at solving them

What is CI/CD

Concept, Tools and Practice.

What is CI/CD

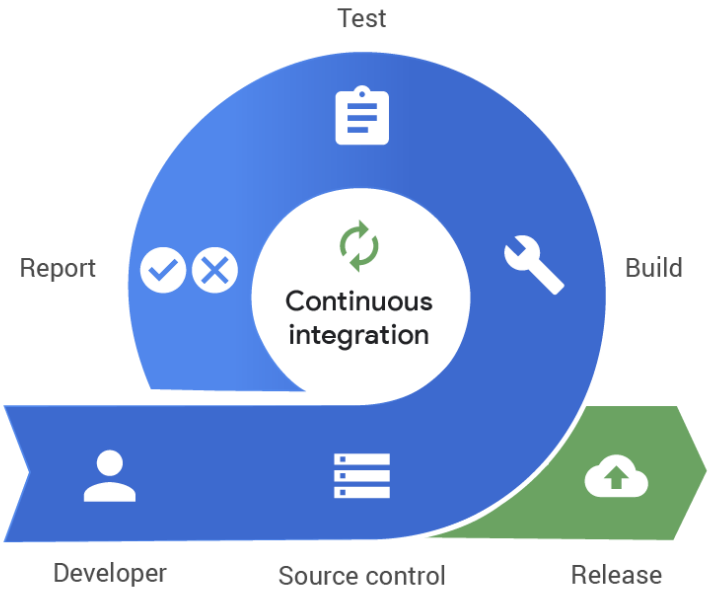


Credit: medium.com

Continuous Integration (CI)

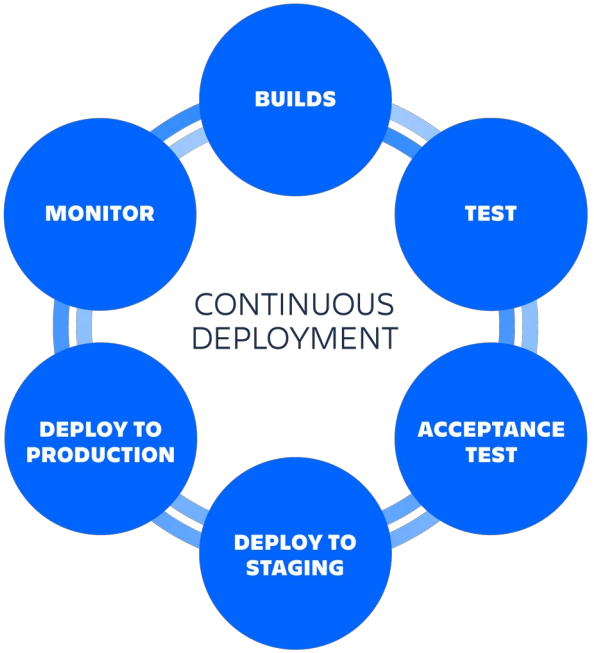
Continuous Integration describes the process of validating and merging completed feature branches in the main branch several times a day.
It emphasizes test automation and quality checks towards achieving high-quality deployable artefacts.

While the input for the continuous integration process is usually developer code, and several configurations, its output is usually deployable artefacts.



Credit: pagerduty.com

Continuous Deployment (CD)



Continuous Describes describes the process of validating the artefacts delivered by the CI process, ensuring it meets required standard viz-a-viz smoke testing, end-to-end testing, infrastructure provisioning, deployment to live environments, and automated rollbacks.

Credit: atlassian.com

Continuous Delivery

Continuous delivery is a process that guides the effective implementation of Continuous Integration (CI) and Continuous Deployment (CD)

It ensure that changes to a software product can be released to the end-user in a timely manner with the best quality, using best in-class CI/CD tools

Why CI/CD

Reliable

Produce reliable builds

Simplicity

Adopt a simpler pattern devoid of sending ftp files by email, and adopt modern practices

Dependable

Ensure only well-tested and validated code make it to the pipeline

Fast

Fail fast, and get the best reviews, and improve the system based on them within a short while.

A future with CI/CD

One-click Deployment

Deploy applications at ease by leveraging robust pipeline automating all the hard stuffs.

Scale like a pro

Increase/decrease resource allocation based on real-time utilization data, and cut excesses,

Stay Informed

Leverage observability tools to stay in the know about faulty resources before your customer calls

Reduced Backlog

Ample time saved on frequent manual deployment fixes can now be channeled into awesome features our customer loved.

Cost Reduction

Invest expensive man-hours previously spent on tedious manual processes on more important stuffs.

Improved Customer Satisfaction

Keep customer happy be providing increasing value consistently.

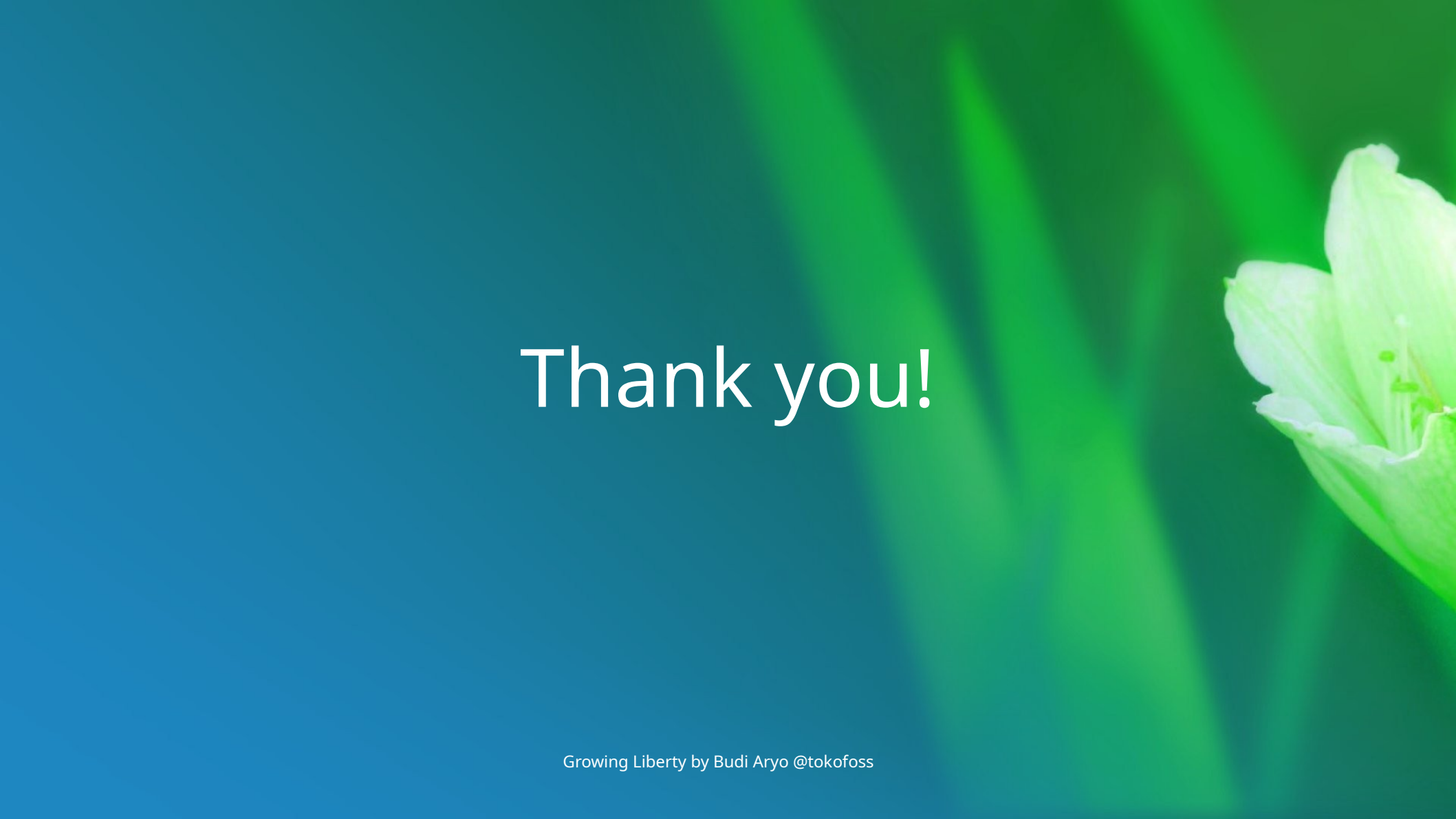
Challenges ahead

Problem 1: Cost and Resource Management
Solution: Leverage on-demand resource, and destroy when not in use.

Problem 2: Steep Learning Curve
Solution: Equip DevOps engineers to learn about modern DevOps practices

Problem 3: Integrating Security Tools
Solution: Implement vulnerability mitigation as part of the CI/CD worklow, and report to the nearest police station.

Problem 4: Ownership Conflict
Solution: Sometimes, especially where there are no dedicated DevOps engineers, CI/CD tends to create some tension, where most folks are not ready to own their domains. Gamifying the adoption by organizing internal hackathons tend to reduce the time to massive adoption in large companies.


Thank you!