CI/CD

Amira Allam

AGENDA

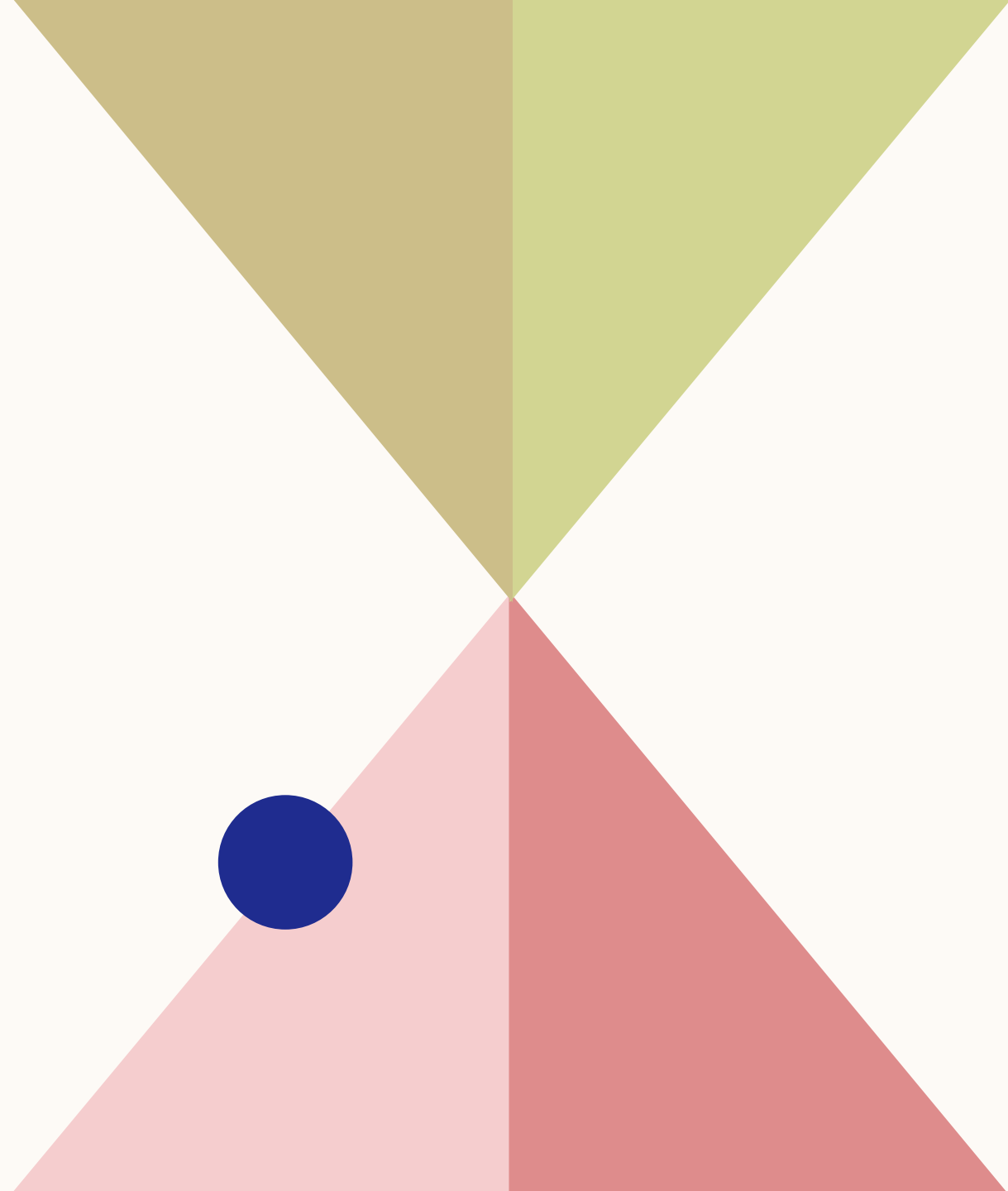
Introduction

Continuous Integration

Continuous Delivery

Benefits

Summary



INTRODUCTION

CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.

CONTINUOUS INTEGRATION / CONTINUOUS DELIVERY

CONTINUOUS INTEGRATION

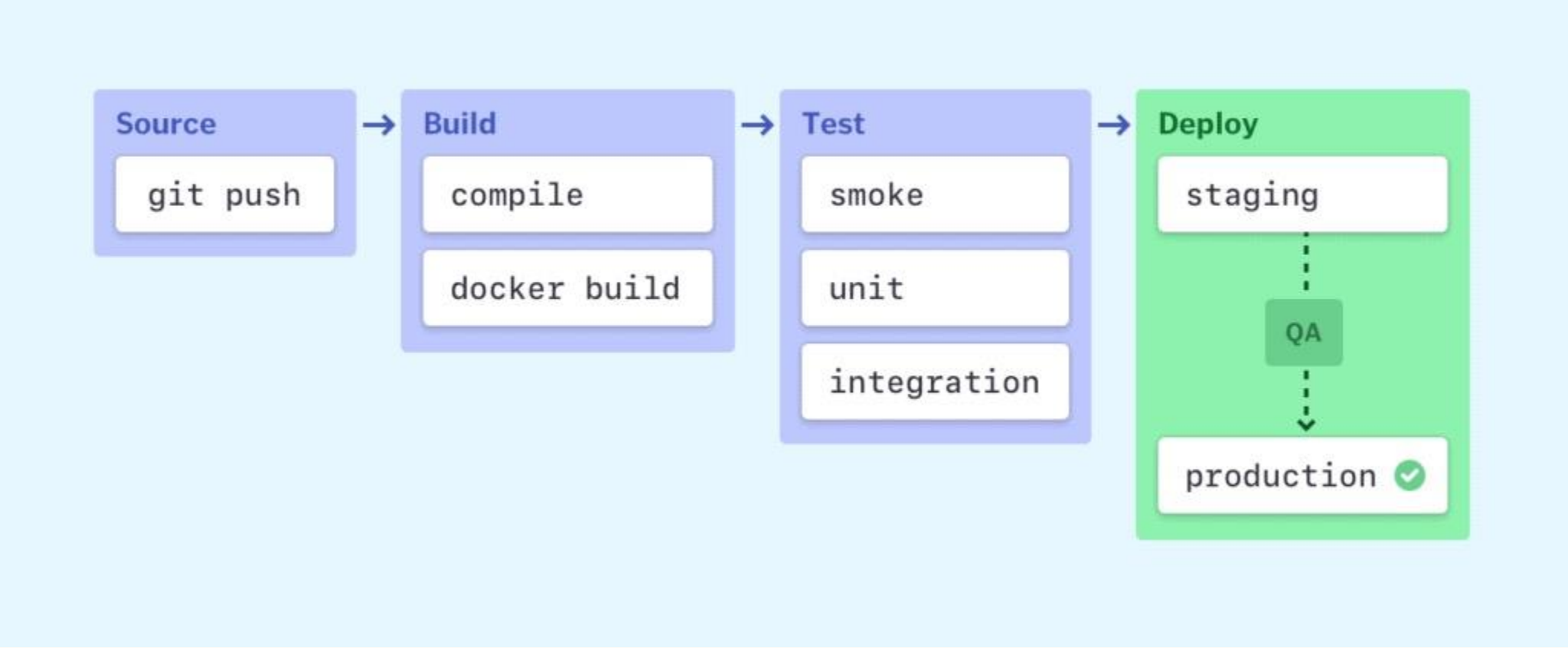
Continuous Integration (CI). Continuous Integration is about automating build and test processes to make sure the resulting software is in a good state, ideally every time a developer changes code. CI helps development teams avoid integration issues where the software works on individual developers' machines, but it fails when all developers combine their code.

CONTINUOUS DELIVERY

Continuous Delivery (CD). Continuous Delivery goes one step further to automate a software release, which typically involves packaging the software for deployment in a production-like environment. The goal of CD is to make sure the software is always ready to go to production, even if the team decides not to do it for business reasons.

ELEMENTS OF A CI/CD PIPELINE

A CI/CD pipeline may sound like overhead, but it isn’t. It’s essentially a runnable specification of the steps that any developer needs to perform to deliver a new version of a software product. In the absence of an automated pipeline, engineers would still need to perform these steps manually, and hence far less productively.
Most software releases go through a couple of typical stages:



WHAT ARE SOME COMMON CI/CD TOOLS?

CI/CD tools can help a team automate their development, deployment, and testing. Some tools specifically handle the integration (CI) side, some manage development and deployment (CD), while others specialize in continuous testing or related functions.

One of the best known open source tools for CI/CD is the automation server Jenkins. Jenkins is designed to handle anything from a simple CI server to a complete CD hub.

BENIFITS

SMALLER CODE CHANGES

it allows you to integrate small pieces of code at one time

FAULT ISOLATIONS

when an error occurs, the negative outcomes are limited in scope

FASTER MEAN TIME TO RESOLUTION

track the amount of time spent to recover from a failure.

MORE TEST RELIABILITY

improves due to the bite-size and specific changes introduced to the system

FASTER RELEASE RATE

Failures are detected faster and as such, can be repaired faster, leading to increasing release rates.

BENIFITS

SMALLER BACKLOG

reduces the number of non-critical defects in your backlog.

CUSTOMER SATISFACTION

Utilizing a CI/CD approach keeps your product up-to-date with the latest technology

INCREASE TEAM TRANSPARENCY

increases the transparency of any problems in the team and encourages responsible accountability.

REDUCE COSTS

increasing code quality with automation also increases your ROI

EASY MAINTENANCE

Don’t take the system down during peak traffic times to update code changes.

SUMMARY

CI and CD stand for continuous integration and continuous delivery/continuous deployment. In very simple terms, CI is a modern software development practice in which incremental code changes are made frequently and reliably. Automated build-and-test steps triggered by CI ensure that code changes being merged into the repository are reliable. The code is then delivered quickly and seamlessly as a part of the CD process. In the software world, the CI/CD pipeline refers to the automation that enables incremental code changes from developers' desktops to be delivered quickly and reliably to production.

THANK YOU