



Jenkins

Fundamentals of

JENKINS

BY ALOK SRIVASTAVA - NETWORK NUTS





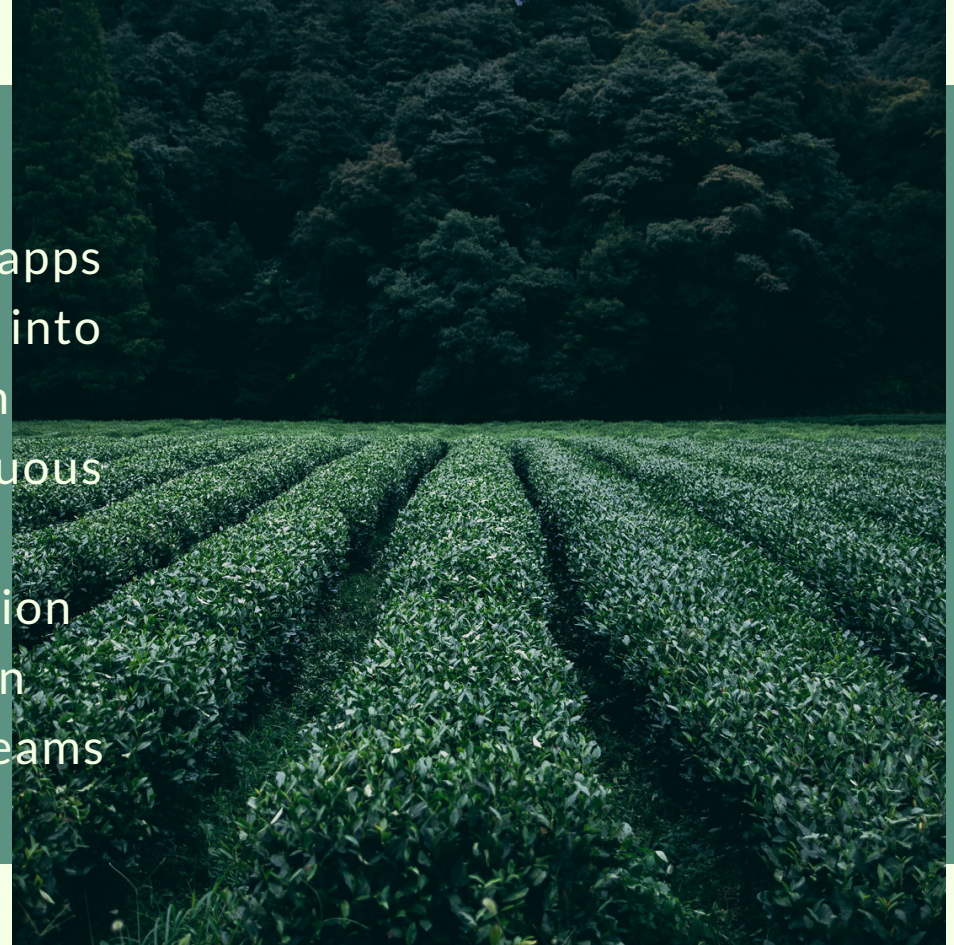
The Basic Elements of

CI/CD



CI/CD

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. CI/CD is a solution to the problems integrating new code can cause for development and operations teams (AKA “integration hell”).



CONTINUOUS INTEGRATION

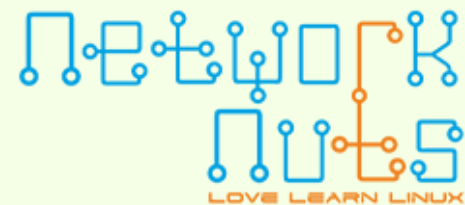
The “CI” in CI/CD always refers to continuous integration, which is an automation process for developers. Successful CI means new code changes to an app are regularly built, tested, and merged to a shared repository.

CONTINUOUS DELIVERY

The “CD” in CI/CD refers to continuous delivery. Continuous delivery usually means a developer’s changes to an application are automatically bug tested and uploaded to a repository (like GitHub or a container registry), where they can then be deployed to a live production environment by the operations team.

CONTINUOUS DEPLOYMENT

Continuous deployment (the other possible “CD”) can refer to automatically releasing a developer’s changes from the repository to production, where it is usable by customers. It addresses the problem of overloading operations teams with manual processes that slow down app delivery. It builds on the benefits of continuous delivery by automating the next stage in the pipeline.



CONTINUOUS INTEGRATION

CONTINUOUS DELIVERY

CONTINUOUS DEPLOYMENT

BUILD

TEST

MERGE

AUTOMATICALLY
RELEASE TO
REPOSITORY

AUTOMATICALLY
DEPLOY TO
PRODUCTION

ROLE OF JENKINS

Jenkins is one of the leading open source automation servers, providing countless plugins to support the building, deployment, and automation of any project. Jenkins can almost be described as a CI/CD framework, because much of the logic needed to automate a project is achieved through plugins. This gives the server a greater advantage over other CI/CD servers, because any new language, framework, or tool can be easily set up on Jenkins through these plugins.

ROLE OF JENKINS

Jenkins also provides an effective separation of concerns for everyone involved in the software's life cycle. For instance, developers get to focus on their code; the quality assurance team is solely responsible for quality measures because testing is automated through Jenkins.

JENKINS ADVANTAGES OVER SOFTWARE DEVELOPMENT

- It is open source and completely free.
- Issues with tests and builds are detected easily and reported almost immediately.
- Jenkins is platform independent, available on Windows, macOS, and Linux.
- It is easily configurable and customizable for any project.

