

Continuous

... Integration

... Delivery

... Deployment

..... DevOps

...

CI/CD

A collection of Continuous Delivery reference architectures

“Jenkins, JIRA, Nexus, Sonar, Puppet, Chef, and Maven seem to be common themes...”

(VP and DevOps Advocate)

CI/CD

Jenkins – an open source tool written in Java, which provides continuous integration services for software development. It is a server-based Build system running in a servlet container (Tomcat) which can execute arbitrary shell/batch commands. Builds can be start triggered by manual button clicks, commits in a version control system, scheduling via a cron-like mechanism, building when other builds have completed, and by requesting specific URL.

JIRA – an issue tracking product, developed by Atlassian in 2002. It provides bug tracking, issue tracking, and project management functions. It is written in Java and uses Web technology stack.

Nexus – a repository manager to store and organize binary software components. It provides a central point for management of all binary software components and their dependencies by acting as a proxy between the organization and public repositories (Maven), as well as a deployment destination (staging area) for internally developed binary components.

CI/CD

Sonar – (SonarQube) is a web-based application to manage code quality by collecting and combining metrics and presenting them in dashboards, charts and plots

Puppet/Chef – IT configuration management automation tools to manage various stages of the IT infrastructure lifecycle, including the provisioning, patching, configuration, and management of operating system and application components across enterprise data centers and cloud infrastructures

Maven – is a build automation tool used primarily for Java projects. It promotes the "Convention over Configuration" doctrine, and uses conventions for the build procedures. It uses an XML file (POM) to describe the software project, its dependencies on other external modules and components. During the Build, Maven dynamically downloads the required Java libraries (internal and 3rd party) from one or more remote repositories (e.g. Nexus)

CI/CD

- Jenkins - *A general purpose Job scheduler.*
- JIRA - *issue tracking tool with some project management, including agile functions*
- Nexus - *place to store and manage yours, as well as 3rd party binary artifacts*
- Sonar - *collect and present code metrics*
- Puppet, Chef - *IT CM tools for automated OS provisioning, configuration and patching*
- Maven - *software build tool*

CI/CD

Continuous integration (CI) “is the practice, in software engineering, of merging all developer working copies with a shared mainline several times a day. It was first named and proposed by Grady Booch and later adopted as part of extreme programming (XP). The main aim of CI is to prevent integration problems, referred to as "integration hell" in early descriptions of XP...” (http://en.wikipedia.org/wiki/Continuous_integration)

Martin Fowler

- ThoughtWorks
 - CruiseControl

“... Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible....Continuous Integration is a practice that requires no particular tooling to deploy..., ...it is useful to use a Continuous Integration server.” (2006)

(<http://www.martinfowler.com/articles/continuousIntegration.html>)

CI/CD

Continuous Delivery (CD) is a software engineering approach in which teams keep producing deliverable software in short cycles. CD extends CI by adding automated packaging, deployments to test environments, testing and publishing (staging) software, which results in the ability to reliably and repeatedly release software with a minimal manual overhead.

Continuous delivery implements a 'release' pipeline, where a software application code is compiled and packaged by a build server every time a change is committed to a source control repository, then tested by a number of different techniques and passes through a number of quality gates before it is marked as releasable.

CI/CD

Continuous Deployment (CD²) is the natural progression of the Continuous Delivery practice, where the 'released' code is automatically deployed to Production as soon as it is ready.

Continuous Deployment relies heavily on a set of fully automated functional, integration, regression, performance, security and other tests, as well as automated IT configuration and deployment procedures.

Continuous Deployment works best in agile environment, where small changes are quickly acknowledged by a CI engine, compiled, unit tested, packaged, deployed into 'test' environments, fully tested, released and immediately deployed to Production.

DevOps

DevOps - is a software development method that stresses communication, collaboration, integration, automation, and measurement of cooperation between software developers and other IT professionals. DevOps acknowledges the interdependence of software development, quality assurance, and IT operations, and aims to help an organization rapidly produce software products and services and to improve operations performance. (Wikipedia)

According to Wikipedia, the term 'DevOps' was popularized by the Agile System Administrators Group on Google through a series of 'DevOps Days' starting in 2009 in Belgium.

DevOps is a build/release/deployment automation philosophy where the automated pipeline can be invoked, monitored and controlled by developers, thus granting developers more control of the environment, and giving infrastructure more application-centric understanding.

- Code and Unit Tests (Developers)
- Automated test procedures (Developers/Testers)
- Automated application deployment procedures (Developers/System Administrators)
- Automated OS/Environment configuration procedures (System Administrators)
- DevOps Engineer ???

CI/CD

- *Scheduler – CruiseControl, Jenkins, BuildForge, RTC, AnthillPro, ...*
- *Issue tracking – Jira, ClearQuest, RTC, StarTeam, ...*
- *Binary repository – Archiva, Artifactory, Nexus, File System, ClearCase, Perforce, StarTeam, ...*
- *Metrics – SonarQube, Jenkins, CruiseControl, FindBugs, LOC, ...*
- *OS provisioning/configuration – Puppet, Chef, none*
- *Build – Ant, Maven, Make, scripts, ...*
- *VCS – anything with a good CLI, auditing, and immutable labels*