Introduction to Jenkins

Internal reference: topics/02-1.md

What is Jenkins?

Jenkins is an open source automation server written in Java. With very active community-based support and a huge number of plugins, it is one of the most popular tools for implementing continuous integration and continuous delivery processes. Formerly known as Hudson, it was renamed after Oracle bought Hudson and decided to develop it as proprietary software. Jenkins was forked from Hudson but remained open source under the MIT license. It is highly valued for its simplicity, flexibility, and versatility.

Characteristics of Jenkins

Jenkins outshines other continuous integration tools and is the most widely used software of its kind. That's all possible because of its features and capabilities.

Let's walk through the most interesting parts of Jenkins:

- `Language-agnostic`: Jenkins has a lot of plugins, which support most programming languages and frameworks. Moreover, since it can use any shell command and any software, it is suitable for every automation process imaginable (agnostic: not specific to a particular programming language aligned (ausgerichtet)).
- `Extensible by plugins`: Jenkins has a great community and a lot of available plugins (over a thousand). It also allows you to write your own plugins in order to customize Jenkins for your needs.
- `Supports most Source Control Management (SCM) tools`: Jenkins integrates with virtually every source code management or build tool that exists.
- `Distributed`: Jenkins has a built-in mechanism for the master/agent mode, which distributes its execution across multiple nodes, located on multiple machines. It can also use heterogeneous environments; for example, different nodes can have different operating systems installed.
- `Simplicity`: The installation and configuration process is simple. There is no need to configure any additional software or the database. It can be configured completely through a GUI, XML, or Groovy scripts.
- `Code-oriented`: Jenkins pipelines are defined as code. Also, Jenkins itself can be configured using YAML/XML files or Groovy scripts. That allows you to keep the configuration in the source code repository and helps in the automation of the Jenkins configuration.

Based on the book: „Continuous Delivery with Docker and Jenkins, 3rd Edition - Third Edition By Leszko“