What is the difference between Nexus and Maven?

Asked 7 years, 9 months ago Active 2 years, 8 months ago Viewed 146k times

152

What is the difference between **Nexus** and **Maven**?

What is a basic scenario of usage only **Maven**? What about a scenario considering only **Nexus**? And how it looks when I want to use both?

46

maven nexus

Share Improve this question Follow

edited Nov 24 '17 at 1:59
Chathuranga Chandrasekara
19.7k 28 95 136

asked Apr 15 '14 at 11:40
ruhungry
4,256 18 53 96

3 Answers

Active Oldest Score

180

Sonatype Nexus and Apache Maven are two pieces of software that often work together but they do very different parts of the job. Nexus provides a repository while Maven uses a repository to build software.

Here's a quote from "What is Nexus?":

> Nexus manages software "artifacts" required for development. If you develop software, your builds can download dependencies from Nexus and can publish artifacts to Nexus creating a new way to share artifacts within an organization. While Central repository has always served as a great convenience for developers you shouldn't be hitting it directly. You should be proxying Central with Nexus and maintaining your own repositories to ensure stability within your organization. With Nexus you can completely control access to, and deployment of, every artifact in your organization from a single location.

And here is a quote from "Maven and Nexus Pro, Made for Each Other" explaining how Maven uses repositories:

> Maven leverages the concept of a repository by retrieving the artifacts necessary to build an application and deploying the result of the build process into a repository. Maven uses the concept of structured repositories so components can be retrieved to support the build. These components or dependencies include libraries, frameworks, containers, etc. Maven can identify components in repositories, understand their dependencies, retrieve all that are needed for a successful build, and deploy its output back to repositories when the build is complete.

So, when you want to use both you will have a repository managed by Nexus and Maven will access this repository.

Share Improve this answer Follow

edited Nov 16 '18 at 8:30
cnorthfield
3,218 13 22

answered Apr 15 '14 at 12:07
blalasaadri
5,843 5 38 58

14

This has a good general description: https://gephi.wordpress.com/tag/maven/

Let me make a few statement that can put the difference in focus:

1. We migrated our code base from Ant to Maven
2. All 3rd party librairies have been uploaded to Nexus. Maven is using Nexus as a source for libraries.
3. Basic functionalities of a repository manager like Sonatype are:
   - Managing project dependencies,
   - Artifacts & Metadata,
   - Proxying external repositories
   - and deployment of packaged binaries and JARs to share those artifacts with other developers and end-users.

Share Improve this answer Follow

edited May 11 '18 at 21:28
Michael Mrozek
158k 26 161 169

answered Dec 22 '15 at 0:14
H M
149 1 2

2

Whatever I understood from my learning and what I think it is is here. I am Quoting some part from a book i learnt this things. Nexus Repository Manager and Nexus Repository Manager OSS started as a repository manager supporting the Maven repository format. While it supports many other repository formats now, the Maven repository format is still the most common and well supported format for build and provisioning tools running on the JVM and beyond. This chapter shows example configurations for using the repository manager with Apache Maven and a number of other tools. The setups take advantage of merging many repositories and exposing them via a repository group. Setting this up is documented in the chapter in addition to the configuration used by specific tools.

Details

Share Improve this answer Follow

edited May 24 '19 at 13:30
Koray Tugay
21.4k 40 164 293

answered Jan 3 '18 at 9:42
Moon
21 1 3