

## Jenkins-Showcase: wie Bareos-Pakete entstehen

**Das Bareos-Projekt stellt aktuell Binärpakete für insgesamt 26 verschiedene Plattformen bereit. Das Bauen, Testen und Paketieren übernimmt eine nicht mehr ganz triviale parametrische Jenkins-Pipeline.**

**Diese automatisiert neben Release- und Pre-Release-Builds der einzelnen Branches auch das bauen und testen von Pull Requests**

**Dieser Vortrag soll zeigen wie unser Build-Prozess in Jenkins implementiert ist und welche Möglichkeiten sich bieten auch große und komplexe Builds zügig, reproduzierbar und flexibel vom Git-Repository bis zum Artefakt auf dem Download-Server zu automatisieren.**

Ausgehend von einer kurzen Einführung was Bareos eigentlich ist, welcher „Tech-Stack“ dahinter steht und welche Plattformen unterstützt werden, werfen wir einen Blick auf den Entwicklungsprozess und die Frage wie die Software überhaupt gebaut und paketiert wird.

Nachdem wir nun eine Vorstellung davon gewonnen haben was eine Build-Pipeline für Bareos grundsätzlich leisten muss, schauen wir uns an wie diese mit Hilfe von Jenkins implementiert worden ist.

Hierbei schauen wir uns Schritt für Schritt die einzelnen Werkzeuge und Bausteine an aus denen unsere Pipeline aufgebaut ist:

- Beschreibung der Pipeline mittels Jenkinsfile
- Build-Agenten und Label
- Datenaustausch zwischen den Agenten
- Parallelle Ausführung von Einzelschritten
- Handhabung von Zugangsdaten und Geheimnissen
- Multibranch-Pipelines
- Dynamische Generierung von Build-Stages

Im Anschluss gebe ich einen kurzen Einblick in den Umfang unserer konkreten Pipeline und gehe auf die zur Ausführung erforderlichen Systeme ein.

Zum Abschluss gibt es noch eine Reihe von Dingen, die wir in den letzten 6 Jahren über Jenkins oder auch Continous Integration / Continous Deployment im Allgemeinen gelernt haben.

## Literaturquellen

### Getting Started with Pipeline

<https://www.jenkins.io/doc/book/pipeline/getting-started/>

### Scaling Jenkins

<https://www.jenkins.io/doc/book/scaling/>