

JENKINS - AUTOMATE YOUR DEVELOPERS LIFE

**A JOURNEY HOW MOEBEL INTRODUCED
JENKINS AND LEARNINGS OUT OF IT**

AGENDA

- ▶ /me
- ▶ What is Jenkins?
- ▶ Basics about build / automation servers
- ▶ Use Cases @moebel
- ▶ Installation / Configuration
- ▶ Used plugins @moebel
- ▶ Security and Authorization / Roles
- ▶ Integration - Slack, Bitbucket
- ▶ Tips and Tricks and Learnings

/ME

- ▶ Lars Eler
- ▶ Team Lead @moebel
- ▶ 14 years work experience
 - ▶ Team Lead @Notebooksbilliger,
 - ▶ Team Lead @Lamudi (Rocket Venture),
 - ▶ CTO @Bringmeister (Kaiser's Tengelmann)
- ▶ Passioned about automation, open source, delivering good software for the customer and fixing problems @root cause

WHAT IS JENKINS?

- ▶ "The leading open source *automation** server, Jenkins provides hundreds of plugins to support building, deploying and automating any project."
- ▶ Slogan: "Build great things at any scale"
- ▶ Buzzwords: Continuous Integration and Continuous Delivery, Easy installation, Easy configuration, Plugins, Extensible, Distributed
- ▶ Built in Java, long history (2005 as Sun Hudson), MIT license
- ▶ See Jenkins as your best friend and butler or your oversized script runner with API interface. Whatever you prefer :)

* Emphasis added by /me

BASICS ABOUT BUILD/AUTOMATION SERVER

- ▶ Needed for Continuous Integration
 - ▶ "is the practice of merging all developer working copies to a shared mainline several times a day."*
- ▶ Commonly used to automate builds, tests and deployments
- ▶ Can be used far beyond - „Cronjobs“ and Server Automation like Ansible or Puppet or anything you are too lazy to do on your own several times a week ;)

* <https://www.thoughtworks.com/continuous-integration>

USE CASES @MOEBEL

- ▶ Runs tests on pull requests
- ▶ Builds and deploys our software (IT + ML)
 - ▶ Push deploy with tar.gz file, no git on server, no ssh key on target server
 - ▶ Jenkins is allowed to access target server (like data1) but not the other way around
 - ▶ Same tar.gz is deployed to Stage and Live
- ▶ Provisions our server environments (parallel / selective)
 - ▶ Prod, Stage, Test
- ▶ Does maintenance jobs
 - ▶ Elasticsearch Index Rotation / Management, Minified Backup
- ▶ Runs tests against C24 Shopping APIs
 - ▶ Prod, Stage

INSTALLATION / CONFIGURATION

- ▶ Either package manager or tool like ansible or docker image - easy
- ▶ Jenkins is due to XML configs very automation friendly
 - ▶ Server config.xml
 - ▶ Job(s) config.xml
 - ▶ User(s) config.xml
- ▶ Hint: <https://jenkins.moebel-vergleich.de/job/test-server-tests/config.xml>

INSTALLATION / CONFIGURATION

- ▶ Job Configuration
 - ▶ Pipeline Syntax
 - ▶ Stages
 - ▶ Calls to bash / python scripts via sh call
 - ▶ Post Success / Error
 - ▶ Plugin Usage
- ▶ <https://jenkins.moebel-vergleich.de/job/build/config.xml>
- ▶ We configure Jenkins via UI @local machine, then copy XML to Ansible Role and provision Jenkins Live Machine with it
- ▶ Other option - Pipeline Script Generator <https://jenkins.moebel-vergleich.de/job/build/pipeline-syntax/>

USED PLUGINS @MOEBEL

- ▶ Jobs: Pipeline / Workflow Aggregator, Build Blocker Plugin
- ▶ Security: Role Strategy
- ▶ Communication: Slack
- ▶ Misc: AnsiColor
- ▶ Todo: Build User Vars Plugin

SECURITY AND AUTHENTICATION/ROLES

- ▶ Weakest point of security in whole system
 - ▶ Has access to GIT (best case only read), access to all servers, can deploy, can provision, has access to DB, does a lot of things with root..
- ▶ Use reverse proxy and use only https
- ▶ Limit access on IP Level (IP-Range Filter/DMZ)
- ▶ Log access and forward logs to different machine
- ▶ Users, Roles, limit roles of automatic users
- ▶ Add prefilled known hosts file to avoid security problems
- ▶ Keep SSH key only crypted in git

SECURITY AND AUTHENTICATION/ROLES

- ▶ Disable everything you don't need

- ▶ Configure RSCheck

```
<slaveAgentPort>-1</slaveAgentPort>
<disabledAgentProtocols>
  <string>JNLP4-connect</string>
</disabledAgentProtocols>
```

- ▶ Use vendor package repository / docker to get latest versions

- ▶ https://www.cvedetails.com/vulnerability-list/vendor_id-15865/product_id-34004/Jenkins-Jenkins.html

```
- name: Modify /etc/default/jenkins config file
  lineinfile:
    dest: "{{ jenkins_init_file }}"
    regexp: '^JENKINS_ARGS='
    line: 'JENKINS_ARGS="--webroot=/var/cache/$NAME/war --httpPort=$HTTP_PORT --httpListenAddress=127.0.0.1"'

- name: Modify /etc/default/jenkins config file
  lineinfile:
    dest: "{{ jenkins_init_file }}"
    regexp: '^JAVA_ARGS='
    line: 'JAVA_ARGS="--Djava.awt.headless=true -Dhudson.udp=-1 -Dhudson.DNSMultiCast.disabled=true"'
```

INTEGRATION

- ▶ Slack Plugin (easy)
- ▶ Bitbucket -> Jenkins
 - ▶ Basically HTTPS call with dedicated user, token for each operation and limit rights due to api-user role configured
- ▶ Jenkins -> Bitbucket
 - ▶ does not work yet (if anyone can help, please..)

TIPS AND TRICKS AND LEARNINGS

- ▶ Please no vendor lock-in
 - ▶ Keep core logic inside own scripts, e.g bash/python/nameit
 - ▶ In worst case you can run those commands by hand..
 - ▶ Use only common functionality (git, communication), which is available as standard in servers like that in case you change Jenkins for ?
- ▶ Use git shallow copies with depth 1 to save bandwidth and time and disk space (ML)

TIPS AND TRICKS AND LEARNINGS

- ▶ Use plain old unix error return codes ($\neq 0$) to stop job working
 - ▶ Good advice anyway :P
- ▶ Think about, when you want to be informed or not, e.g. failing tests are enough, success should not be informed
- ▶ You can schedule jobs with cron syntax, even with automatic load distribution
- ▶ Limit which jobs can run parallel

- ▶ Questions, Remarks, Concerns?