



# Continuous Integration using Docker & Jenkins

LinuxCon Europe 2014

October 13-15, 2014



Mattias Giese  
Solutions Architect  
B1 Systems GmbH  
giese@b1-systems.de

# Introducing B1 Systems

- founded in 2004
- operating both nationally and internationally
- more than 60 employees; low employee turnover
- Provider for IBM, SUSE, Oracle & HP
- vendor-independent (hardware and software)
- Focus:
  - Consulting
  - Support
  - Development
  - Training
  - Operations
  - Solutions

## Areas of Expertise

- Virtualization (XEN, KVM & RHEV)
- Systems management (Spacewalk, Red Hat Satellite, SUSE Manager)
- Configuration management (Puppet & Chef)
- Monitoring (Nagios & Icinga)
- IaaS Cloud (OpenStack & SUSE Cloud)
- High availability (Pacemaker)
- Shared Storage (GPFS, OCFS2, DRBD & CEPH)
- File Sharing (ownCloud)
- Packaging (Open Build Service)
- Providing on-site systems administration and/or development

# Partners



ARISTA





# Deployment Stack

## Deployment Stack – Technologies Used

**Docker** an open platform for developers and sysadmins to build, ship, and run distributed applications

**Fig** a Docker orchestration tool

**Gitlab** an Open Source software to collaborate on code

**Jenkins** an Open Source continuous integration system

**Puppet/r10k** an Open Source configuration management system to define the state of an IT infrastructure and to automatically enforce this state

# Docker – Build, Ship and Run Applications



# What is Docker?

- an open platform for developers and sysadmins
- Open Source Engine to standardize LXC
- build, ship and run (distributed) applications
- easy to use
- create and share images
- chroot on steroids
- all you need is inside the container
- *not* a Virtual Server – less overhead



# Technologies Used

- Linux Containers (LXC)
- chroot
- use of Linux Kernel features:
  - cgroups
  - kernel namespaces
  - ...

# Features

- can run any distribution
- “if it will run on Linux it will run in Docker”
- limited to the same architecture as the host
- all you need is inside the container
  - libraries
  - dependencies
  - ...

## Fig – A Docker Orchestration Tool



- simple orchestration tool for Docker
- easy to deploy and use
- helps to define and control a multi-container service

# GitLab – Collaboration on Code



# Features

- completely free and Open Source
- manage and browse Git repositories
- keep your code secure on own server
- manage access permissions
- perform code review and merge requests
- hooks
- much more!

# Gitlab 1/3

Dashboard

Activity
**Projects**
Issues 0
Merge Requests 0
Help

## My Projects

All projects you have access to are listed here. Public projects are not included here unless you are a member sort: Name ▾

All 5	<div style="display: flex; align-items: center;"> <div> <p><a href="#">apps / owncloud</a></p> <p style="font-size: small;">Last activity: 34 minutes ago</p> </div> </div>
Personal 0	<div style="display: flex; align-items: center;"> <div> <p><a href="#">docker / mysql</a></p> <p style="font-size: small;">Last activity: about an hour ago</p> </div> </div>
Joined 5	<div style="display: flex; align-items: center;"> <div> <p><a href="#">docker / owncloud</a></p> <p style="font-size: small;">Last activity: 17 minutes ago</p> </div> </div>
Owned 5	<div style="display: flex; align-items: center;"> <div> <p><a href="#">docker / toolbox</a></p> <p style="font-size: small;">Last activity: 14 days ago</p> </div> </div>
<b>Visibility</b>	
<input type="checkbox"/> Private	
<input type="checkbox"/> Internal	
<input checked="" type="checkbox"/> Public	
<b>Groups</b>	
<input type="checkbox"/> apps 1	<div style="display: flex; align-items: center;"> <div> <p><a href="#">puppet / global</a></p> <p style="font-size: small;">Last activity: 17 days ago</p> </div> </div>
<input type="checkbox"/> docker 3	
<input type="checkbox"/> puppet 1	

# Gitlab 2/3

apps / owncloud
Search in this project

Activity
Files
Commits
Network
Graphs
Issues 0
Merge Requests 0
Wiki
Settings

apps / owncloud

SSH

HTTP

git@docker-test.example.com:apps/owncloud.git

☰

– Edit
8 commits 1 branch 0 tags 0.16 MB

---

- Administrator pushed to branch **master** at apps / owncloud

36fc76666 fixing image urls

34 minutes ago
- Administrator pushed to branch **master** at apps / owncloud

9e68ed598 even more

about 2 hours ago
- Administrator pushed to branch **master** at apps / owncloud

eccc54922 fixing owncloud http port

about 2 hours ago
- Administrator pushed to branch **master** at apps / owncloud

a41c2c6a3 fig does not support tags, bumper.

about 2 hours ago
- Administrator pushed to branch **master** at apps / owncloud

553ed93e8 foo

about 4 hours ago
- Administrator pushed to branch **master** at apps / owncloud

bf87c4277 foo

about 4 hours ago

★ Star 0

F Fork repository 0

D Download zip ▼

C Compare code

R README

Created on Oct 09, 2014

Owned by apps group

# Gitlab 3/3

apps / owncloud

Search in this project

Activity Files Commits **Network** Graphs Issues 0 Merge Requests 0 Wiki Settings

master

You can move around the graph by using the arrow keys.

Input an extended SHA1 syntax    Begin with the selected commit

Oct 10

- fixing image urls
- even more
- fixing owncloud http port
- fig does not support tags, bumner.
- foo
- foo
- added readme
- initial



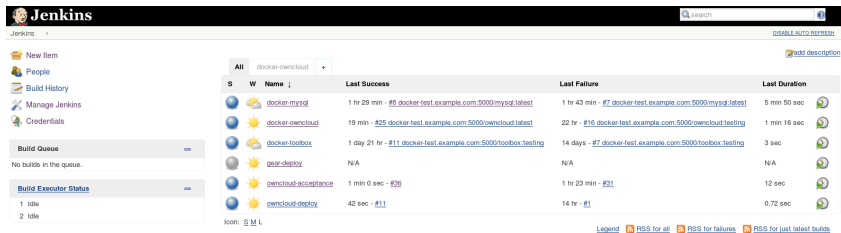
# Jenkins – Continuous Integration



# What is Jenkins?

- Jenkins is a server-based system for continuous integration running in a servlet container (like Apache Tomcat).
- Jenkins supports SCM tools including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC, and can execute Apache Ant and Apache Maven.
- Jenkins is free software, released under the MIT License.
- Builds can be started by various means, including being triggered by commit in a version control system.
- Jenkins monitors executions of repeated jobs, such as building a software project or jobs run by cron.
- Jenkins is written in Java and released under the MIT License.

# Jenkins 1/3

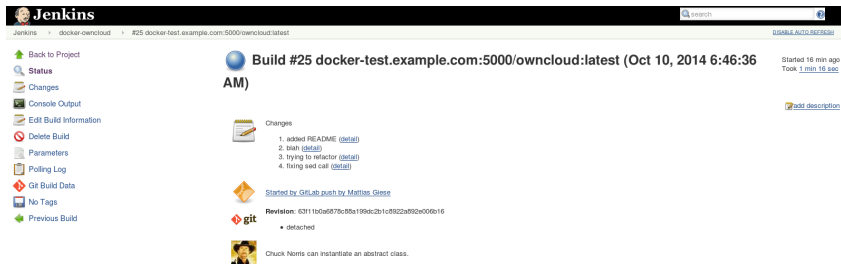


The screenshot shows the Jenkins dashboard with a sidebar on the left containing navigation links: New Item, People, Build History, Manage Jenkins, and Credentials. Below these are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main area displays a table of build jobs for the 'docker-owncloud' workspace.

S	W	Name	Last Success	Last Failure	Last Duration
		<a href="#">docker-mysql</a>	1 hr 29 min - #8 <a href="#">docker-test.example.com:5000/mysql:latest</a>	1 hr 43 min - #7 <a href="#">docker-test.example.com:5000/mysql:latest</a>	5 min 50 sec
		<a href="#">docker-owncloud</a>	19 min - #25 <a href="#">docker-test.example.com:5000/owncloud:latest</a>	22 hr - #16 <a href="#">docker-test.example.com:5000/owncloud:testing</a>	1 min 16 sec
		<a href="#">docker-toolbox</a>	1 day 21 hr - #11 <a href="#">docker-test.example.com:5000/toolbox:testing</a>	14 days - #7 <a href="#">docker-test.example.com:5000/toolbox:testing</a>	3 sec
		<a href="#">gear-deploy</a>	N/A	N/A	N/A
		<a href="#">owncloud-acceptance</a>	1 min 0 sec - #36	1 hr 23 min - #31	12 sec
		<a href="#">owncloud-deploy</a>	42 sec - #11	14 hr - #1	0.72 sec

At the bottom of the table, there are links for 'icon: S M L' and a 'Legend' section with links for 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'.

# Jenkins 2/3



**Jenkins** [search] [DISABLE AUTO-RECEIVE]

Jenkins > docker-owncloud > #25 docker-test.example.com:5000/owncloud/latest [DISABLE AUTO-RECEIVE]

- Back to Project
- Status
- Changes
- Console Output
- Edit Build Information
- Delete Build
- Parameters
- Polling Log
- Git Build Data
- No Tags
- Previous Build

**Build #25 docker-test.example.com:5000/owncloud:latest (Oct 10, 2014 6:46:36 AM)** Started 16 min ago  
Took 1 min 16 sec

[add description](#)

**Changes**

1. added README [\(detail\)](#)
2. blah [\(detail\)](#)
3. trying to refactor [\(detail\)](#)
4. fixing sed call [\(detail\)](#)

**Started by** [GitHub push by Matthias Glese](#)

**Revision:** 63111b0a6876c88a189dc2b1c8922a892e006b16

- detached

**git**

Chuck Norris can instantiate an abstract class.

# Jenkins 3/3

The screenshot shows the Jenkins web interface for a build named 'docker-owncloud'. The left sidebar contains navigation links: Back to Project, Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete Build, Parameters, Polling Log, Git Build Data, No Tags, and Previous Build. The main area displays the 'Console Output' for the build.

```

Started by GitLab push by Mattias Giese
Building in workspace /var/lib/jenkins/jobs/docker-owncloud/workspace
> git rev-parse --is-inside-work-tree # timeout=10
> git fetch --tags --progress git@docker-test.example.com:docker/owncloud.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url git@docker-test.example.com:docker/owncloud.git # timeout=10
Fetching upstream changes from git@docker-test.example.com:docker/owncloud.git
> git --version # timeout=10
using GIT_SSH to set credentials Jenkins CI default
> git fetch --tags --progress git@docker-test.example.com:docker/owncloud.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse 63f11b8a6878c88a199dc2b1c8922a892e006b16^{commit} # timeout=10
Checking out Revision 63f11b8a6878c88a199dc2b1c8922a892e006b16 (detached)
> git config core.sparsecheckout # timeout=10
> git checkout -f 63f11b8a6878c88a199dc2b1c8922a892e006b16
> git rev-list 0484fcc2ee379b3a9e4366e0be9256f37fe2288 # timeout=10
No credentials provided, so not logging in to the registry.
(workspace) $ docker build -t docker-test.example.com:5000/owncloud:latest .
Sending build context to Docker daemon 129 kB

Sending build context to Docker daemon
Step 0 : FROM hachque/opensuse
--> 042c8bf64d0e
Step 1 : MAINTAINER Mattias Giese <giese@b1-systems.de>
--> Using cache
--> 621047f6688c
Step 2 : ENV REFRESHED_AT 20141010
--> Using cache
--> f1e8fed03852
Step 3 : RUN zypper mr -ae
--> Using cache
--> 7853bd4956e0
Step 4 : RUN zypper --non-interactive ar 'http://download.opensuse.org/repositories/ix/:ownCloud:/community/7.9/openSUSE_13.1/' owncloud
--> Using cache
--> 102a1169c682
Step 5 : RUN zypper --non-interactive --pgp-auto-import-keys ref -f
--> Using cache
--> 008b0aceb8d8
Step 6 : RUN zypper --non-interactive update --auto-agree-with-licenses
--> Using cache
  
```

# Jenkins Plugins

**Docker Buildstep** allows to add various docker commands into a job as a build step.

**Docker publish** provides the ability to build projects with a Dockerfile and publish them to the docker registry.

**Git** allows the use of Git as a build SCM.

**Gitlab** a build trigger that makes GitLab think Jenkins is a GitLab CI.

**Build Pipeline Plugin** provides a Build Pipeline View of upstream and downstream connected jobs that typically form a build pipeline.

**Downstream-Ext Plugin** supports extended configuration for triggering downstream builds.

**Publish Over SSH Plugin** transfers files and data secured by SSH.

Chuck Norris ;) ...

# Jenkins – Build Pipeline Plugin

- provides a Build Pipeline View of upstream and downstream connected jobs that typically form a build pipeline.
- offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.

## Jenkins – Downstream-Ext Plugin

This plugin supports extended configuration for triggering downstream builds:

- triggers build only if downstream job has SCM changes
- triggers build if upstream build result is better/equal/worse than any given result (SUCCESS, UNSTABLE, FAILURE, ABORTED)
- for Matrix (alias multi-configuration) jobs, you can decide which part of the job should trigger the downstream job: parent only, configurations only, or both



## Jenkins – Publish Over SSH Plugin

- SCP – send files over SSH (SFTP)
- execute commands on a remote server
- username and password or public key authentication
- passwords & -phrases encryption in configuration files and UI
- SSH SFTP/SSH Exec as a build step during the build process
- SSH before a (maven) project build, or to run after a build whether the build was successful or not
- send files directly from the artifacts directory of the build that is being promoted ("promotion aware")
- optionally override the authentication credentials for each server in the job configuration
- optionally retry if the transfer of files fails
- enable the command/script to be executed in a pseudo TTY

# Puppet – Configuration Management



# What is Puppet?

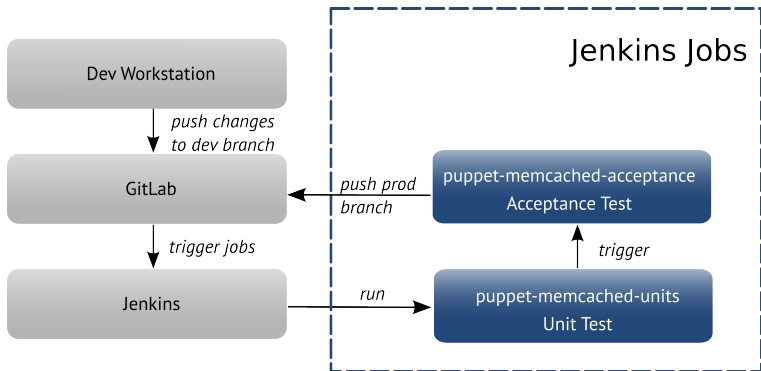
- a configuration management system to define a certain state of an IT infrastructure
- developed since 2005 by Puppet Labs
- describes resources and their state in manifests
- uses its own declarative language
- distributes these manifests through a server program called master
- Agents on the target systems enforce the desired state.
- System specific information will be discovered using `facter` for a dynamic configuration.
- Agents also send a report on the taken actions back to the Puppet master.
- Puppet's open API can send and receive data to/from third-party tools.

# Puppet – r10k

- deployment helper for Puppet modules (internal/Puppet Forge)
- uses a cache directory to preserve space
- may use a so called Puppetfile for complex deployment needs (⇒ Gemfile)

# Use Case 1: Automatic Testing of a Puppet Module

# Use Case 1 – Automatic Testing of a puppet module



## Use Case 1 – Prerequisites

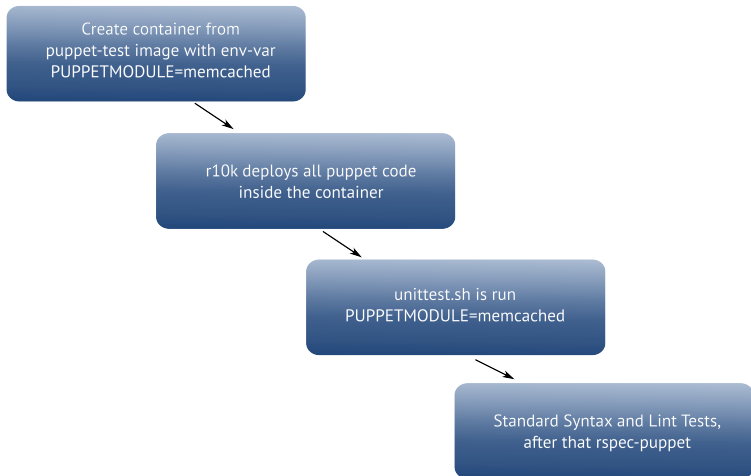
- Requirements:
  - base Docker container for every supported OS of a module
  - are being built as separate jobs
- Preparation:
  - 1 git push to development branch
  - 2 Gitlab triggers Jenkins Webhook
  - 3 Jenkins merges dev with test branch

## Use Case 1 – puppet-memcached-units

- 1 Jenkins creates a new container.
- 2 r10k deploys all puppet code.
- 3 simple syntax and style(lint) checks
- 4 rspec-puppet is run.



# puppet-memcached-units

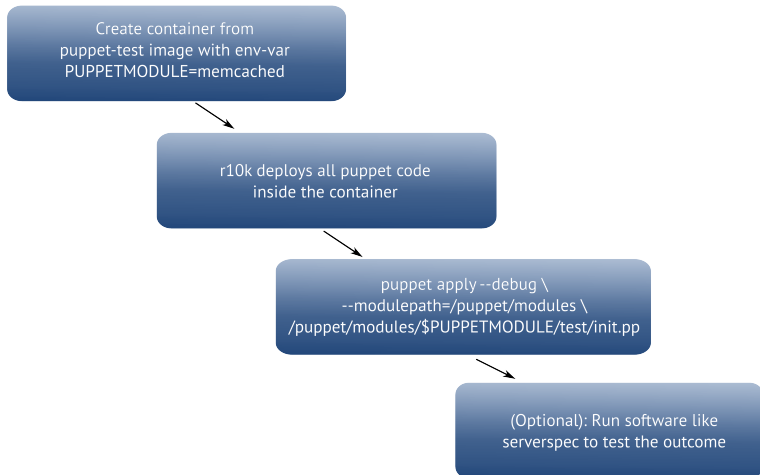


## Use Case 1 – Job: puppet-memcached-acceptance

- 1 Jenkins starts a fresh container from a puppet-enabled base image
- 2 r10k deploys all needed Puppet code
- 3 puppet apply is run with the specified module:

```
PUPPETENV=/puppet/environments/dev/  
puppet apply -debug -modulepath $PUPPETENV/modules  
$PUPPETENV/modules/$PUPPETMODULE/test/init.pp
```

# puppet-memcached-acceptance



## Use Case 2: Integration/Acceptance Testing of a Simple Webapp

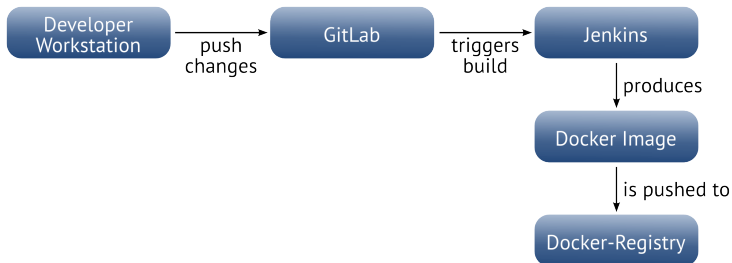
## Use Case 2: Integration/Acceptance Testing of a Simple Webapp

- simple use case for a multi-tier app
- httpd + webapp and mysql
- need to be linked together
- should be automatically deployed if possible
- two containers: owncloud and mysql
- automatic rebuild on change
- after changes, integration testing should be done
- if integration tests succeed, deploy app on staging host

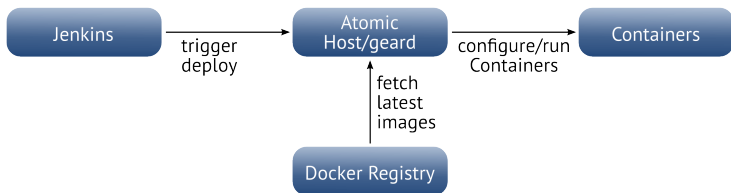
# Jenkins Job Overview



# Docker Build Pipeline 1



# Docker Build Pipeline 2







## Next Steps

## Next Steps

- Most Docker image jobs are identical:
  - implement flow/job dsl plugin
- create a dynamic test matrix for Puppet modules
  - multi-configuration jobs may help with that
  - use Packer for building base Docker images (Puppet Provisioner)
- integrate with CoreOS/Project Atomic/OpenStack
- implement a better global orchestration scheme (TerraForm, Ansible, SaltStack maybe?)
- integrate with clustered configuration through Serf/Consul or others
- automatic handoff to QA, tests with real world data in staging environments
- easy push button tagging of Docker images to 'production'
  - may be implemented through 'promoted builds'



Thank you for your attention!

For further information, please contact:  
info@b1-systems.de or +49 (0)8457 - 931096