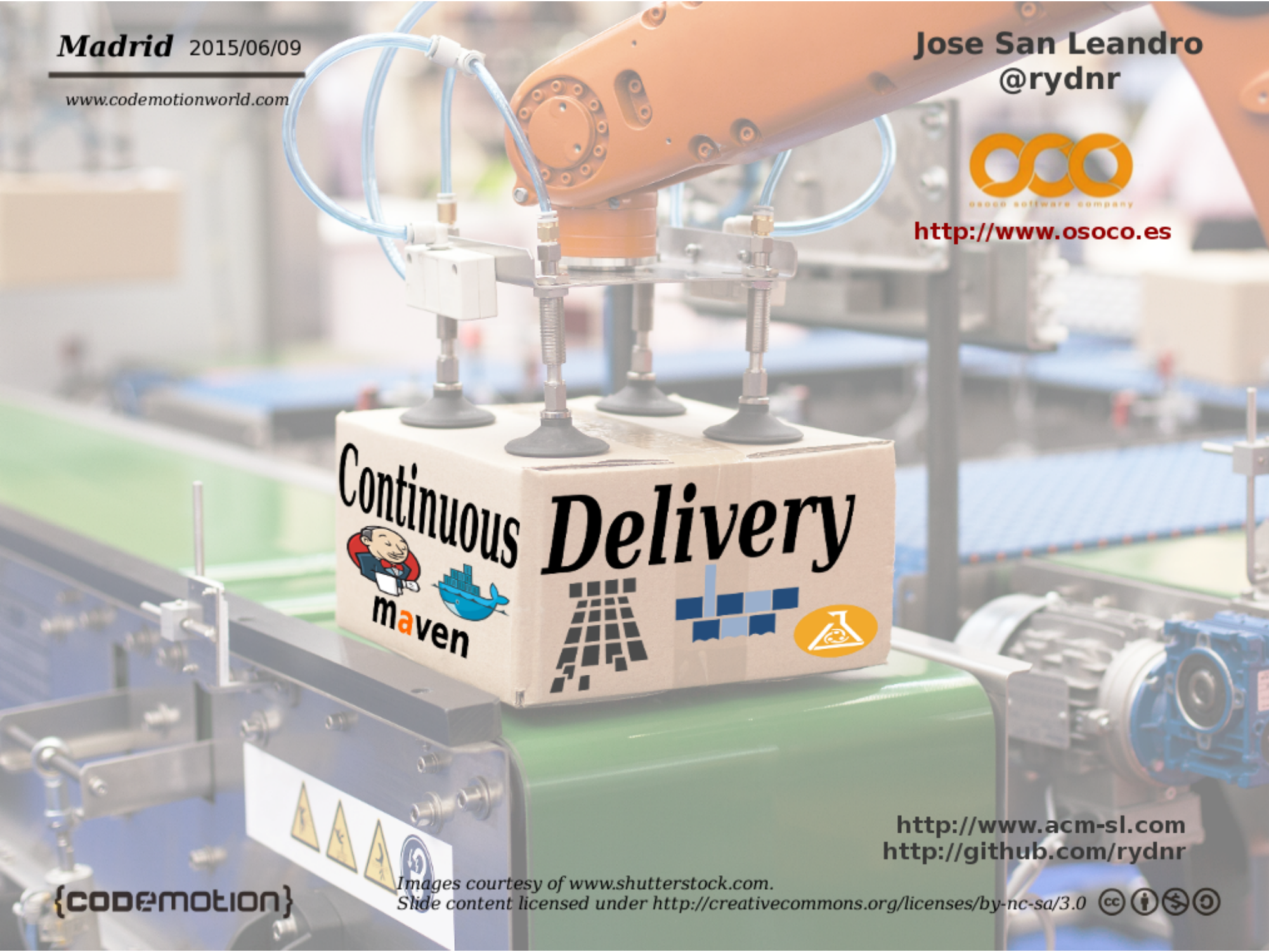


Madrid 2015/06/09

www.codemotionworld.com

Jose San Leandro
@rydnr

osoco software company

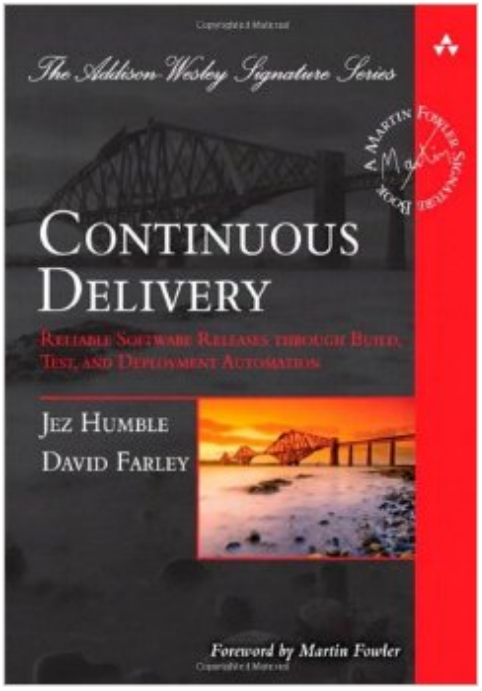
http://www.osoco.es

http://www.acm-sl.com
http://github.com/rydnr

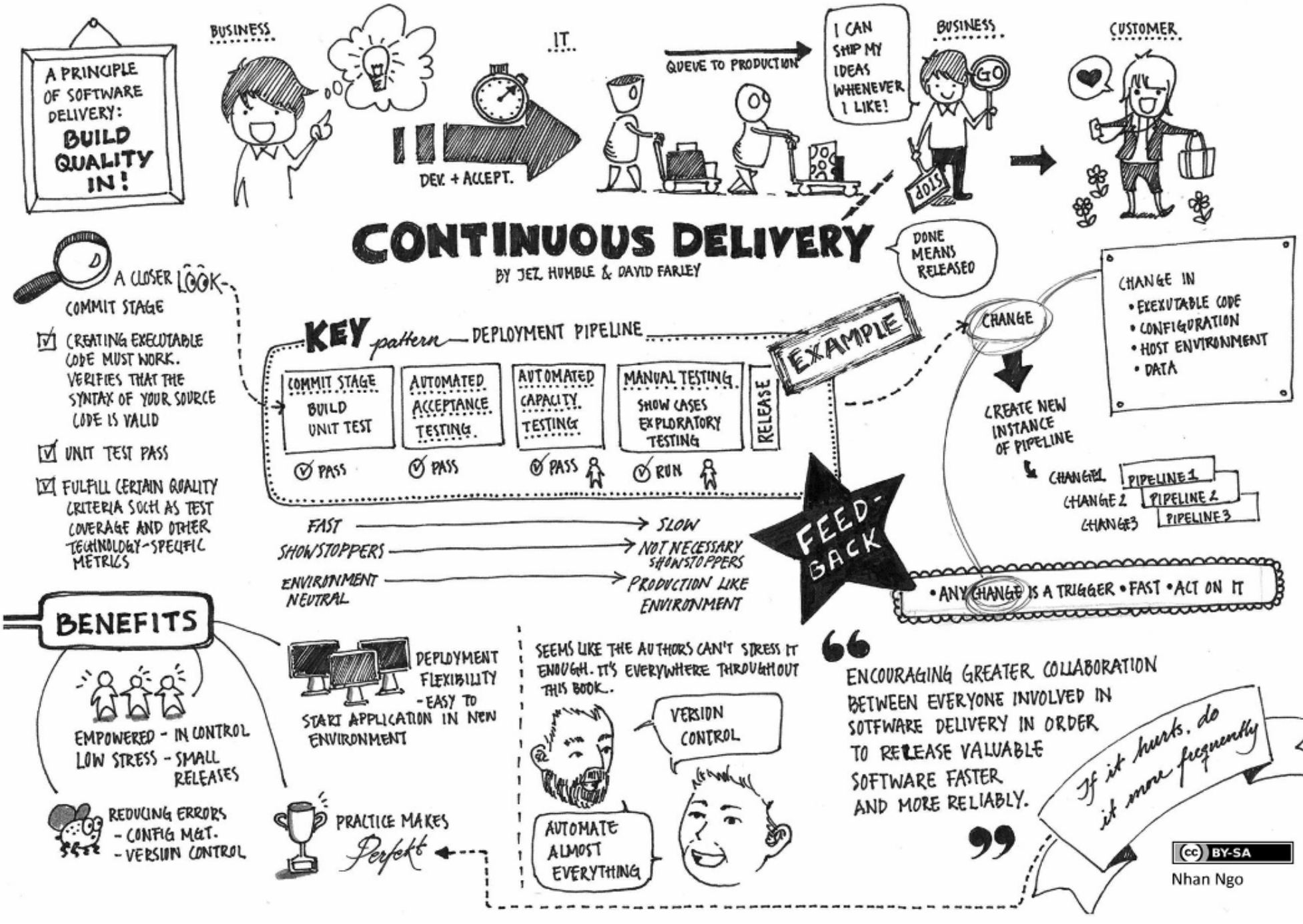
{codemotion}

What is this all about?

- A brief introduction to Continuous Delivery.
- How each tool fits in the big picture.
- The approach I propose, without the pain.
- Help to get you started on your own.
- Advance towards Continuous Deployment.



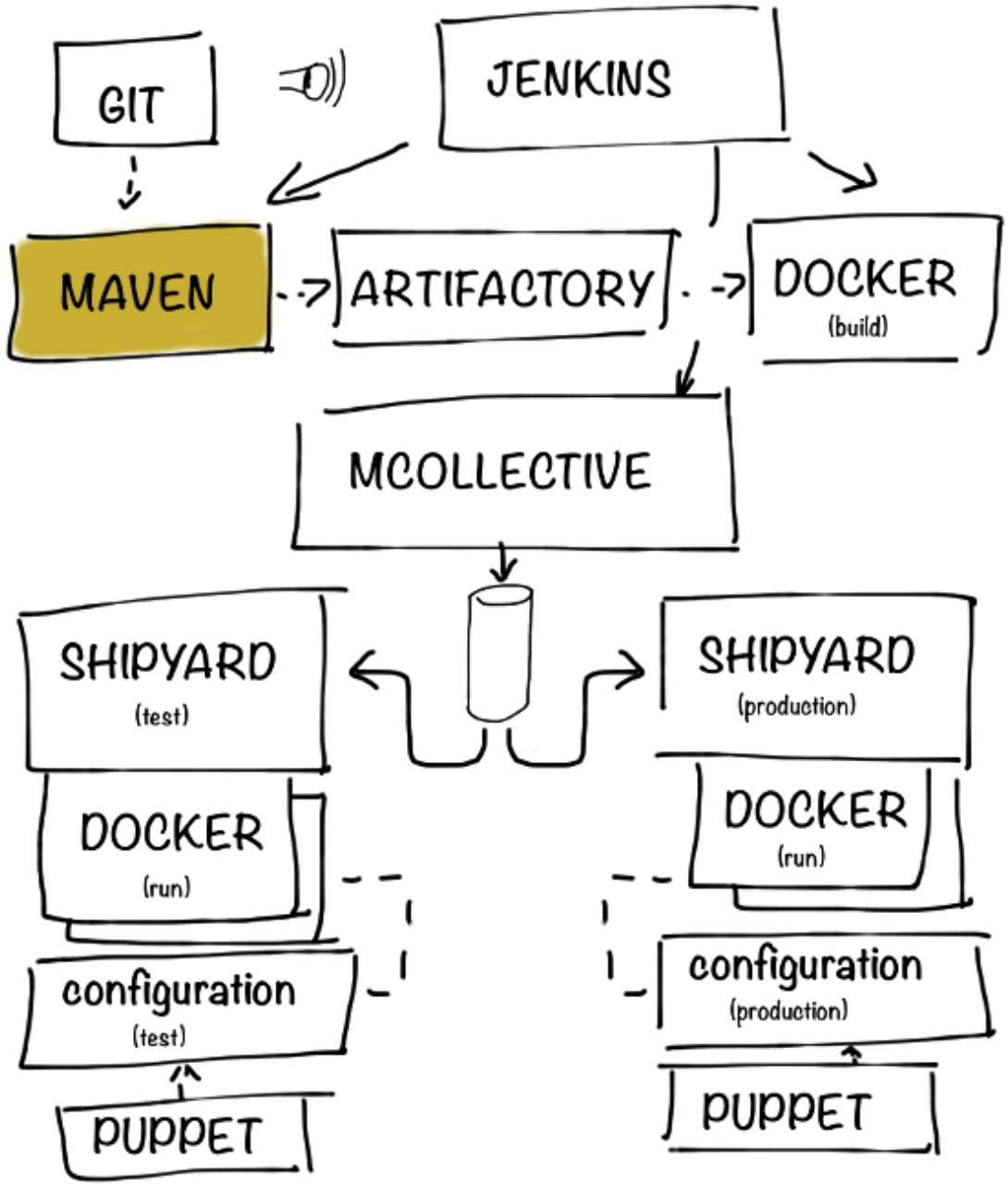
"Encouraging greater collaboration between everyone involved in software delivery in order to release valuable software faster and more reliably."


A PRINCIPLE OF SOFTWARE DELIVERY: BUILD QUALITY IN!
BUSINESS
IT
DEV. + ACCEPT.
QUEUE TO PRODUCTION
I CAN SHIP MY IDEAS WHENEVER I LIKE!
BUSINESS
GO
STOP
CUSTOMER
CONTINUOUS DELIVERY
BY JEZ HUMBLE & DAVID FARLEY
DONE MEANS RELEASED
A CLOSER LOOK-
COMMIT STAGE
☑ CREATING EXECUTABLE CODE MUST WORK. VERIFIES THAT THE SYNTAX OF YOUR SOURCE CODE IS VALID
☑ UNIT TEST PASS
☑ FULFILL CERTAIN QUALITY CRITERIA SUCH AS TEST COVERAGE AND OTHER TECHNOLOGY-SPECIFIC METRICS
KEY pattern — DEPLOYMENT PIPELINE
COMMIT STAGE BUILD UNIT TEST
AUTOMATED ACCEPTANCE TESTING.
AUTOMATED CAPACITY TESTING
MANUAL TESTING. SHOW CASES EXPLORATORY TESTING
RELEASE
PASS
PASS
PASS
RUN
EXAMPLE
CHANGE
CHANGE IN
• EXECUTABLE CODE
• CONFIGURATION
• HOST ENVIRONMENT
• DATA
CREATE NEW INSTANCE OF PIPELINE
CHANGE1 PIPELINE 1
CHANGE 2 PIPELINE 2
CHANGE3 PIPELINE 3
FAST → SLOW
SHOWSTOPPERS → NOT NECESSARY SHOWSTOPPERS
ENVIRONMENT NEUTRAL → PRODUCTION LIKE ENVIRONMENT
FEED-BACK
• ANY CHANGE IS A TRIGGER • FAST • ACT ON IT
BENEFITS
EMPOWERED - IN CONTROL
LOW STRESS - SMALL RELEASES
DEPLOYMENT FLEXIBILITY - EASY TO START APPLICATION IN NEW ENVIRONMENT
REDUCING ERRORS - CONFIG MGT. - VERSION CONTROL
PRACTICE MAKES Perfekt
SEEMS LIKE THE AUTHORS CAN'T STRESS IT ENOUGH. IT'S EVERYWHERE THROUGHOUT THIS BOOK.
VERSION CONTROL
AUTOMATE ALMOST EVERYTHING
“ENCOURAGING GREATER COLLABORATION BETWEEN EVERYONE INVOLVED IN SOFTWARE DELIVERY IN ORDER TO RELEASE VALUABLE SOFTWARE FASTER AND MORE RELIABLY.”
If it hurts, do it more frequently
BY-SA
Nhan Ngo

Automation

Bots welcome

- Speed up the release of new features.
- Special focus on risk: automate everything!
- Advance towards Continuous Deployment.
- No need for code freeze.
- Automated tagging.



"Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information."

maven

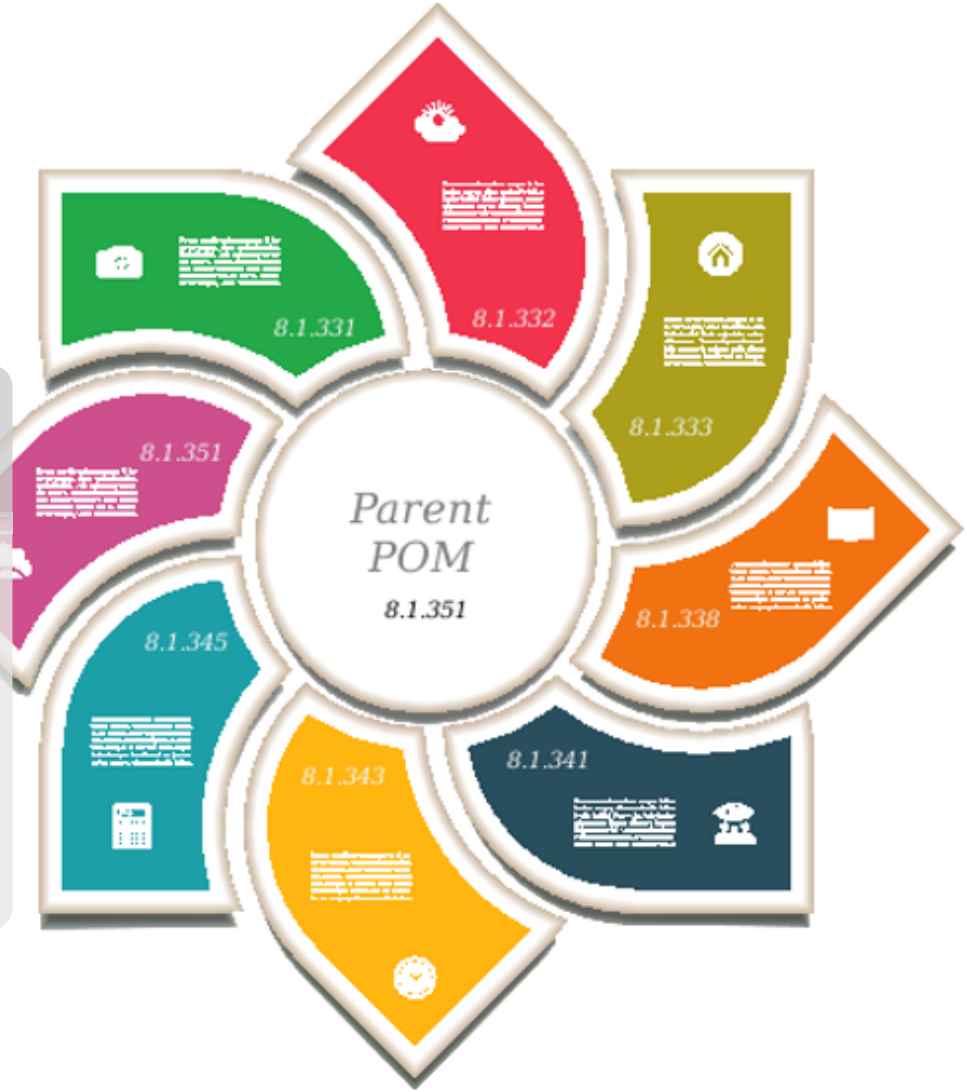
http://maven.apache.org

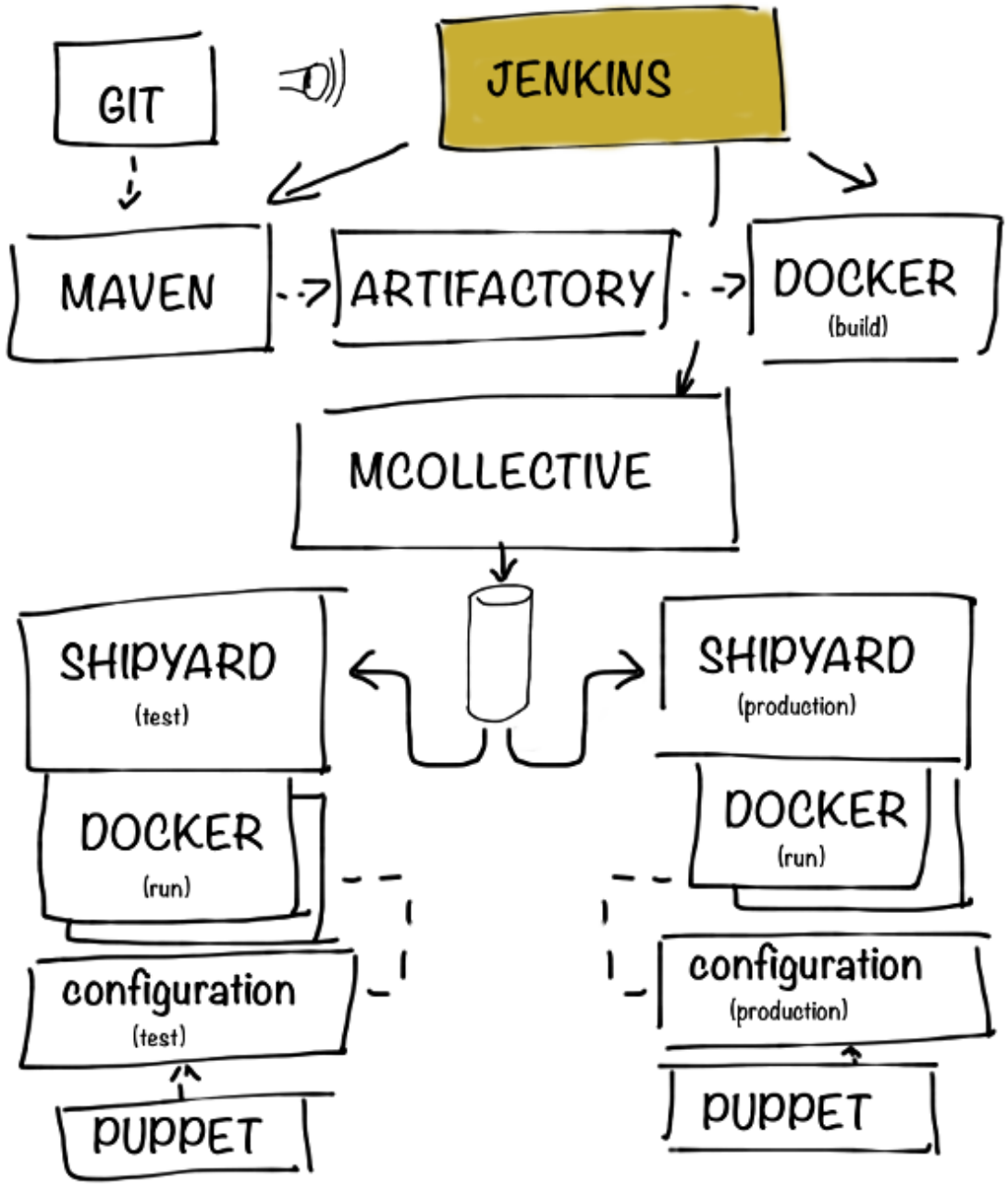
Convention

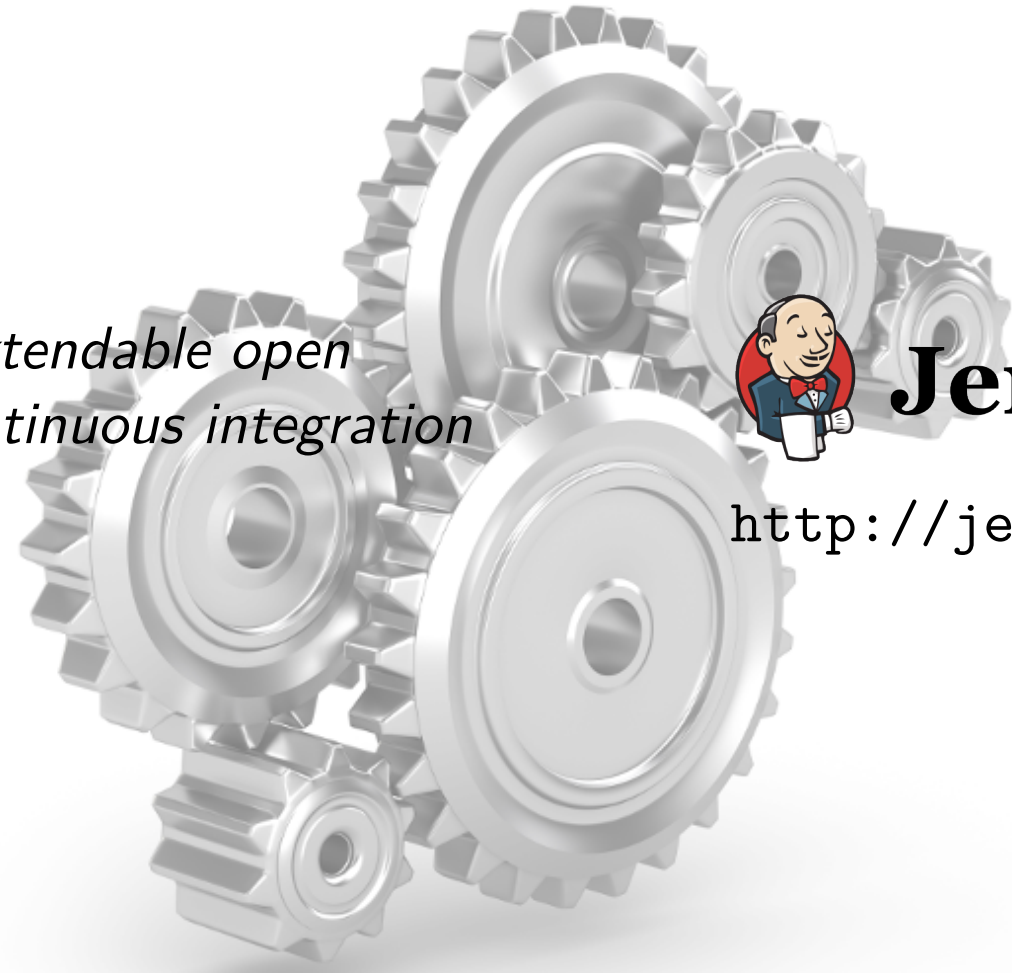
- All logic is isolated in its own module.
- No multi-module projects, unless for WARs.
- All modules inherit from a common, logic-less module: the parent POM.

Versions are unique and time-ordered

- All in-house modules share the same version (`latest-SNAPSHOT`).
- Actual versions are resolved when generating releases.



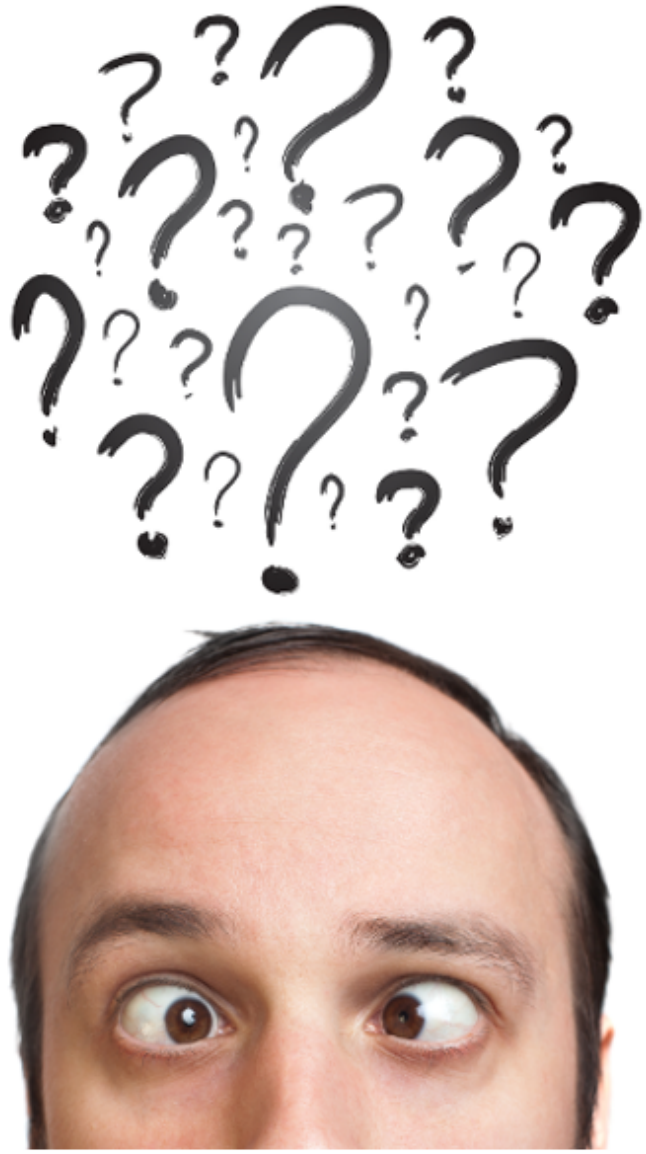




"An extendable open source continuous integration server."

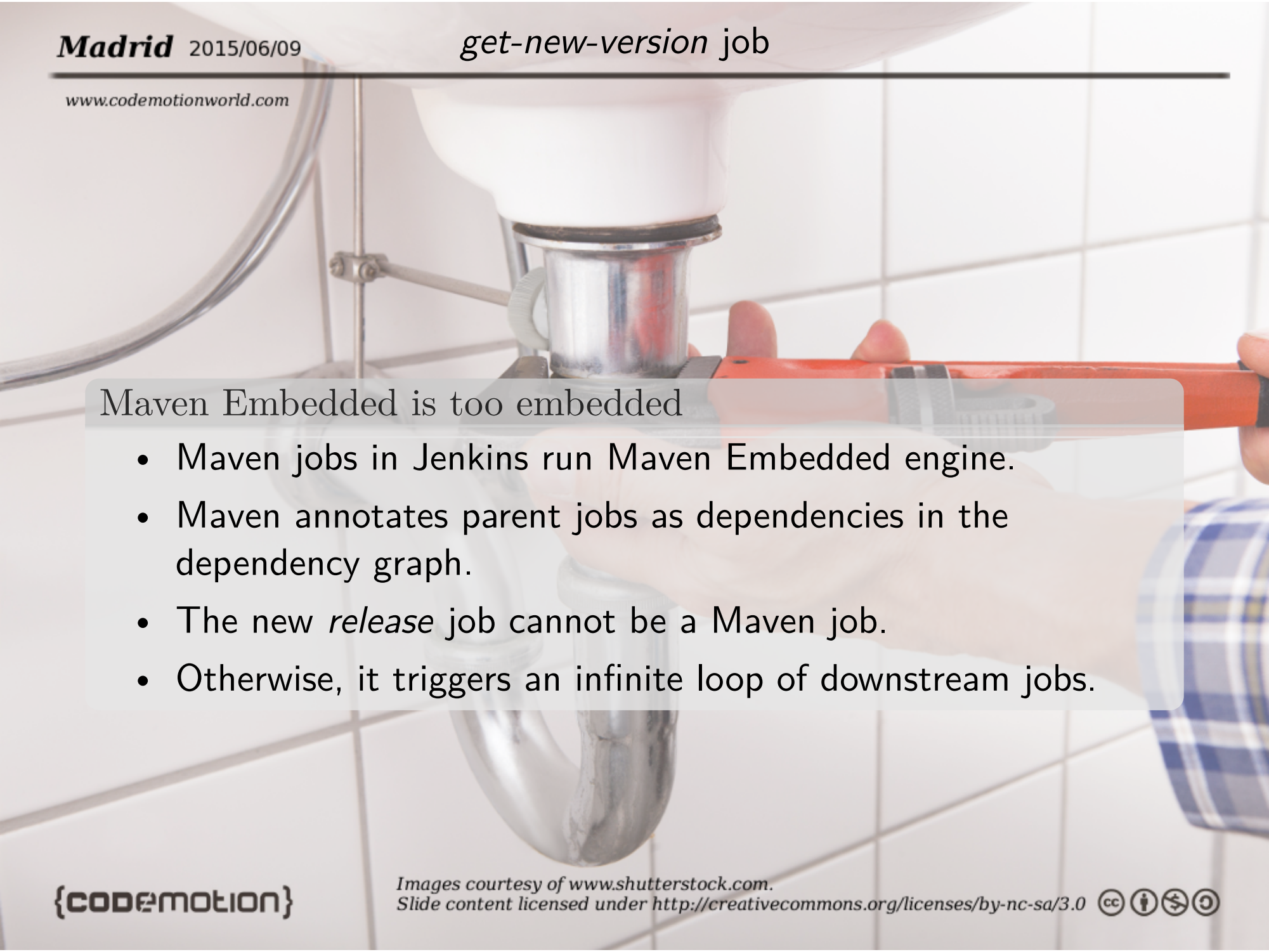
Jenkins

http://jenkins-ci.org



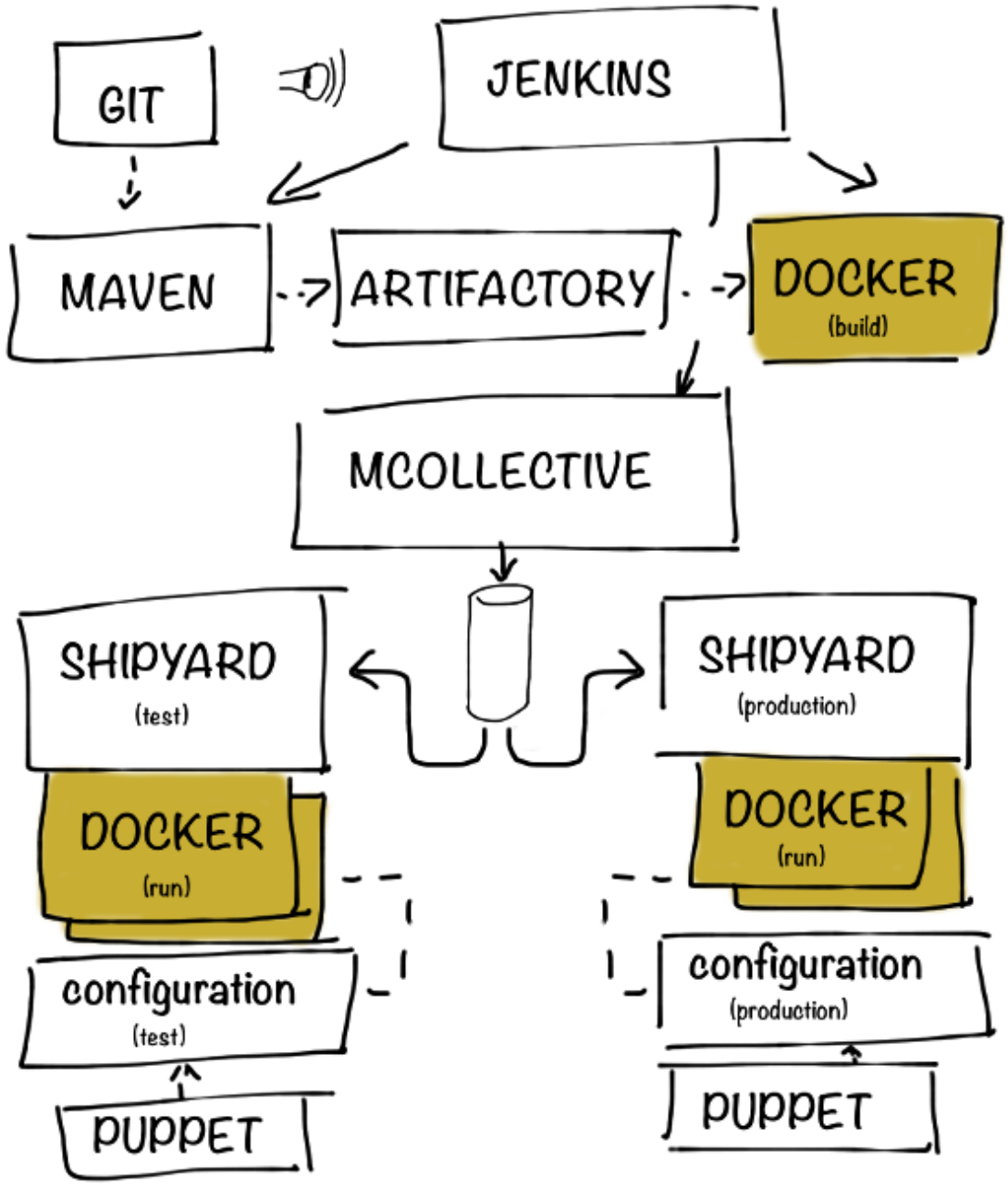
One **job** to rule them all

- Helper job to automate the tagging and packaging process.
- Checks out parent-pom code.
- Gets launched for every change, and generates a new tagged release.



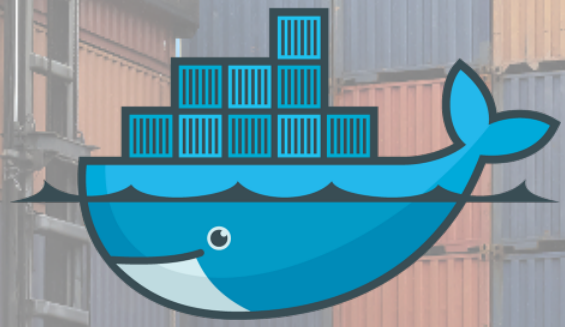
Maven Embedded is too embedded

- Maven jobs in Jenkins run Maven Embedded engine.
- Maven annotates parent jobs as dependencies in the dependency graph.
- The new *release* job cannot be a Maven job.
- Otherwise, it triggers an infinite loop of downstream jobs.

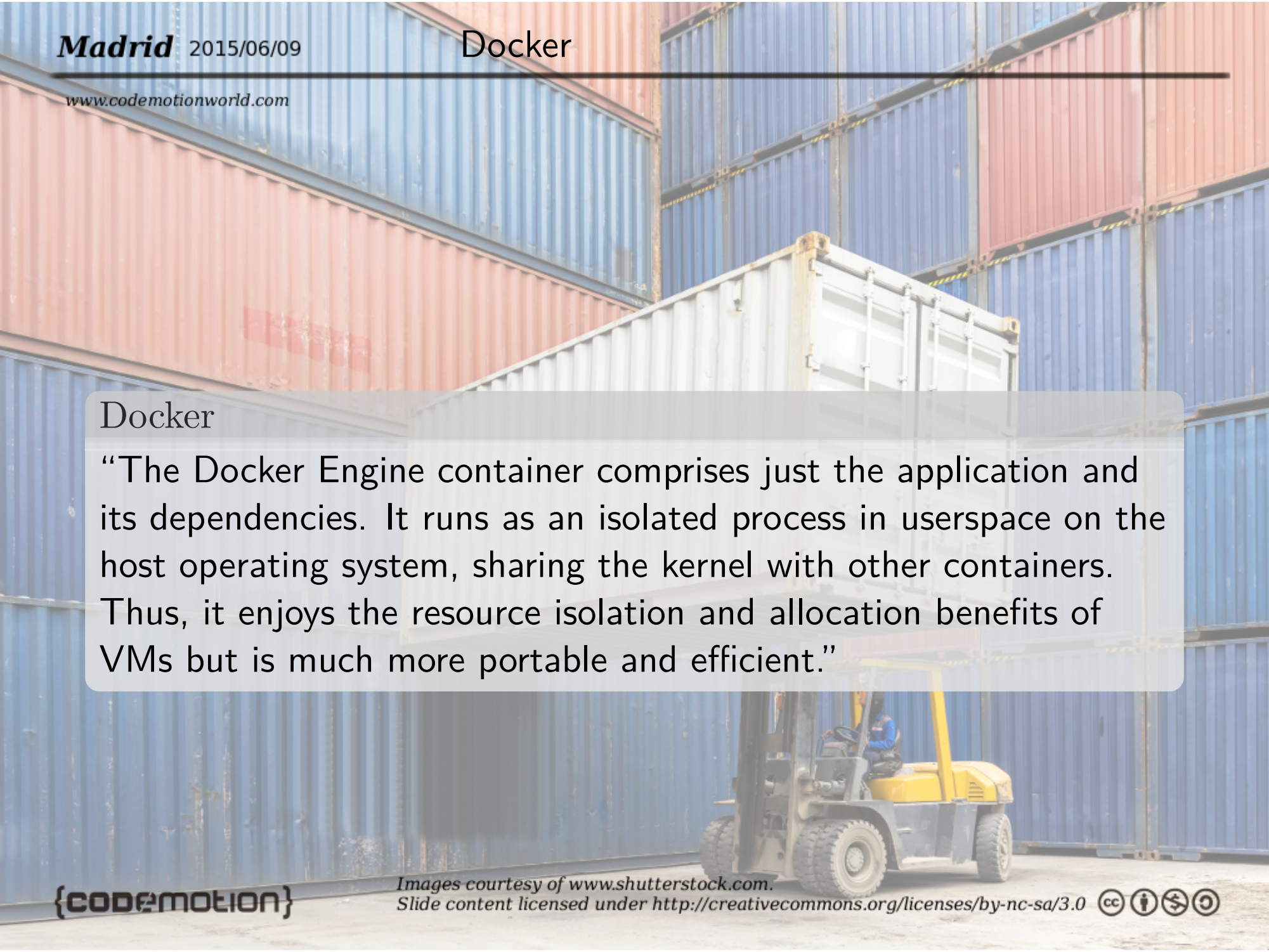


Docker

"An open platform for distributed applications for developers and sysadmins."



`http://www.docker.com`



Docker

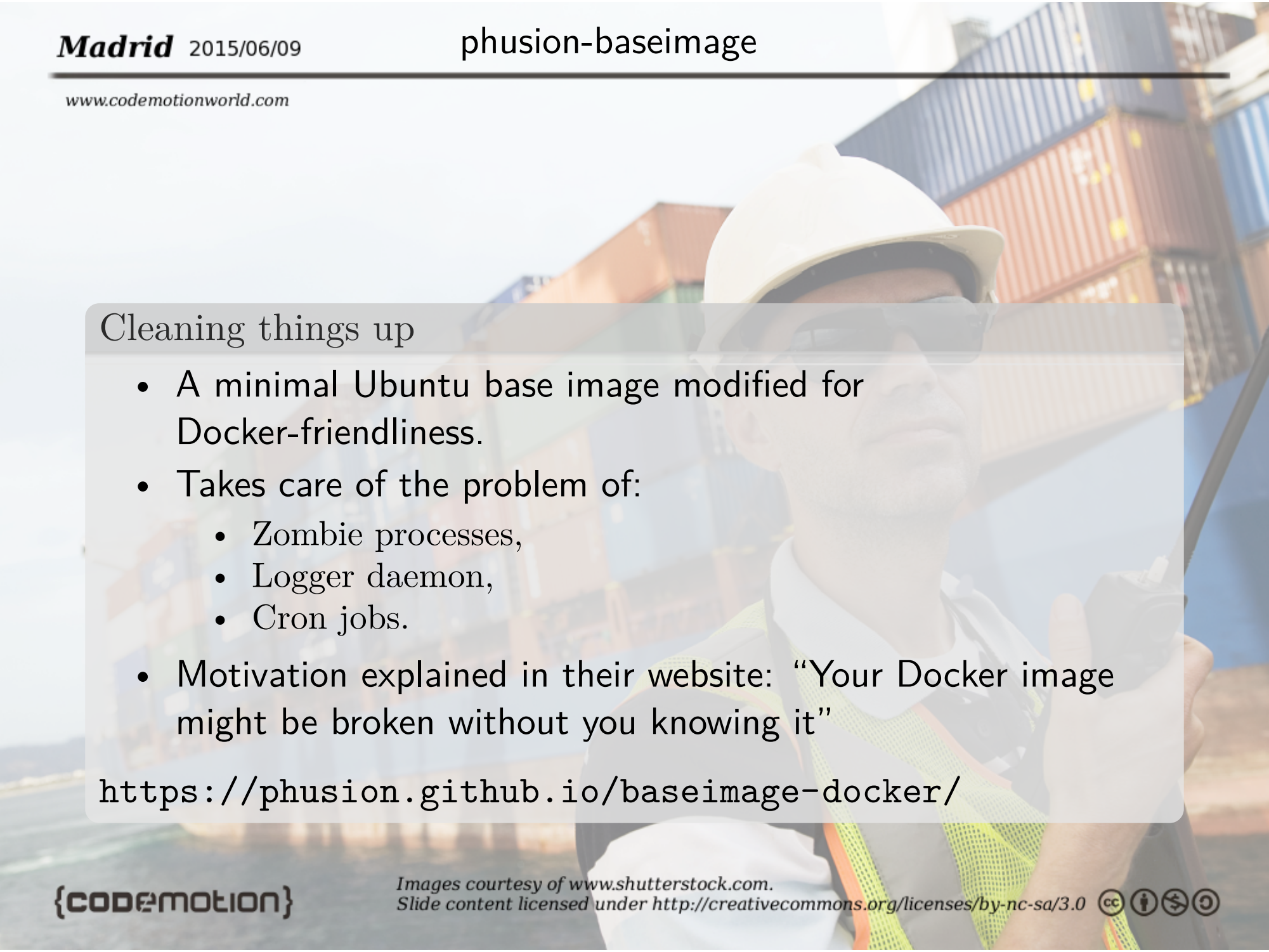
"The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient."

Packaging applications

- **Image**: Packaged application and dependencies. Ready to launch.
- **Container**: An isolated (process, memory, network, etc.) environment, running an *image*.
- **Volume**: A folder within a container, accessible from the host. Can be directly mapped to a folder in the host.

Running applications

- **Link**: Docker mechanism to help containers communicate with each other. It's defined as `-link container:alias`:
  - *container*: the name of the external, already running container,
  - *alias*: the name used locally in the new container, pointing to the external container. Docker adds it to /etc/hosts, and defines some environment properties.
- **Exposed port**: Docker service can map host ports to internal ports, when the container starts.



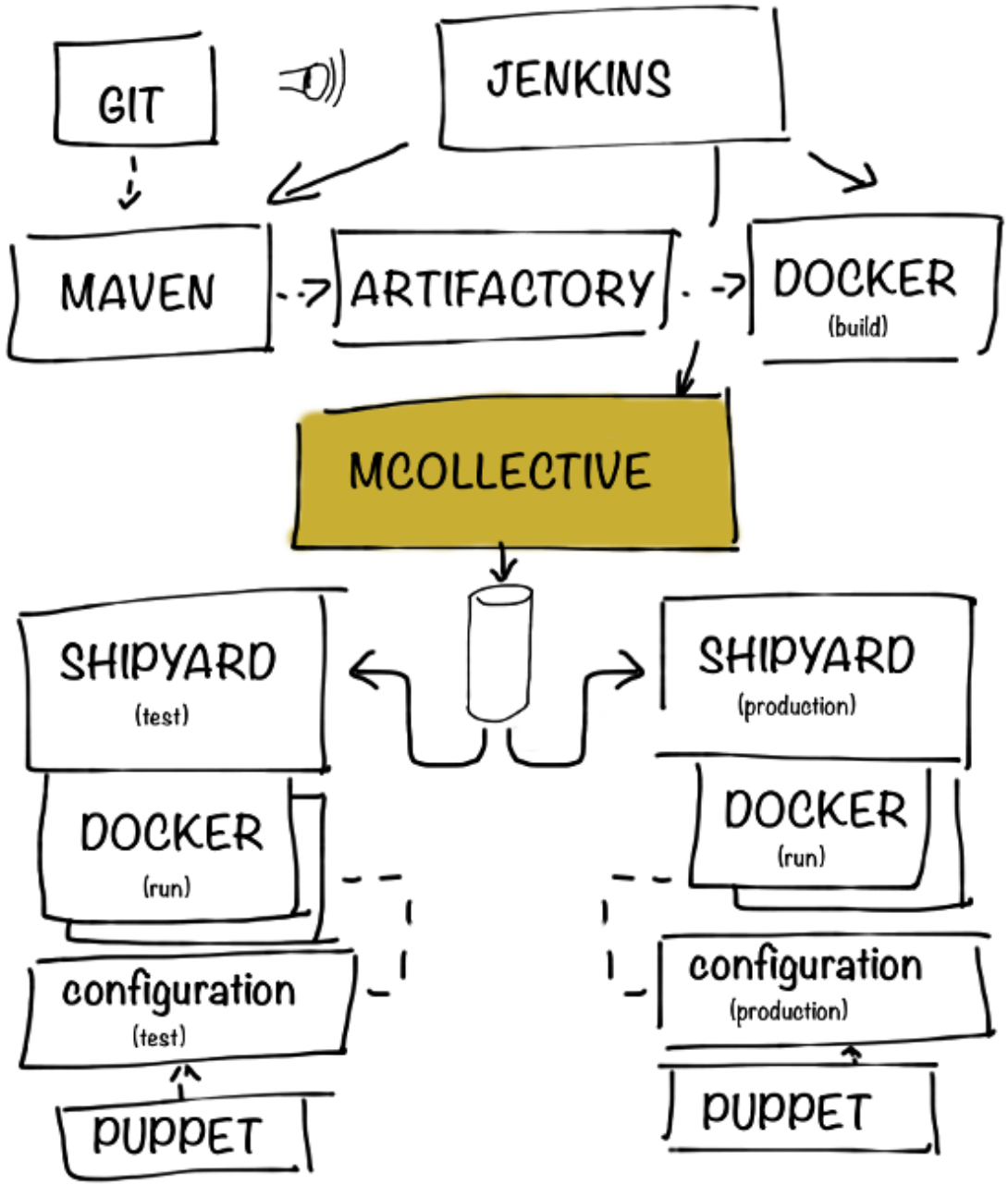
Cleaning things up

- A minimal Ubuntu base image modified for Docker-friendliness.
- Takes care of the problem of:
  - Zombie processes,
  - Logger daemon,
  - Cron jobs.
- Motivation explained in their website: "Your Docker image might be broken without you knowing it"

`https://phusion.github.io/baseimage-docker/`

Variables in Dockerfiles

- Based on wking's approach and code for Gentoo-based images: `https://github.com/wking/dockerfile`
- Modified for phusion-baseimage.
- Enhanced with in-house bash scripting framework: dry-wit.
- Allows placeholders in Dockerfiles.



MCollective

"MCollective is a powerful orchestration framework.
Run actions on thousands of servers simultaneously, using existing plugins or writing your own."

http://www.puppetlabs.com

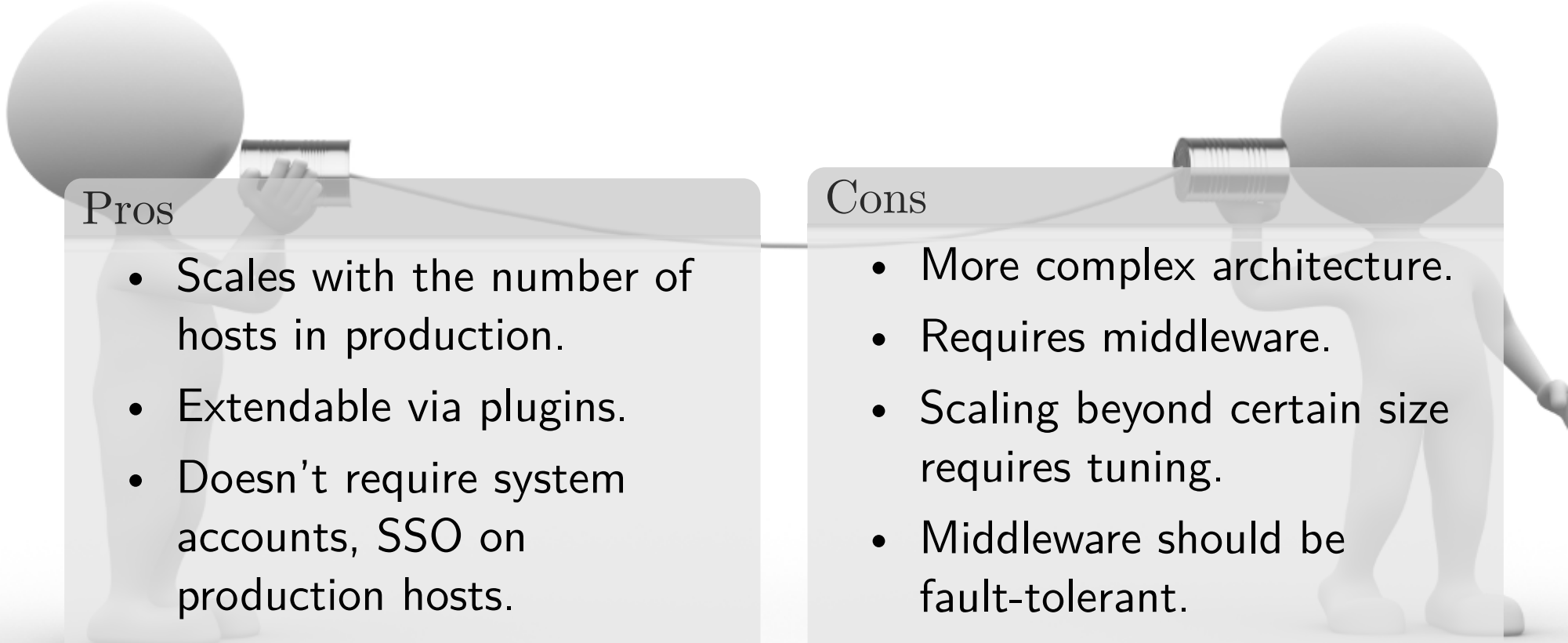
ssh in a loop (1)

Pros

- Simple and straightforward.
- Fast enough up to a certain number of hosts.
- Easy and cheap to adapt to perform different tasks.
- Scriptable.

Cons

- Scripts with hard-coded host names or IPs.
- Requires way too much information about the production environment.
- Cannot easily run remote commands which expect some kind of interaction.
- When the number of host grows, the risk of overlook reported problems increases.
- Requires dealing with account permissions, SSO, etc.



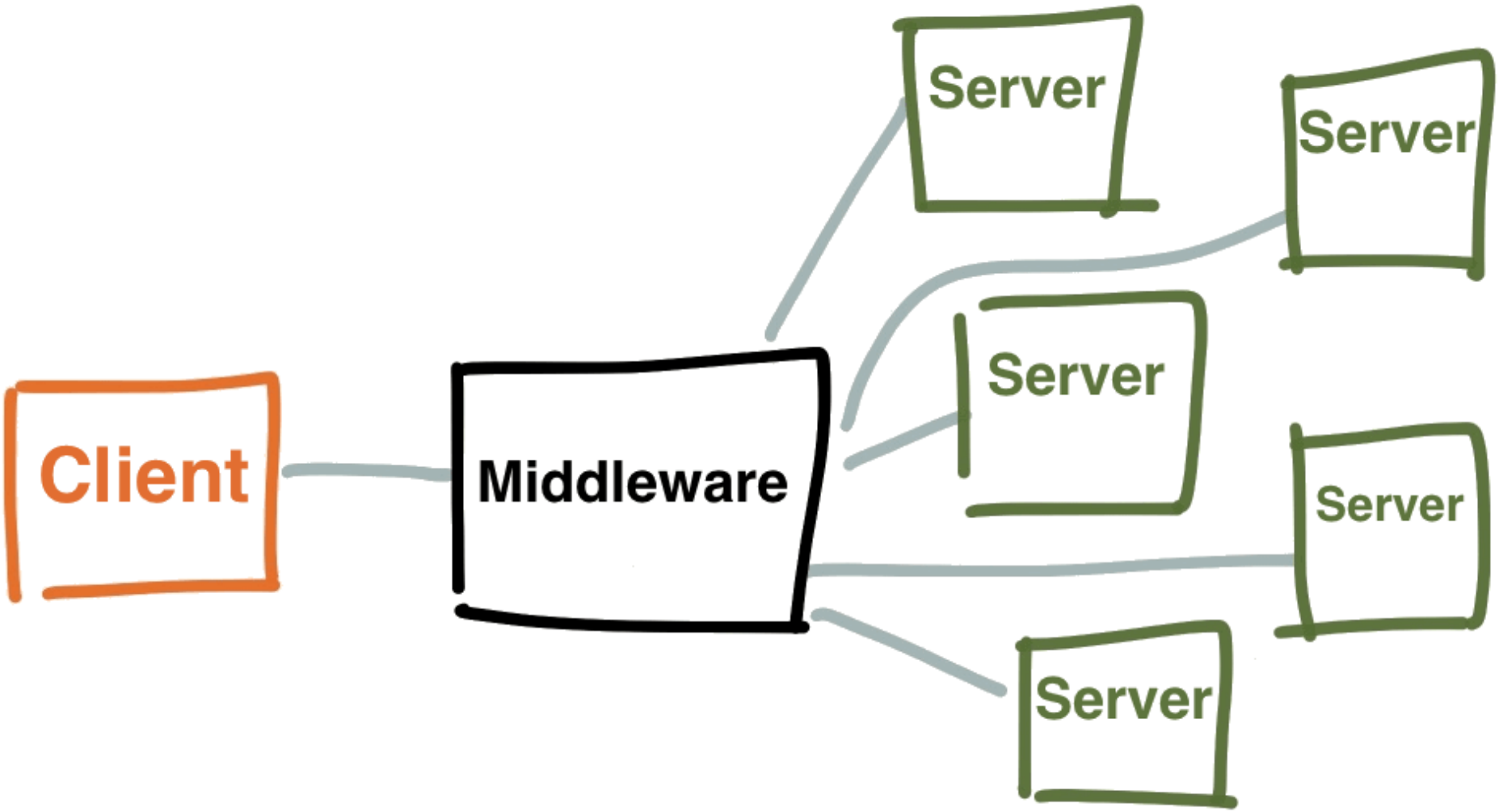
Pros

- Scales with the number of hosts in production.
- Extendable via plugins.
- Doesn't require system accounts, SSO on production hosts.
- Puppet module available for servers.

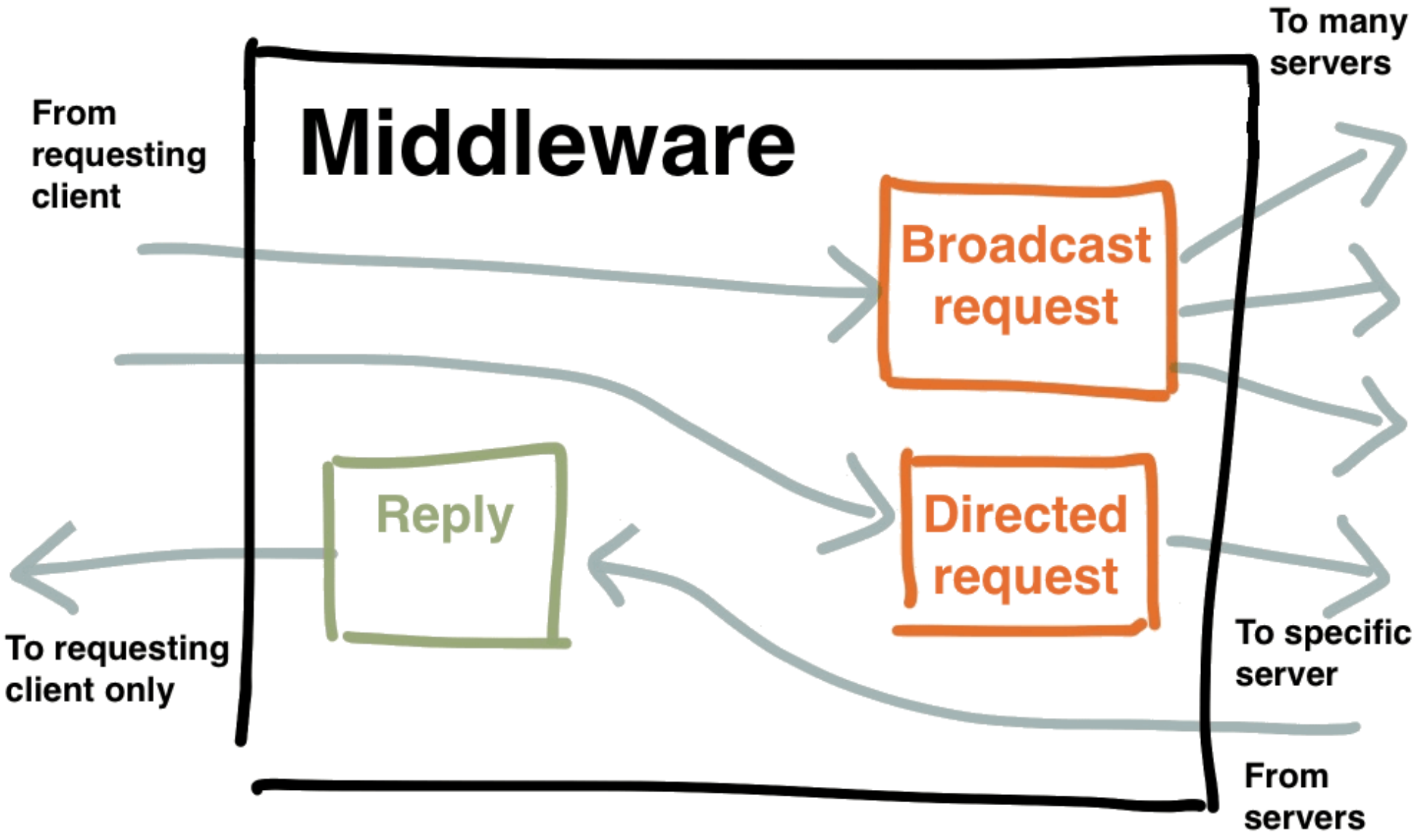
Cons

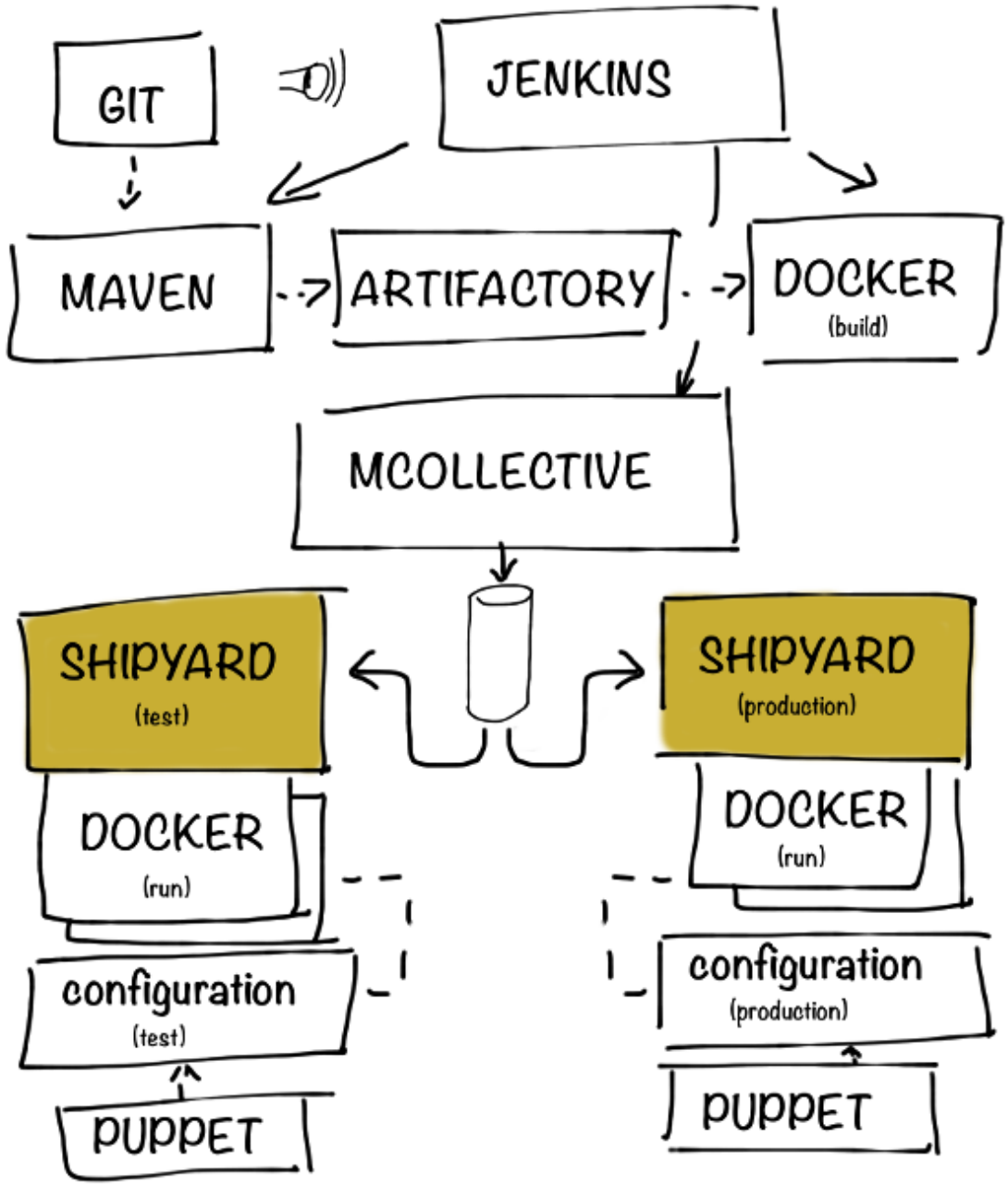
- More complex architecture.
- Requires middleware.
- Scaling beyond certain size requires tuning.
- Middleware should be fault-tolerant.
- Misconfigured setups can generate excessive traffic.

Architecture



Middleware



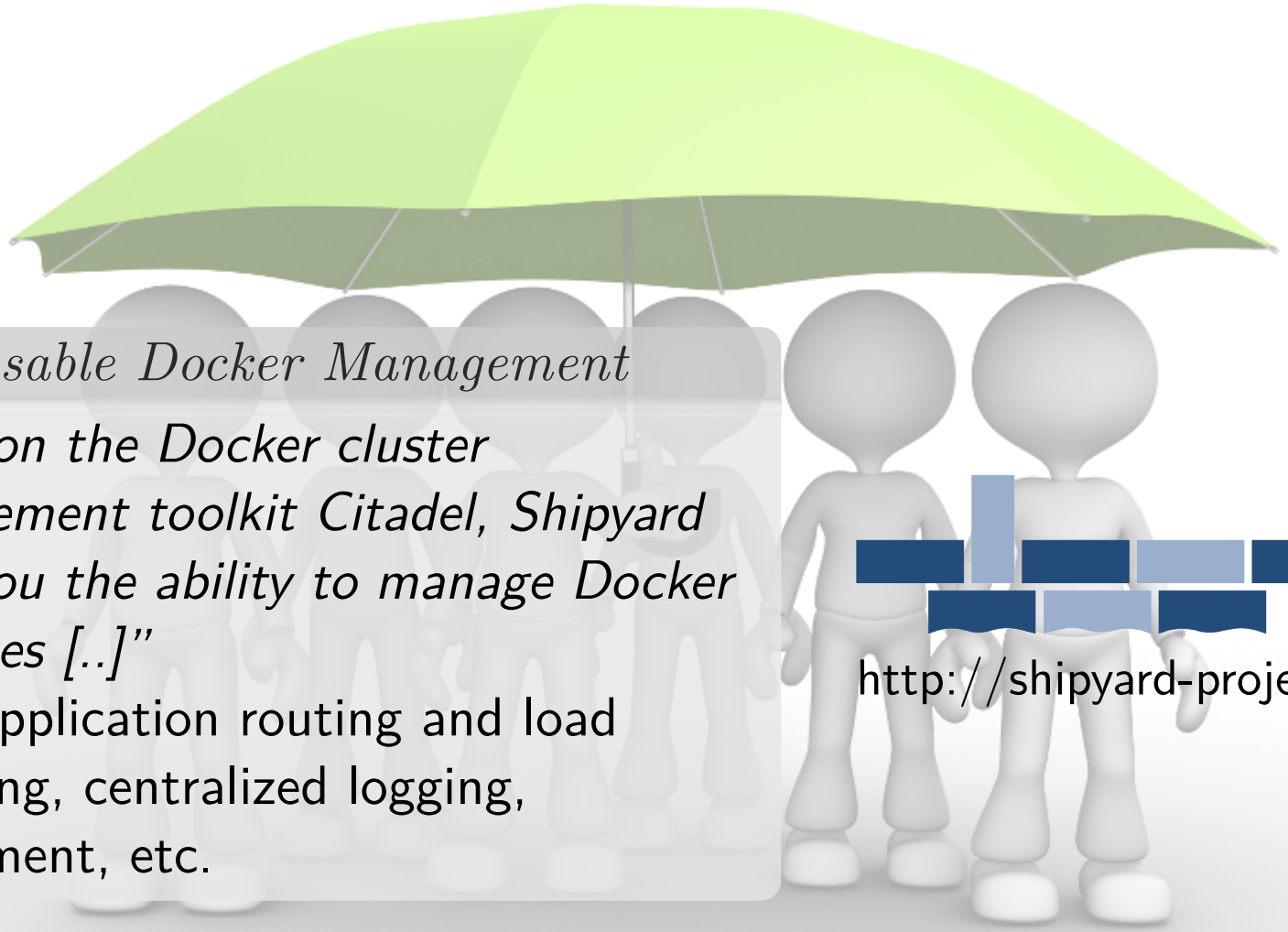


Citadel

"Citadel is a toolkit for scheduling containers on a Docker cluster."

http://citadeltoolkit.org

Shipyard

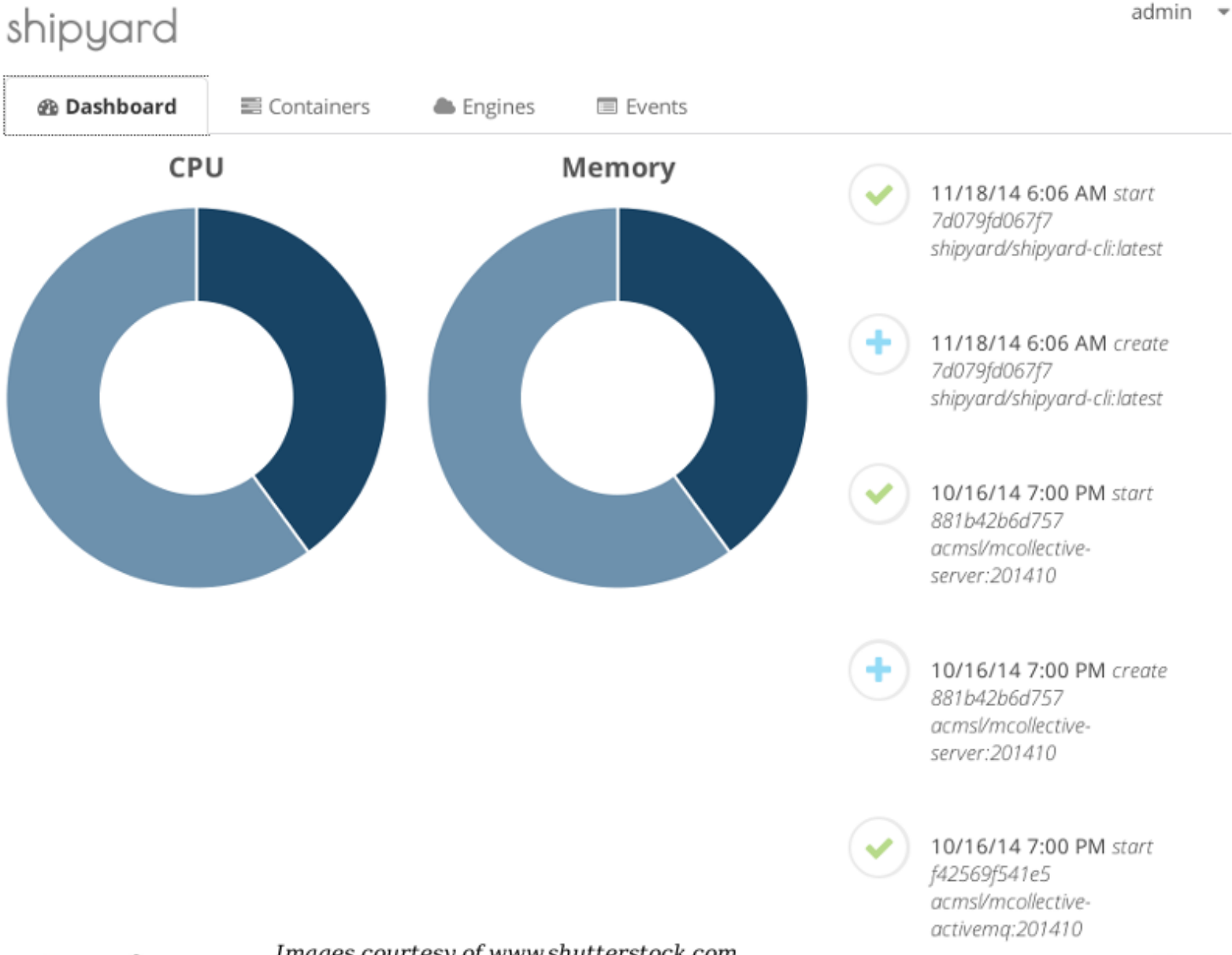


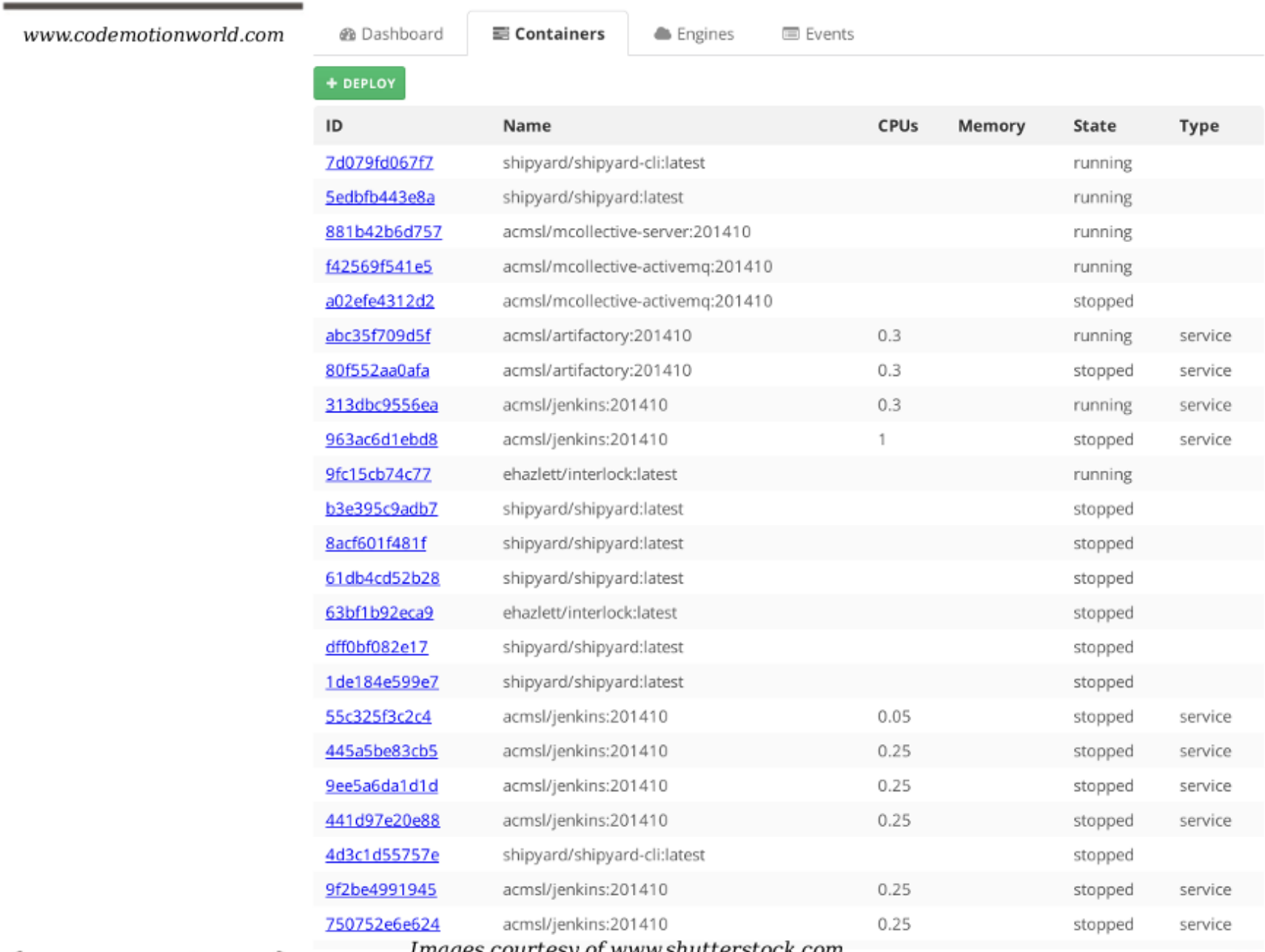
Composable Docker Management

"Built on the Docker cluster management toolkit Citadel, Shipyard gives you the ability to manage Docker resources [..]"

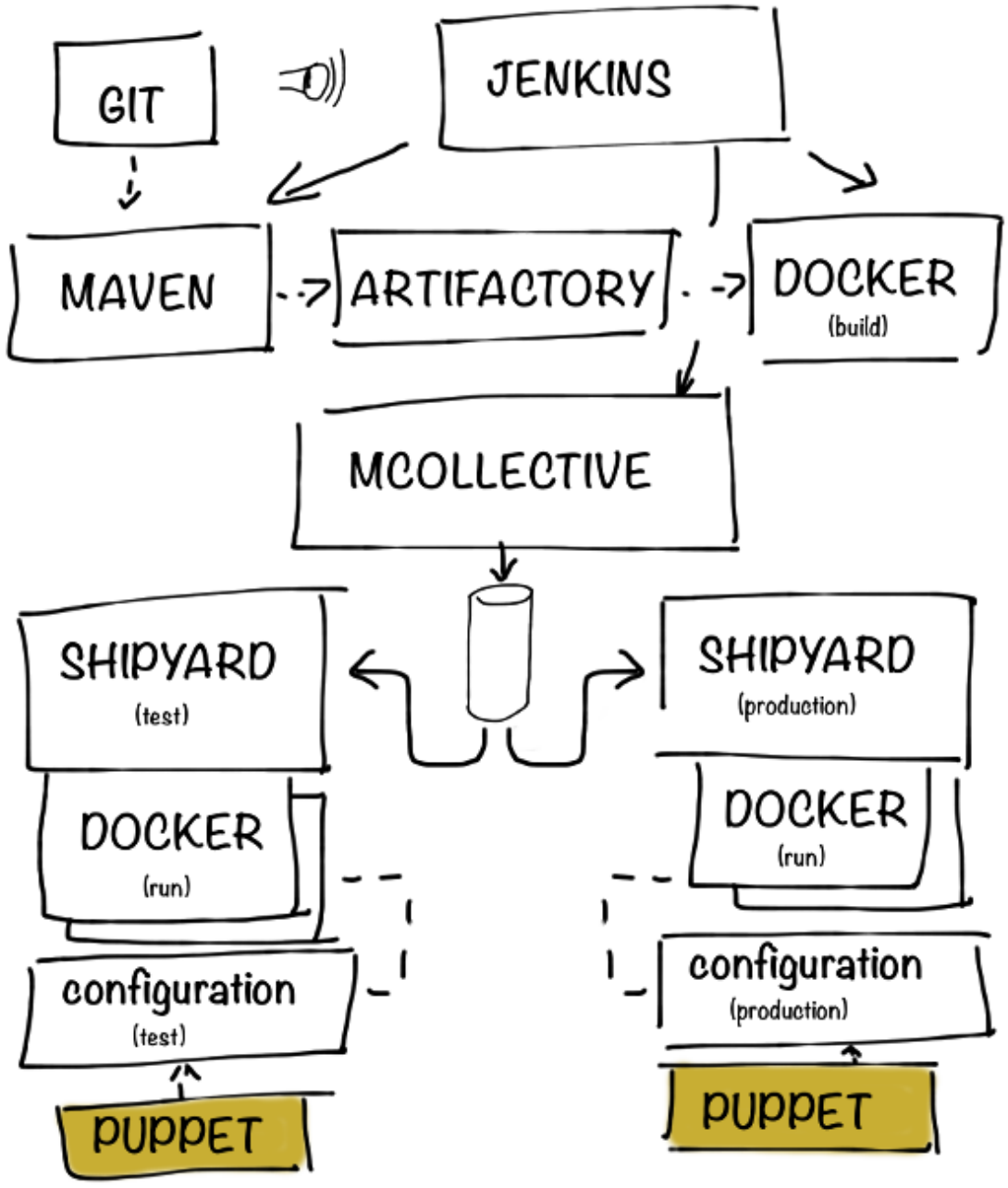
Plus: application routing and load balancing, centralized logging, deployment, etc.

http://shipyard-project.com





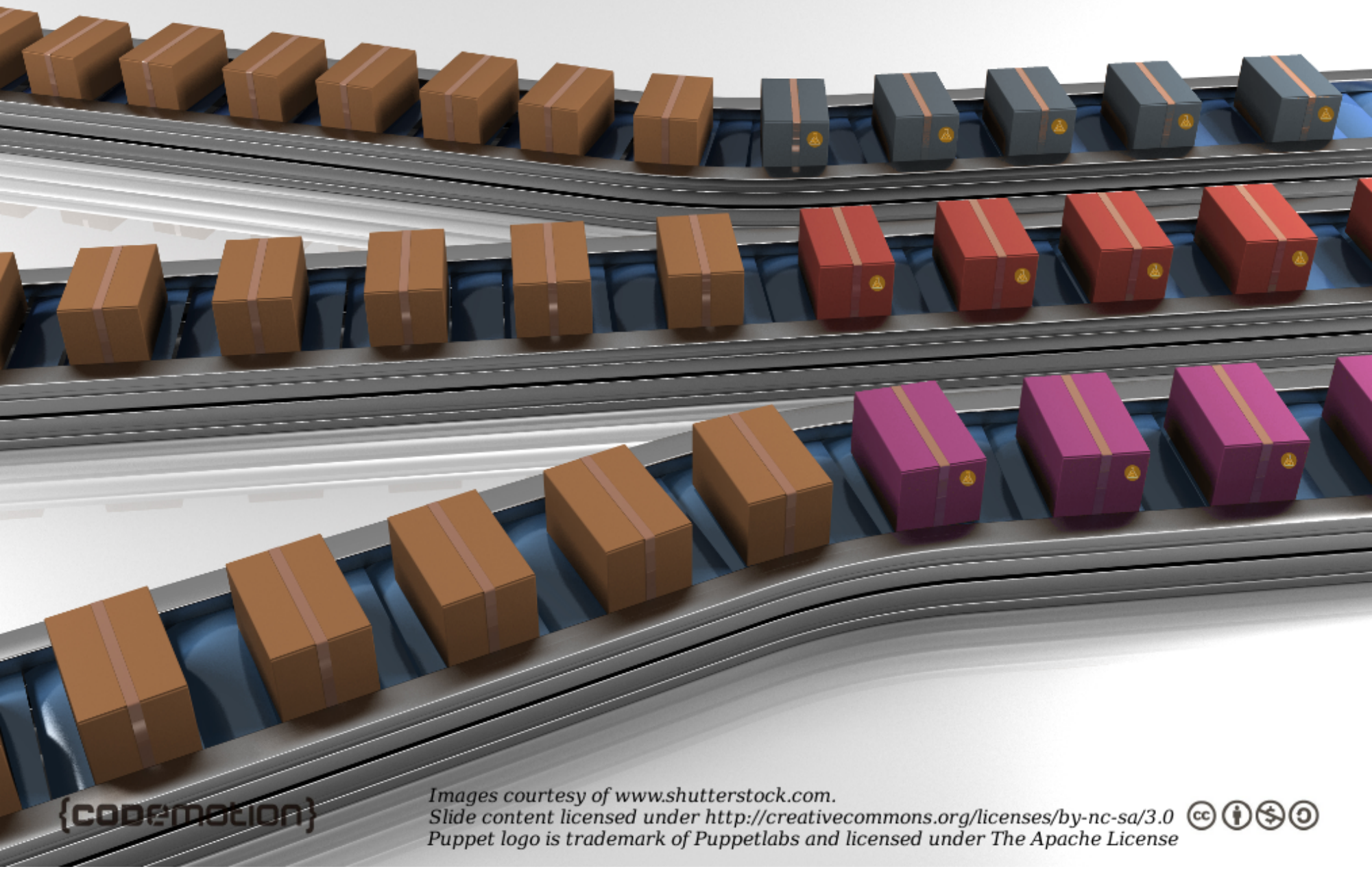
ID	Name	CPUs	Memory	State	Type
7d079fd067f7	shipyard/shipyard-cli:latest			running	
5edbfb443e8a	shipyard/shipyard:latest			running	
881b42b6d757	acmsl/mcollective-server:201410			running	
f42569f541e5	acmsl/mcollective-activemq:201410			running	
a02efe4312d2	acmsl/mcollective-activemq:201410			stopped	
abc35f709d5f	acmsl/artifactory:201410	0.3		running	service
80f552aa0afa	acmsl/artifactory:201410	0.3		stopped	service
313dbc9556ea	acmsl/jenkins:201410	0.3		running	service
963ac6d1ebd8	acmsl/jenkins:201410	1		stopped	service
9fc15cb74c77	ehazlett/interlock:latest			running	
b3e395c9adb7	shipyard/shipyard:latest			stopped	
8acf601f481f	shipyard/shipyard:latest			stopped	
61db4cd52b28	shipyard/shipyard:latest			stopped	
63bf1b92eca9	ehazlett/interlock:latest			stopped	
dff0bf082e17	shipyard/shipyard:latest			stopped	
1de184e599e7	shipyard/shipyard:latest			stopped	
55c325f3c2c4	acmsl/jenkins:201410	0.05		stopped	service
445a5be83cb5	acmsl/jenkins:201410	0.25		stopped	service
9ee5a6da1d1d	acmsl/jenkins:201410	0.25		stopped	service
441d97e20e88	acmsl/jenkins:201410	0.25		stopped	service
4d3c1d55757e	shipyard/shipyard-cli:latest			stopped	
9f2be4991945	acmsl/jenkins:201410	0.25		stopped	service
750752e6e624	acmsl/jenkins:201410	0.25		stopped	service



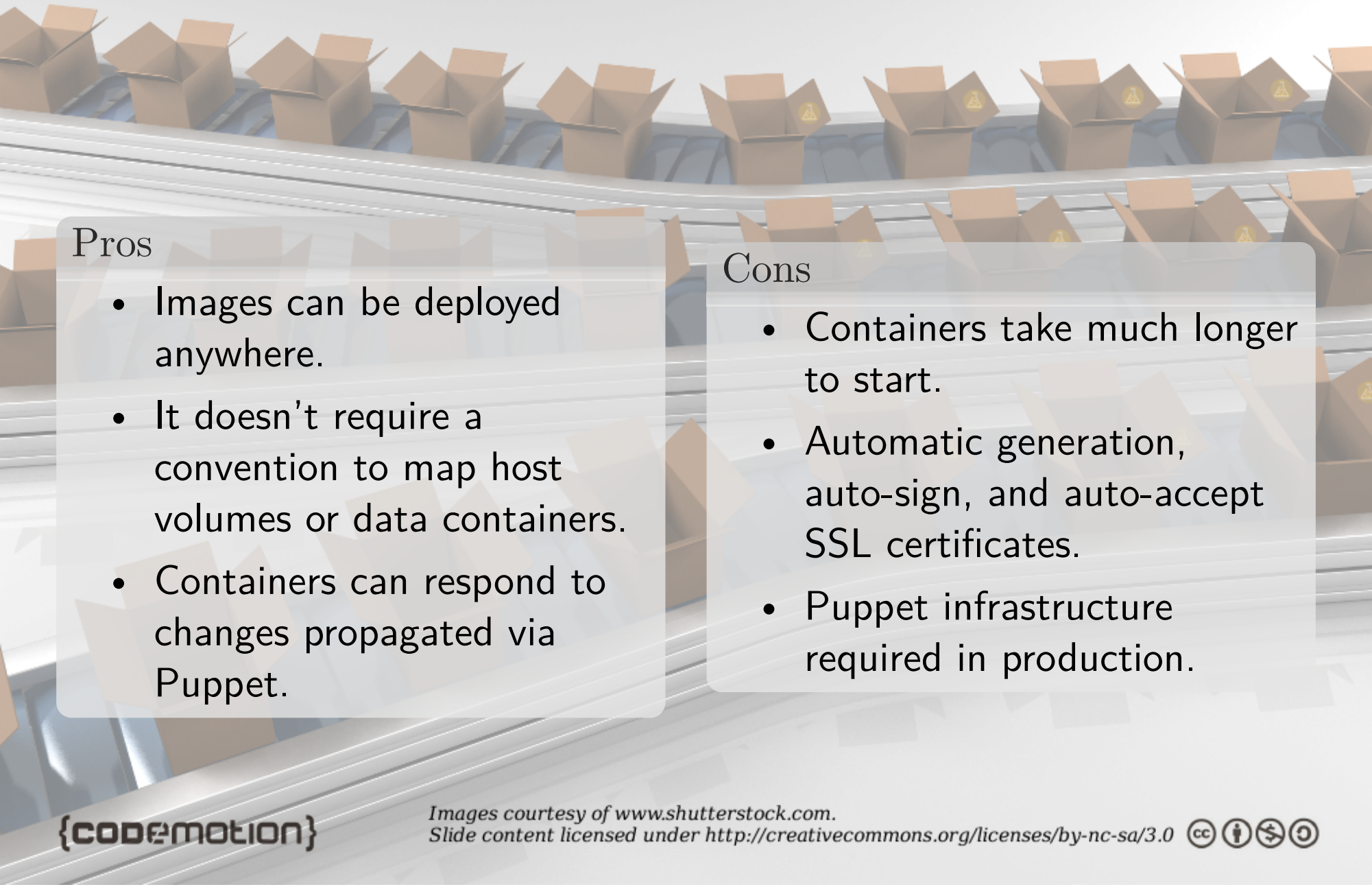
Puppet

"Puppet manages your servers: you describe machine configurations in an easy-to-read declarative language, and Puppet will bring your systems into the desired state and keep them there."

http://www.puppetlabs.com



Puppet on guests



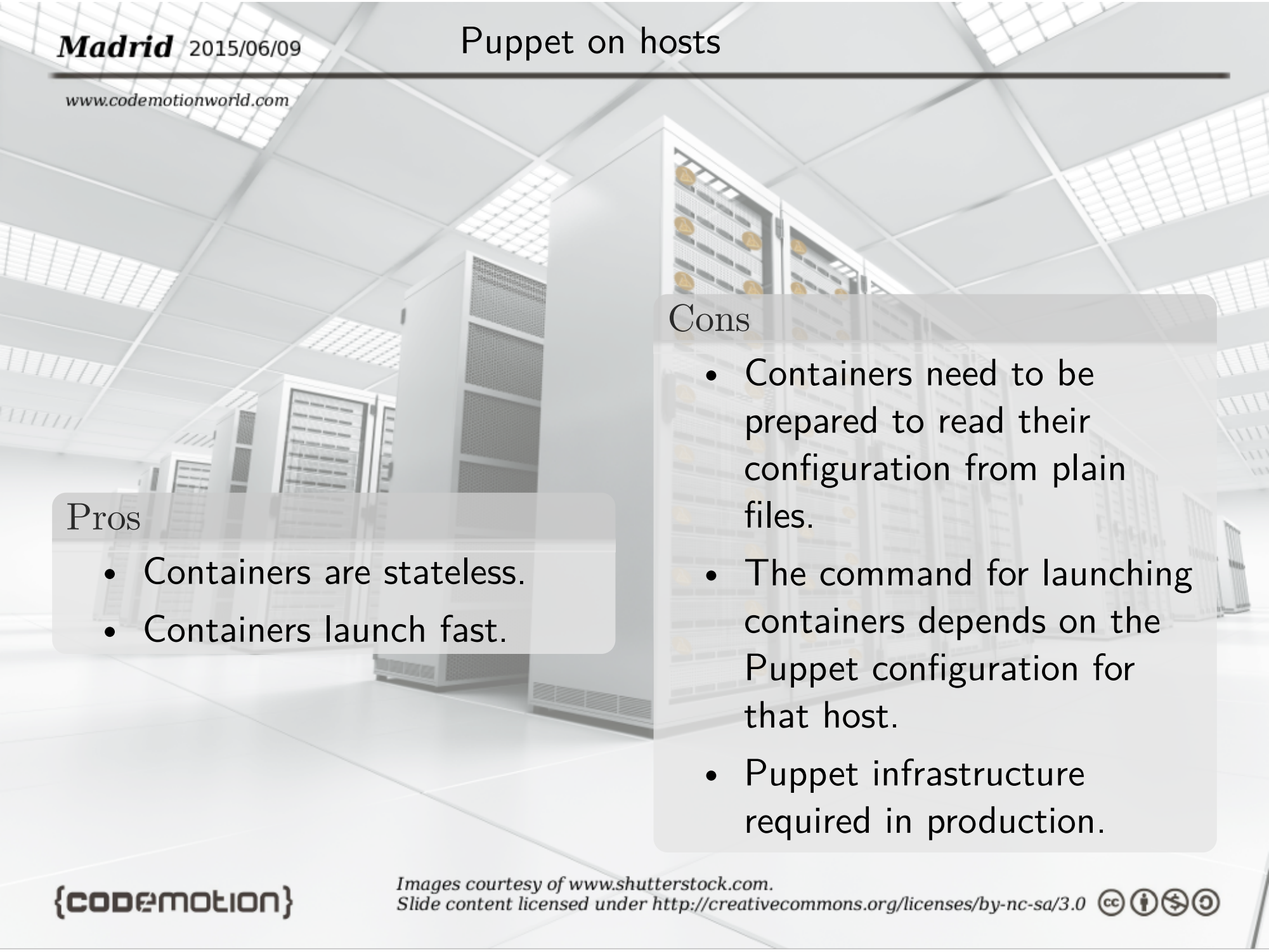
Pros

- Images can be deployed anywhere.
- It doesn't require a convention to map host volumes or data containers.
- Containers can respond to changes propagated via Puppet.

Cons

- Containers take much longer to start.
- Automatic generation, auto-sign, and auto-accept SSL certificates.
- Puppet infrastructure required in production.

Puppet on hosts



Pros

- Containers are stateless.
- Containers launch fast.

Cons

- Containers need to be prepared to read their configuration from plain files.
- The command for launching containers depends on the Puppet configuration for that host.
- Puppet infrastructure required in production.

Puppet to manage data-container images

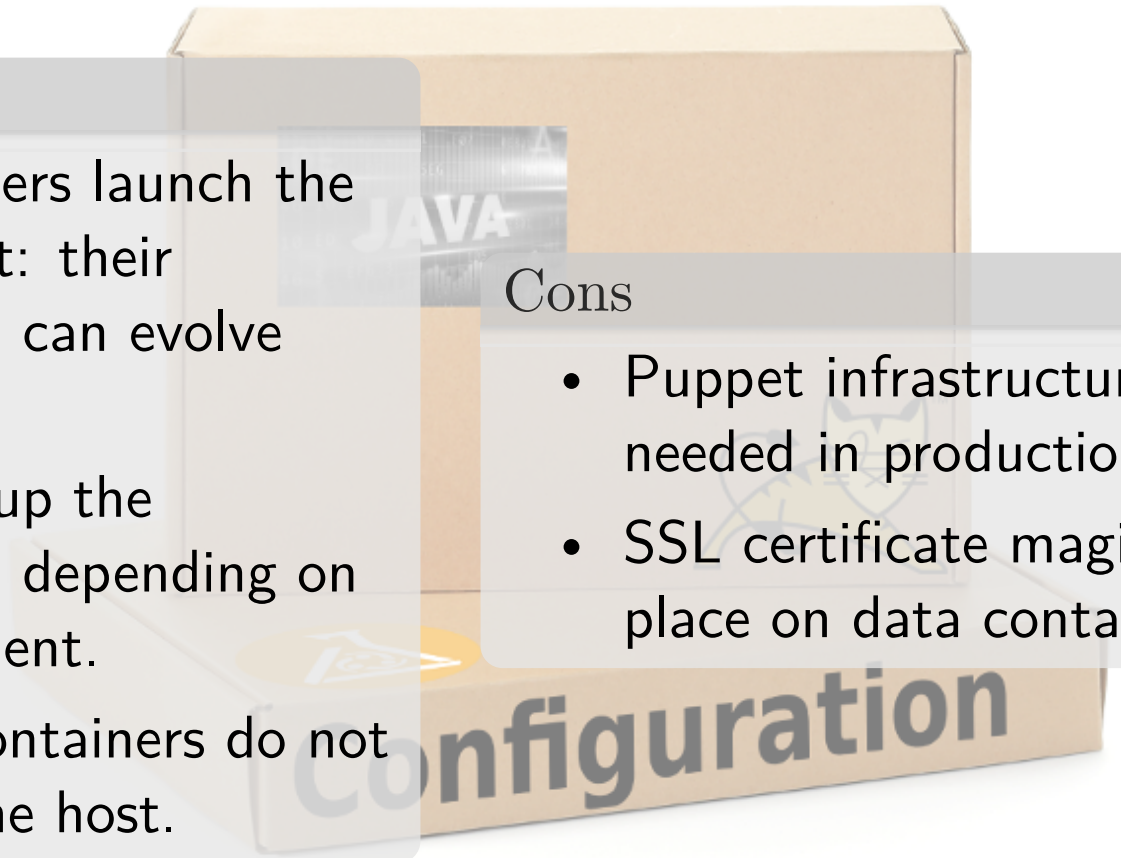
Environment isolated in data containers

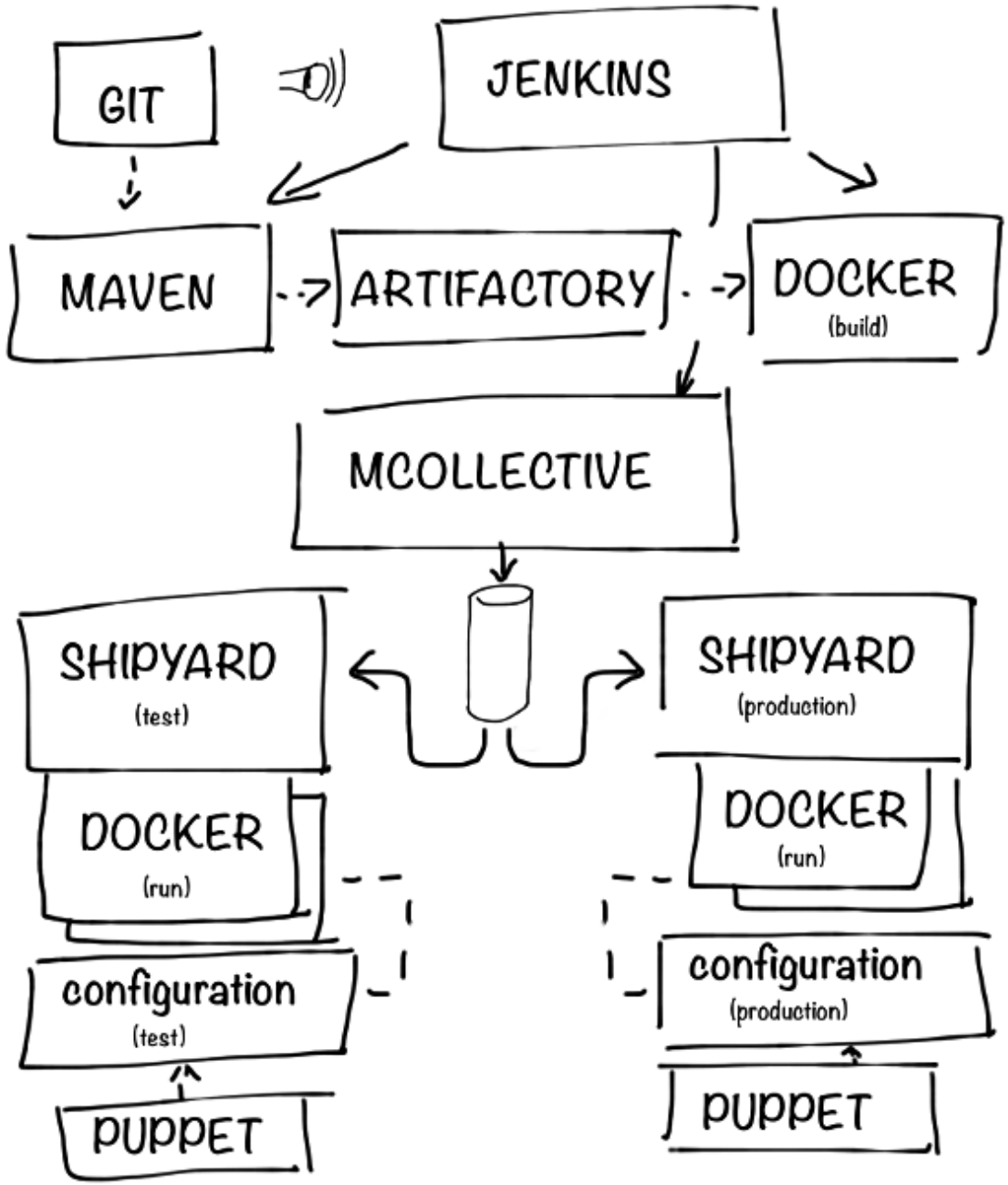
Pros

- Data containers launch the Puppet agent: their configuration can evolve over time.
- Puppet sets up the configuration depending on the environment.
- Launching containers do not depend on the host.

Cons

- Puppet infrastructure needed in production.
- SSL certificate magic takes place on data containers.





Now what?

First things first

- Clone my repos: `http://github.com/rydnr/dockerfile` and `http://github.com/rydnr/dry-wit`
- Take `http://github.com/rydnr/acmsl-jenkins-configs` as a template for **get-new-version** job.
- Build your custom Delivery Pipeline.
- Make Jenkins generate Docker images and push them to a private index.

And then?

Customize and test

- Build mcollective-client and mcollective-server images.
- Install shipyard and mcollective server agent in a test environment.
- Launch docker containers from the mcollective client, via mcollective shell agent.
- Try Interlock in the path to Continuous Deployment!