Using Docker for Testing

Carlos Sanchez
@csanchez

About

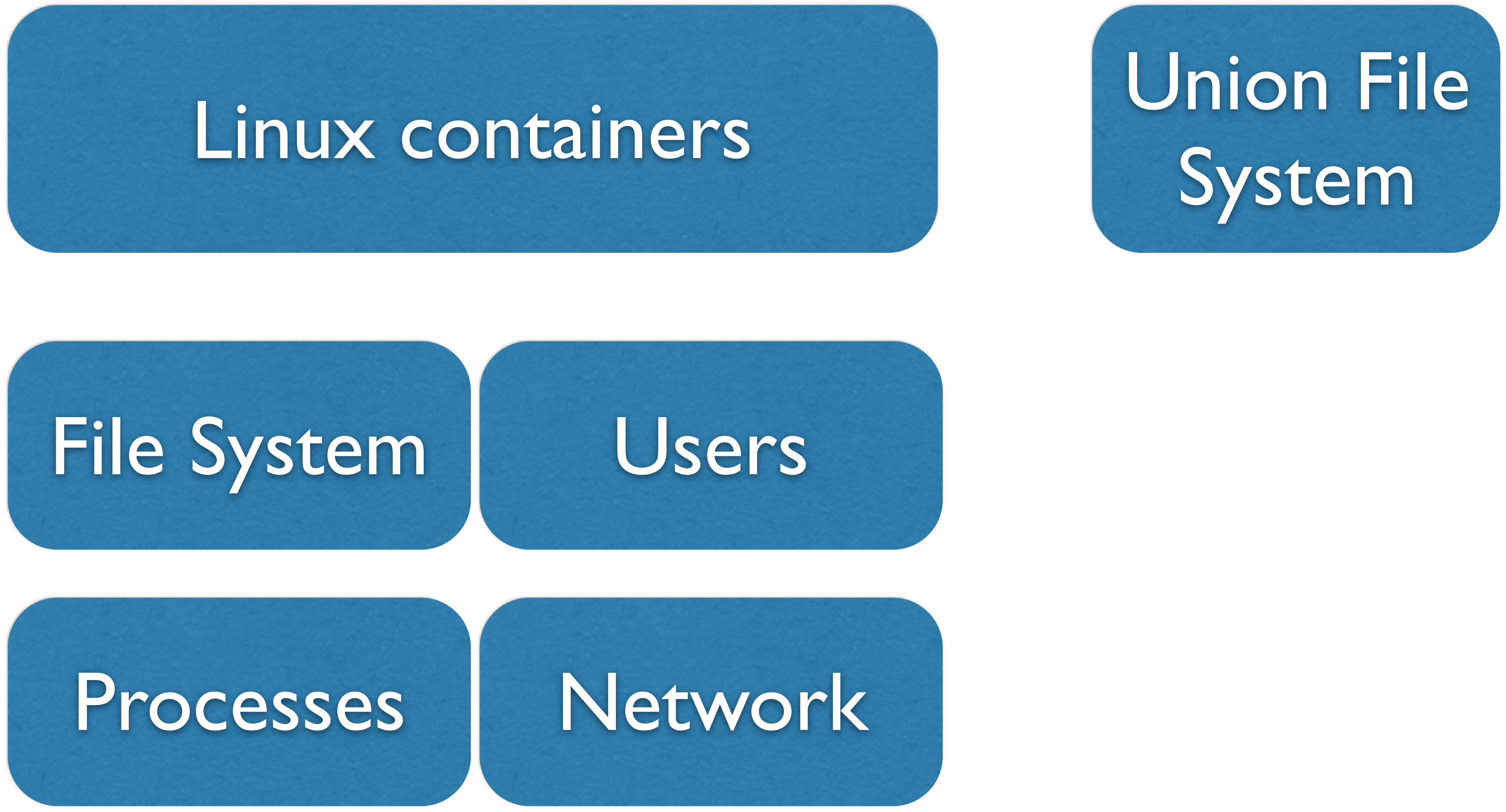
Senior Software Engineer @ CloudBees

Author of Jenkins Kubernetes plugin

Long time OSS contributor at Apache Maven, Eclipse, Puppet,…

Containers & micro services

Docker


Linux containers
Union File System
File System
Users
Processes
Network

But it is not trivial

Docker

Linux required

but

Docker Machine (formerly Boot2Docker) to the rescue

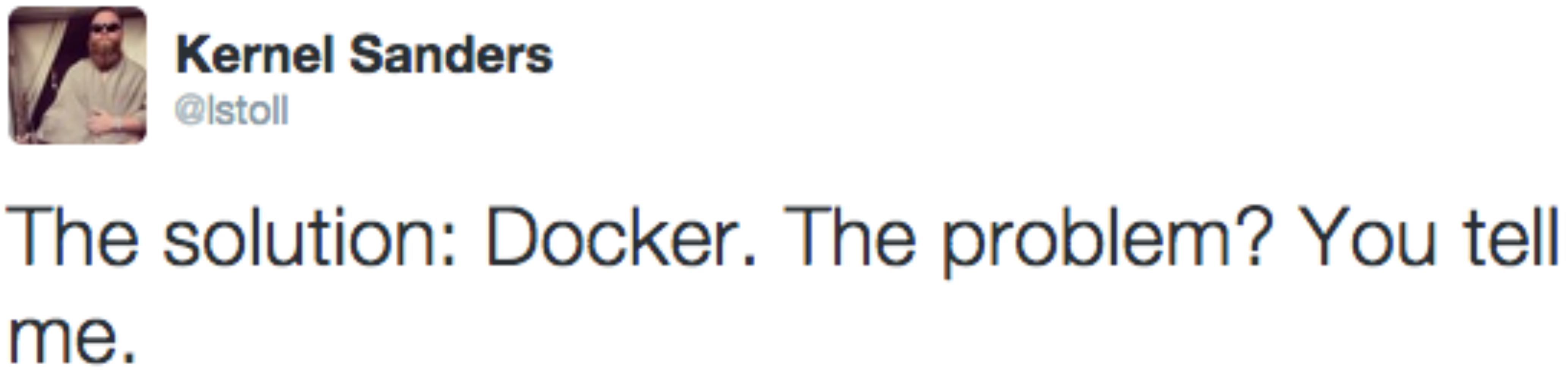
OS X

Windows

Docker

Build once, run anywhere (kind of)

- Bare metal
- Virtual Machines
- Cloud
- Docker



Kernel Sanders
@lstoll

The solution: Docker. The problem? You tell me.

developer oriented

Dependency hell

installation nightmares

“it ran on my machine”

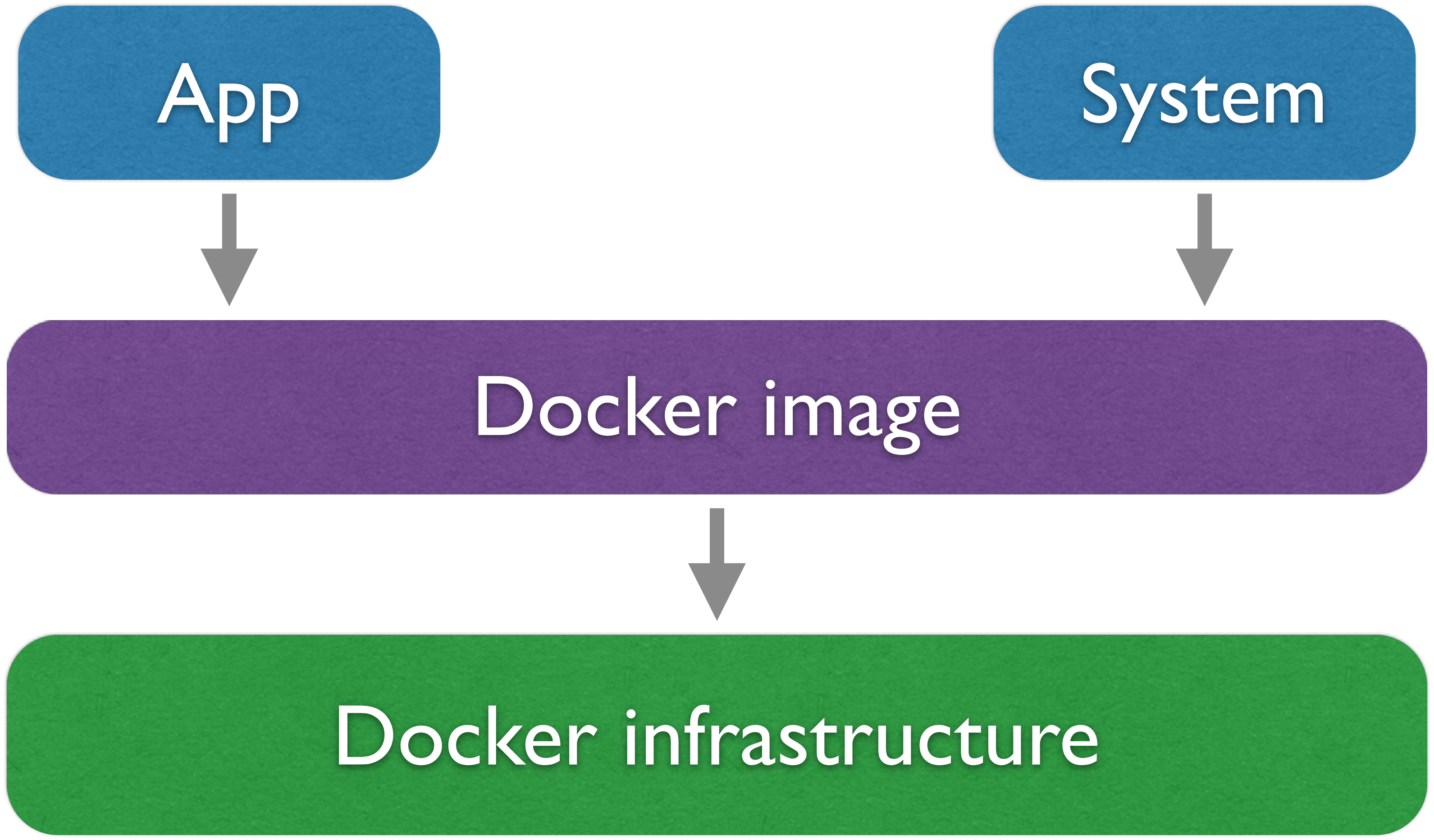
ops oriented

no need to know internals of apps

focus on OPs problems

- (scale, monitoring,…)
- clearer deliverables from dev

Docker delivery



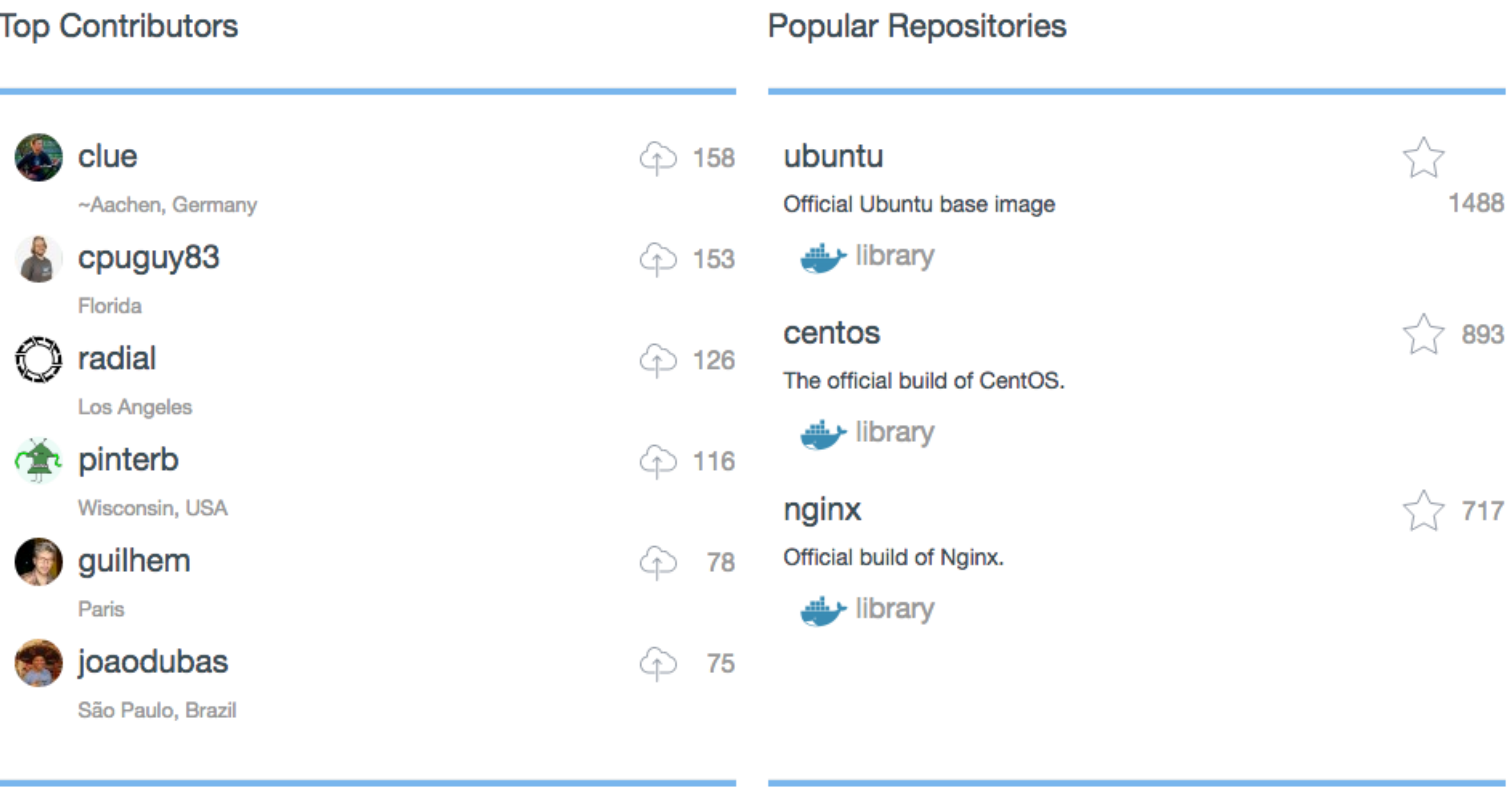
Official Repositories

MySQL
mongoDB
PostgreSQL

RAILS
Ruby
Java
WordPress

redis
NGINX
node

ubuntu
debian
GNU/Linux
CentOS



Top Contributors

clue	~Aachen, Germany	158
cpuguy83	Florida	153
radial	Los Angeles	126
pinterb	Wisconsin, USA	116
guilhem	Paris	78
joaodubas	São Paulo, Brazil	75

Popular Repositories

ubuntu	Official Ubuntu base image	library	1488
centos	The official build of CentOS.	library	893
nginx	Official build of Nginx.	library	717

Related projects

Docker Machine

Provision Docker engines

- VirtualBox, replaces boot2docker !
- Amazon EC2
- Microsoft Azure
- Google Compute Engine
- OpenStack
- Rackspace
- VMware
- …

Docker Swarm

- Clustering for Docker containers
- Using the same API
- Integrates with Mesos / Mesosphere
- And planned
    - Amazon EC2 Container Service (ECS)
    - Google Kubernetes
    - IBM Bluemix Container Service
    - Joyent Smart Data Center
    - Microsoft Azure

Docker Compose

Orchestration of multi-container apps

Based on Fig

Defined by:

- containers
- configuration
- links
- volumes

Apache Mesos

A *distributed systems kernel*

Docker Containerizer

Marathon & Chronos

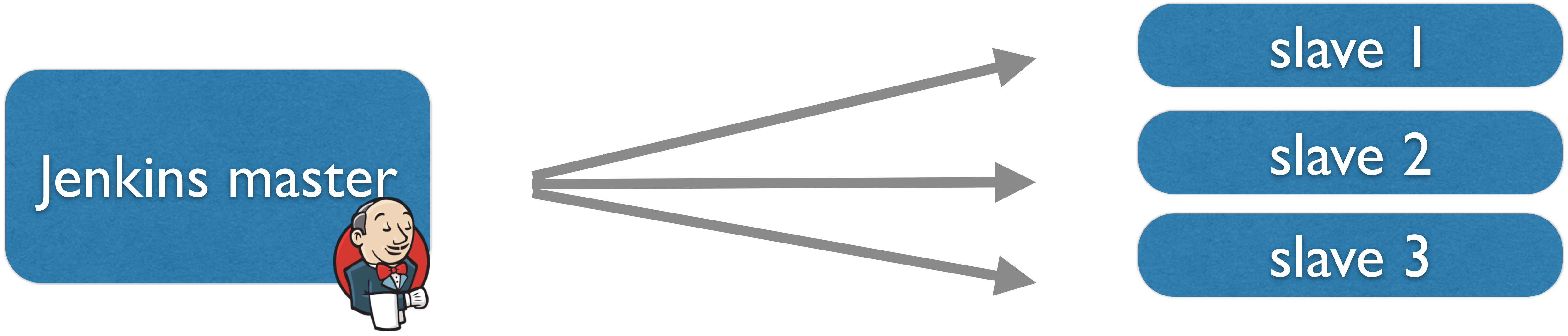
Docker & Jenkins

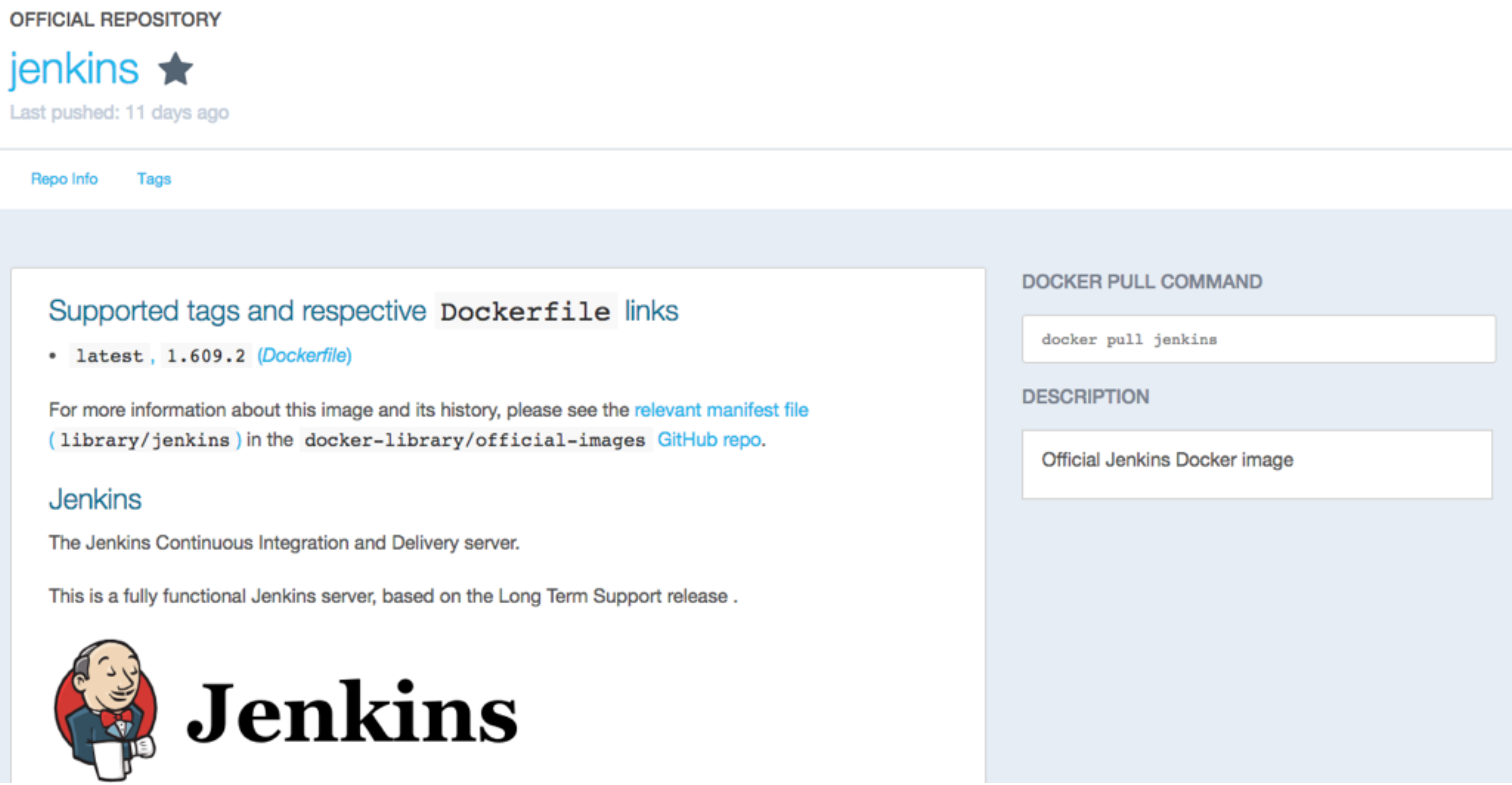
What are we trying to solve

Different projects, different requirements

- languages (Java, Ruby,…)
- tools (Maven, Ant, …)
- system libraries (OpenSSL, …)
- operating systems (Debian, Red Hat,…)
- external dependencies (MySQL, Postgres)

Initial solution





OFFICIAL REPOSITORY

jenkins ★

Last pushed: 11 days ago

Repo Info   Tags

Supported tags and respective `Dockerfile` links

- `latest`, `1.609.2` (Dockerfile)

For more information about this image and its history, please see the relevant manifest file (`library/jenkins`) in the `docker-library/official-images` GitHub repo.

Jenkins

The Jenkins Continuous Integration and Delivery server.

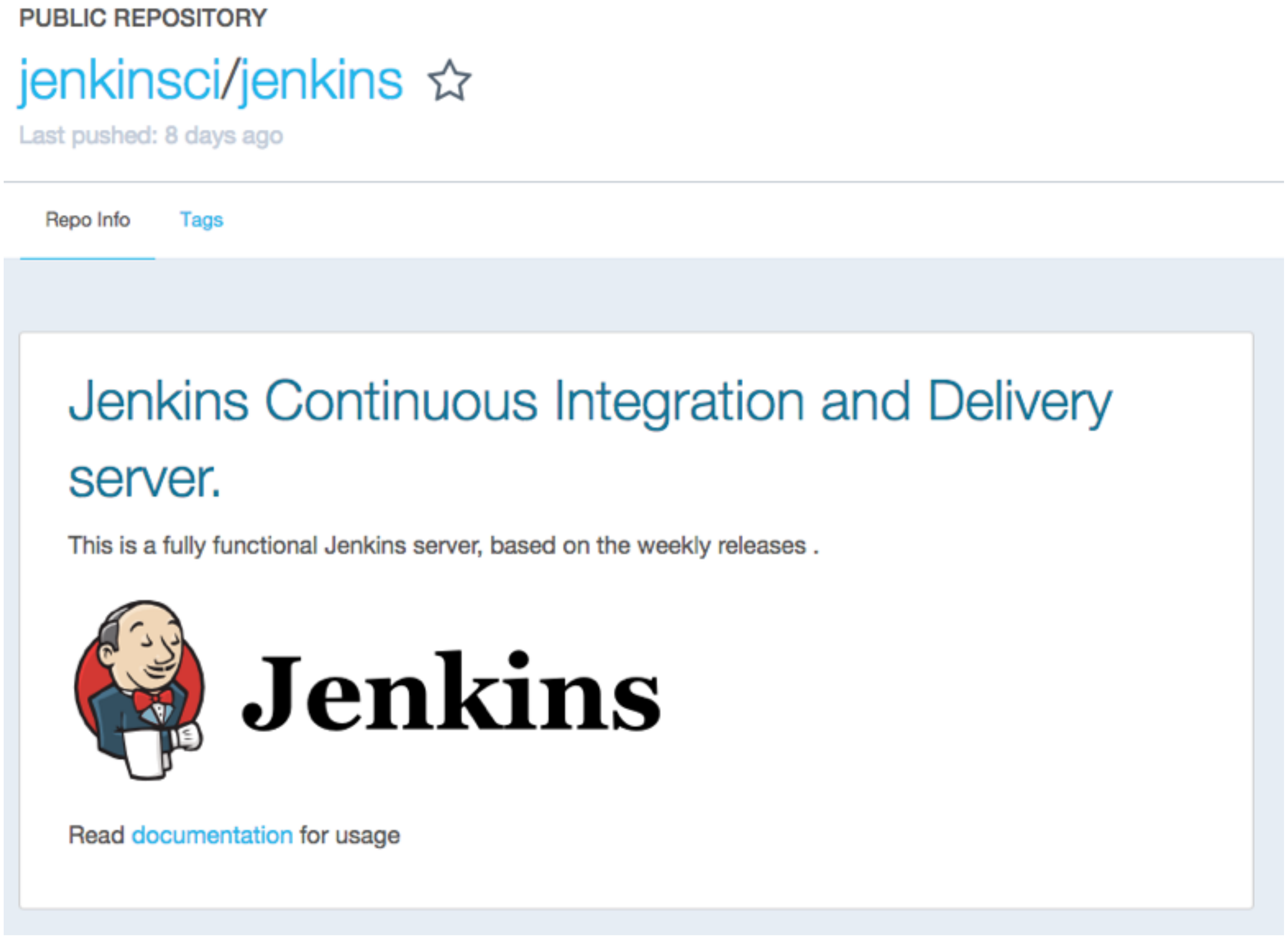
This is a fully functional Jenkins server, based on the Long Term Support release .

DOCKER PULL COMMAND

```
docker pull jenkins
```

DESCRIPTION

Official Jenkins Docker image


PUBLIC REPOSITORY

jenkinsci/jenkins ☆

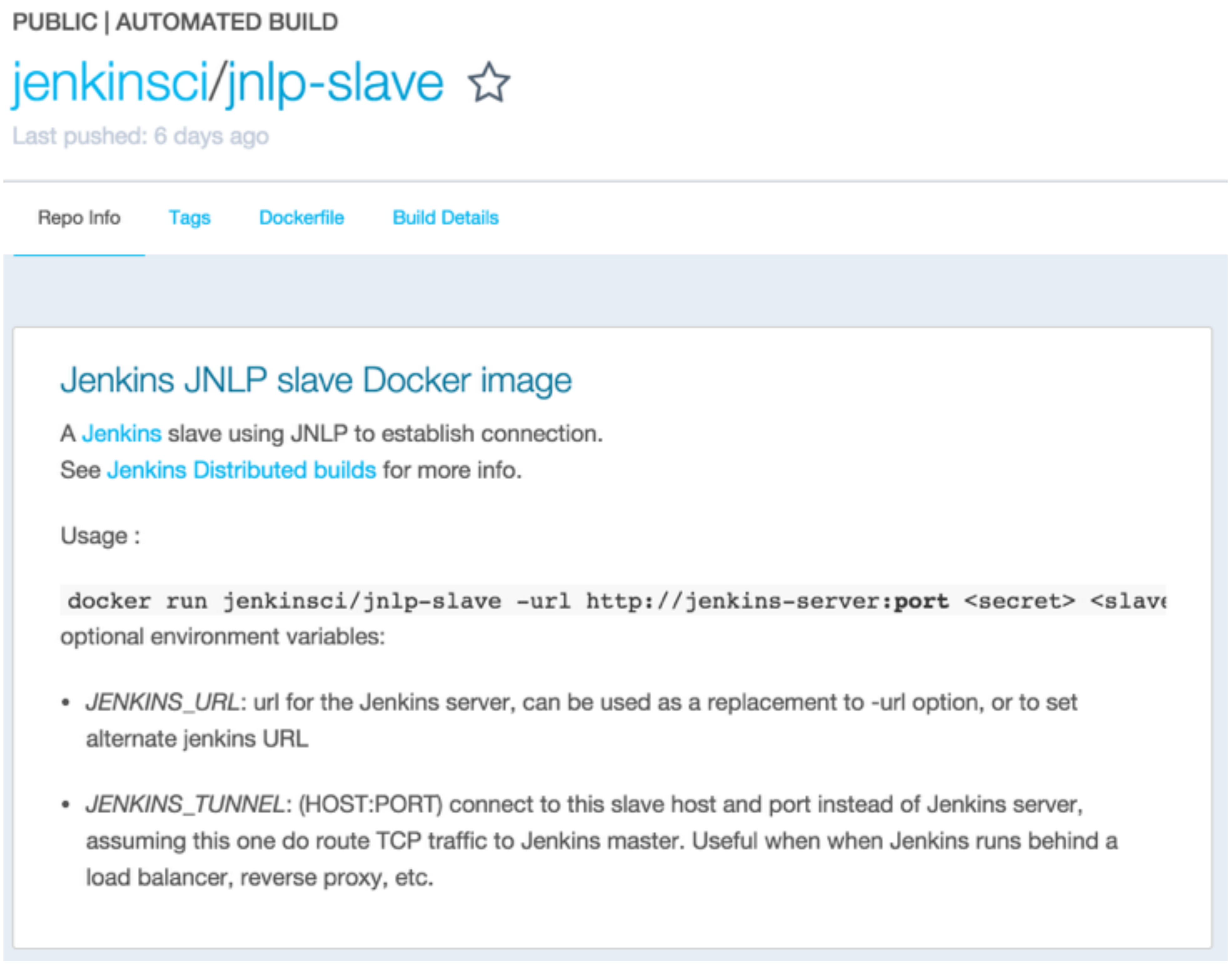
Last pushed: 8 days ago

Repo Info Tags

Jenkins Continuous Integration and Delivery server.

This is a fully functional Jenkins server, based on the weekly releases .

Read documentation for usage



Jenkins JNLP slave Docker image

A Jenkins slave using JNLP to establish connection.
See Jenkins Distributed builds for more info.

Usage :

```
docker run jenkinsci/jnlp-slave -url http://jenkins-server:port <secret> <slave
```

optional environment variables:

- *JENKINS_URL*: url for the Jenkins server, can be used as a replacement to -url option, or to set alternate jenkins URL
- *JENKINS_TUNNEL*: (HOST:PORT) connect to this slave host and port instead of Jenkins server, assuming this one do route TCP traffic to Jenkins master. Useful when when Jenkins runs behind a load balancer, reverse proxy, etc.

Dockerfile

```
# A Debian based image
FROM jenkinsci/jnlp-slave

RUN apt-get update \
    && apt-get install -y mysql \
    && rm -rf /var/lib/apt/lists/*
```

Dockerfile

```
FROM centos

RUN yum -y install openjdk-8 mysql

ENV JENKINS_REMOTING_VERSION 2.52
ENV HOME /home/jenkins

RUN useradd -c "Jenkins user" -d $HOME -m jenkins
RUN curl --create-dirs -sSLo /usr/share/jenkins/remoting-
$JENKINS_REMOTING_VERSION.jar http://repo.jenkins-ci.org/public/
org/jenkins-ci/main/remoting/$JENKINS_REMOTING_VERSION/remoting-
$JENKINS_REMOTING_VERSION.jar \
  && chmod 755 /usr/share/jenkins

COPY jenkins-slave.sh /usr/local/bin/jenkins-slave.sh

USER jenkins

VOLUME /home/jenkins

ENTRYPOINT ["/usr/local/bin/jenkins-slave.sh"]
```

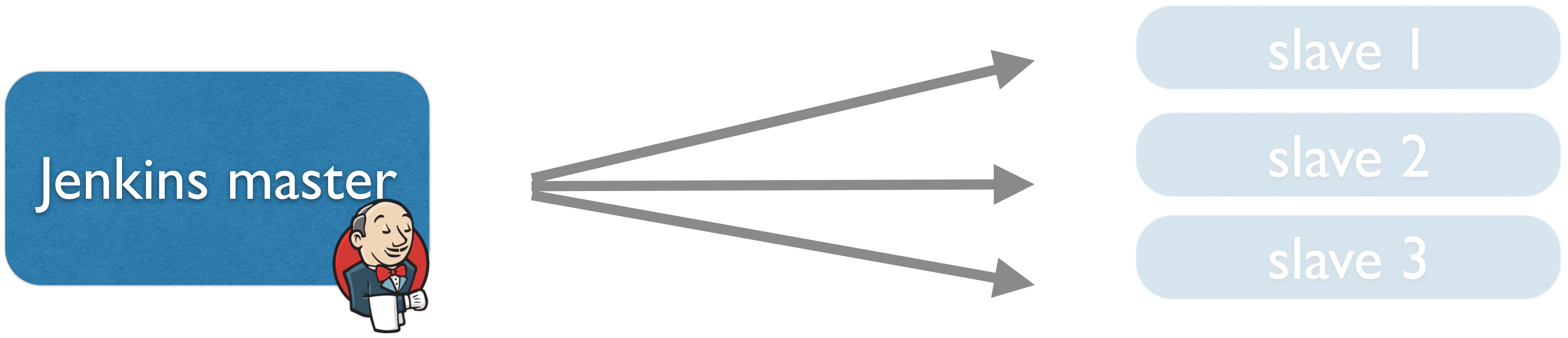
Docker plugin

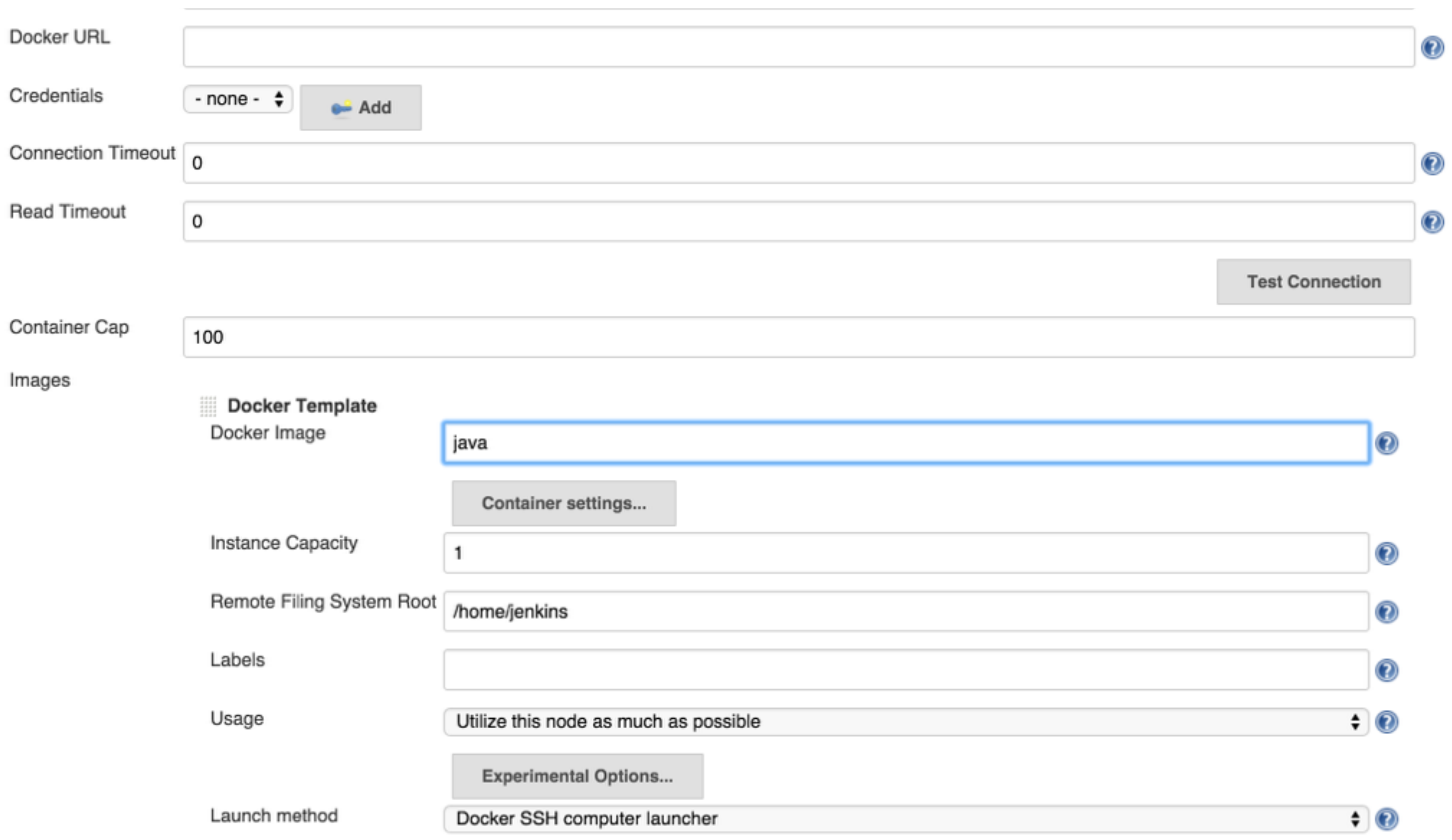
Docker plugin

As a plugin

on demand slaves

https://github.com/jenkinsci/docker-plugin


Jenkins master
slave 1
slave 2
slave 3



Docker URL	
Credentials	- none -
Connection Timeout	0
Read Timeout	0
Container Cap	100

Images

Docker Template

Docker Image	java
Instance Capacity	1
Remote Filing System Root	/home/jenkins
Labels	
Usage	Utilize this node as much as possible
Launch method	Docker SSH computer launcher

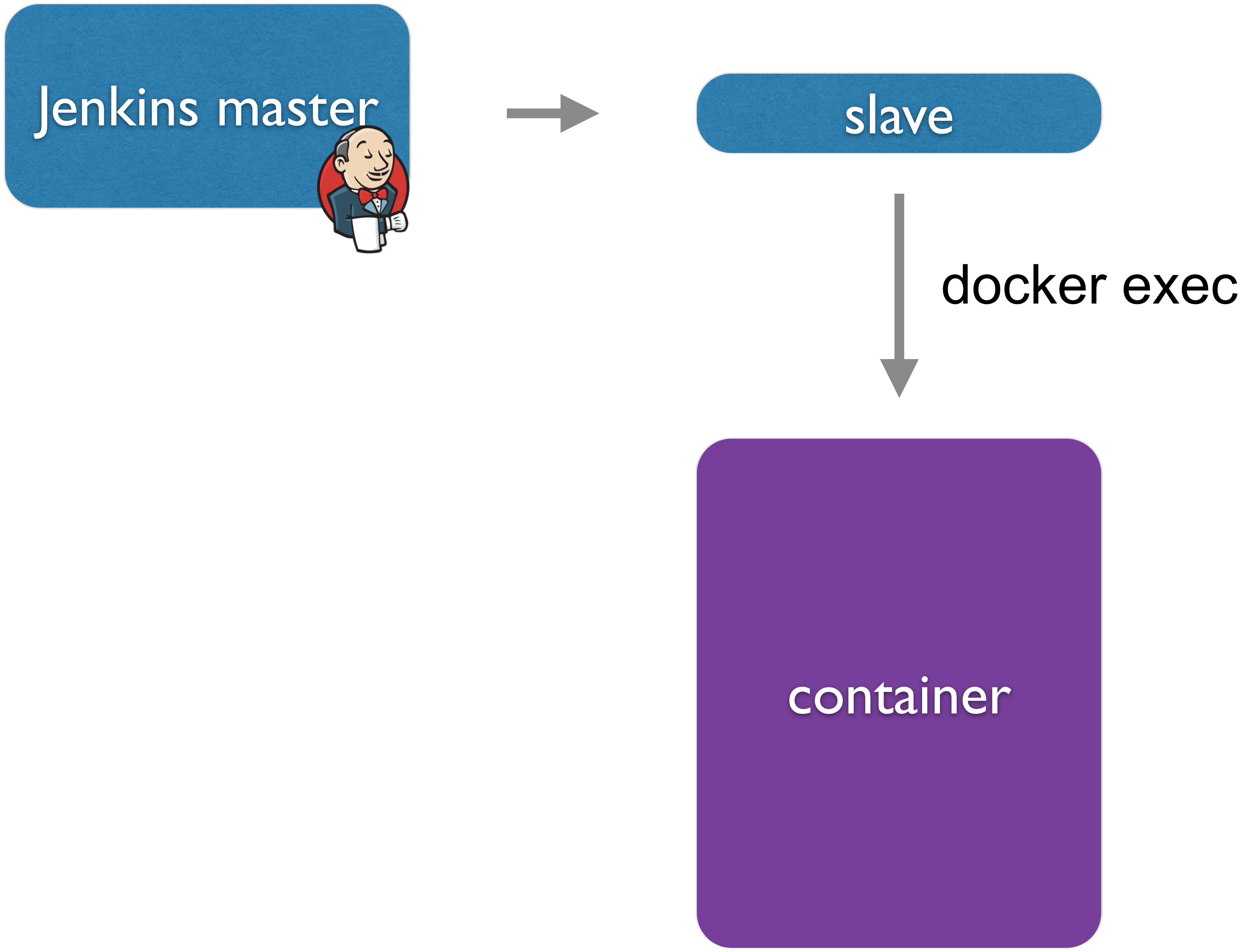
CloudBees Docker Custom Build Environment Plugin

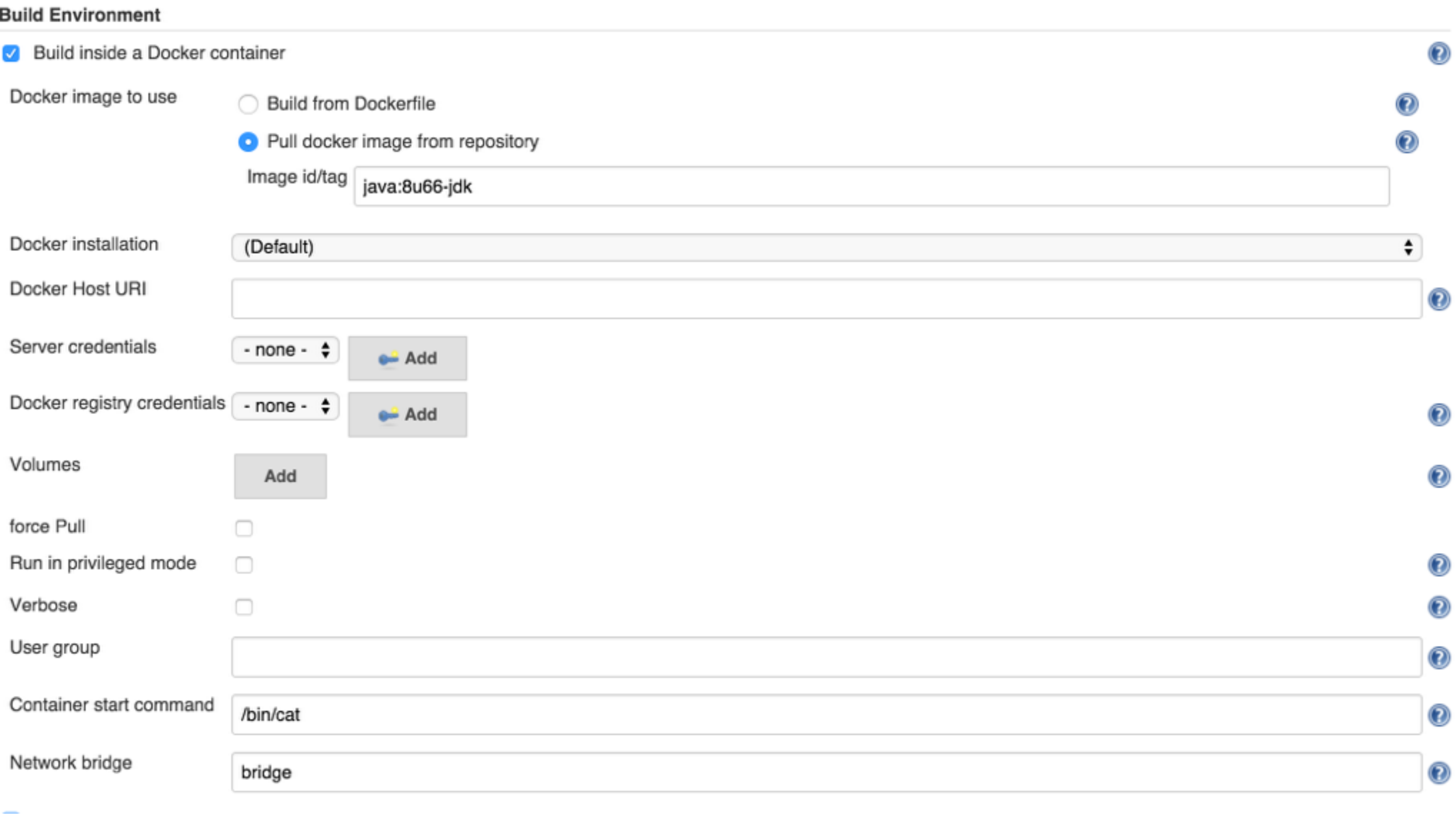
Custom Build Environment

Avoid dependencies in Jenkins

Containers are completely isolated

Use any executor/slave


Jenkins master
slave
docker exec
container



Build Environment

☑ Build inside a Docker container

Docker image to use	○ Build from Dockerfile
	● Pull docker image from repository
	Image id/tag: java:8u66-jdk
Docker installation	(Default)
Docker Host URI	
Server credentials	- none -
Docker registry credentials	- none -
Volumes	
force Pull	☐
Run in privileged mode	☐
Verbose	☐
User group	
Container start command	/bin/cat
Network bridge	bridge

Dockerfile

```
FROM centos

RUN yum -y install openjdk-8 mysql

~~ENV JENKINS_REMOTING_VERSION 2.52~~
~~ENV HOME /home/jenkins~~

~~RUN useradd -c "Jenkins user" -d $HOME -m jenkins~~
~~RUN curl --create-dirs -sSLo /usr/share/jenkins/remoting-~~
~~$JENKINS_REMOTING_VERSION.jar http://repo.jenkins-ci.org/public/~~
~~org/jenkins-ci/main/remoting/$JENKINS_REMOTING_VERSION/remoting-~~
~~$JENKINS_REMOTING_VERSION.jar \~~
~~  && chmod 755 /usr/share/jenkins~~

~~COPY jenkins-slave.sh /usr/local/bin/jenkins-slave.sh~~

~~USER jenkins~~

~~VOLUME /home/jenkins~~

~~ENTRYPOINT ["/usr/local/bin/jenkins-slave.sh"]~~
```

Custom Build Environment

Take advantage of all pre-built Docker images

- java, ruby, python, maven,…

Docker images are now a deliverable

Docker images are part of the pipeline

Build/test/deploy images

Deliver as any other artifact

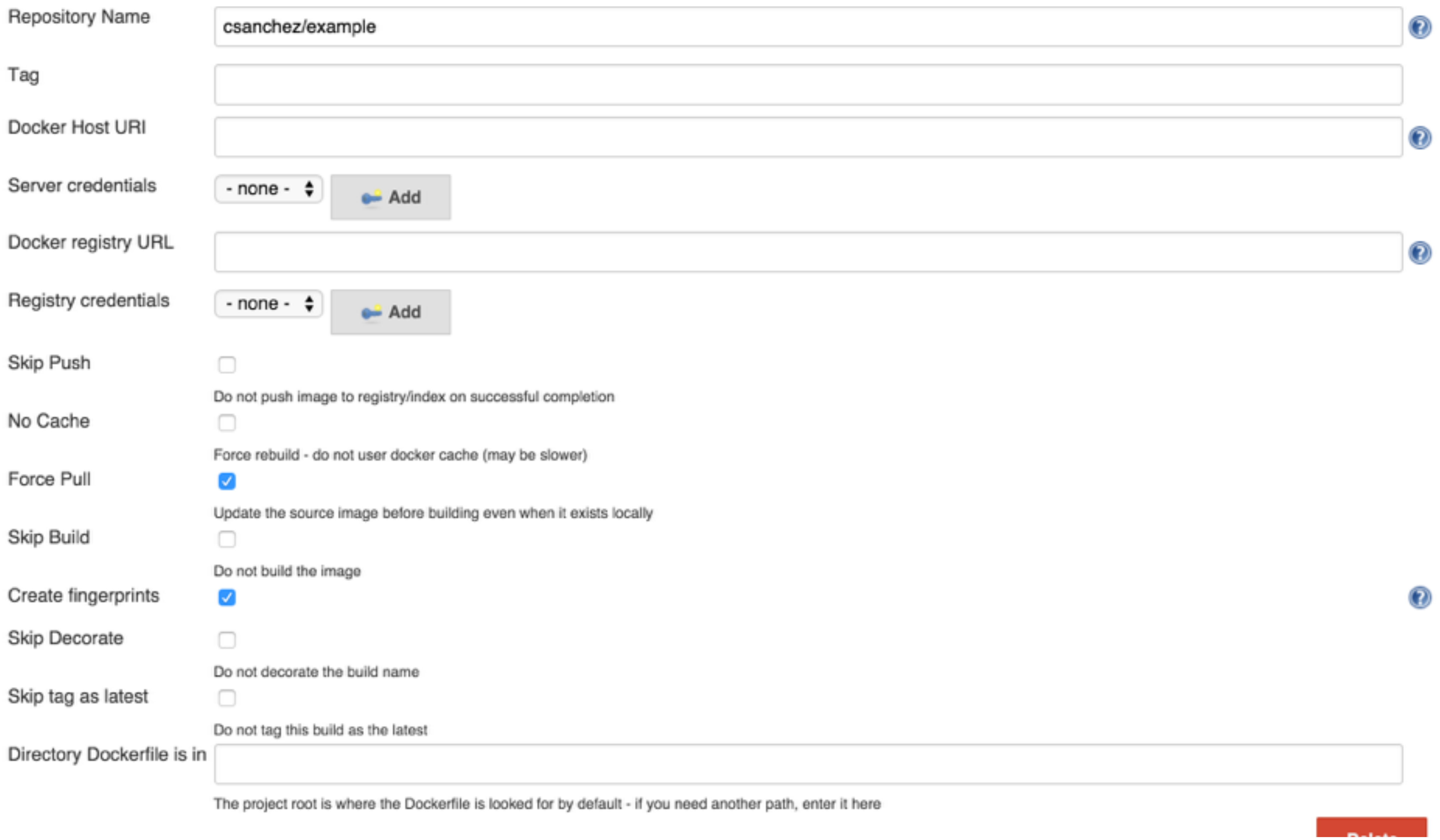
- even if not used to run production systems

More Docker!

Jenkins plugins

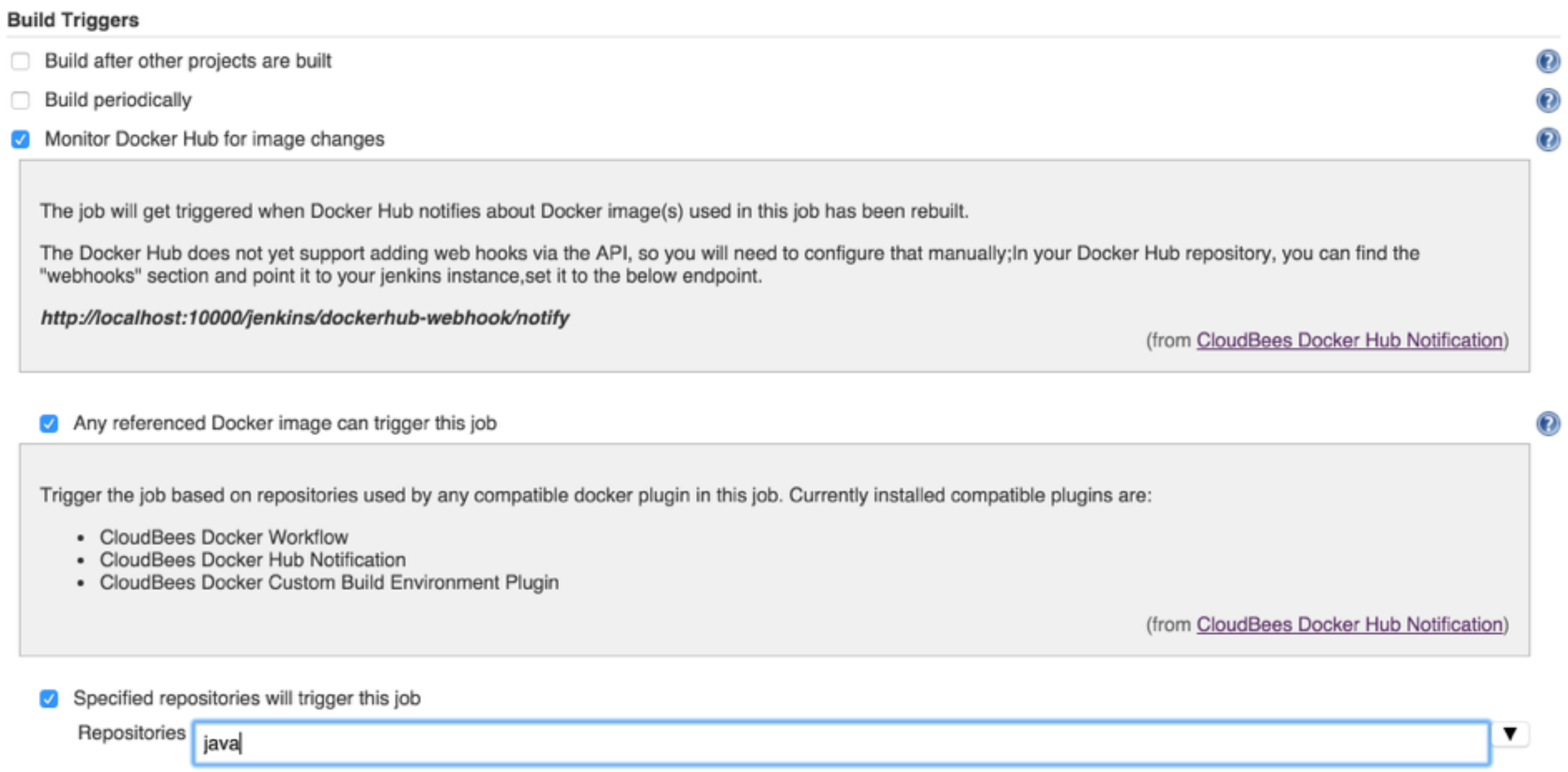
- Docker
- CloudBees Docker Custom Build Environment
- CloudBees Docker Build and Publish
- CloudBees Docker Hub Notification
- CloudBees Docker Traceability
- docker-build-step
- Docker workflow
- Kubernetes
- Mesos

Build and Publish



Repository Name	csanchez/example
Tag	
Docker Host URI	
Server credentials	- none - Add
Docker registry URL	
Registry credentials	- none - Add
Skip Push	☐ Do not push image to registry/index on successful completion
No Cache	☐ Force rebuild - do not user docker cache (may be slower)
Force Pull	✓ Update the source image before building even when it exists locally
Skip Build	☐ Do not build the image
Create fingerprints	✓
Skip Decorate	☐ Do not decorate the build name
Skip tag as latest	☐ Do not tag this build as the latest
Directory Dockerfile is in	The project root is where the Dockerfile is looked for by default - if you need another path, enter it here

Docker Hub Notification



Build Triggers

- [ ] Build after other projects are built
- [ ] Build periodically
- [✓] Monitor Docker Hub for image changes

The job will get triggered when Docker Hub notifies about Docker image(s) used in this job has been rebuilt.

The Docker Hub does not yet support adding web hooks via the API, so you will need to configure that manually;In your Docker Hub repository, you can find the "webhooks" section and point it to your jenkins instance,set it to the below endpoint.

http://localhost:10000/jenkins/dockerhub-webhook/notify

(from CloudBees Docker Hub Notification)

- [✓] Any referenced Docker image can trigger this job

Trigger the job based on repositories used by any compatible docker plugin in this job. Currently installed compatible plugins are:

- CloudBees Docker Workflow
- CloudBees Docker Hub Notification
- CloudBees Docker Custom Build Environment Plugin

(from CloudBees Docker Hub Notification)

- [✓] Specified repositories will trigger this job

Repositories: java

Docker & Selenium

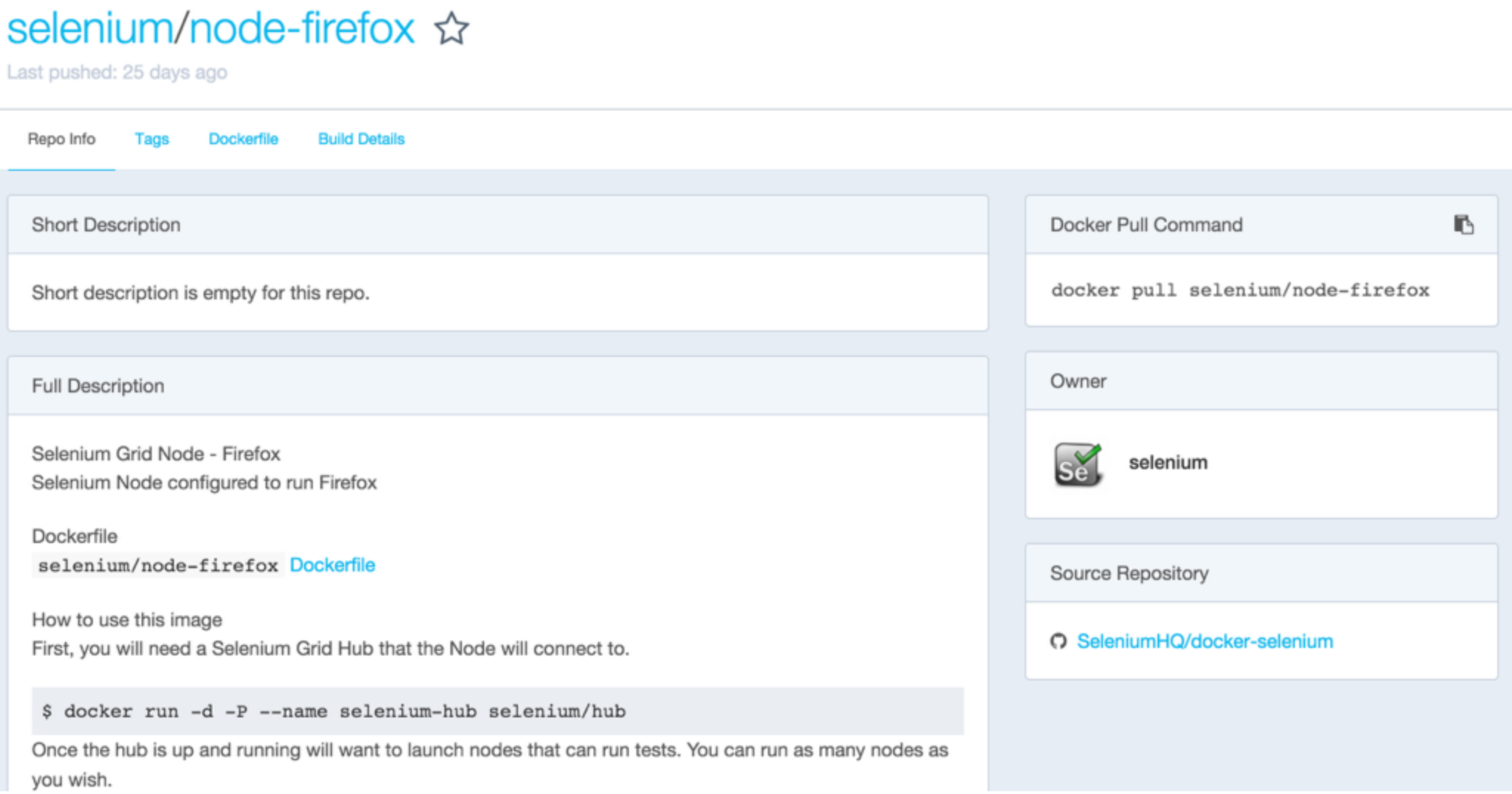
Selenium

Manage multiple combinations of browsers

Any number of them

Standalone or Selenium Hub

- even with VNC



selenium/node-firefox ☆

Last pushed: 25 days ago

Repo Info | Tags | Dockerfile | Build Details

Short Description

Short description is empty for this repo.

Docker Pull Command

```
docker pull selenium/node-firefox
```

Full Description

Selenium Grid Node - Firefox
Selenium Node configured to run Firefox

Dockerfile
`selenium/node-firefox` Dockerfile

How to use this image
First, you will need a Selenium Grid Hub that the Node will connect to.

```
$ docker run -d -P --name selenium-hub selenium/hub
```

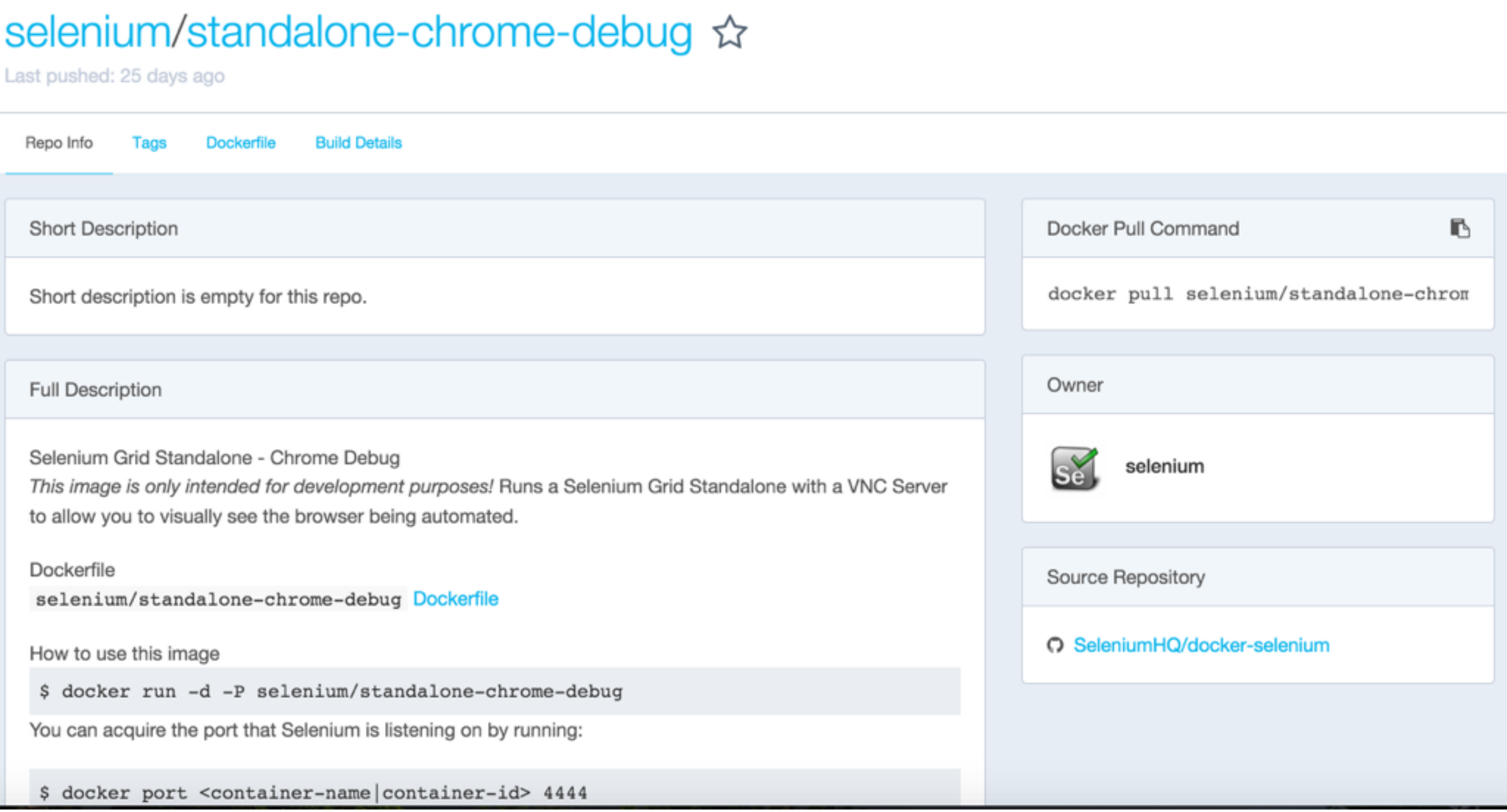
Once the hub is up and running will want to launch nodes that can run tests. You can run as many nodes as you wish.

Owner

selenium

Source Repository

SeleniumHQ/docker-selenium


selenium/standalone-chrome-debug ☆

Last pushed: 25 days ago

Short Description

Short description is empty for this repo.

Full Description

Selenium Grid Standalone - Chrome Debug

This image is only intended for development purposes! Runs a Selenium Grid Standalone with a VNC Server to allow you to visually see the browser being automated.

Dockerfile

`selenium/standalone-chrome-debug` Dockerfile

How to use this image

```
$ docker run -d -P selenium/standalone-chrome-debug
```

You can acquire the port that Selenium is listening on by running:

```
$ docker port <container-name|container-id> 4444
```

Docker Pull Command

```
docker pull selenium/standalone-chrom
```

Owner

selenium

Source Repository

SeleniumHQ/docker-selenium

Selenium Hub

```
hub:
  image: selenium/hub:2.48.2
  ports:
    - "4444:4444"
firefox:
  image: selenium/node-firefox-debug:2.48.2
  links:
    - hub
  ports:
    - "5901:5900"
chrome:
  image: selenium/node-chrome-debug:2.48.1
  links:
    - hub
  ports:
    - "5902:5900"
```

CloudBees

Kubernetes

Kubernetes

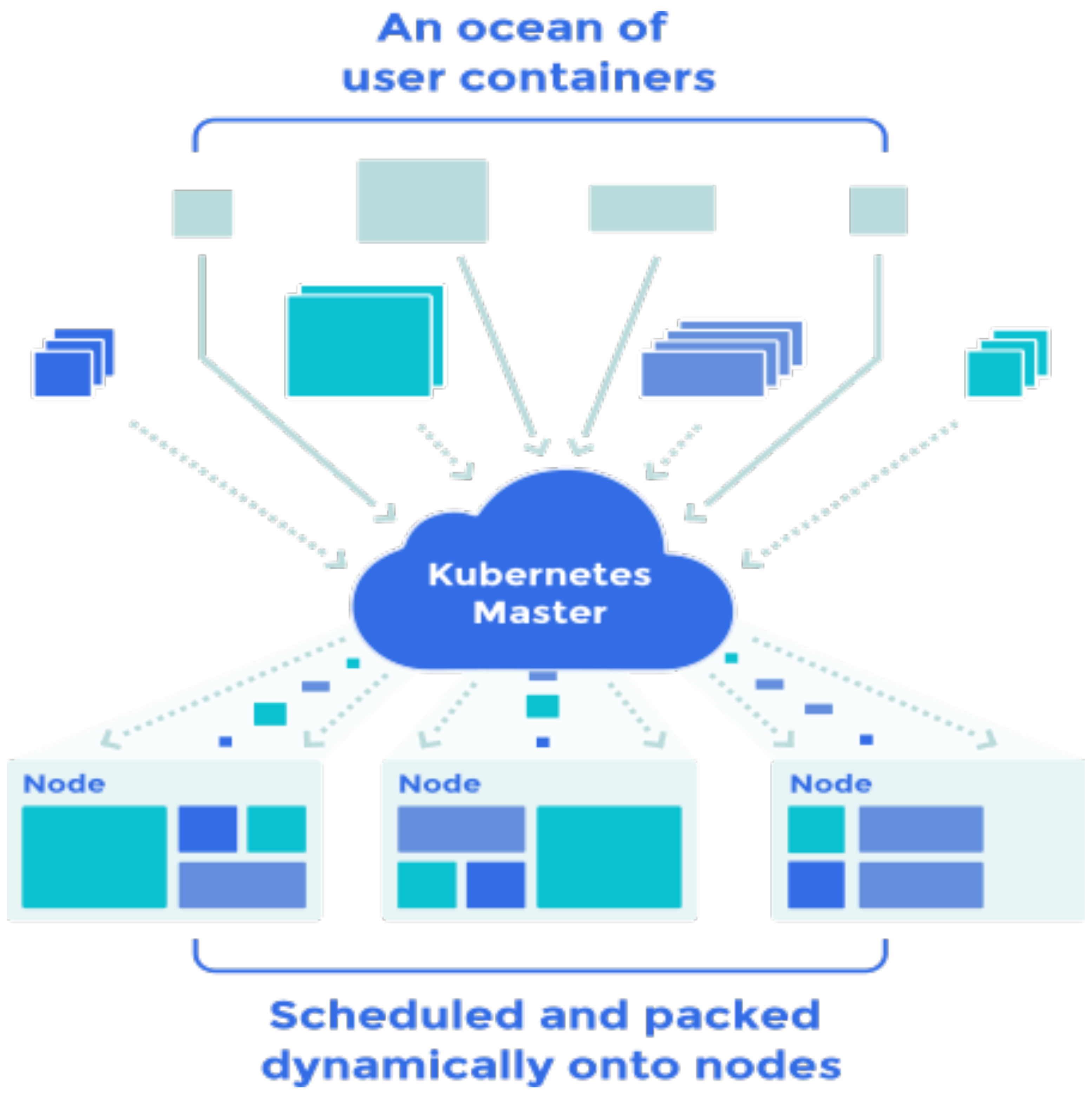
Container cluster orchestration

Docker containers across multiple hosts (nodes or minions)

Higher level API

Enforced state

Monitoring of endpoints


An ocean of
user containers
Kubernetes
Master
Node
Node
Node
Scheduled and packed
dynamically onto nodes

Providers

GKE

Azure

Vmware

Rackspace

oVirt

Vagrant

CloudStack

Ubuntu

Pod

Group of colocated containers

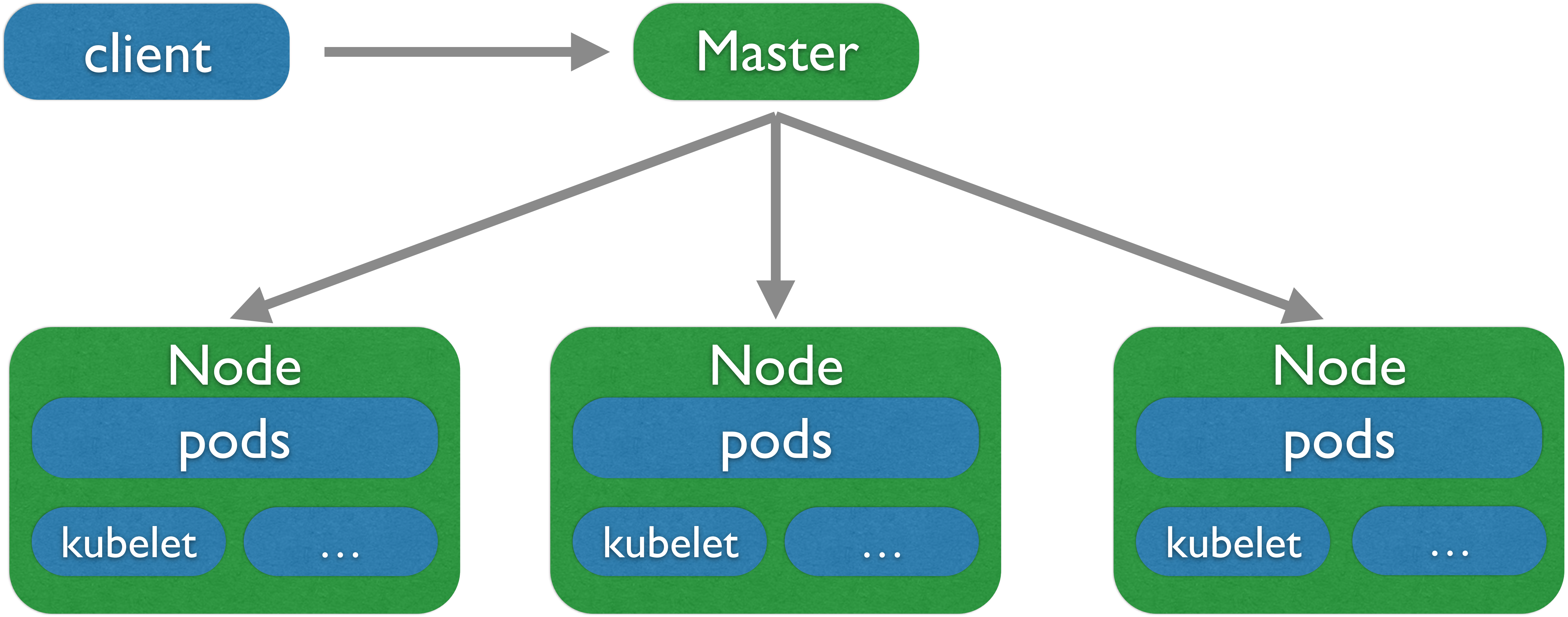
Same network namespace/IP

Environment variables

Shared volumes

- host mounted
- empty volumes
- GCE data disks
- AWS EBS volumes
- nfs
- glusterfs
- secrets

Pods


client
Master
Node
pods
kubelet
...
Node
pods
kubelet
...
Node
pods
kubelet
...

Pod

```
kind: "Pod"
apiVersion: "v1"
metadata:
  name: "jenkins"
  labels:
    name: "jenkins"
spec:
  containers:
    -
      name: "jenkins"
      image: "csanchez/jenkins-swarm:
1.625.1-for-volumes"
      ports:
        - containerPort: 8080
        - containerPort: 50000
     volumeMounts:
        - name: "jenkins-data"
          mountPath: "/var/jenkins_home"
  volumes:
    - name: "jenkins-data"
      hostPath:
        path: "/home/docker/jenkins"
```

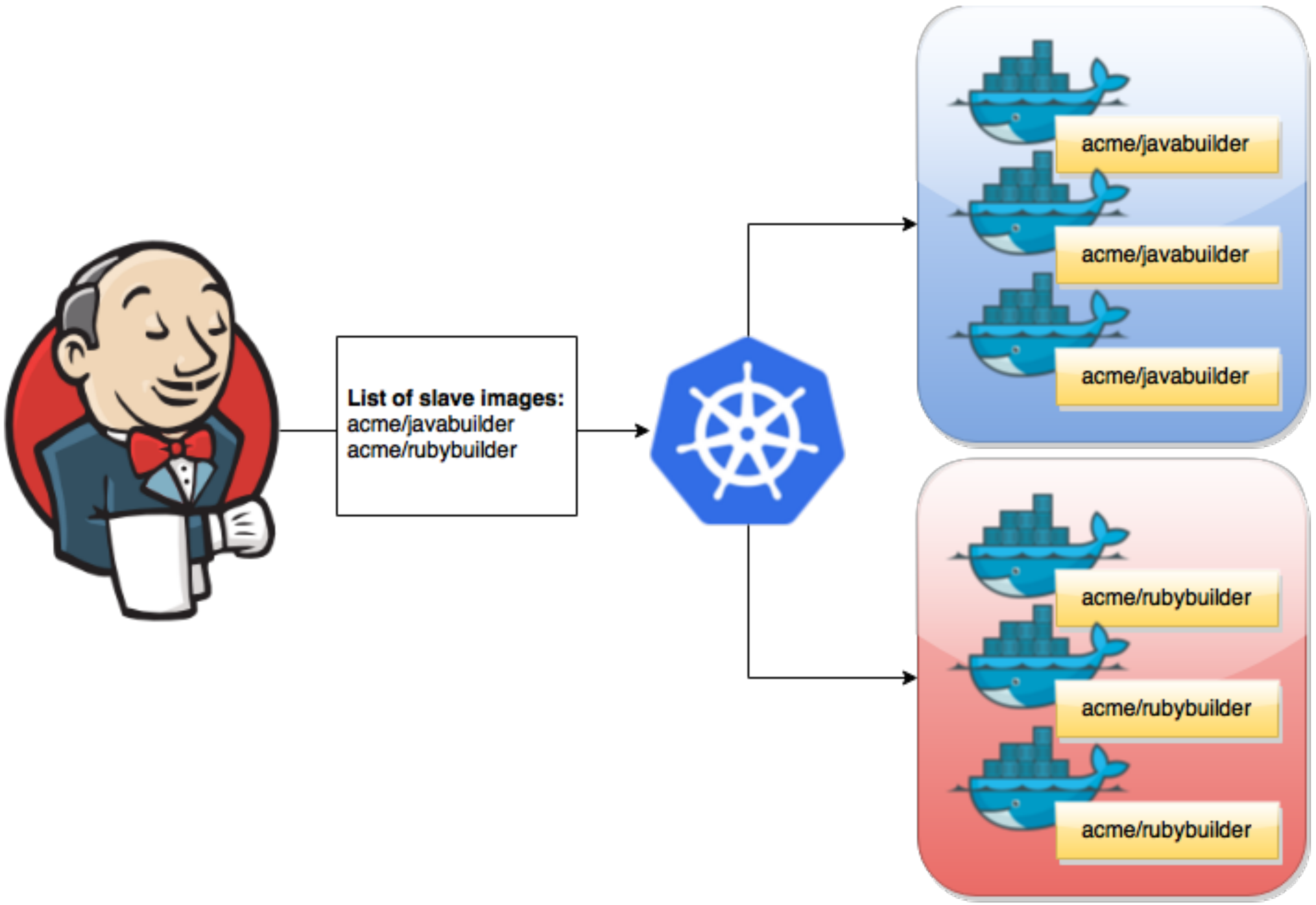
Kubernetes Jenkins plugin

Kubernetes Jenkins plugin

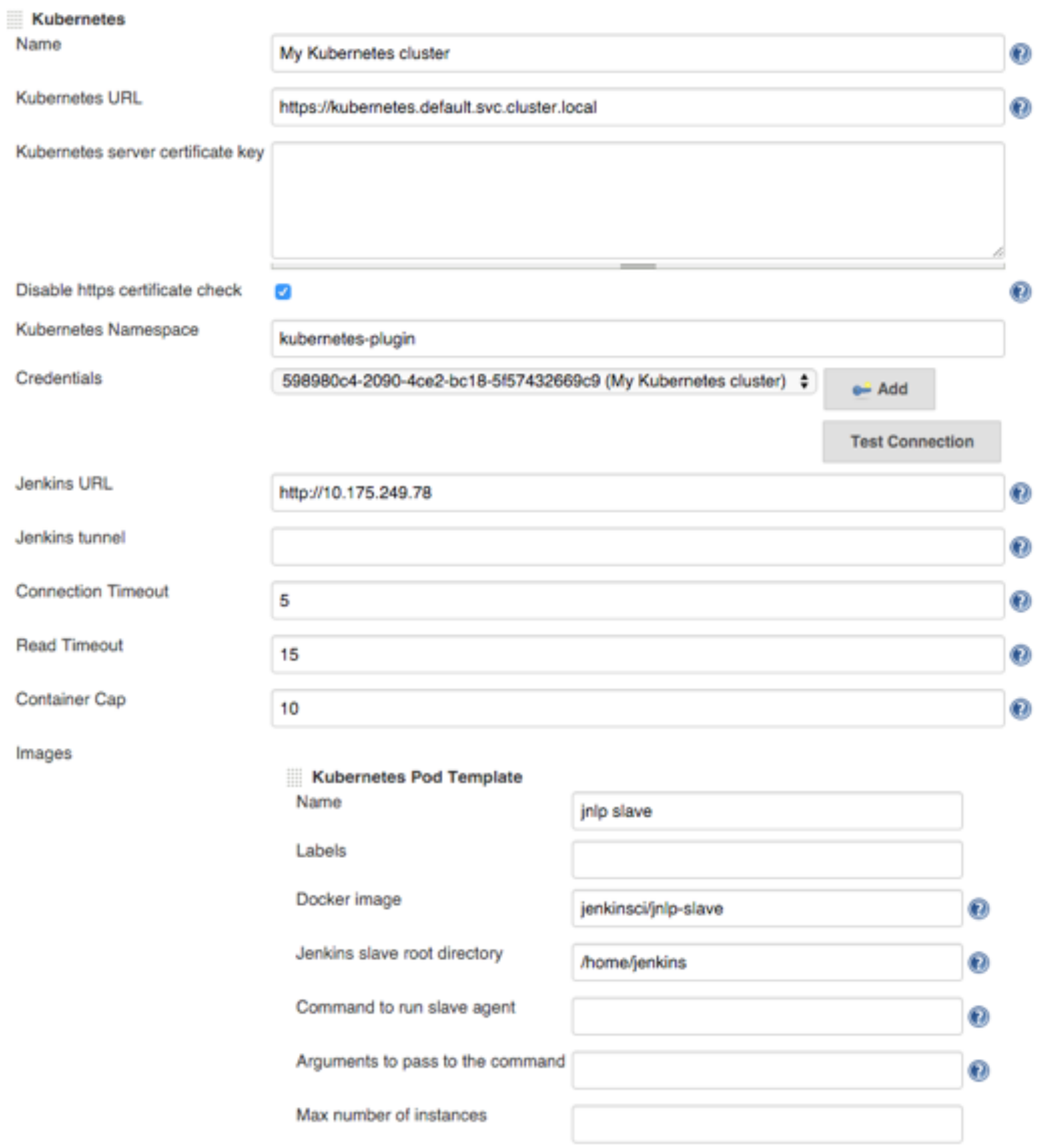
As a plugin

on demand slaves

https://github.com/jenkinsci/kubernetes-plugin



List of slave images:
acme/javabuilder
acme/rubybuilder
acme/javabuilder
acme/javabuilder
acme/javabuilder
acme/rubybuilder
acme/rubybuilder
acme/rubybuilder



Kubernetes	
Name	My Kubernetes cluster
Kubernetes URL	https://kubernetes.default.svc.cluster.local
Kubernetes server certificate key	
Disable https certificate check	✓
Kubernetes Namespace	kubernetes-plugin
Credentials	598980c4-2090-4ce2-bc18-5f57432669c9 (My Kubernetes cluster)
Jenkins URL	http://10.175.249.78
Jenkins tunnel	
Connection Timeout	5
Read Timeout	15
Container Cap	10
Images	

Kubernetes Pod Template	
Name	jnlp slave
Labels	
Docker image	jenkinsci/jnlp-slave
Jenkins slave root directory	/home/jenkins
Command to run slave agent	
Arguments to pass to the command	
Max number of instances	

Danke!

RATE THIS SESSIONS ON WWW.AGILETESTINGDAYS.COM