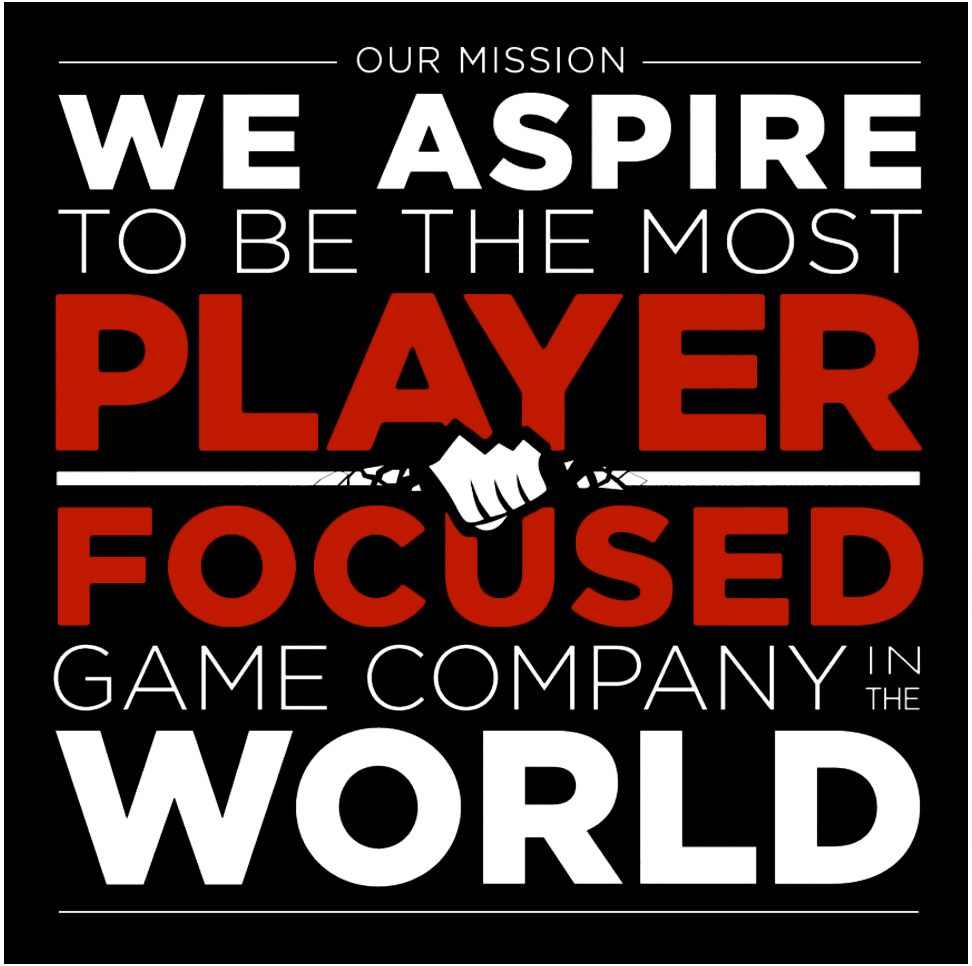
Thinking Inside the Container: A Continuous Delivery Story

Jenkins World

2016

Who’s This Guy?

RIOT GAMES
Sapient
paragon
Goldman Sachs
Linden Lab
NETDEVIL
Sprint
gazillion
JPMorganChase



OUR MISSION

WE ASPIRE TO BE THE MOST PLAYER FOCUSED GAME COMPANY IN THE WORLD

A Quick Note on Riot Games

The Scale of League

Jenkins World
2016

MORE THAN
67 MILLION

MONTHLY
ACTIVE
PLAYERS

MORE THAN
27 MILLION

DAILY ACTIVE
PLAYERS

7.5 MILLION

PEAK
CONCURRENT
PLAYERS

1.25 Million Builds a Year

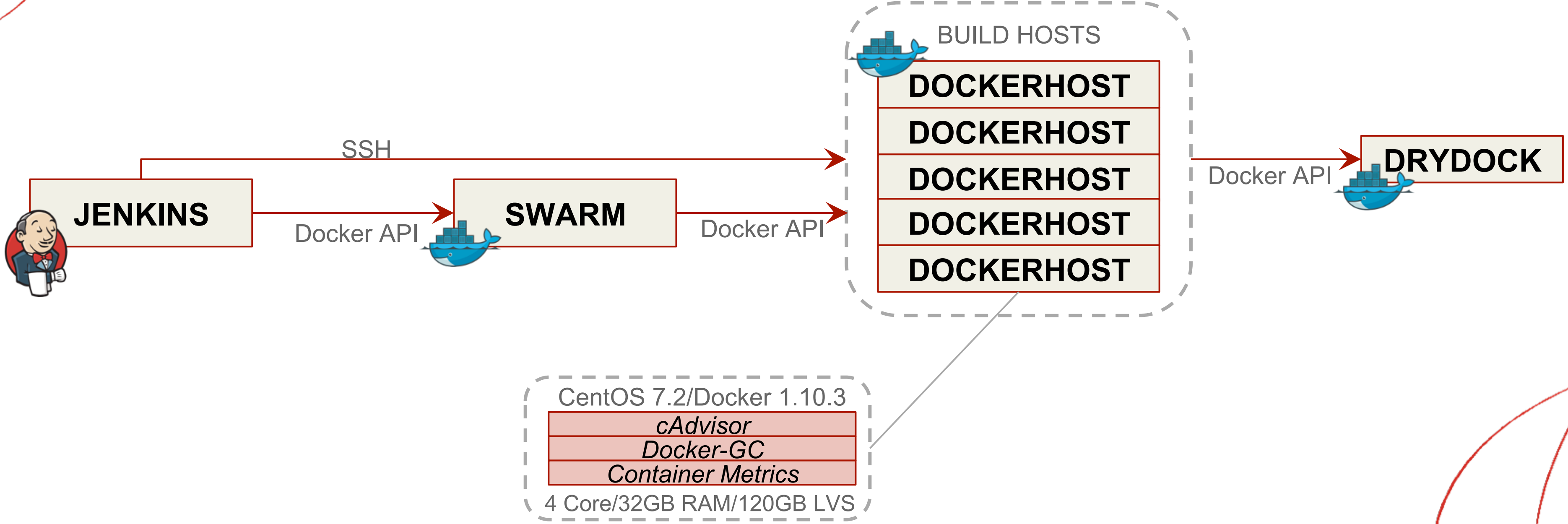
Thinking Inside the Container

10,000 - 14,000 Containers A Week

Jenkins World 2016

120 Jobs An Hour

A Containerized Build Farm



STORY TIME!

(aka. Presenting the Problem)

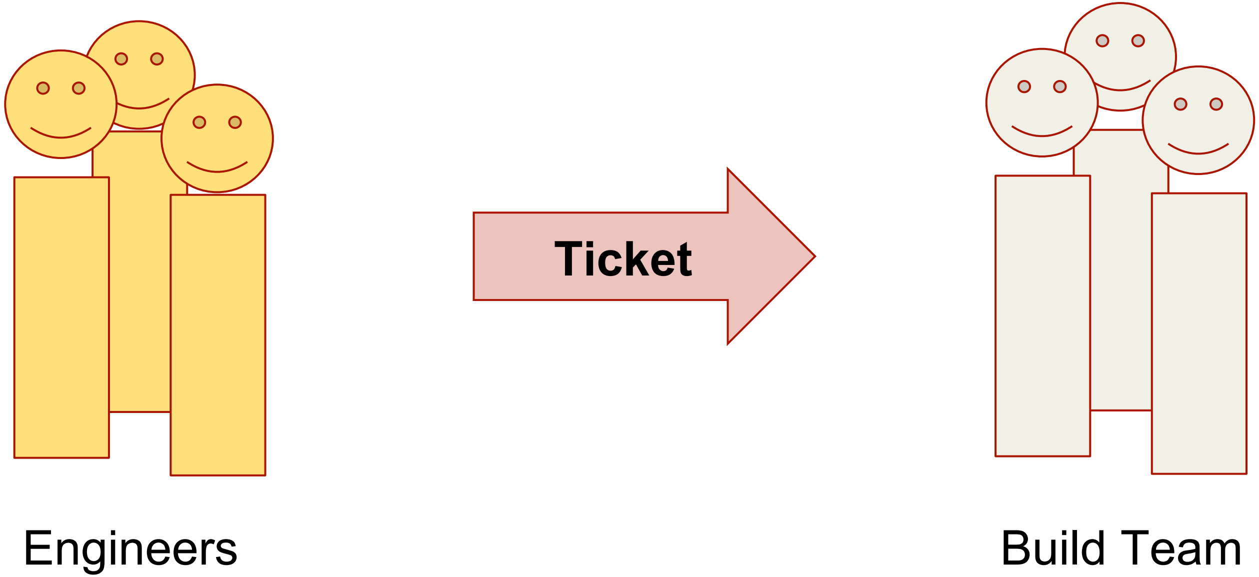
3500+ Unique Build Jobs

650+ Builds an Hour

90+ Build Slaves

Expecting 100 More!

Story Time....

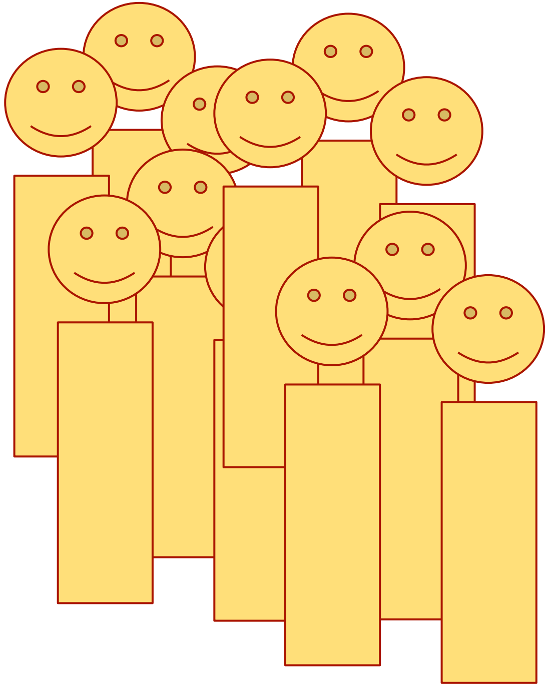


Engineers

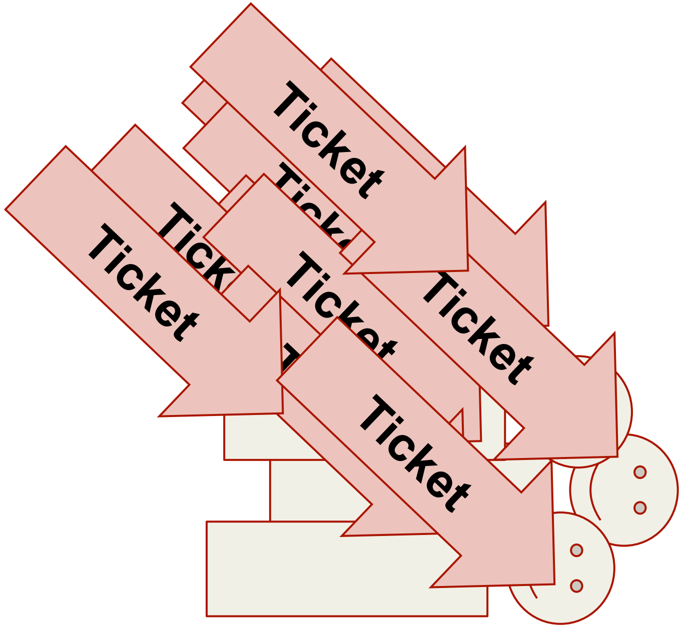
Ticket

Build Team

Story Time....



Engineers



Build Team

What Did We Want?

TEAMS NEEDED TO MOVE FAST

PRODUCTS HAD TO OWN THEIR STACK

CONFIGURATION AS CODE

Maybe We Want...

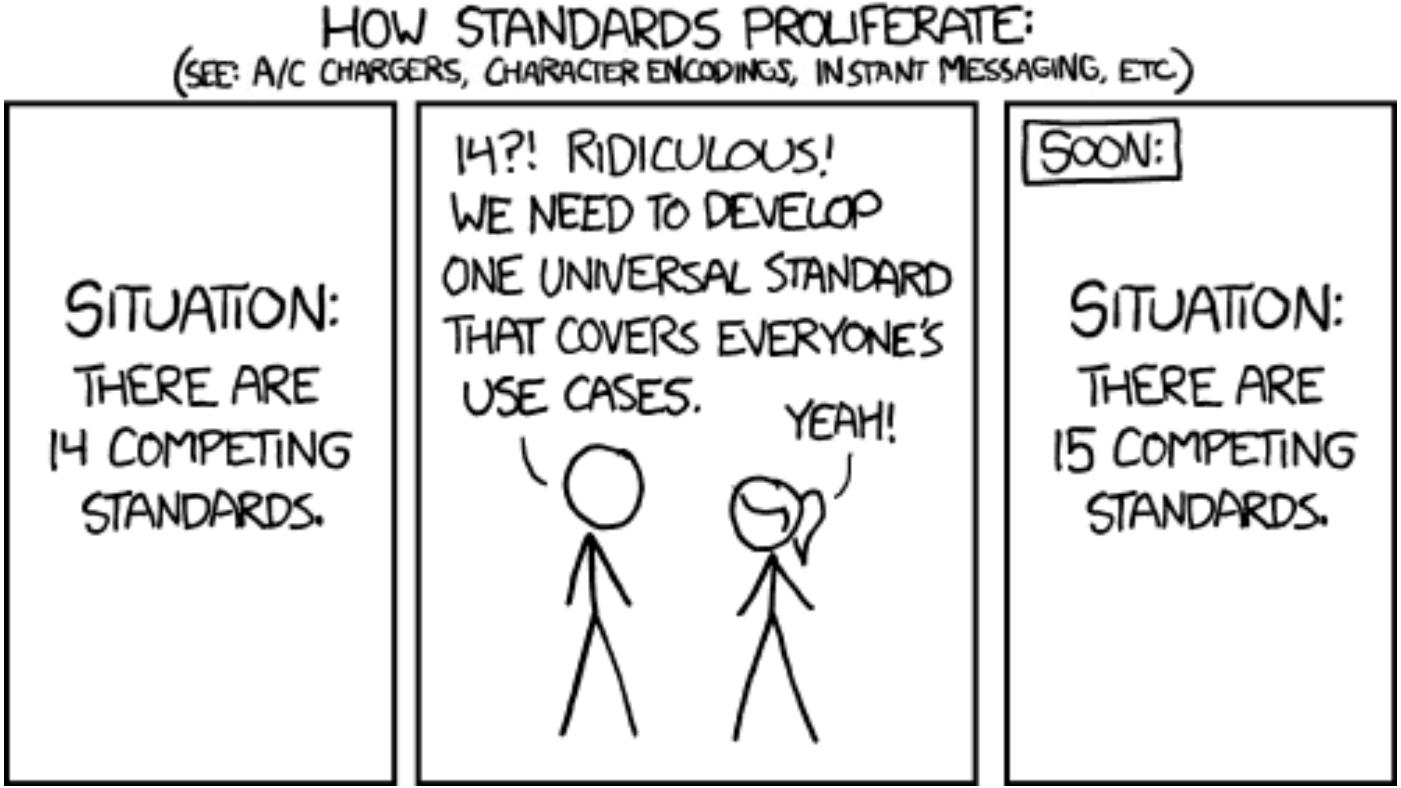
PACKER
VAGRANT
amazon web services™
EC2
openstack®
CLOUD SOFTWARE
vmware®
Microsoft Azure

“Along Came a Whale”

(More Story Time!)

YEAH, IF YOU COULD PUT DOCKER IN THE BUILD FARM
THAT'D BE GREAT.
memegenerator.net

Oh Look! Another Way to Deploy!

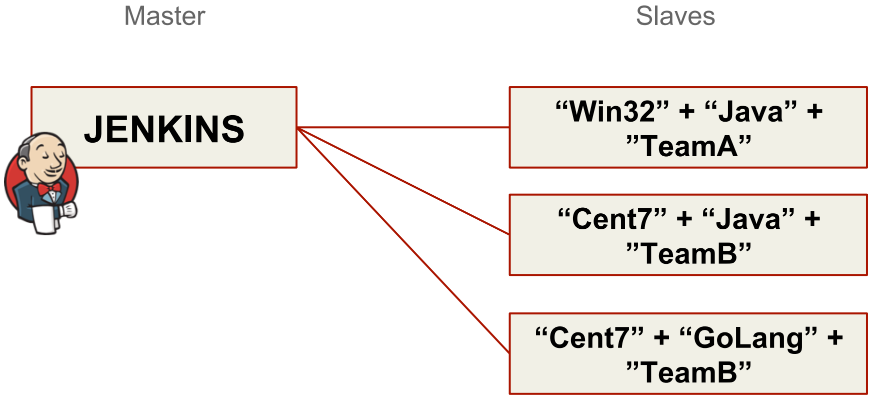


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION:
THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION:
THERE ARE 15 COMPETING STANDARDS.

Docker For Newbz

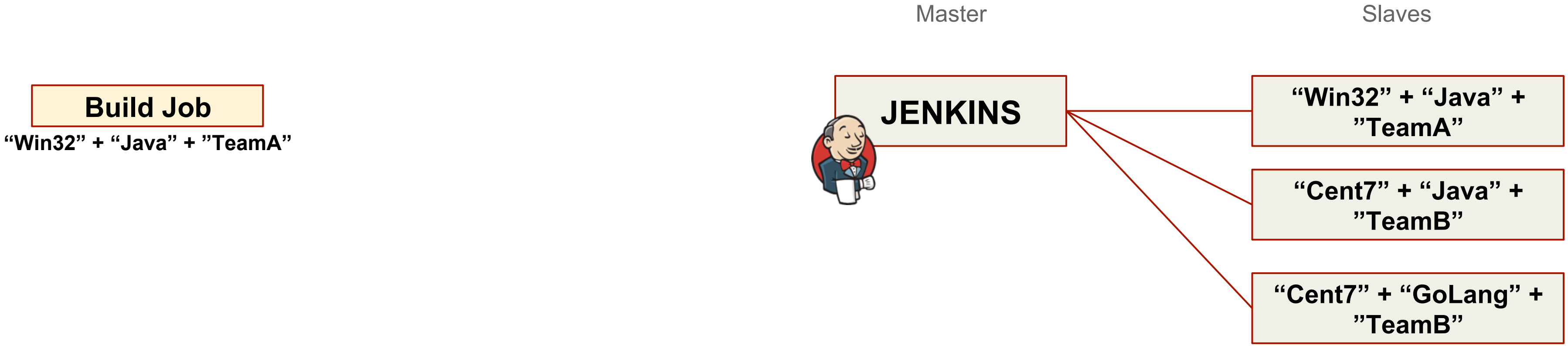
Jenkins Primer

Jenkins Primer



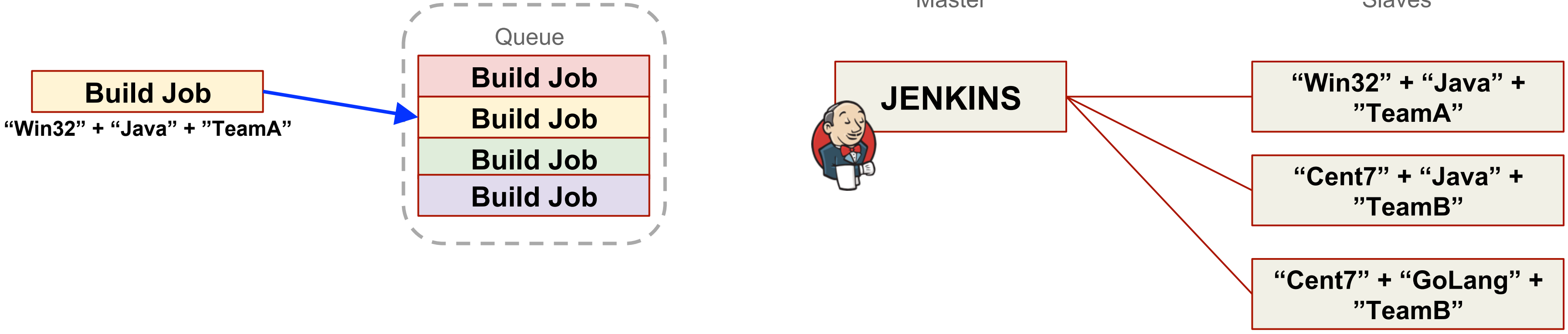
Master
Slaves
JENKINS
“Win32” + “Java” + ”TeamA”
“Cent7” + “Java” + ”TeamB”
“Cent7” + “GoLang” + ”TeamB”

Jenkins Primer

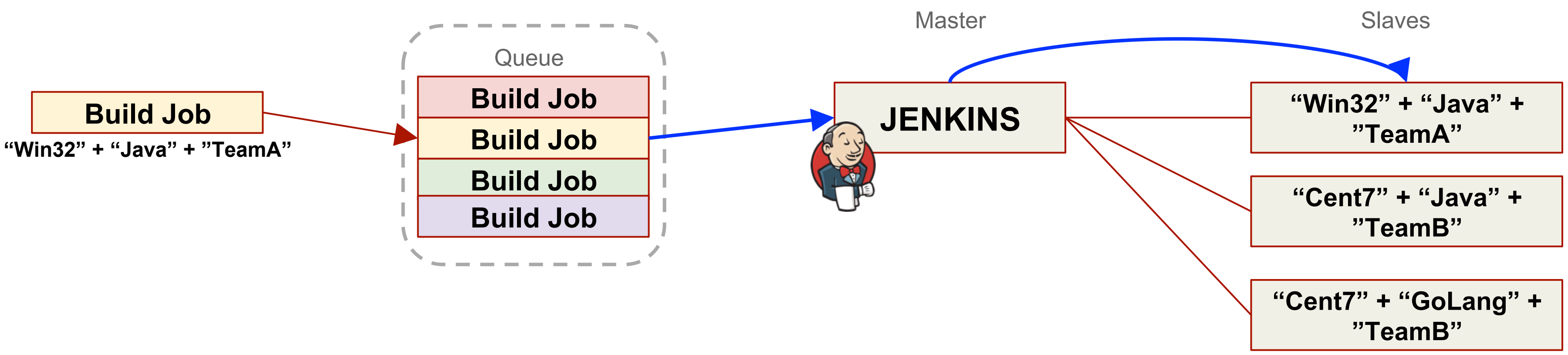


Master
Slaves
Build Job
“Win32” + “Java” + ”TeamA”
JENKINS
“Win32” + “Java” + ”TeamA”
“Cent7” + “Java” + ”TeamB”
“Cent7” + “GoLang” + ”TeamB”

Jenkins Primer


Build Job
"Win32" + "Java" + "TeamA"
Queue
Build Job
Build Job
Build Job
Build Job
Master
JENKINS
Slaves
"Win32" + "Java" + "TeamA"
"Cent7" + "Java" + "TeamB"
"Cent7" + "GoLang" + "TeamB"

Jenkins Primer


Build Job
"Win32" + "Java" + "TeamA"
Queue
Build Job
Build Job
Build Job
Build Job
Master
JENKINS
Slaves
"Win32" + "Java" + "TeamA"
"Cent7" + "Java" + "TeamB"
"Cent7" + "GoLang" + "TeamB"

What Did We Need?

CONTAINER RUNNING AS A BUILD SLAVE

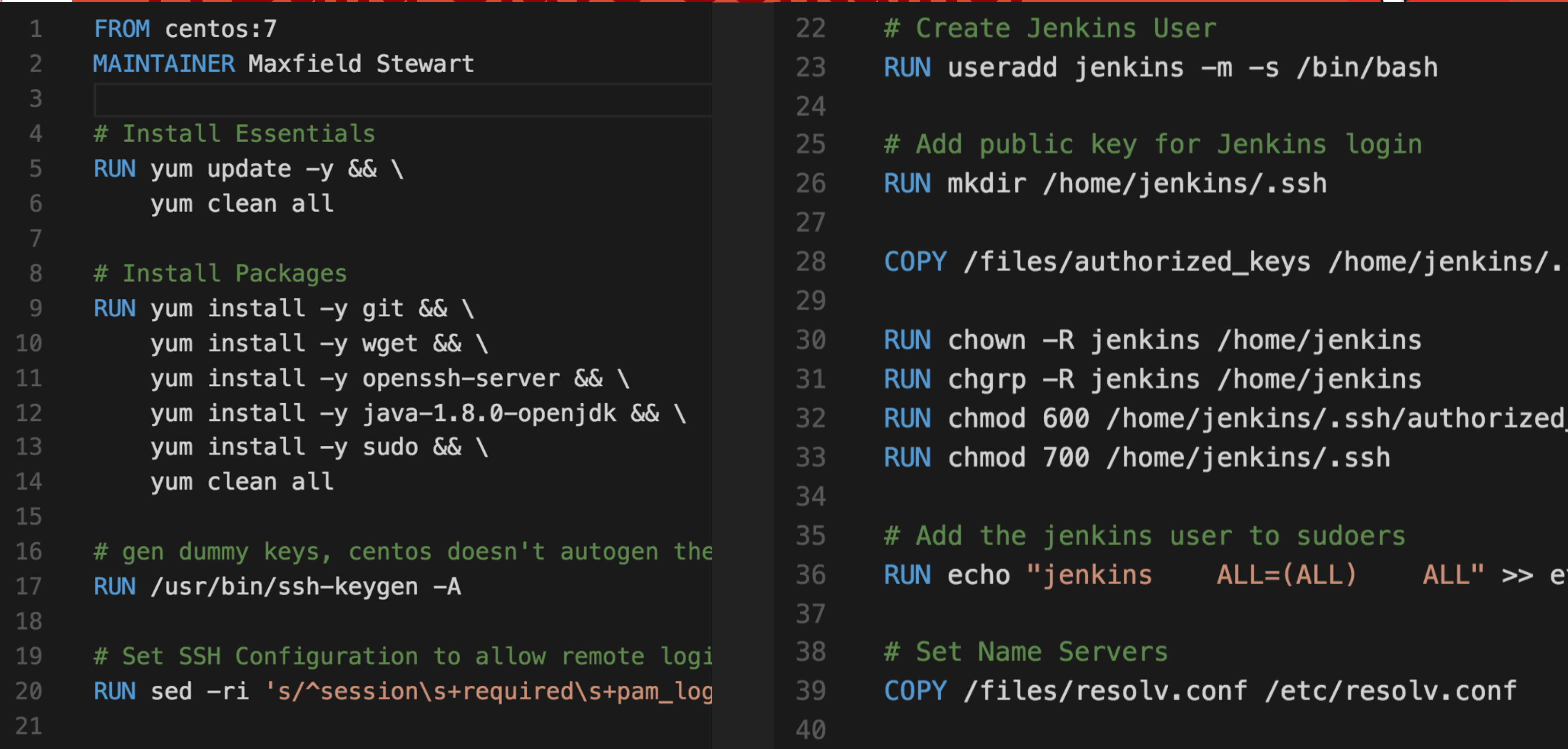
WAY TO HAVE ENGINEERS PROVIDE DOCKERFILES

HOW TO SCALE IT

⋮

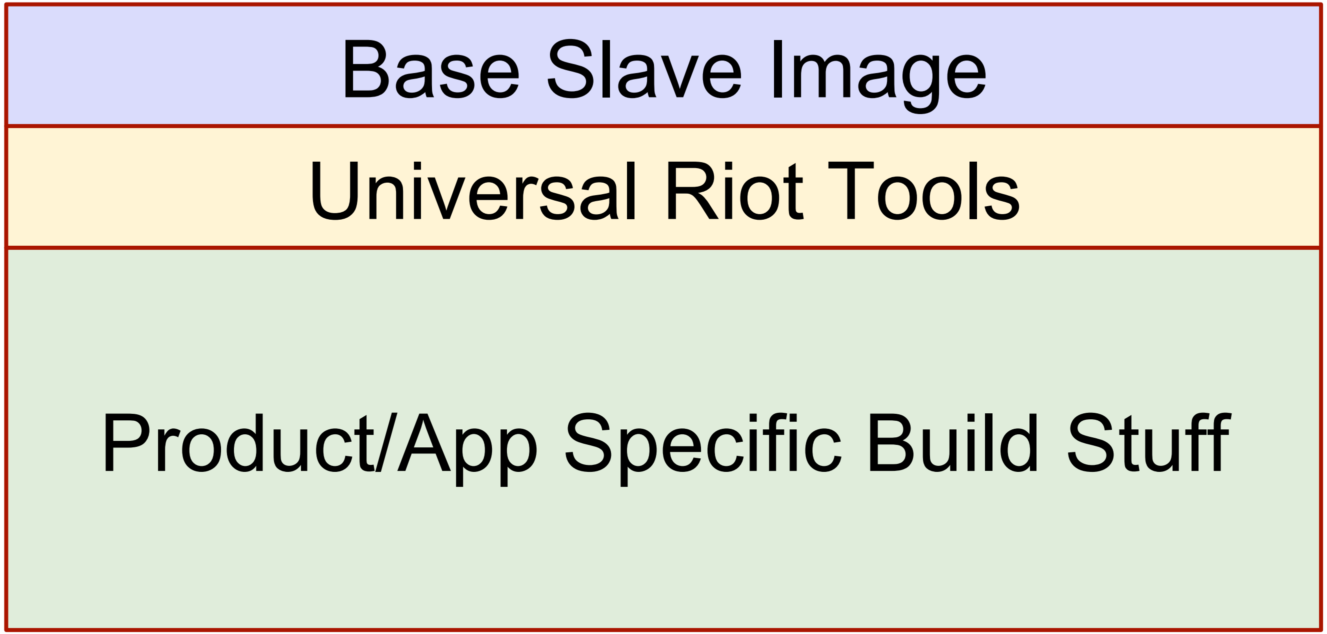
PROFIT?

A Build Slave Container



```
FROM centos:7
MAINTAINER Maxfield Stewart

# Install Essentials
RUN yum update -y && \
    yum clean all

# Install Packages
RUN yum install -y git && \
    yum install -y wget && \
    yum install -y openssh-server && \
    yum install -y java-1.8.0-openjdk && \
    yum install -y sudo && \
    yum clean all

# gen dummy keys, centos doesn't autogen the
RUN /usr/bin/ssh-keygen -A

# Set SSH Configuration to allow remote logi
RUN sed -ri 's/^session\s+required\s+pam_log

# Create Jenkins User
RUN useradd jenkins -m -s /bin/bash

# Add public key for Jenkins login
RUN mkdir /home/jenkins/.ssh

COPY /files/authorized_keys /home/jenkins/.

RUN chown -R jenkins /home/jenkins
RUN chgrp -R jenkins /home/jenkins
RUN chmod 600 /home/jenkins/.ssh/authorized
RUN chmod 700 /home/jenkins/.ssh

# Add the jenkins user to sudoers
RUN echo "jenkins    ALL=(ALL)    ALL" >> e

# Set Name Servers
COPY /files/resolv.conf /etc/resolv.conf

```

Add a Bit of Secret Sauce...

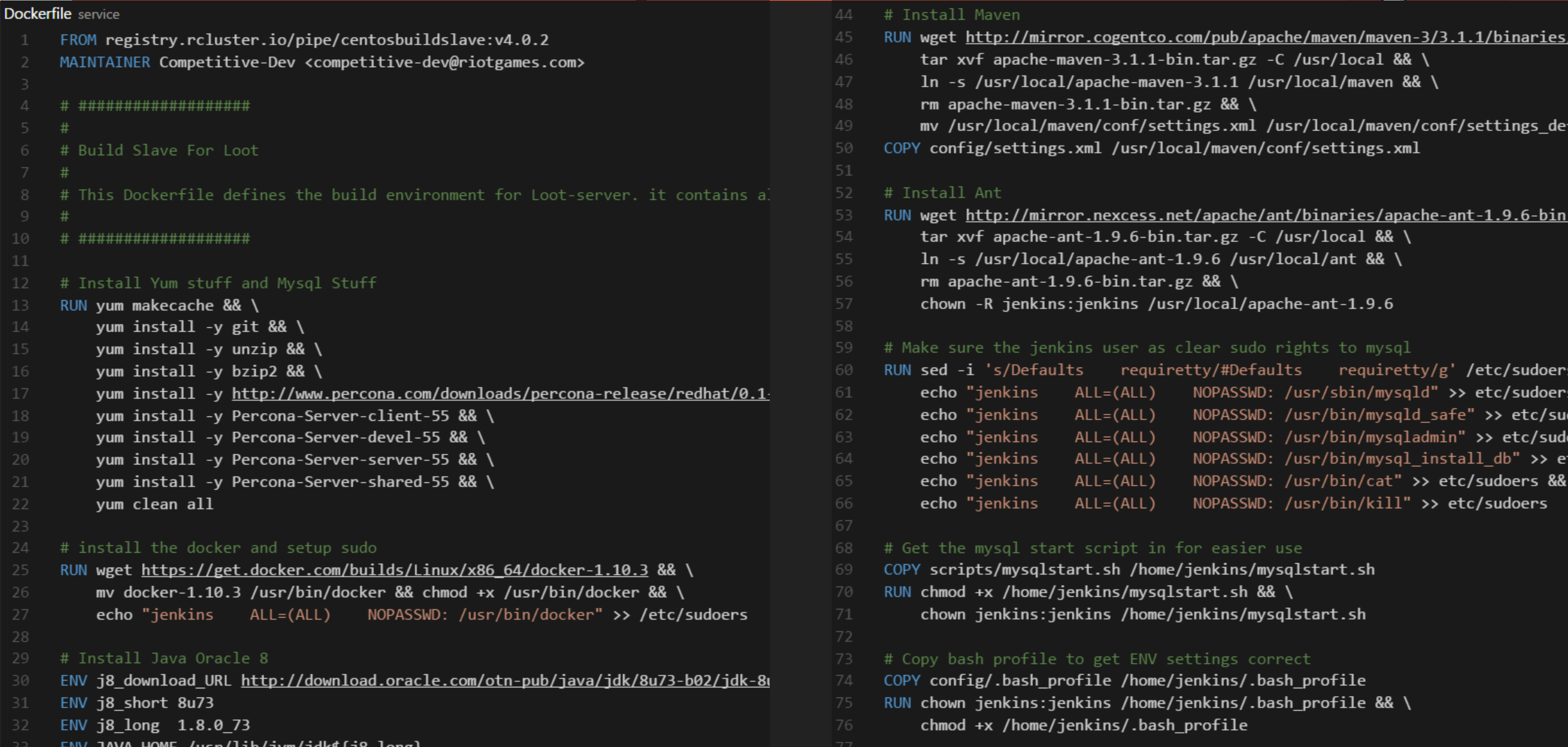


Base Slave Image

Universal Riot Tools

Product/App Specific Build Stuff

A Real Example

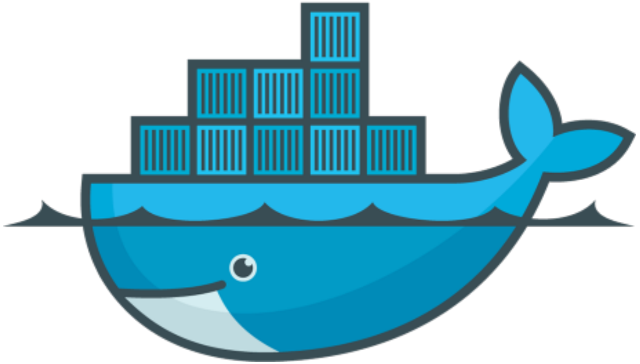


Dockerfile service

```
FROM registry.rcluster.io/pipe/centosbuildslave:v4.0.2
MAINTAINER Competitive-Dev <competitive-dev@riotgames.com>

# ###################
#
# Build Slave For Loot
#
# This Dockerfile defines the build environment for Loot-server. it contains a
#
# ###################

# Install Yum stuff and Mysql Stuff
RUN yum makecache && \
    yum install -y git && \
    yum install -y unzip && \
    yum install -y bzip2 && \
    yum install -y http://www.percona.com/downloads/percona-release/redhat/0.1-
    yum install -y Percona-Server-client-55 && \
    yum install -y Percona-Server-devel-55 && \
    yum install -y Percona-Server-server-55 && \
    yum install -y Percona-Server-shared-55 && \
    yum clean all

# install the docker and setup sudo
RUN wget https://get.docker.com/builds/Linux/x86_64/docker-1.10.3 && \
    mv docker-1.10.3 /usr/bin/docker && chmod +x /usr/bin/docker && \
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/bin/docker" >> /etc/sudoers

# Install Java Oracle 8
ENV j8_download_URL http://download.oracle.com/otn-pub/java/jdk/8u73-b02/jdk-8u
ENV j8_short 8u73
ENV j8_long  1.8.0_73
ENV JAVA_HOME /usr/lib/jvm/jdk${j8_long}
```

```
# Install Maven
RUN wget http://mirror.cogentco.com/pub/apache/maven/maven-3/3.1.1/binaries
    tar xvf apache-maven-3.1.1-bin.tar.gz -C /usr/local && \
    ln -s /usr/local/apache-maven-3.1.1 /usr/local/maven && \
    rm apache-maven-3.1.1-bin.tar.gz && \
    mv /usr/local/maven/conf/settings.xml /usr/local/maven/conf/settings_de
COPY config/settings.xml /usr/local/maven/conf/settings.xml

# Install Ant
RUN wget http://mirror.nexcess.net/apache/ant/binaries/apache-ant-1.9.6-bin
    tar xvf apache-ant-1.9.6-bin.tar.gz -C /usr/local && \
    ln -s /usr/local/apache-ant-1.9.6 /usr/local/ant && \
    rm apache-ant-1.9.6-bin.tar.gz && \
    chown -R jenkins:jenkins /usr/local/apache-ant-1.9.6

# Make sure the jenkins user as clear sudo rights to mysql
RUN sed -i 's/Defaults    requiretty/#Defaults    requiretty/g' /etc/sudoer
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/sbin/mysqld" >> etc/sudoer
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/bin/mysqld_safe" >> etc/su
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/bin/mysqladmin" >> etc/sud
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/bin/mysql_install_db" >> e
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/bin/cat" >> etc/sudoers &&
    echo "jenkins    ALL=(ALL)    NOPASSWD: /usr/bin/kill" >> etc/sudoers

# Get the mysql start script in for easier use
COPY scripts/mysqlstart.sh /home/jenkins/mysqlstart.sh
RUN chmod +x /home/jenkins/mysqlstart.sh && \
    chown jenkins:jenkins /home/jenkins/mysqlstart.sh

# Copy bash profile to get ENV settings correct
COPY config/.bash_profile /home/jenkins/.bash_profile
RUN chown jenkins:jenkins /home/jenkins/.bash_profile && \
    chmod +x /home/jenkins/.bash_profile
```

Provisioning and Plugins



+

= ?

Of Whales and Plugins

DOCKER PLUGIN

MESOS PLUGIN

KUBERNETES PLUGIN

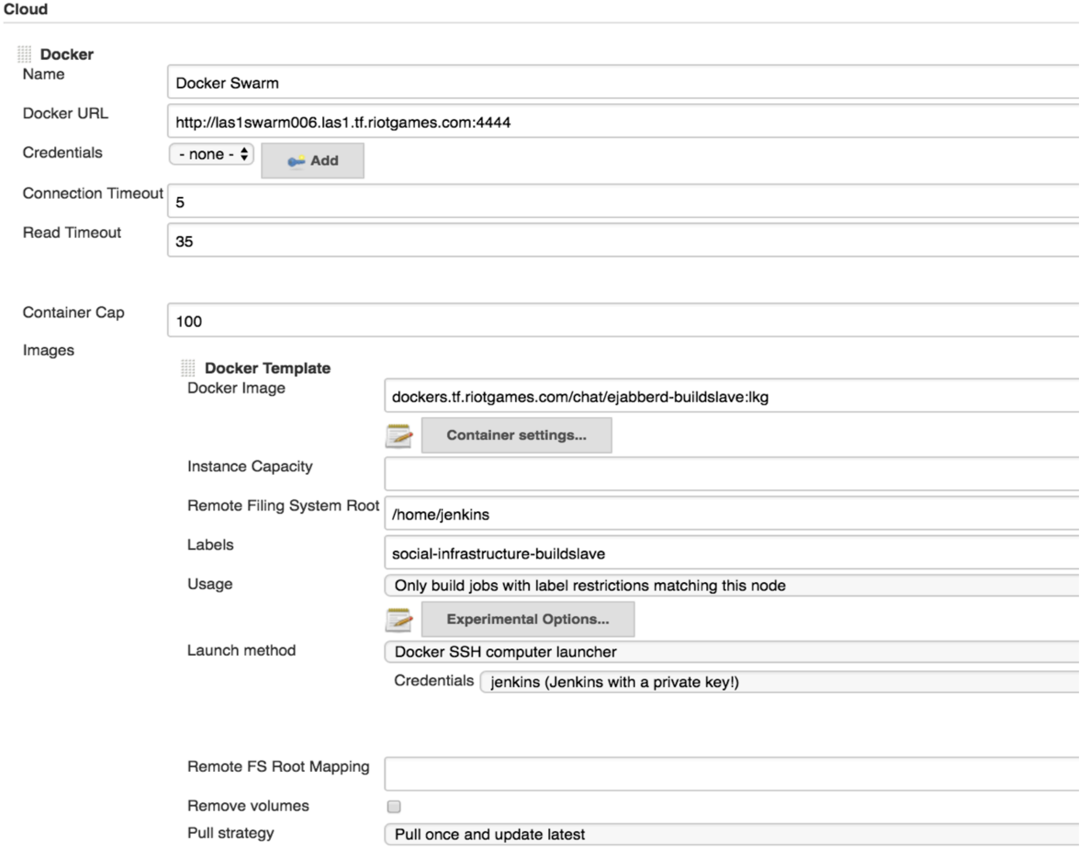
Of Whales and Plugins

DOCKER PLUGIN

MESOS PLUGIN

KUBERNETES PLUGIN

Quick Look



Cloud

Docker

Name: Docker Swarm

Docker URL: http://las1swarm006.las1.tf.riotgames.com:4444

Credentials: - none -

Connection Timeout: 5

Read Timeout: 35

Container Cap: 100

Images

Docker Template

Docker Image: dockers.tf.riotgames.com/chat/ejabberd-buildslave:lkg

Instance Capacity:

Remote Filing System Root: /home/jenkins

Labels: social-infrastructure-buildslave

Usage: Only build jobs with label restrictions matching this node

Launch method: Docker SSH computer launcher

Credentials: jenkins (Jenkins with a private key!)

Remote FS Root Mapping:

Remove volumes: □

Pull strategy: Pull once and update latest

Provisioning and Plugins

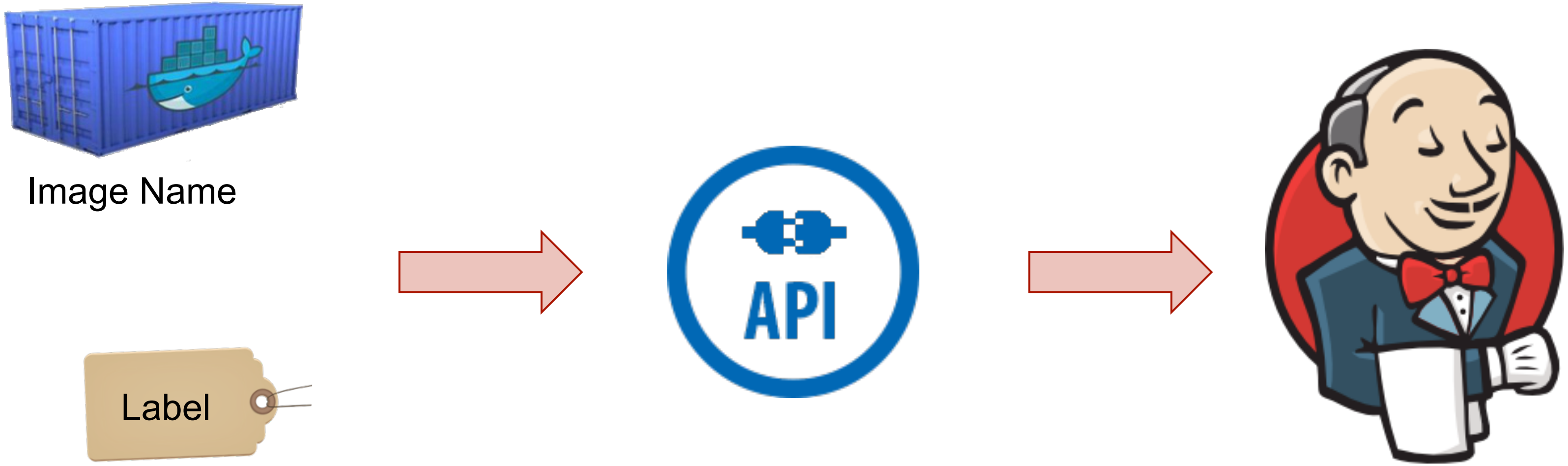
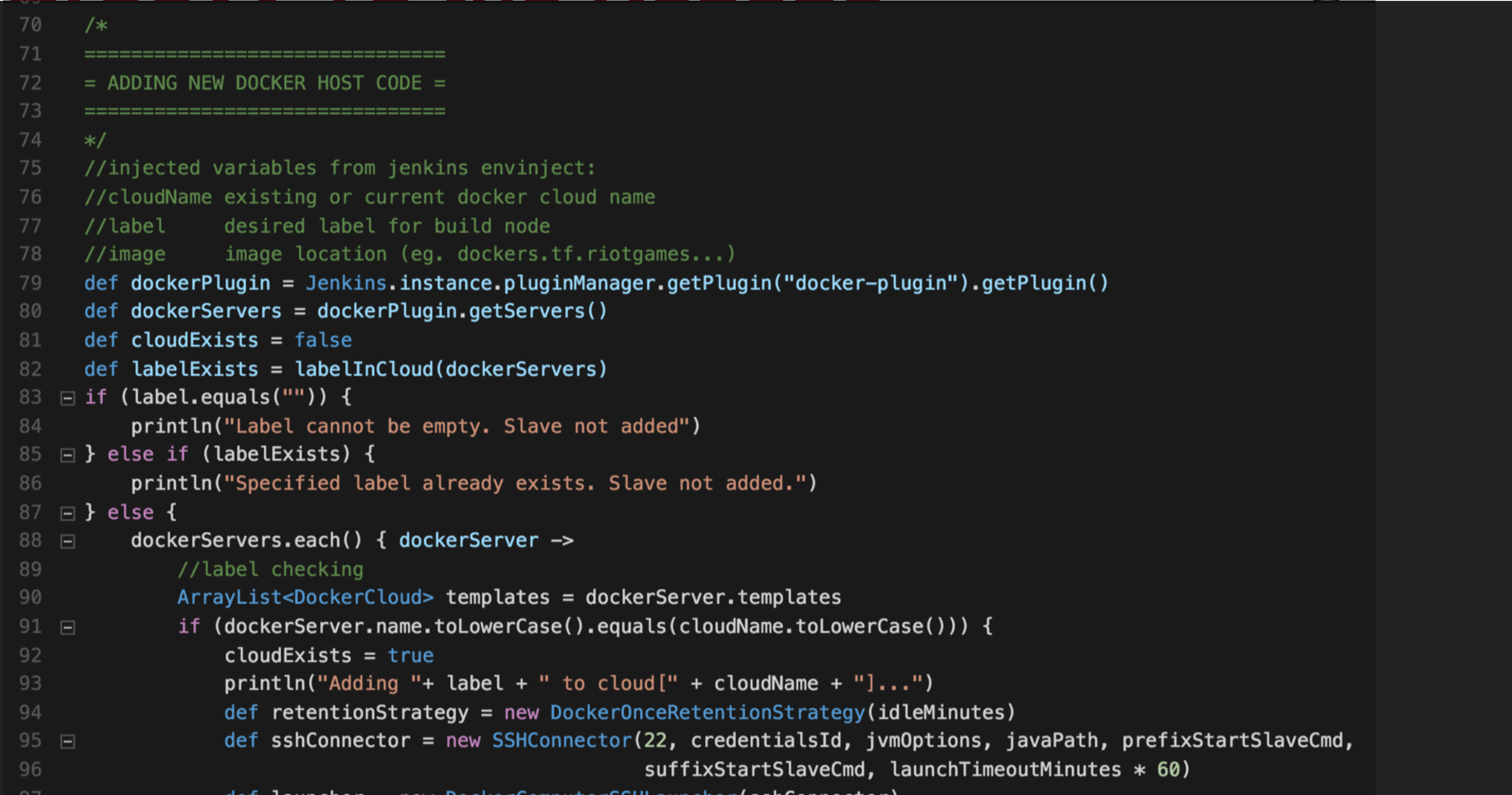


Image Name

Label

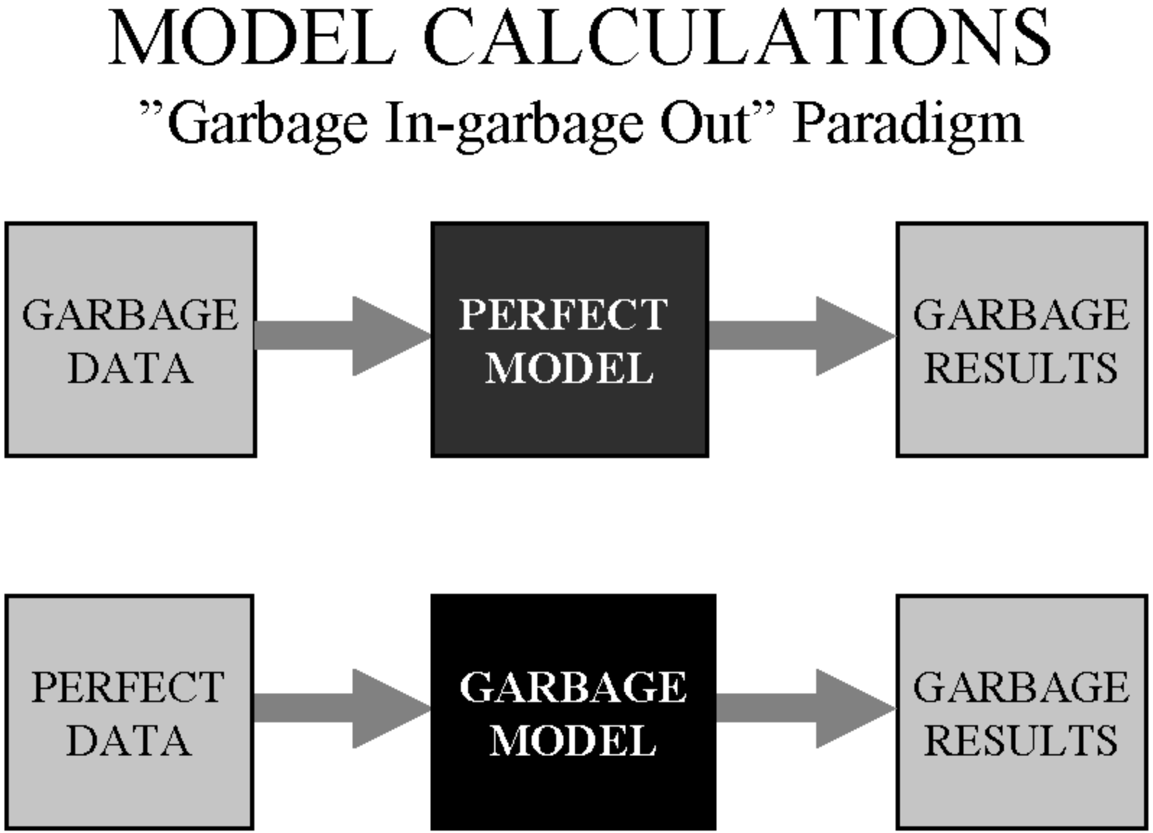
Groovy To the Rescue

https://github.com/maxfields2000/dockerjenkins_tutorial/tree/master/jenkinsscripts



```
/*
===============================
= ADDING NEW DOCKER HOST CODE =
===============================
*/
//injected variables from jenkins envinject:
//cloudName existing or current docker cloud name
//label     desired label for build node
//image     image location (eg. dockers.tf.riotgames...)
def dockerPlugin = Jenkins.instance.pluginManager.getPlugin("docker-plugin").getPlugin()
def dockerServers = dockerPlugin.getServers()
def cloudExists = false
def labelExists = labelInCloud(dockerServers)
if (label.equals("")) {
    println("Label cannot be empty. Slave not added")
} else if (labelExists) {
    println("Specified label already exists. Slave not added.")
} else {
    dockerServers.each() { dockerServer ->
        //label checking
        ArrayList<DockerCloud> templates = dockerServer.templates
        if (dockerServer.name.toLowerCase().equals(cloudName.toLowerCase())) {
            cloudExists = true
            println("Adding "+ label + " to cloud[" + cloudName + "]...")
            def retentionStrategy = new DockerOnceRetentionStrategy(idleMinutes)
            def sshConnector = new SSHConnector(22, credentialsId, jvmOptions, javaPath, prefixStartSlaveCmd,
                                                suffixStartSlaveCmd, launchTimeoutMinutes * 60)
            def launcher = new DockerComputerSSHLauncher(sshConnector)
```

We Created A Monster



MODEL CALCULATIONS

”Garbage In-garbage Out” Paradigm

GARBAGE DATA → PERFECT MODEL → GARBAGE RESULTS

PERFECT DATA → GARBAGE MODEL → GARBAGE RESULTS

We Need to Inspect Our Containers

Photo: Singapore Customs

O'REILLY

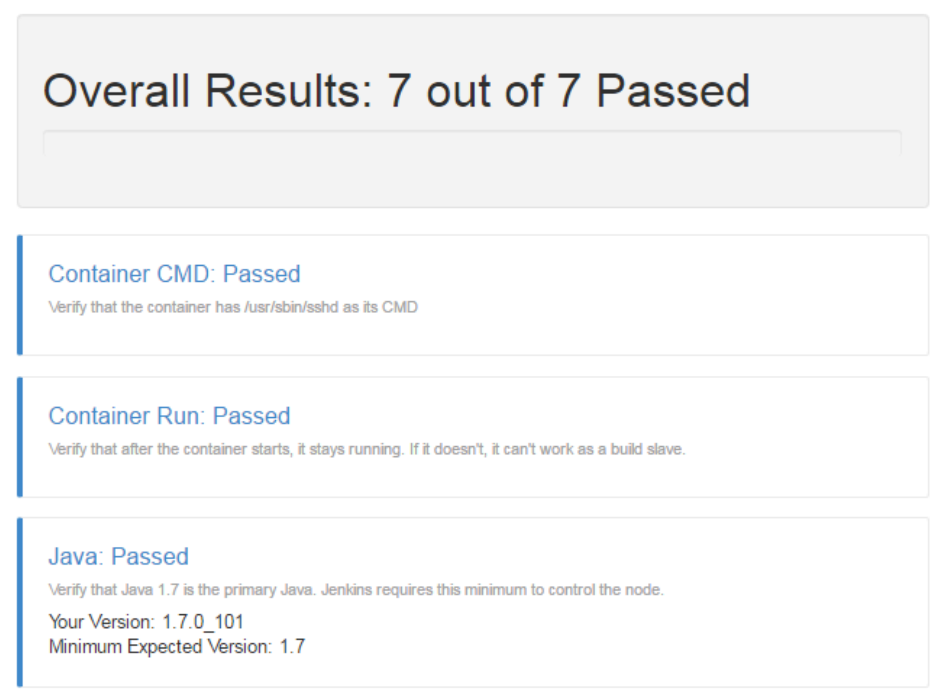
Infrastructure as Code

MANAGING SERVERS IN THE CLOUD

Kief M

We Need to Inspect Our Containers

```
./harbormaster --registry_user "$REGISTRY_USER" --registry_token "$REGISTRY_TOKEN" --jenkins_user $USER --
jenkins_pass $PASSWORD --jenkins --dockerhost $TEST_DOCKER_HOST "$IMAGENAME" "$CLOUDNAME" "$LABELNAME"
```



Overall Results: 7 out of 7 Passed

Container CMD: Passed

Verify that the container has /usr/sbin/sshd as its CMD

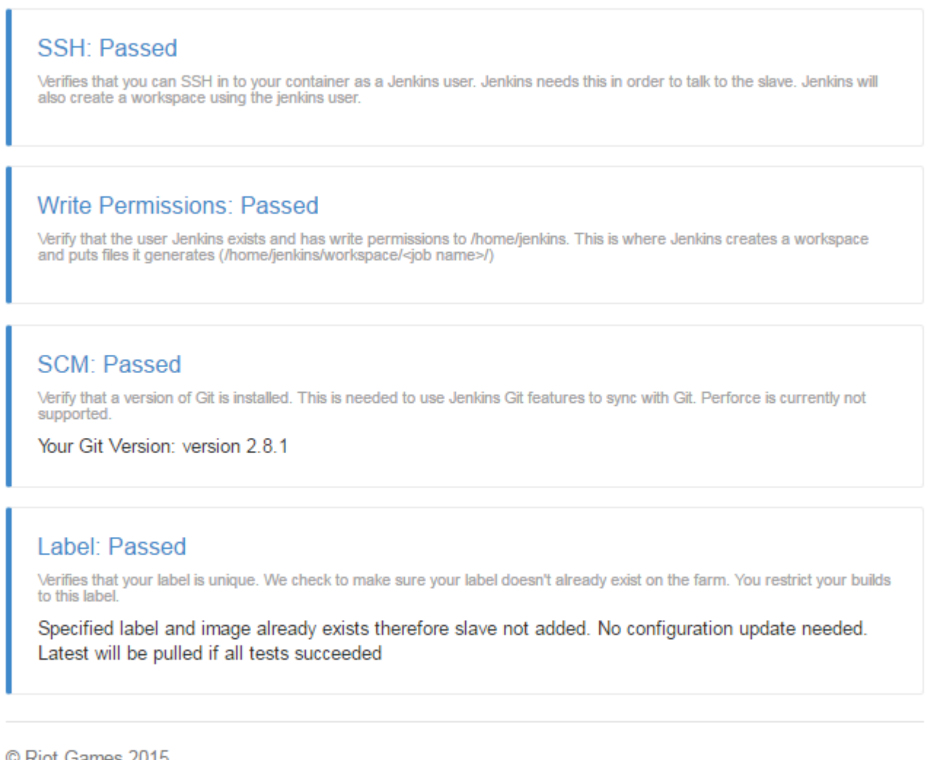
Container Run: Passed

Verify that after the container starts, it stays running. If it doesn't, it can't work as a build slave.

Java: Passed

Verify that Java 1.7 is the primary Java. Jenkins requires this minimum to control the node.

Your Version: 1.7.0_101
Minimum Expected Version: 1.7



SSH: Passed

Verifies that you can SSH in to your container as a Jenkins user. Jenkins needs this in order to talk to the slave. Jenkins will also create a workspace using the jenkins user.

Write Permissions: Passed

Verify that the user Jenkins exists and has write permissions to /home/jenkins. This is where Jenkins creates a workspace and puts files it generates (/home/jenkins/workspace/<job name>/)

SCM: Passed

Verify that a version of Git is installed. This is needed to use Jenkins Git features to sync with Git. Perforce is currently not supported.

Your Git Version: version 2.8.1

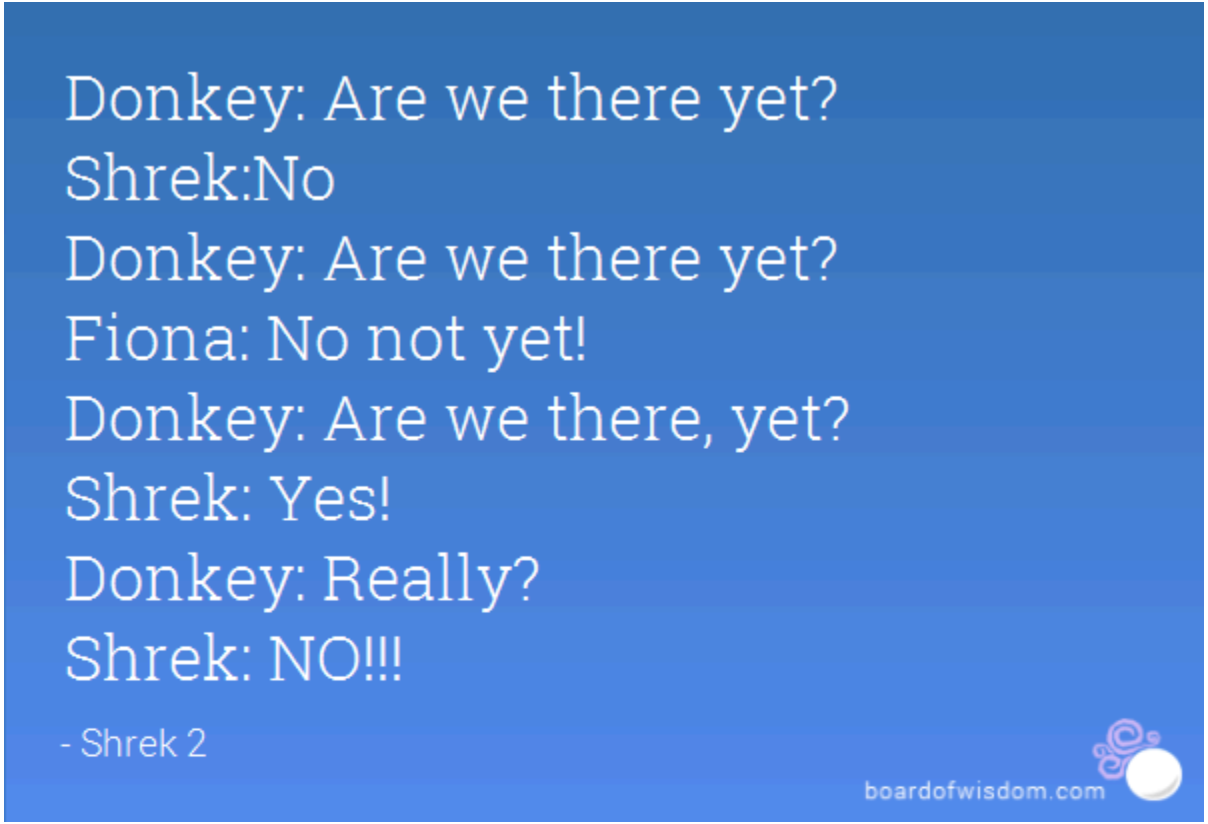
Label: Passed

Verifies that your label is unique. We check to make sure your label doesn't already exist on the farm. You restrict your builds to this label.

Specified label and image already exists therefore slave not added. No configuration update needed. Latest will be pulled if all tests succeeded

© Riot Games 2015

Are We Done?



Donkey: Are we there yet?

Shrek:No

Donkey: Are we there yet?

Fiona: No not yet!

Donkey: Are we there, yet?

Shrek: Yes!

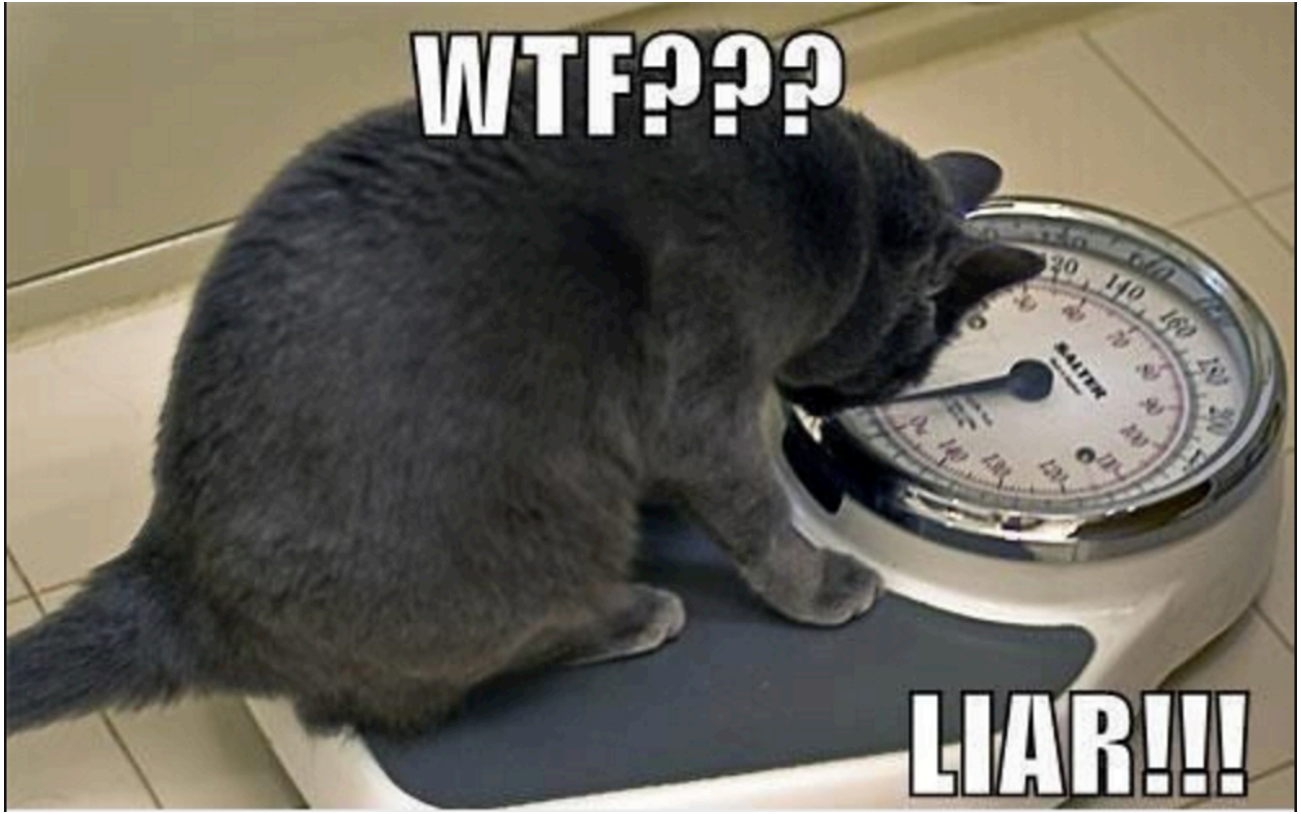
Donkey: Really?

Shrek: NO!!!

- Shrek 2

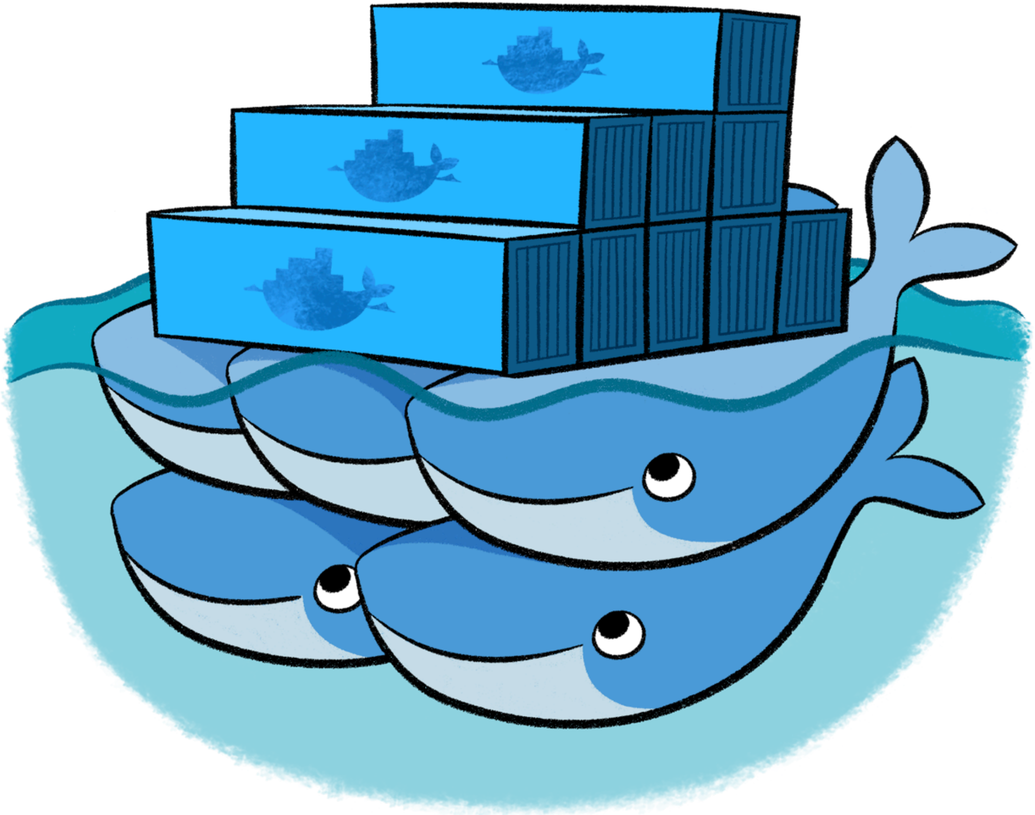
boardofwisdom.com

But It Needs to Scale!

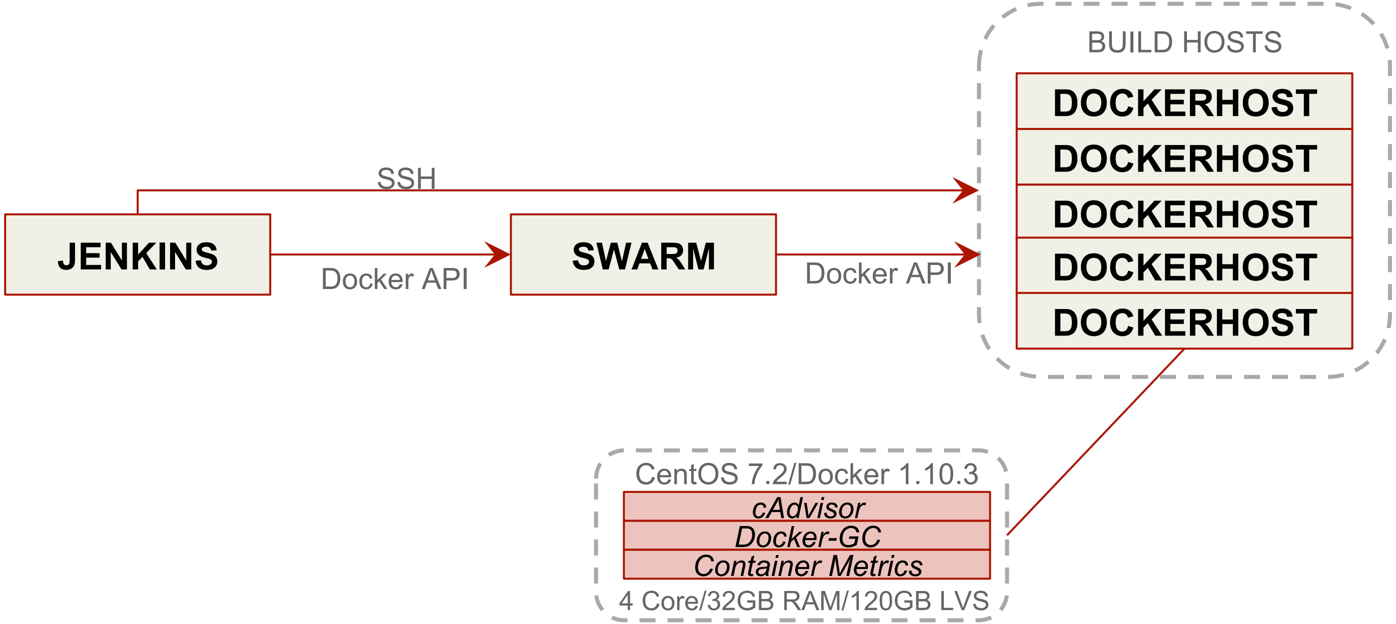


If Only...

If Only...



But It Needs to Scale!



BUILD HOSTS

DOCKERHOST

DOCKERHOST

DOCKERHOST

DOCKERHOST

DOCKERHOST

SSH

JENKINS

Docker API

SWARM

Docker API

CentOS 7.2/Docker 1.10.3

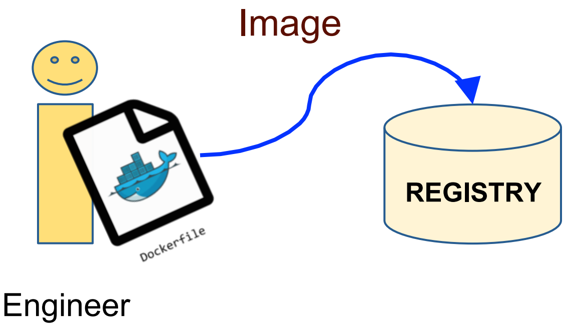
cAdvisor

Docker-GC

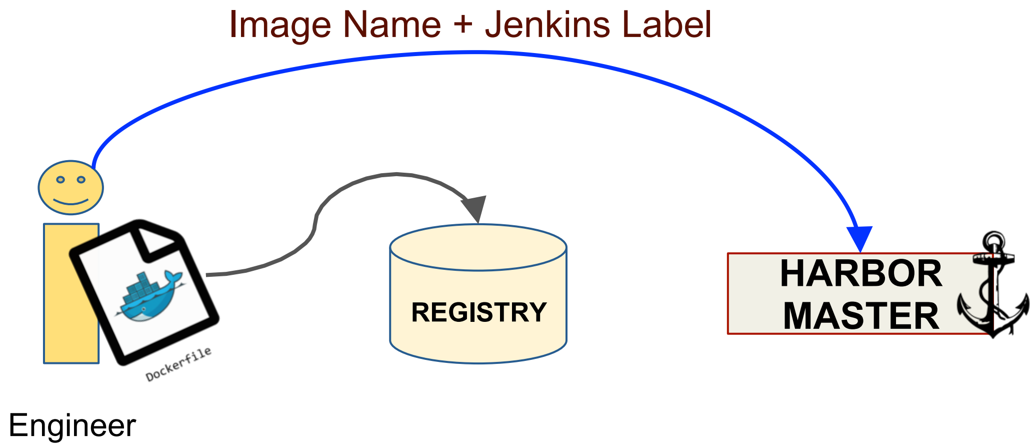
Container Metrics

4 Core/32GB RAM/120GB LVS

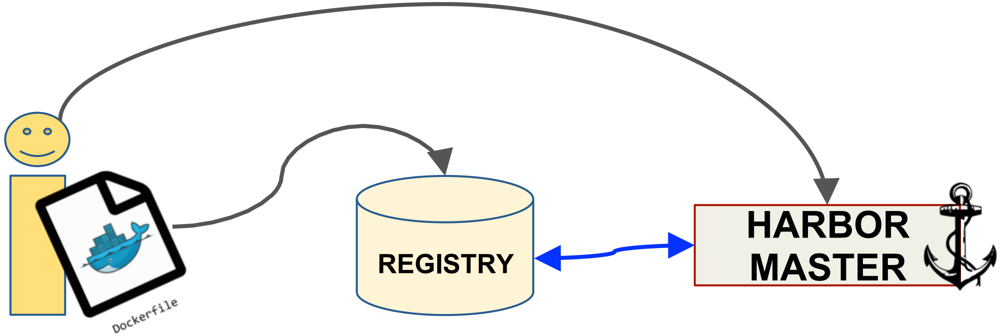
Putting It All Together



Putting It All Together



Putting It All Together



Engineer

Runs Tests

Putting It All Together

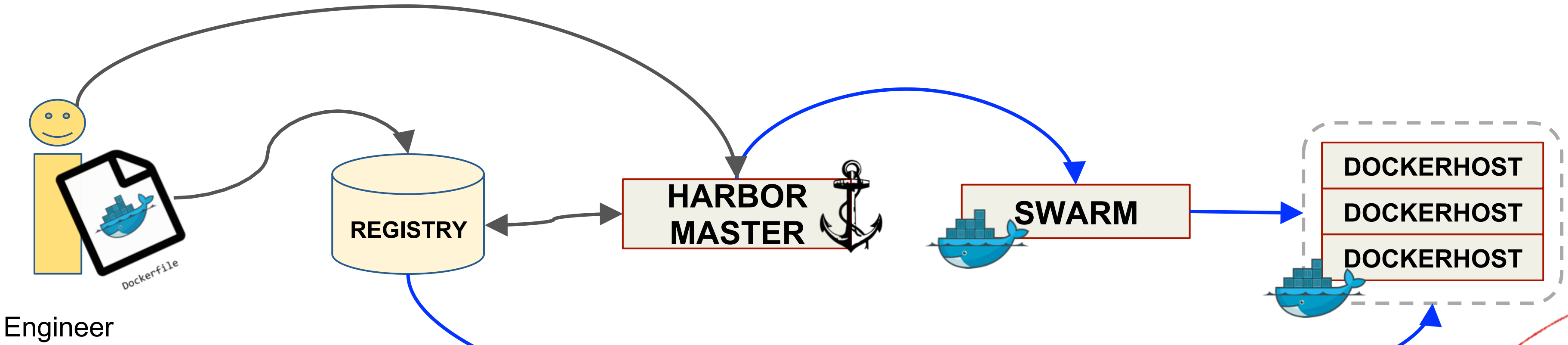
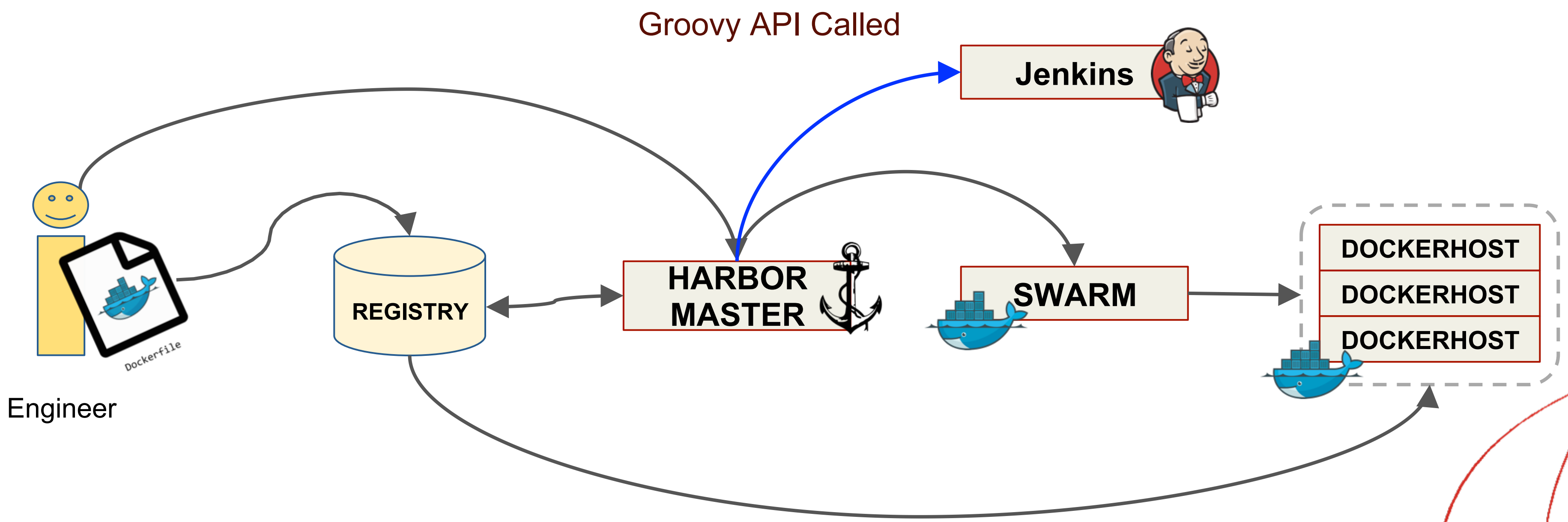
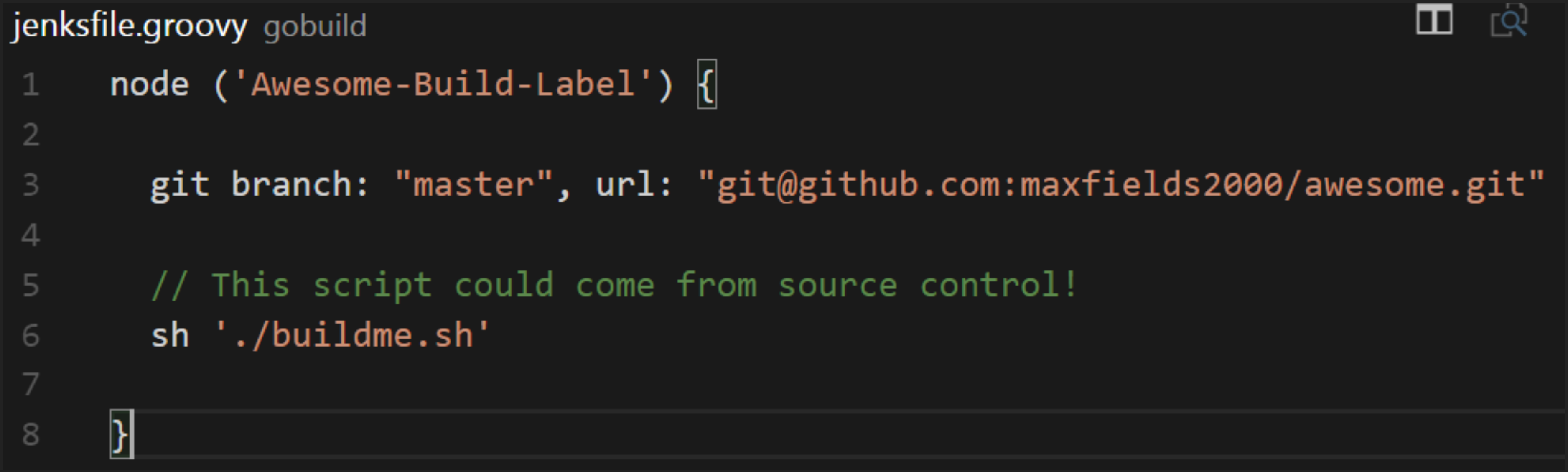


Image Pulled to Swarm

Putting It All Together



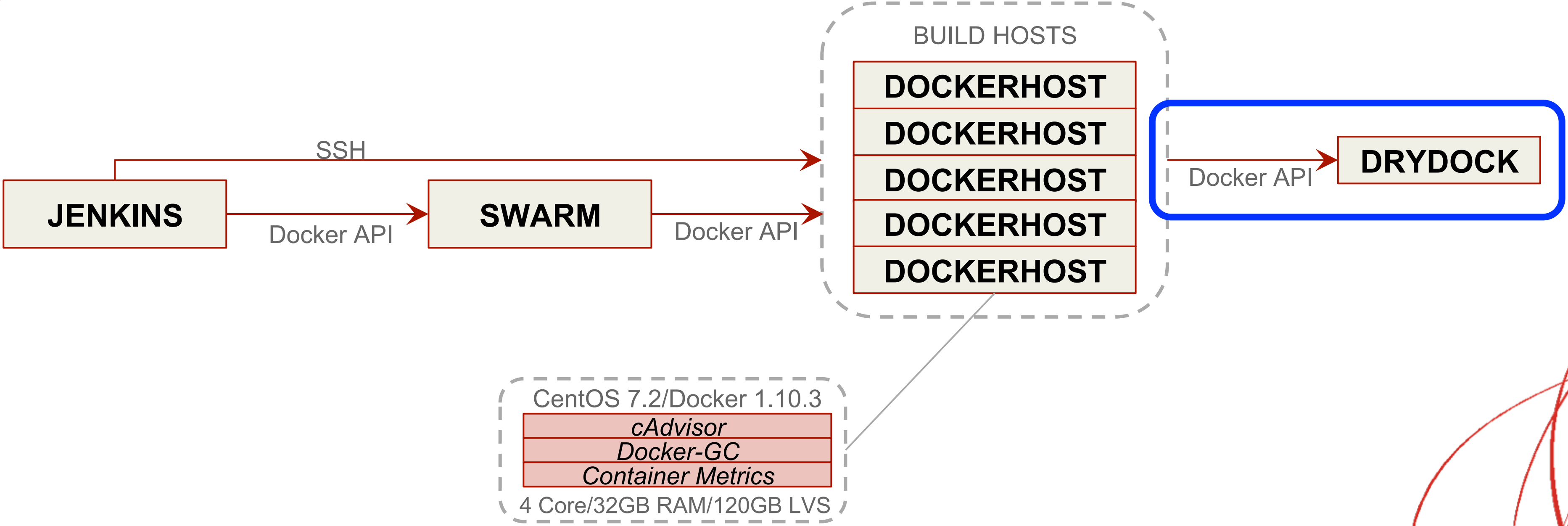
Build Job Quick Look



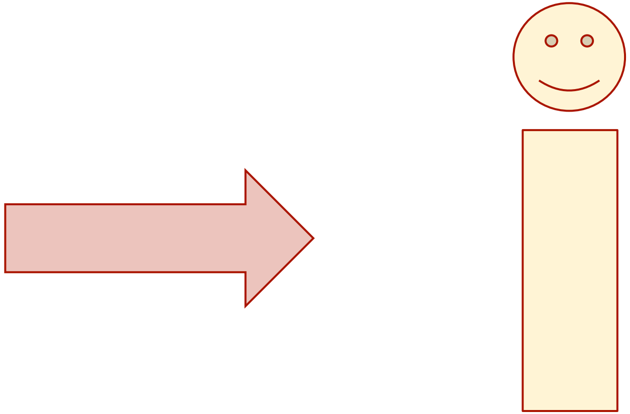
```groovy
node ('Awesome-Build-Label') {

  git branch: "master", url: "git@github.com:maxfields2000/awesome.git"

  // This script could come from source control!
  sh './buildme.sh'

}
```

Dockerception

Where to Build Containers?



Mai Tai’s On the Beach



Build Engineer

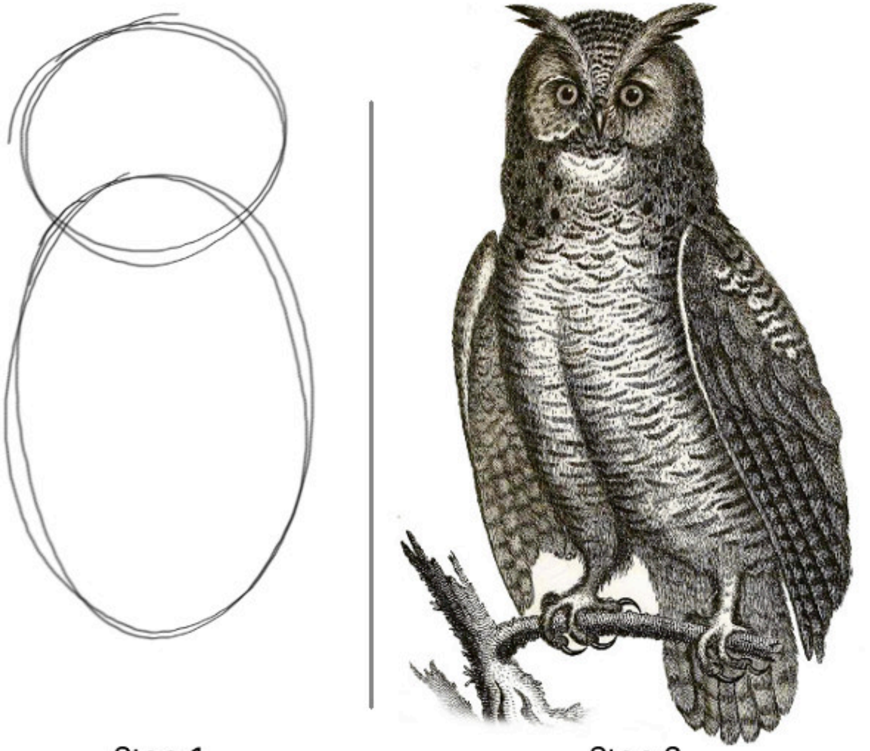
How Do You Actually Build It?

How Do Y Build It?

(engineering.riotgames.com)

https://github.com/maxfields2000/dockerjenkins_tutorial

You Make This Sound Simple.



Step 1
Draw two circles

Step 2
Draw the rest of the owl

Lesson 1 - Docker Isn’t “Simple”

- DOCKERFILES REQUIRE DECENT SYSTEMS ADMIN KNOWLEDGE
- DOCKER IMAGES THAT BUILD DOCKER IMAGES IS DOCKERCEPTION
- DOCKER “VOODOO AND BLACK MAGIC” QUESTIONS

Lesson 2 - Containers != VM’s

CANNOT MOUNT REMOTE FILE SYSTEMS

EPHEMERAL CONTAINERS DON’T MAINTAIN STATE

HAVE RULES FOR DOCKERFILE VS RUN TIME

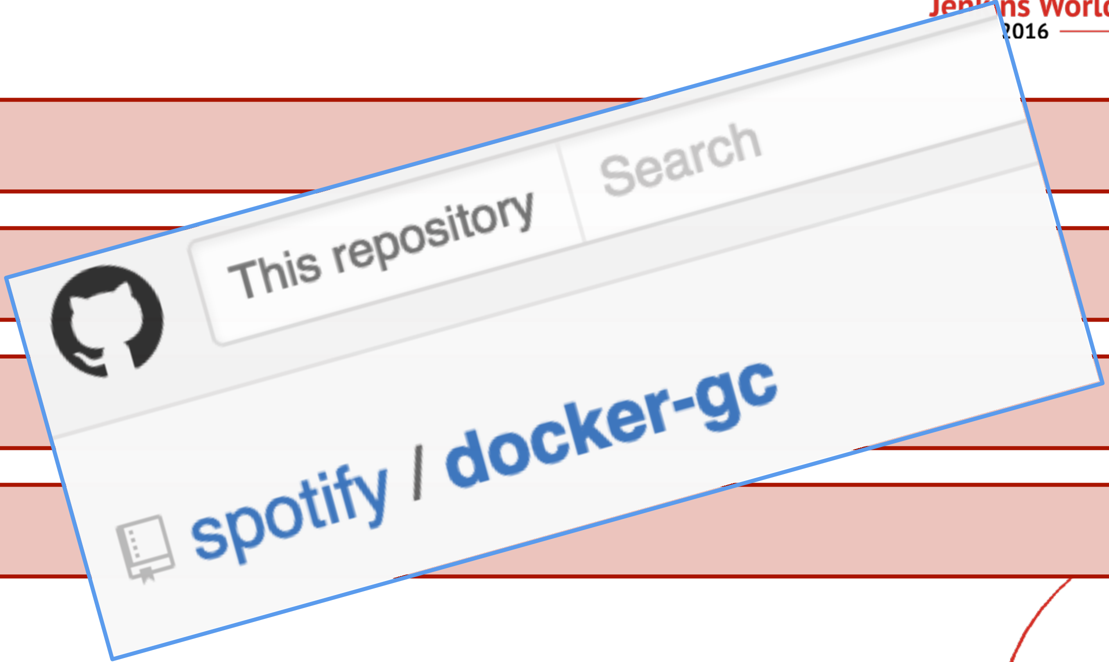
Lesson 3 - Garbage Collection

DOCKER BUILD

DOCKER RUN

DOCKER PULL

VOLUME



spotify / docker-gc

Lesson 4 - Maintenance/Failure

PULL HOSTS ON/OFFLINE

UPDATE ALL IMAGES

ROLLING RESTARTS

Lesson 5 - How Will You Upgrade?

Lesson 6 - Credentials & Security

WHAT SHOULD YOU PUT IN YOUR BASE IMAGE?

SSH KEYS IN YOUR REGISTRY?

PASSWORDS IN YOUR SOURCE CODE?

PASSING EVERYTHING IN AS A ENVIRONMENT VARIABLE?

PROFITS
SALES

“What if we don’t change at all ...
and something magical just happens?”

Over 1200 New Build Jobs Created

30% of All Environments Are Containers

Environment Change/Create/Fix

Tickets Dissappeared

We No Longer Have “Just” A Build Team

QUESTIONS?